Solar Panel Capstone

Optimizing cost efficiency for residential installations

By Zachary Brown

Abstract

Solar panel installations are one of the most effective ways that an individual can lower their carbon footprint, however, they often cost over $10,000 and are unaffordable for many. My goal in this project was to use solar panel installation data from the Lawrence Berkeley National Laboratory's Tracking the Sun program to identify what aspects of a solar panel installation can be controlled to maximize cost efficiency for a person living in Austin Texas. After compiling the data, expanding the features, and comparing a wide range of regression models, an XGBoost regression model was developed that identified the following guidance:

- Design the solar panel installation with a relatively high inverter loading ratio
- Identify and secure any rebate or grant available
- Consider buying a solar panel model with lower conversion efficiency
- Purchase the largest configuration of solar panels possible for the house, sizing the inverter quantity to maintain that high inverter loading ratio
- Consider scheduling the installation for July or December

The model had root mean squared error (RMSE) of 35 million which was due primarily to three major outliers that appear to be typos in the total installation cost. To expand on this work in the future I think it would be valuable to design a tool that takes weather, location, this historical installation data, as well as average electricity use to make a recommendation on how large someone should build their solar panel array, what rebates or grants may be available, and how long it will take for the solar panels to pay for themselves.

Introduction

As global temperatures rise, many people are looking for ways they can affordably reduce their carbon footprint to help slow climate change. One effective change homeowners can make is to install solar panels for electricity generation, however, the average cost for a residential solar panel installation is around $16,000 according to Forbes.com. For many homeowners the only way to invest in solar panels is to ensure that they are as cost-effective as possible so that they can potentially make their money back over the life of the solar panels.

The Lawrence Berkeley National Laboratory maintains a program called Tracking the Sun which collects nationwide solar panel installation data. In addition to creating a yearly report with their insights, they host the data for public use. This dataset includes features such as the total price of the installation, the system size in direct current (DC) kilowatts (KW), any rebates or grants obtained,

the solar panel tilt angle, rotation angle, inverter size, inverter loading ratio, installer, installation state, city, and zip code, electric utility territory, etc.

For some background on how a solar panel system works, first the photovoltaic module (solar panels) absorb sunlight and convert it into DC electricity. An inverter then converts the DC electricity to alternating current (AC), which is then transmitted to the house first, and the grid second if there is excess. Solar panels can be mounted either on the ground, or more typically for residential installations, on the roof. They are typically angled to maximize the amount of sunlight they can capture, factoring in the tilt angle (facing straight up vs facing sideways) and the azimuth angle (360° rotation along the north, south, east, west axis).

My goal in this project was to take the most recent residential installation data from this Tracking the Sun dataset and first to calculate the cost per KW which would be my metric for cost efficiency. I would then develop regression models to help identify what customers in Austin Texas can do to minimize that metric for their own solar panel installation.

Methodology

Loading Data

I began this project by importing and joining the parquet datafiles hosted by the Tracking the Sun program. I looped through the files for each state, saved the parquet file, loaded them into individual dataframes, added a column identifying the state for each, and then joined them into one unified dataframe. I then reduced the data to only residential installations to keep the data relevant to this project goal. The data was then split by installation year, and since there were only 2100 entries from the year 2021, the data was reduced to just the years 2020 and 2021, leaving ~240,000 entries to work with. I then broke out installation month into a separate column to capture any seasonality associated with pricing.

Creating the key metric

Since the metric of this project is price per KW, any field missing the total installed price value is of no use to the project, so I dropped any rows missing that information. The dataset encodes missing values as -1, so I recoded missing values in the rebate or grant column to 0 so that those missing values didn't impact the final price per KW metric. I then limited the dataset to only rows where the system size was greater than zero, because A) it doesn't make sense that someone would install solar panels that generate zero or negative electricity and B) that will be the denominator in the equation calculating price per KW. At this point I created the key metric column: price_per_kw using the following equation:

$$price\ per\ KW\ = \frac{total\ installed\ price - rebate\ or\ grant}{system\ size\ DC}$$

Tuning location features

I then created dummy columns for each state present in the dataset. Digging deeper into location, I trimmed zip codes to only the first 5 digits since there were some 9 digit zip codes, and then I created dummy columns for each unique zip code with more than 30 entries. The zip code column was then dropped to avoid collinearity. Finally I checked the individual cities within Texas and found that there were no entries for Austin, so I decided to drop the city column entirely and use zip code as my smallest location grouping.

Feature engineering

To avoid data leakage I dropped the total installed price column from the data. This was important because it is a key driver of price per KW that is essentially out of the control of the customer. I also dropped the customer segment after reducing the data to only residential installations since there was no information to glean from that feature anymore. Next I checked the proportion of each feature that was missing and I dropped any features missing 30% or more of the data.

I checked categorical columns with ‘object’ type for the number of unique values in each. I proceeded to drop system ID because there was almost one unique value for every entry in the data, and then I created dummy columns for each value in any columns with fewer than 25 unique values. For any remaining features I created dummy columns for any unique value with more than 30 occurrences and then dropped the original column to avoid collinearity.

To wrap up the data preparation I performed a 75%/25% train-test split on the data. I then fit scikit-learn’s simple imputer to impute the most common value in the train set, then transformed both the train and test set using this imputer. Next I fit a standard scaler on the train set and then transformed the train and test sets. At this point I narrowed the features down from almost 2900 to the 400 most important using f regression. The x_train, x_test, y_train, and y_test sets were then saved separately to avoid data leakage throughout model development.

Modeling

My plan for this project was to screen a wide range of regression models to identify the best model for this dataset. As a first step I loaded just the x_train and y_train datasets and then split them into 10% train and 90% test subsets. I then used random search CV with 60 iterations to tune hyperparameters for each model of interest. Figure 1 is a graphical visualization of the resulting RMSE values for easy comparison. Table 1 shows the models tested, hyperparameters that were tuned, and those same RMSE values for this initial screening.

