Solar Power Plant Generation Prediction Report

Tarun Kaushik
December 2023
Data Sourced from: Kaggle.org, OpenWeatherMap.org

Introduction:

Solar power is now more widely used than ever, in all parts of the world there are new solar power plants being created to make power grids more green. India has recently committed to generating more solar energy and is currently fifth in the world for highest percentage of solar energy generated. Two plants, one in Gandikota and the other in Nashik, have gathered data over a 34 day period to find trends and make predictions for future power generation. Both plants have 22 sets of arrays, each with their own unique identification number and individual generation data. To make accurate predictions, the plant needs to know how each individual array will perform along with the plant as a whole to determine an accurate yearly quota.

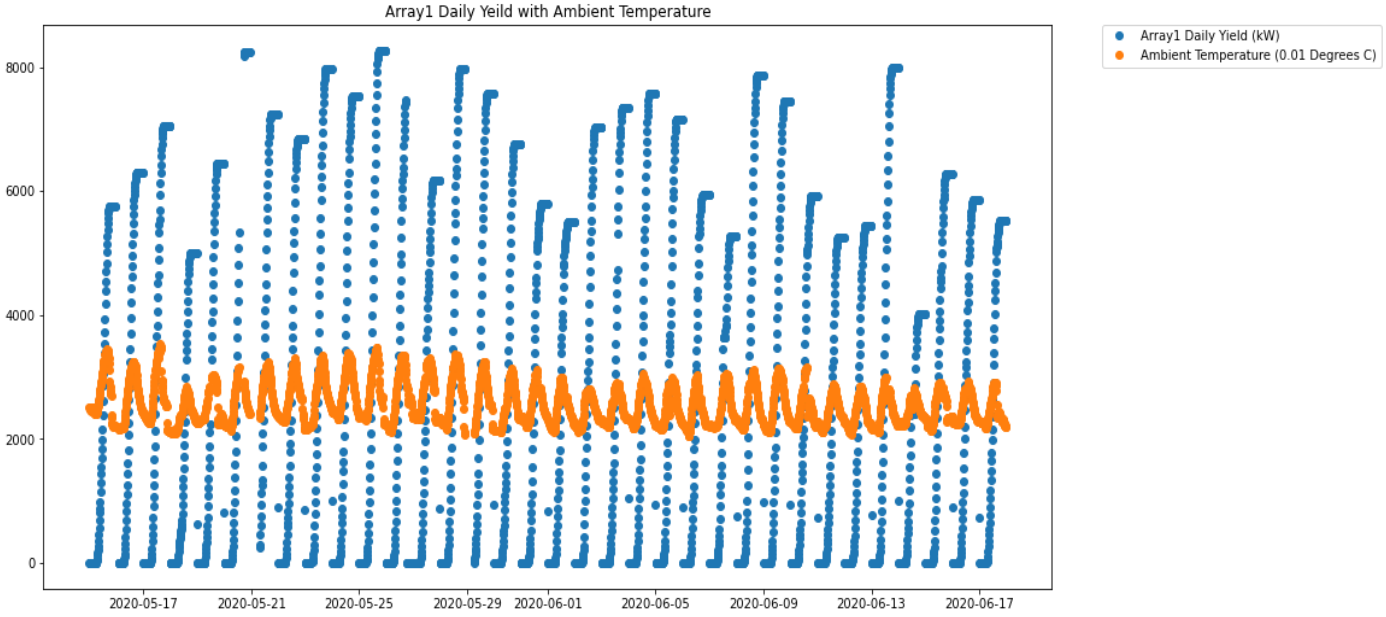
Problem:

Solar plants are a great, clean way to generate energy, however they can be unreliable. It's impossible to know exactly how much power an individual panel or an entire plant will produce in a day. This can make running a solar power plant an incredibly difficult and unpredictable task. How can we use data to better predict the output of a solar power plant on any given day?