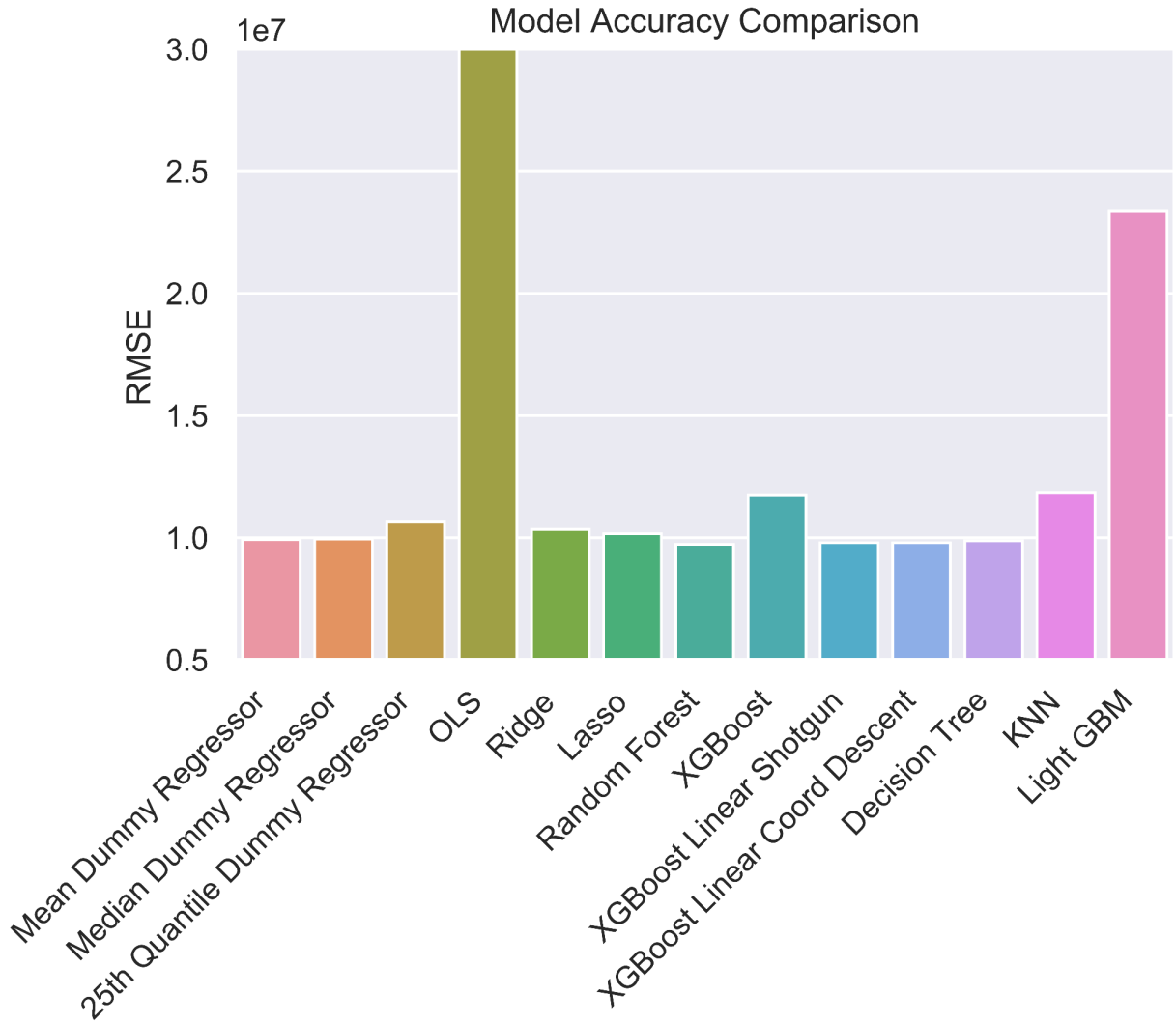

Figure 1. RMSE is plotted for each model using the best hyperparameters identified through randomized search CV. Models were trained on only 10% of the available data.

Table 1: The models initially trained on 10% of the data are shown with the hyperparameters that were tuned and the best resulting RMSE when tested against the remaining 90%.

Model	Hyperparameters	RMSE
Mean Dummy Regressor	N/A	9,923,409
Median Dummy Regressor	N/A	9,941,871
25th Quantile Dummy Regressor	N/A	10,670,605
Ordinary Least Squares	N/A	1.2650560e+19
Ridge Regression	alpha	10,327,496
Lasso Regression	alpha	10,155,465
Random Forest Regression	max_features, max_depth, min_samples_leaf, n_estimators	9,720,312
XGBoost Regressor	n_estimators, max_depth, eta, colsample_bytree	11,759,799
Linear XGBoost Regressor - shotgun updater	reg_lambda, reg_alpha, feature_selector	9,788,061
Linear XGBoost Regressor - coordinate descent updater	reg_lambda, reg_alpha, feature_selector	9,789,561
Decision Tree Regressor	max_depth, min_samples_leaf	9,865,568
K Nearest Neighbors Regressor	n_neighbors	11,856,353
Light GBM Regressor	num_leaves, n_estimators, max_depth, learning_rate	23,385,497

After comparing these initial results and tuning the hyperparameters, most of the models were retrained using the same hyperparameters, this time on 80% of the data. The resulting RMSE for each is compared in Figure 2.

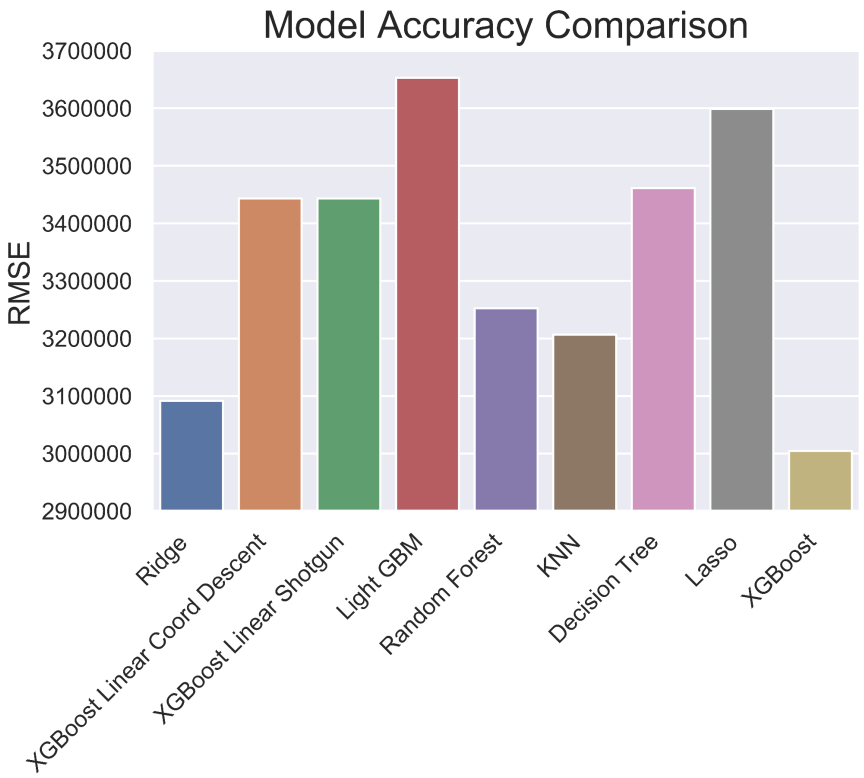

Figure 2. RMSE is plotted for various models using hyperparameter settings from the initial tuning and trained on 80% of the available data.

Now, having identified the best options as ridge, KNN, and XGBoost regressor (I included Light GBM because in earlier iterations of this notebook it performed much better than this), I explored the hyperparameter tuning results to identify any that appeared to have room for improvement. Figure 3 shows the hyperparameter tuning results for XGBoost and Figure 4 shows the same for Light GBM.

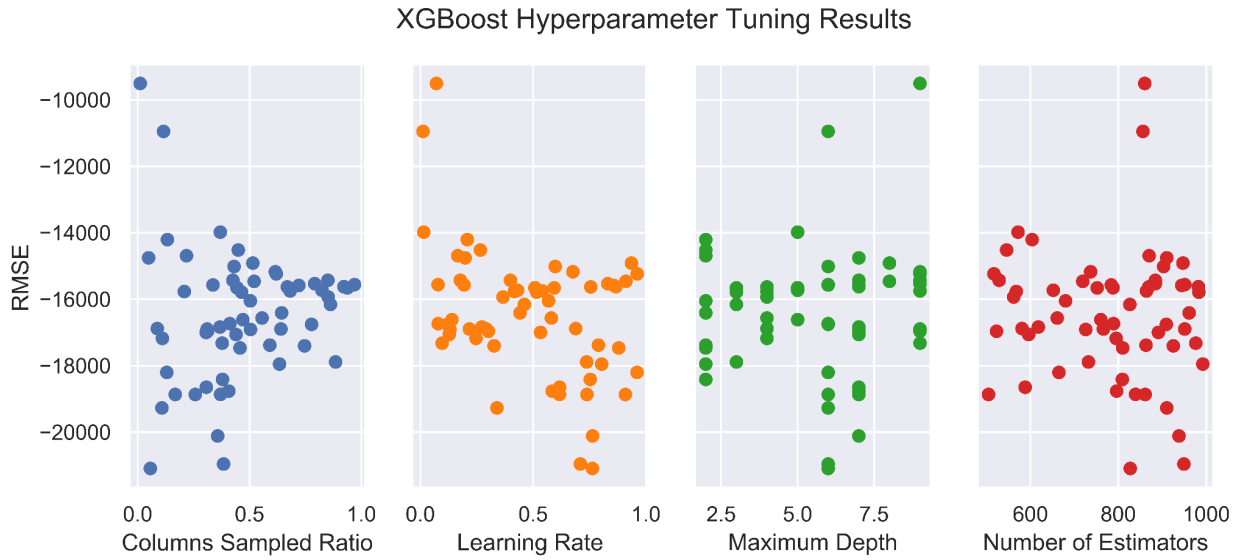

Figure 3. Hyperparameter tuning values for XGBoost are plotted against the resulting RMSE values.