Data Wrangling

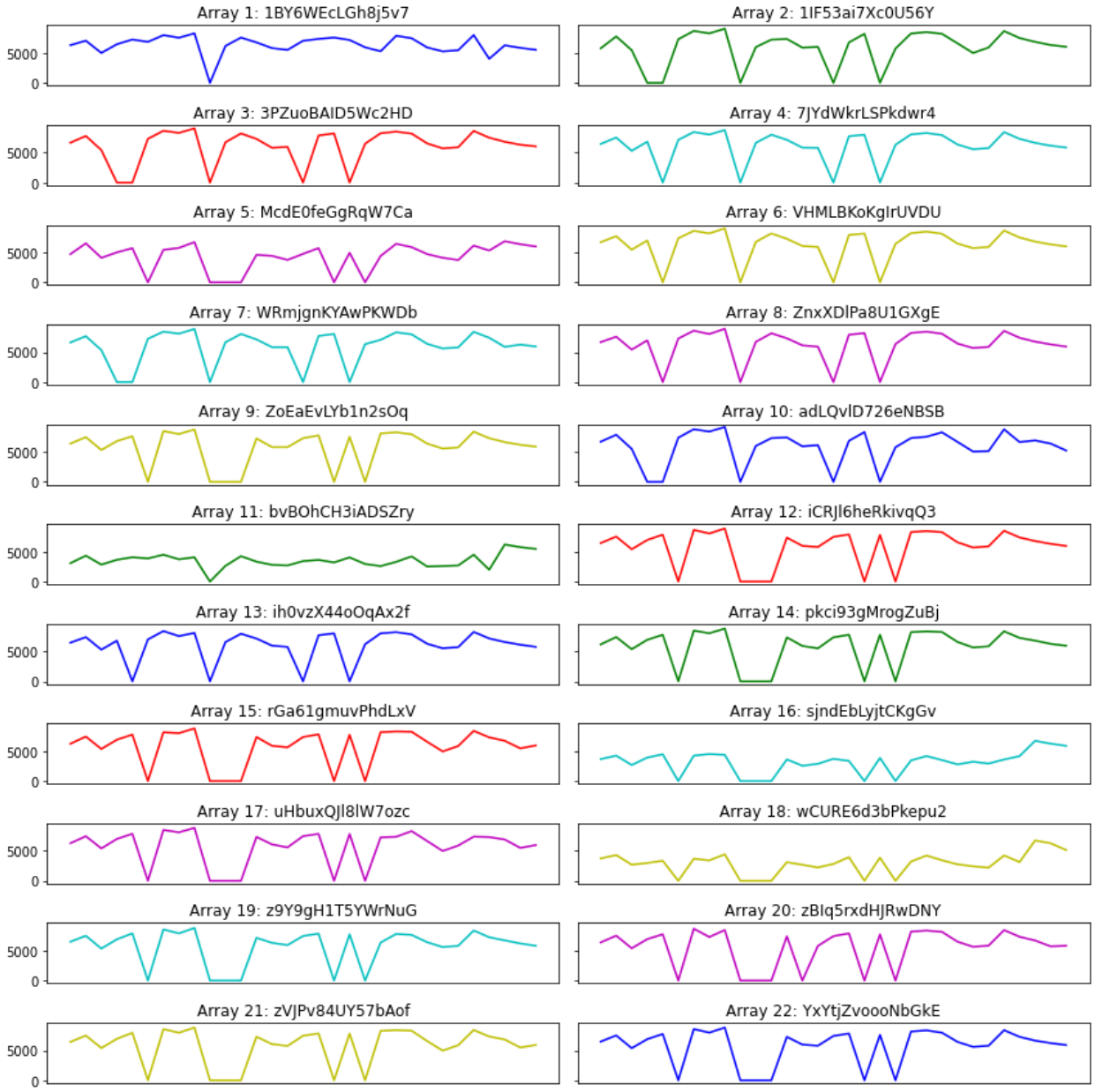
The data originally came as four csv files, two for weather data and two for power generation data. To make things easier, the weather and power datasets were merged together so that there was just one file for both locations. Both datasets were already relatively cleaned and only had missing data for a couple days, I handled this by converting the missing data to null values, even if some array had data while a different one didn't. If one array was missing data it would mess up calculations for all the other arrays, so it made sense to make everything null values. Because each dataset had around 60,000 entries, 100-200 null entries wouldn't make a significant impact.

After renaming some columns and cleaning up the structure of the data, I was able to make a first look at what the total power generation vs ambient temperature looked like:

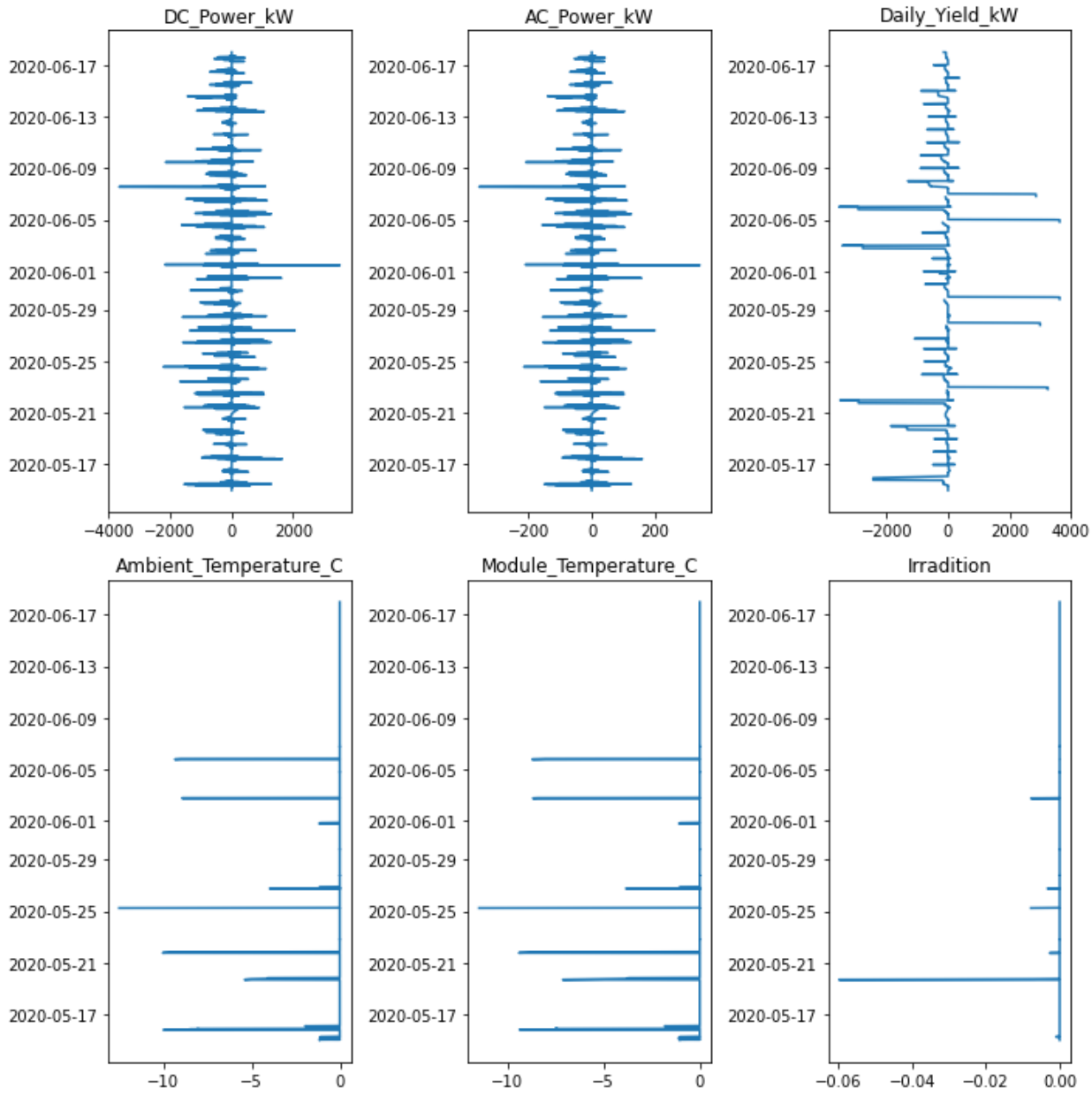


Exploratory Data Analysis

I started off by taking a look at what the generation data looked like for each individual solar array. If there were any major outliers or underperformers it would be easy to spot them visually.



There was no single array that stuck out at first, but when compared to the average daily power generation it was easy to spot arrays 1 and 12 as outliers. After closer inspection of each variable I was able to determine that these two arrays had more instances of lower temperatures and, therefore, lower total power generation overall.



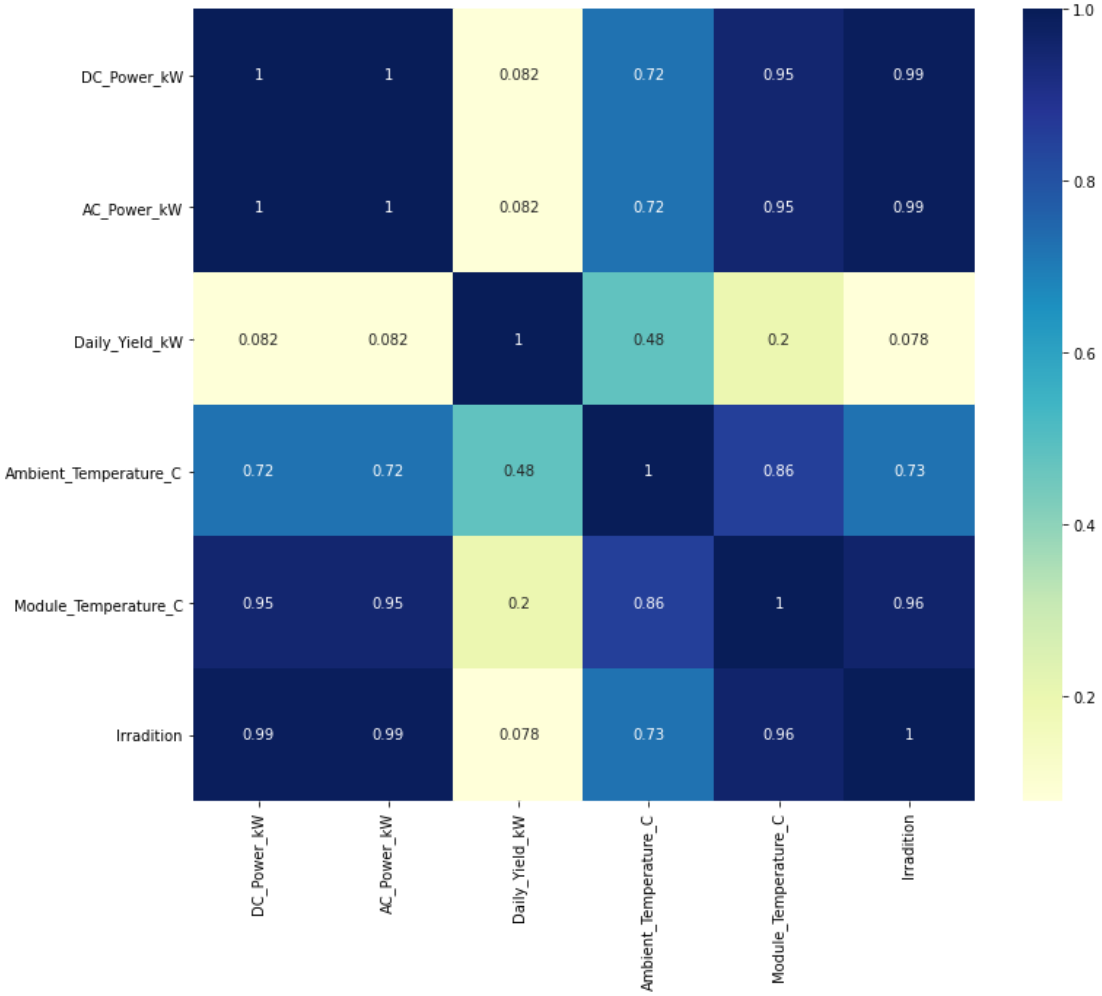
It was difficult to compare the arrays just by looking at graphs, so a new metric needed to be found. Luckily, the Nominal Operating Cell Temperature (NOCT) is a great way to determine how each array performs. The NOCT is how much power a solar panel will generate at certain conditions - Air Temperature at 20°C, Solar Irradiance at 800 W/m^2, and Wind Velocity at 1 m/s. The higher the NOCT, the more power you'll generate at any given temperature.