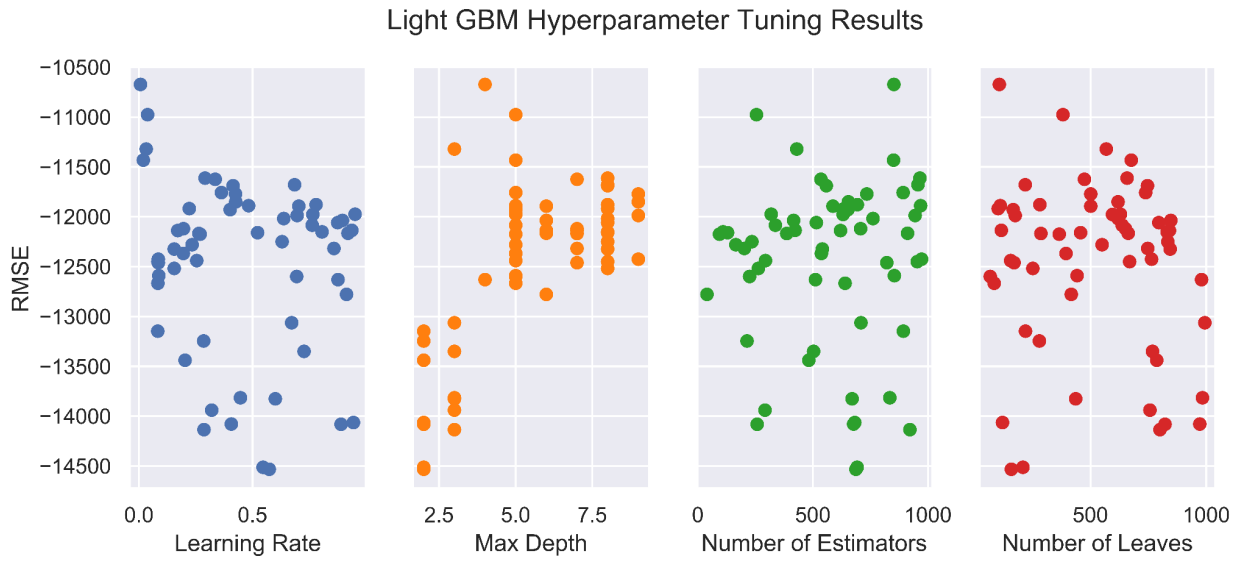

Figure 4. Hyperparameter tuning values for Light GBM are plotted against the resulting RMSE values.

These results suggested that I probably found a sweet spot for max_depth and n_estimators in the XGBoost model since there are no obvious trends, so I decided to hold those where they are. I decided to hold the num_leaves and n_estimators in Light GBM for the same reason. I then proceeded to tune the remaining hyperparameters in those four models this time fitting 80% of the data to try to squeeze out any remaining improvement possible. Those four results are plotted in Figure 5.

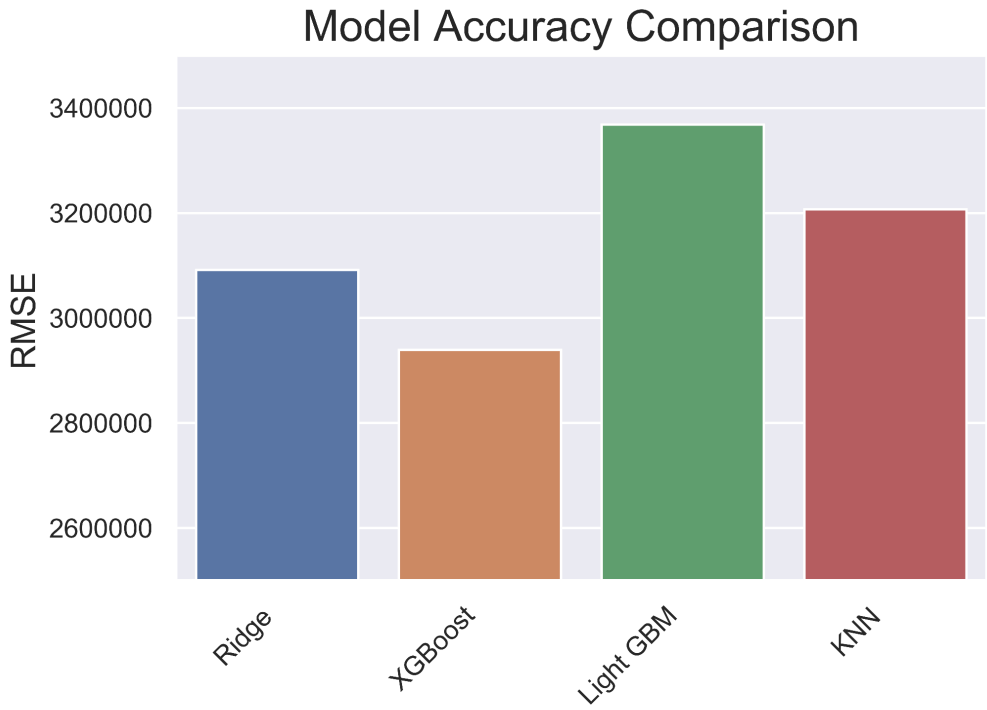

Figure 5. The four remaining models are plotted showing their RMSE after hyperparameter tuning on 80% of the data.

Insights and Performance

Exploratory data analysis findings

To begin my exploration of the data I began with a simple box plot of the cost efficiency to get a feel for the distribution of the data (Figure 6).

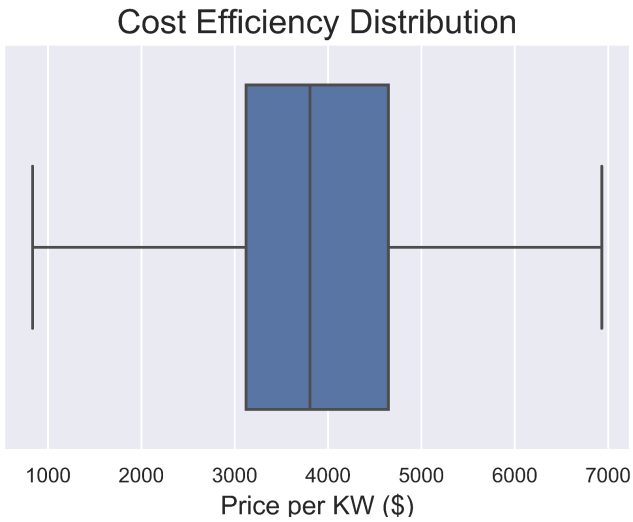

Figure 6. The distribution of price per KW for this dataset, excluding outliers.

Next, to get an idea of whether Texans should expect to pay more or less per KW compared to the rest of the US, I plotted empirical cumulative distribution functions (ECDFs) for both (Figure 7). I then verified with a Shapiro-Wilk test that both portions of the data were normal and confirmed with an independent t-test that the two are significantly (t statistic = -8.33, p-value = 8e-17).