$$T_{Cell} = T_{Air} + \frac{NOCT - 20}{80} S$$

By finding values at or very close to those conditions, we were able to determine the NOCT for each individual array. For the Gandikota power plant it was around 18.34 with no major outliers, and for Nashik it was around 19.54 with no major outliers. This was surprising though because an average NOCT is around 48 for modern arrays - this led me to believe that each array was either using outdated technology or operating very inefficiently.

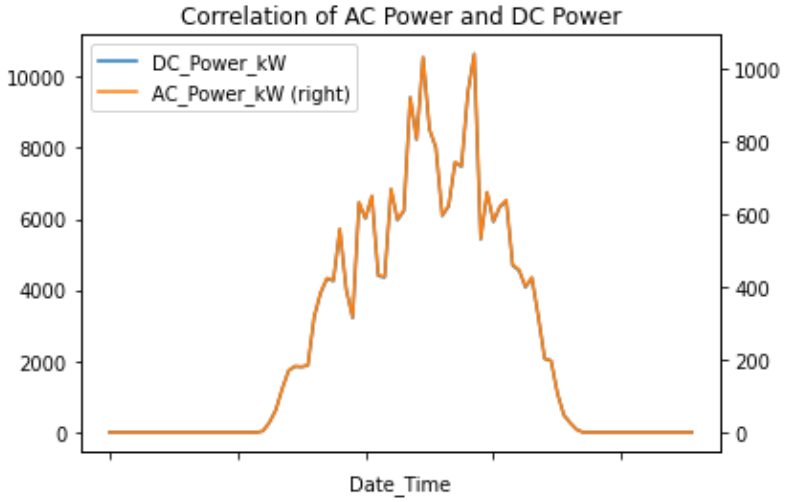
Before moving on, I wanted to see how each variable related to power generation correlated so that we could build our model around predicted power output. By making a correlation heatmap we were able to find the four most important features in:

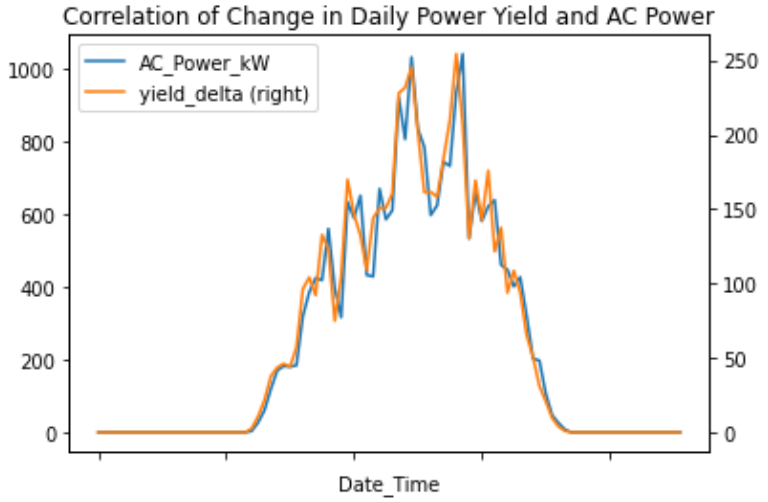
- DC Power (kW)
- Ambient Temperature (C°)
- Module Temperature (C°)
- Solar Irradiation (W/m^2)

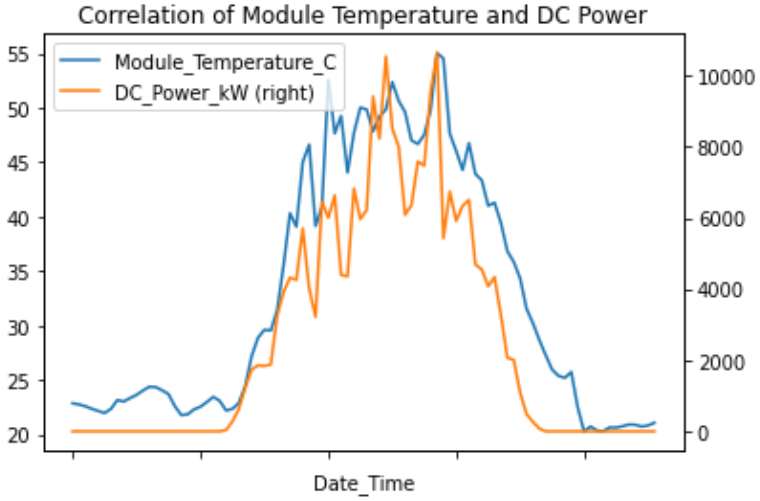


Pre-Processing and Training Data

In order to complete the equation to convert Ambient Temperature to DC Power, and find the relationships between other variables, I needed to find the coefficients between each variable. AC Power and DC Power had a perfect correlation with a Pearson's Correlation Coefficient of 1. AC Power and Daily Yield had a very high correlation with a PCC of 0.985 and finally, Module Temperature and DC Power had a high correlation with a PCC of 0.944. The last two variables weren't perfect, but PCCs above 0.9 are still very statistically similar so these coefficients should work well.

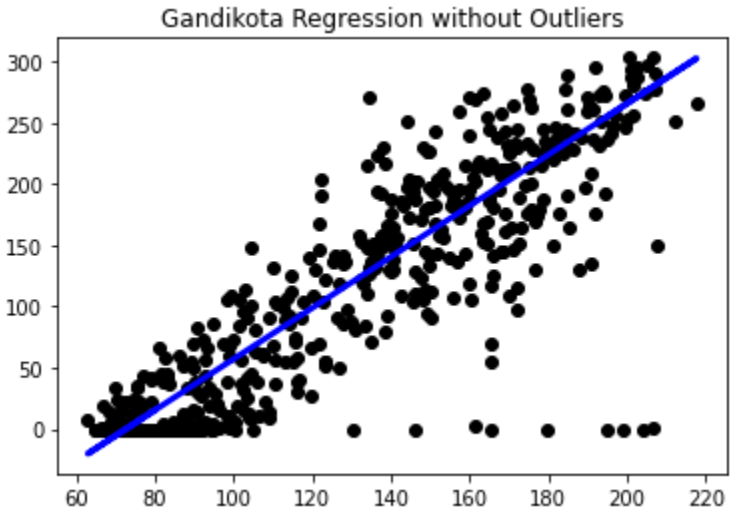






Now we had an accurate equation to convert module temperature to daily yield in power. This, in combination with the NOCT equation, allows us to convert ambient temperature

and solar irradiation directly to power output. To test the effectiveness of the equation, I performed a simple linear regression to compare the real output data to the predicted output (as seen by the blue line)



This is just the result of using one out of two equations to predict power. This prediction, in addition to the NOCT equation should yield very accurate results.

Modeling

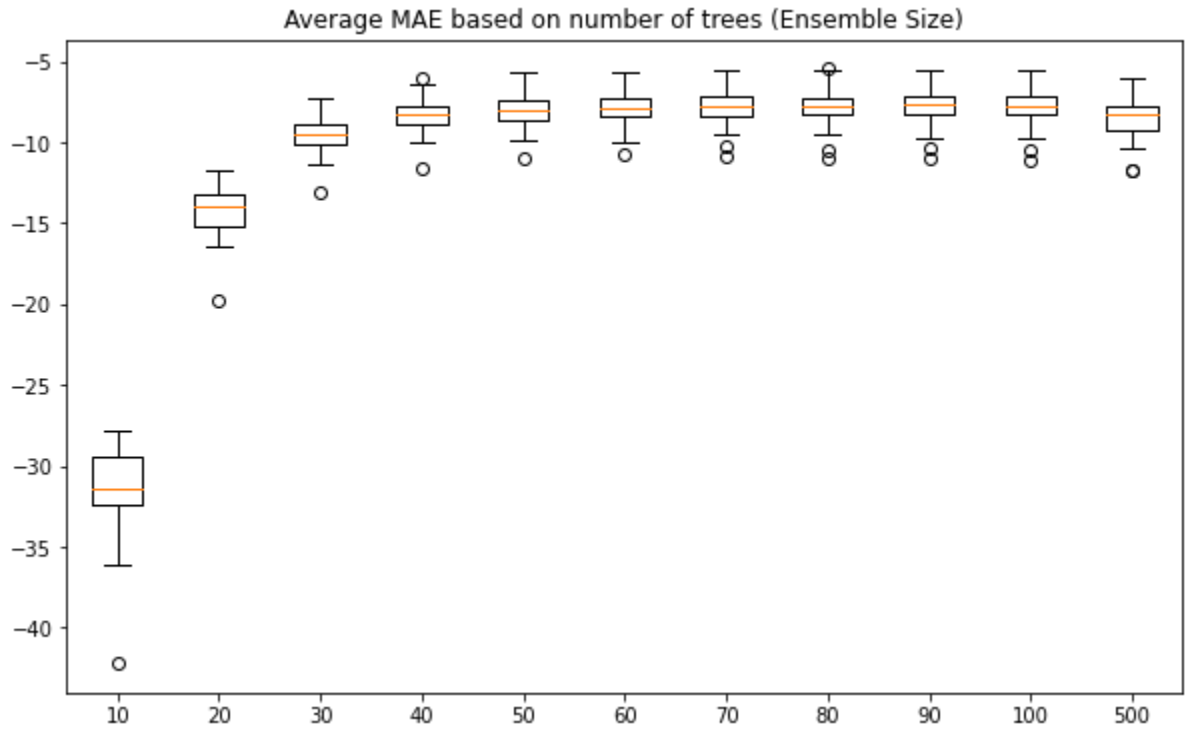
To finish up our equation's prediction we combined the NOCT equation with the new coefficients to find a "predicted sum". The results were surprisingly spot on:

```
Real average power output:  10303078.718452102
Predicted average power output:  10303075.359881079
Total difference (in kW) 3.358571022748947
Coefficient:  0.9999996740225796
```

Once I confirmed that this equation worked with each individual array (the lowest coefficient was only 0.97 which is still very similar), I decided to move on to model selection.