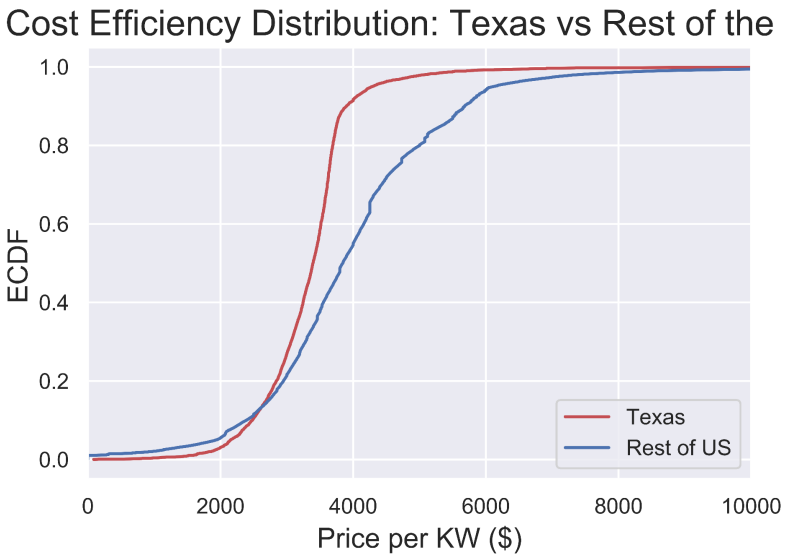

Figure 7. ECDFs for Texas installations compared to non-Texas installations showing that Texas has a more narrow distribution at a lower average.

Next I prepared boxplots of installations by month (Figure 8) and then verified via chi-squared test that there is a significant correlation between the installation month and price per KW (test statistic: 8730078, p-value: 0.0).

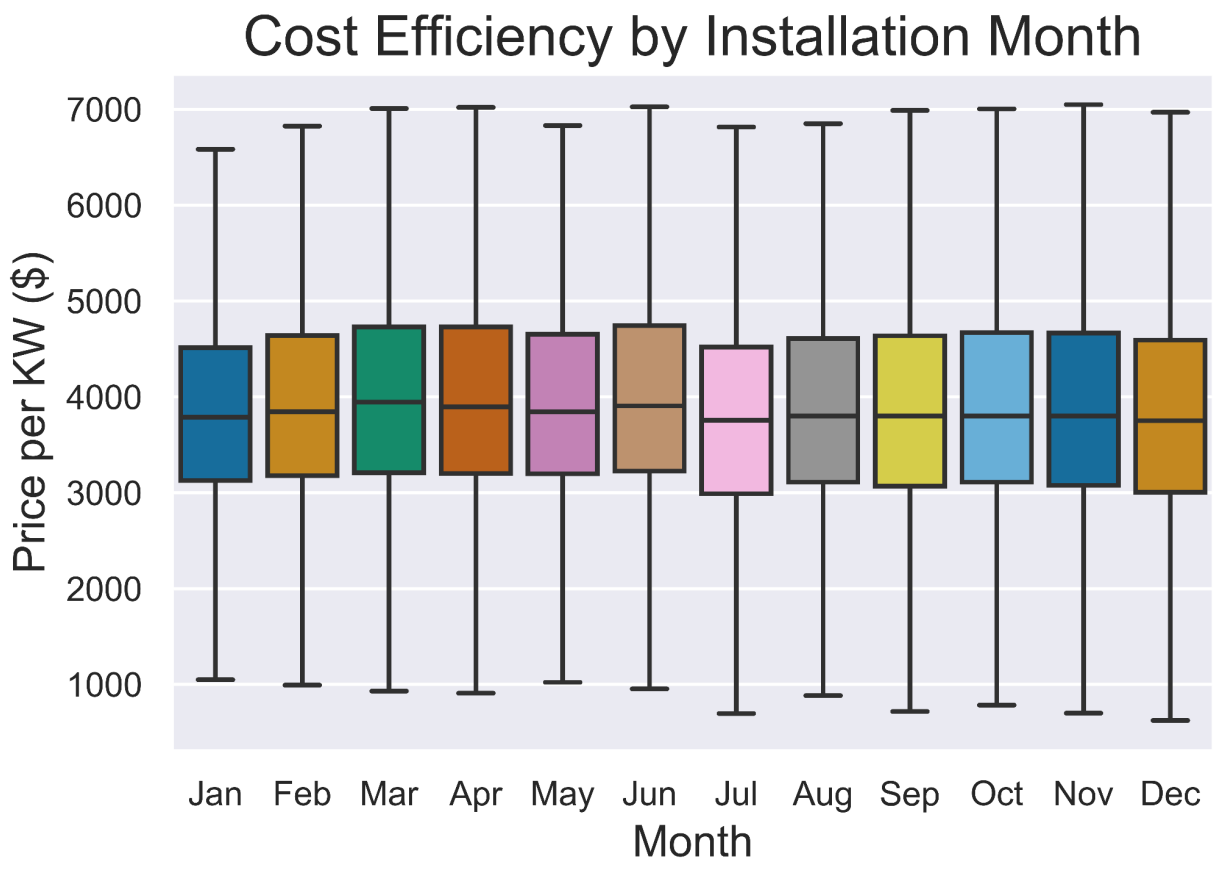

Figure 8. The distributions of cost efficiency vs installation month are plotted with outliers hidden.

I proceeded to loop through any categorical variables that allowed yes/no/missing responses and plotted boxplots of each followed by t-tests to confirm significant correlation. These variables included whether the installation was an expansion of an existing system or not, whether the system installed was multiple phase, whether the system includes tracking equipment, if the system is ground or roof mounted, whether the system is owned by a third party, if it was self installed, whether there were additional modules installed, if the modules were building integrated, if the panels can absorb light from both sides, if there were additional inverters installed, if the inverter is a micro-inverter, a hybrid inverter, or is built in to the photovoltaic system, and finally, whether there is a DC optimizer built in. Of these features, self installation, additional modules, building integration, additional inverters, and hybrid inverters all had significant correlations with cost efficiency.

Next I looped through the continuous features and prepared scatterplots to visualize any obvious correlations with cost efficiency. Two that immediately showed obvious negative correlations were number of inverters (Figure 9) and system size (Figure 10).