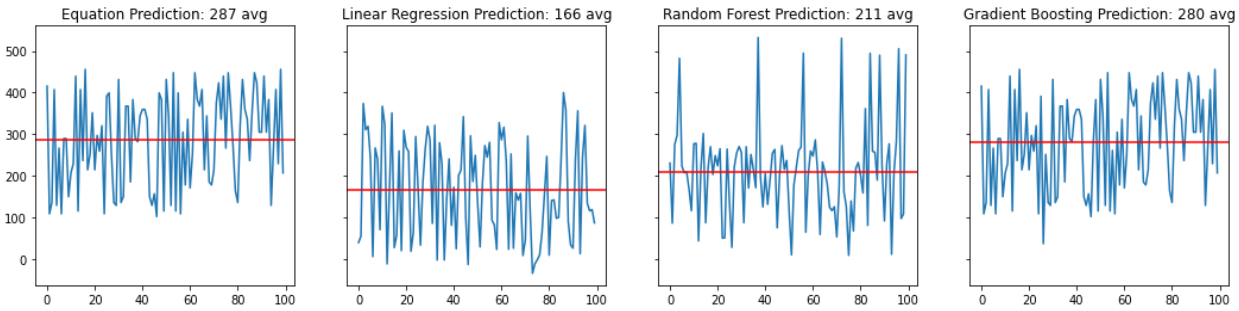
The goal with this model was to use a machine learning algorithm to produce results that were as good as the results of the NOCT and power generation equation. By using the results from both equations and the data from the original dataset we were able to make train/test splits to accurately compare the performance of each model. Three different types of models were tested: Linear Regression, Random Forest, and Gradient Boosting. In each test, I made sure to improve hyperparameters to make the results as accurate as possible. For example, I determined that the ideal number of decision trees for the Gradient

Boosting model was 70 by testing results with variety of different decision tree totals (MAE = Mean Absolute Error):

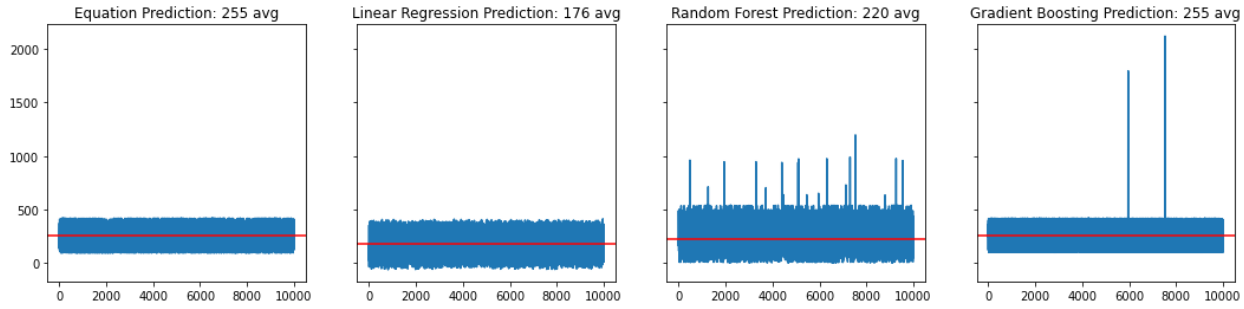


Each model had relatively high scores and low mean absolute errors, but to determine which one would be the best at accurately predicting total output I decided to test each one with random data.

Model Performances: 100 random values:



Model Performances: 10000 random values:

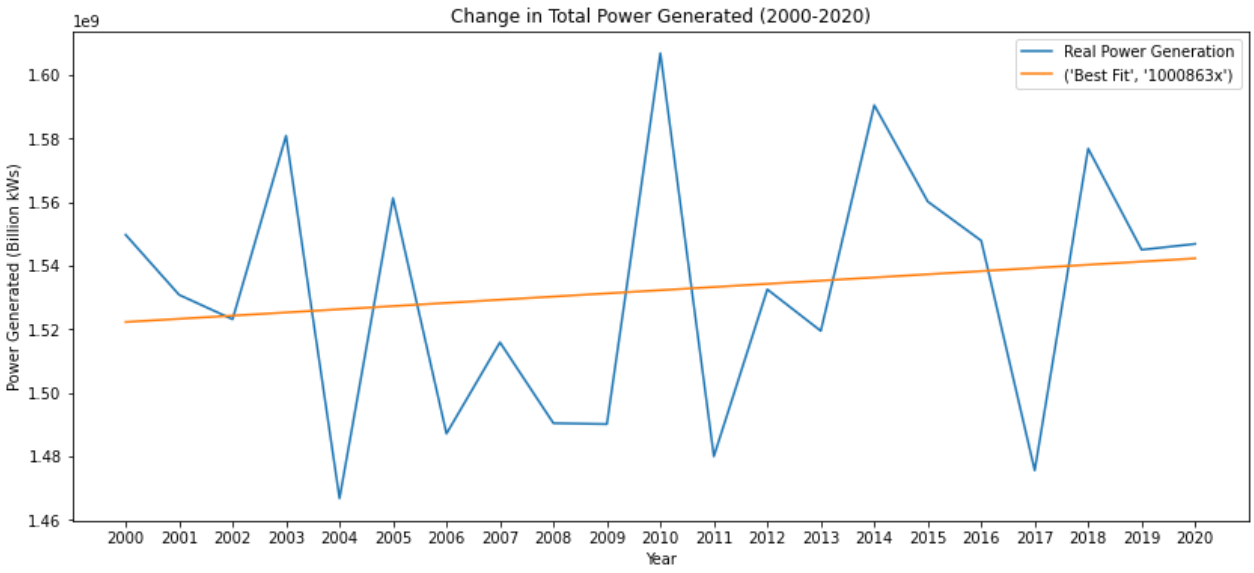


As you can probably see, the Gradient Boosting model is the closest to the prediction equation - which we proved is almost identical to actual expected output. Because of this, the Gradient Boosting model was selected as the most accurate prediction model.

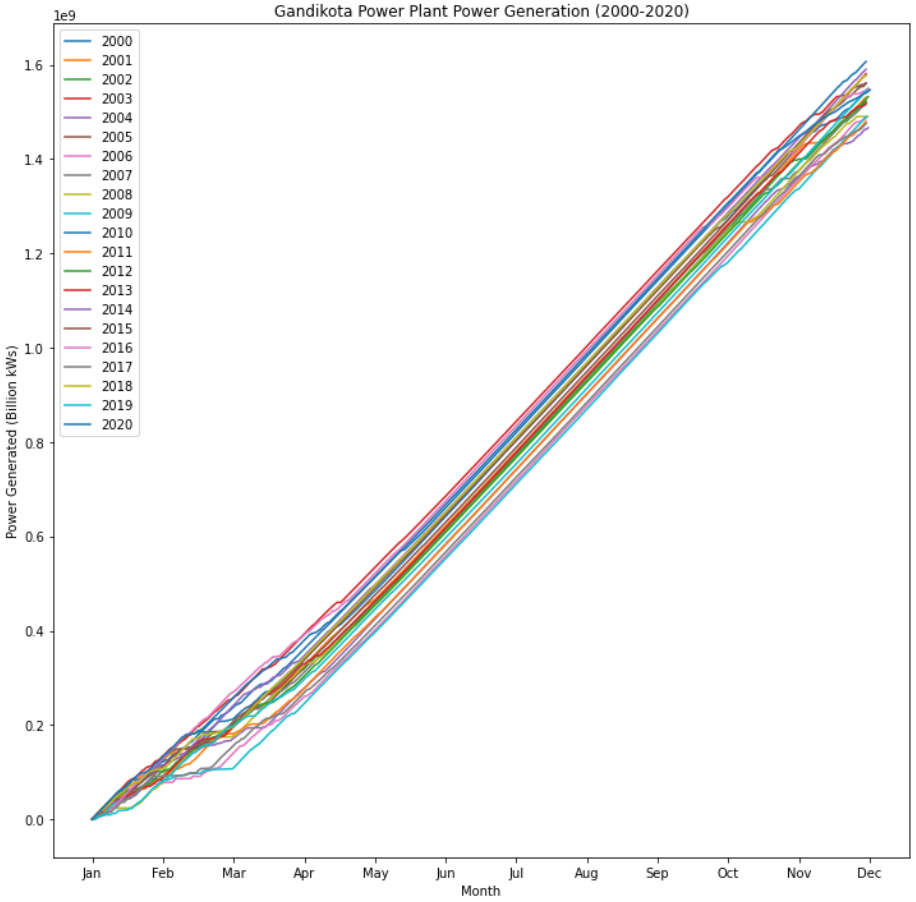
Now that we can predict what the output should be, given weather data, the next step would be to predict what that weather data might look like. This isn't as easy as it sounds, and my method is a huge oversimplification, but at the very least we should be able to determine a good yearly quota for the powerplant to meet. To predict what the weather data should look like, I found a dataset that contained meteorological data from Gandikota for the past 20 years. This was a huge dataset, I actually had to pay $10 to get it.

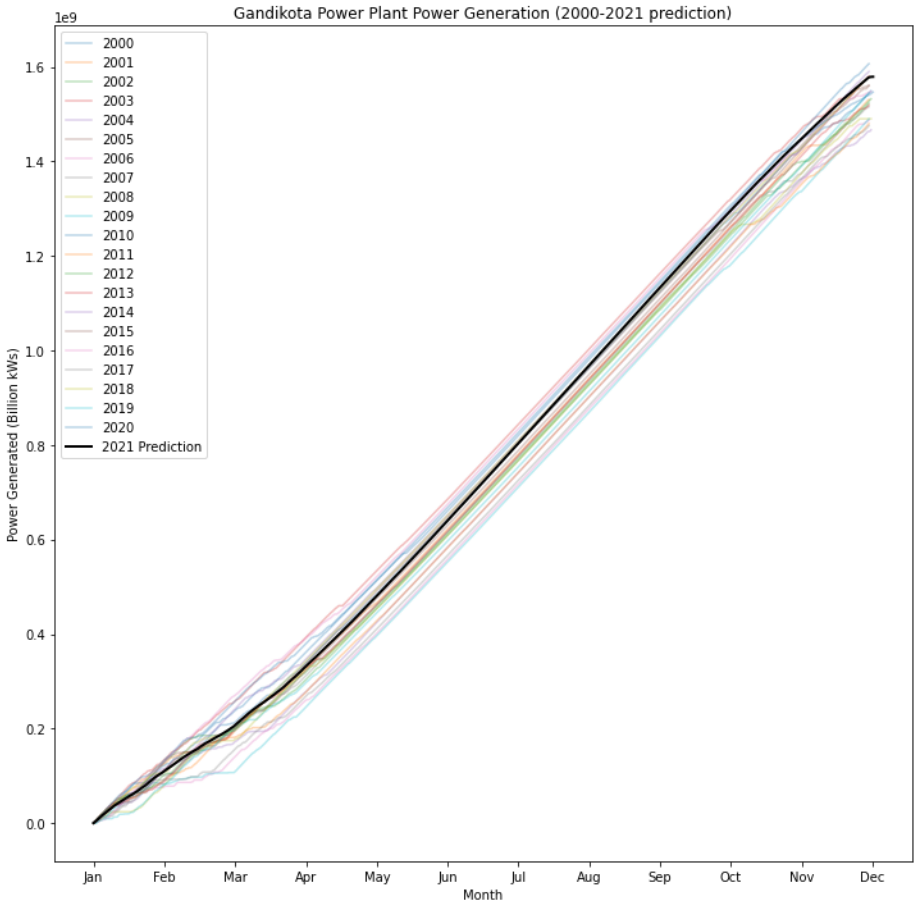
Note at this point I decided to only move forward with the Gandikota power plant for two reasons: 1. I already proved that these plants faced a similar problem - low NOCT values which is leading to inefficient array generation - which means the solution would be the same. 2. I didn't want to pay for the Nashik data as well.