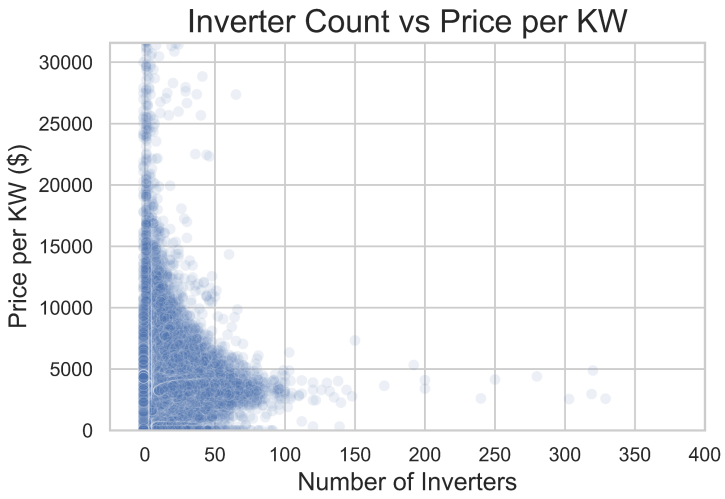

Figure 9. Inverter size is plotted against cost efficiency, showing a clear negative correlation.

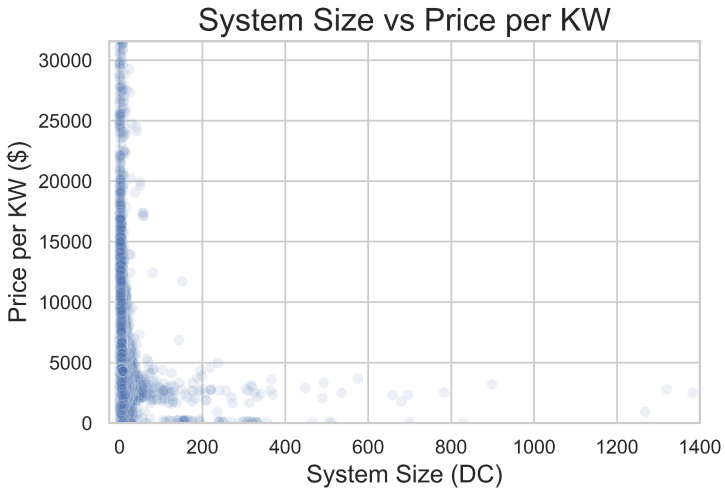

Figure 10. System size is plotted against cost efficiency, showing a clear negative correlation.

Other continuous features were less obvious, so OLS regression lines were fitted and the p-values for the intercept and slope were used to determine whether the correlation was significant. In all cases where I performed this analysis the R^2 was less than 0.00, so while it was clear that none of these features singlehandedly predicts cost-efficiency, they may still have some significant correlation. Module efficiency was the first feature I tested this way, with the following parameters (Table 2) and plot (Figure 11).

Table 2. Slope, intercept, and each corresponding p-value are shown for the OLS regression between module efficiency and cost efficiency.

	Statistic	**p-value**
Slope	3880.995467	2.791065e-06
Intercept	3254.944534	5.904827e-87

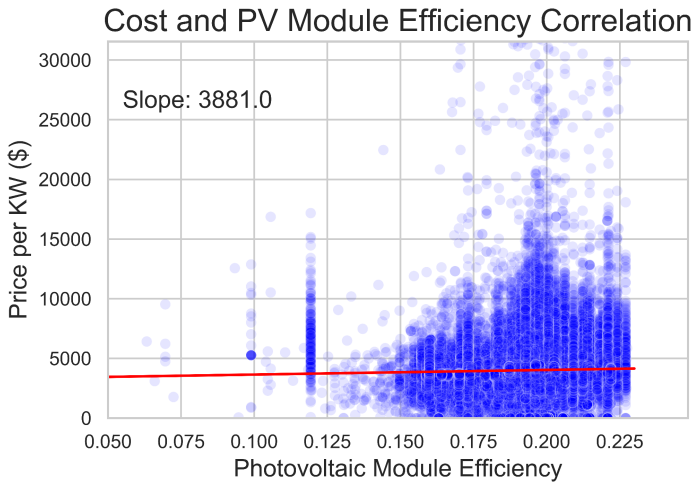

Figure 11. The correlation between module efficiency and cost efficiency is shown with the OLS regression in red.

Advertised capacity was similar, showing a slight positive correlation with cost efficiency, but inverter loading ratio was a particularly surprising feature. When plotted (Figure 12), there is no obvious slope to the data plotting inverter loading ratio against cost efficiency. However, the OLS analysis shows a significant negative correlation as described in Table 3.

Table 3. Slope, intercept, and each corresponding p-value are shown for the OLS regression between inverter loading ratio and cost efficiency.

	Statistic	**p-value**
Slope	-517.405285	2.207527e-37
Intercept	4607.379181	0.000000e+00

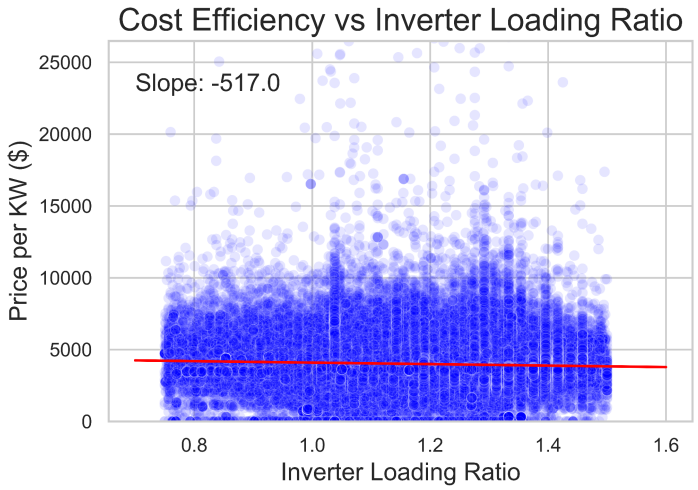

Figure 12. The correlation between inverter loading ratio and cost efficiency is shown with the OLS regression in red.

This concluded my preliminary analysis of the data prior to modeling. These results proved to be quite insightful when analyzing the final model results.

Model results

After training the final XGBoost regressor model on the full training set it was tested on the separately saved test data. The RMSE increased from 2.96 million in the training set to 35.05 million when exposed to the test data. I suspected that there may be outliers triggering this increase, so I plotted the y-predicted values against the y-test and identified 3 outliers (Figure 13).

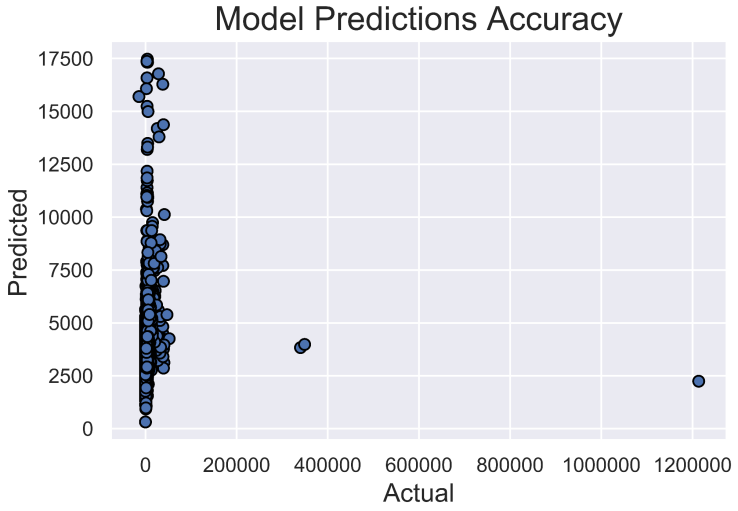

Figure 13. The y-test values are plotted against the predicted values, showing three very significant outliers.

When I identified the outliers and looked through their entries in the original dataset I found that all three listed the total installed price in the millions, whereas most entries were in the tens of thousands, with an average of around $25,000. Given that those three outliers listed typical values for system size but the prices were 3427200, 2631400, and 8255000, it seems likely that they were entered with two extra zeros at the end of their true prices. Nonetheless, I proceeded to call the feature importances for this model to gauge what features it found most important to predicting cost efficiency (Figure 14).

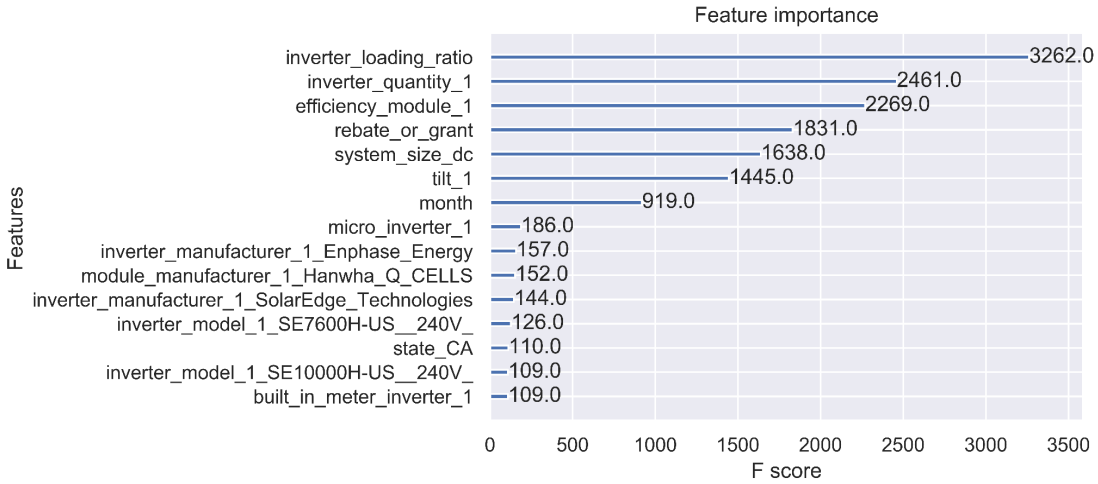

Figure 14. The top 15 features of my XGBoost model are listed by F score.

As shown in the exploratory data analysis, all of the top seven features were identified as having significant correlations with price per KW. Of those top seven features, inverter loading ratio, inverter quantity, rebate or grant, and system size all had negative correlations with price per KW (improving cost efficiency). Module efficiency was the one feature that had a positive correlation with price per KW (decreasing cost efficiency). Month and tilt are slightly more complex.

If we dig deeper into those features with negative correlations, it becomes apparent that there is an industry-wide volume discount, where there is less cost added as the size of the job increases. That discount is directly reflected in inverter quantity and system size. Inverter loading ratio is related, but slightly more complicated. The inverter loading ratio reflects the maximum amount of DC current the solar panels can create vs the maximum amount of DC current the inverters can convert. Say a solar panel installation has an inverter loading ratio of 1, so at peak hours the solar panels can create 5 KW of DC electricity and the inverter can convert that full 5 KW. That's great during peak hours, however, during off-peak hours this system will only capture a fraction of this capacity. In a system with the same inverter capacity but an inverter loading ratio of 1.3, some of that peak energy will be lost because the inverters can't handle it, but during the rest of the day that system will produce 30% more energy than the first system. This is shown in Figure 15.

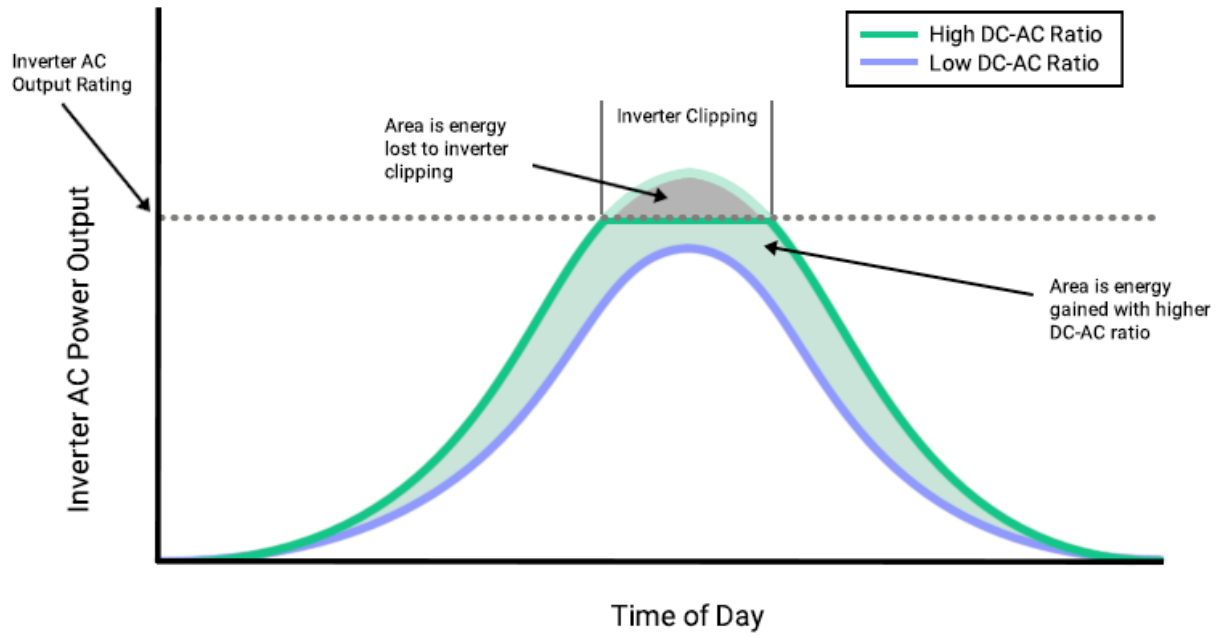

Figure 15. This plot shows the advantage of having an inverter loading ratio greater than 1. **Source**: https://aurorasolar.com/blog/choosing-the-right-size-inverter-for-your-solar-design-a-primer-on-inverter-clipping/

Module efficiency has a positive correlation meaning that as module efficiency increases, the cost per KW does as well. While reduces cost efficiency for the installation, it likely increases the overall cost efficiency for the life of the solar panel array. This would be one feature the

customer would want to dig deeper into for their own installation, because it could have a significant impact on their long-term savings thanks to the solar panels.

Examining installation month, revisiting Figure 8 shows that there are some months with lower average price per KW than others, with July and December standing out as the most inexpensive months of the year. One possible explanation for this could be that holiday specials may help drive down price around the 4th of July and during the December holiday season.

Tilt is much less clear than any other feature. As shown in Figure 16 there are specific angles, multiples of 5, that are much more frequent than others.

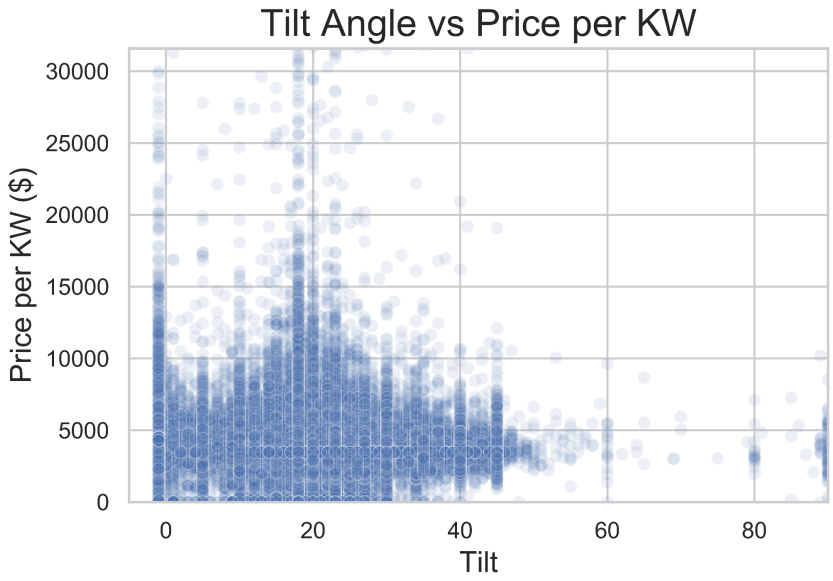

Figure 16. Price per KW is plotted against tilt angle.

It seems most likely that although there is a correlation between tilt angle and cost efficiency, it may be more of a feature of how much data we have at each angle rather than a true direct correlation.

Conclusions

This study has identified multiple features of a solar panel installation that can help a customer maximize the solar energy generated for their money. The most important factor is the overall size of the solar panel array that is chosen. The customer will want to install as large an array as possible, while taking into account their budget and the limits of any money they can recoup by selling excess electricity back to the grid. Along with the array size, they will want to choose a relatively high inverter loading ratio to help minimize inverter cost while still converting a large percentage of that absorbed sunlight. In terms of the up-front installation costs, they will want to take advantage of any rebates or grants available to them, and they should consider whether the added cost of higher efficiency panels will pay for themselves over the long term, or if it makes more sense to stick with cheaper lower efficiency panels. Finally, if the customer is not

on a tight schedule, they should target July or December for their installation, as those months seem to generally offer better cost efficiency.

In the future I would like to take this work a step further and develop a recommendation tool to help make solar panel installations even more accessible for homeowners. Right now there are some tools available to help potential customers predict the output they could get from a solar panel array. However, I'd like to take it even further by taking the customers budget, average monthly electrical use, location, and electrical provider to recommend the size of their solar panel array, the number of inverters they should install, any rebates or grants they are eligible for, and provide the length of time it would take for the solar panels to pay for themselves. There are a lot of factors to consider when deciding whether to install solar panels or not, and I think a tool like this would make that decision much less daunting.