With this new data, I used the model to calculate what the yearly power generation was and then calculated a best fit line to see the trend of the plant's power generation. With that trend I could make a good prediction for what the 2021 generation should look like.



Using this prediction I worked backwards and applied the model to predict the power output for each day in 2021, then compared it to the cumulative sum of the power output for the previous 20 years.

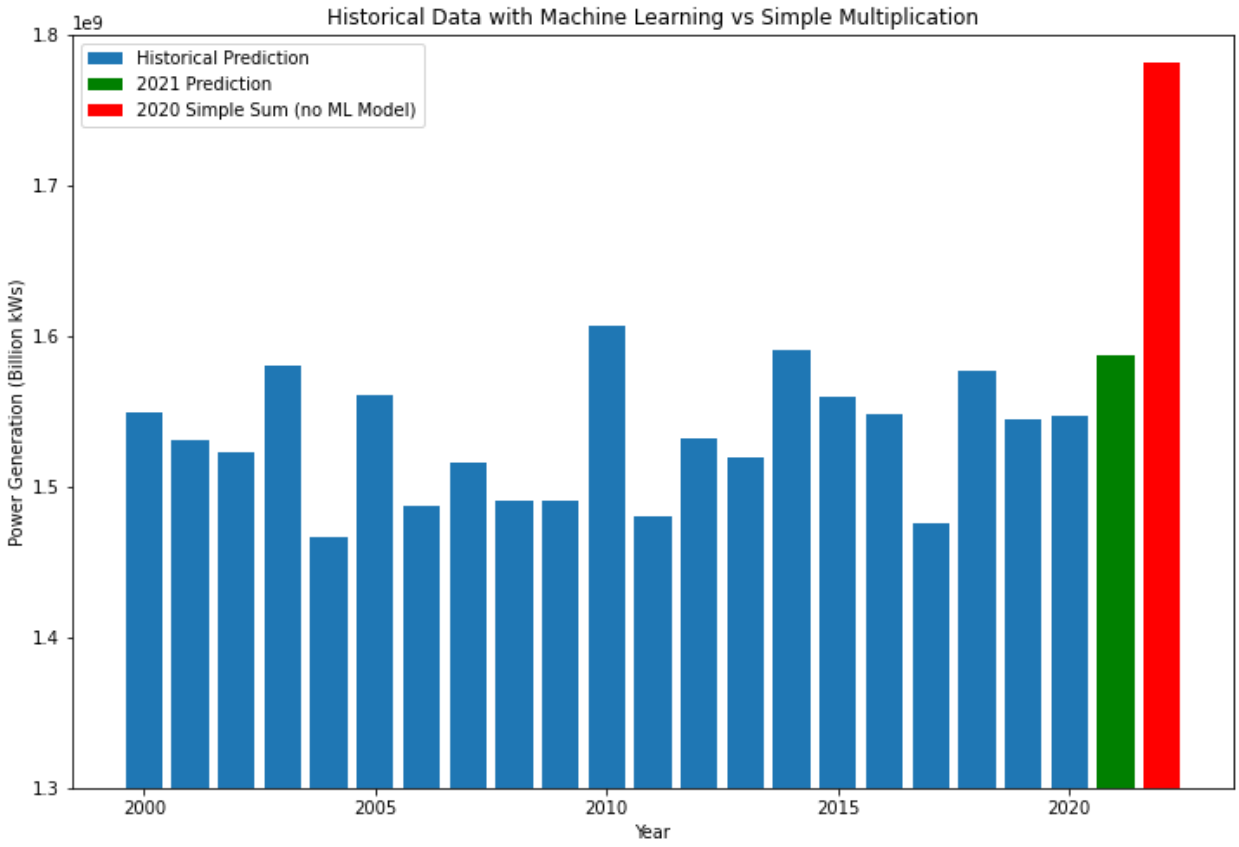




The 2021 prediction fits right in with the rest of the data, it's a little on the high end for total power but that makes sense given that we're following the positive trend line. Now we finally have our output prediction, according to this model the gandikota power plant will produce 1.58 Billion kW/hrs in 2021.

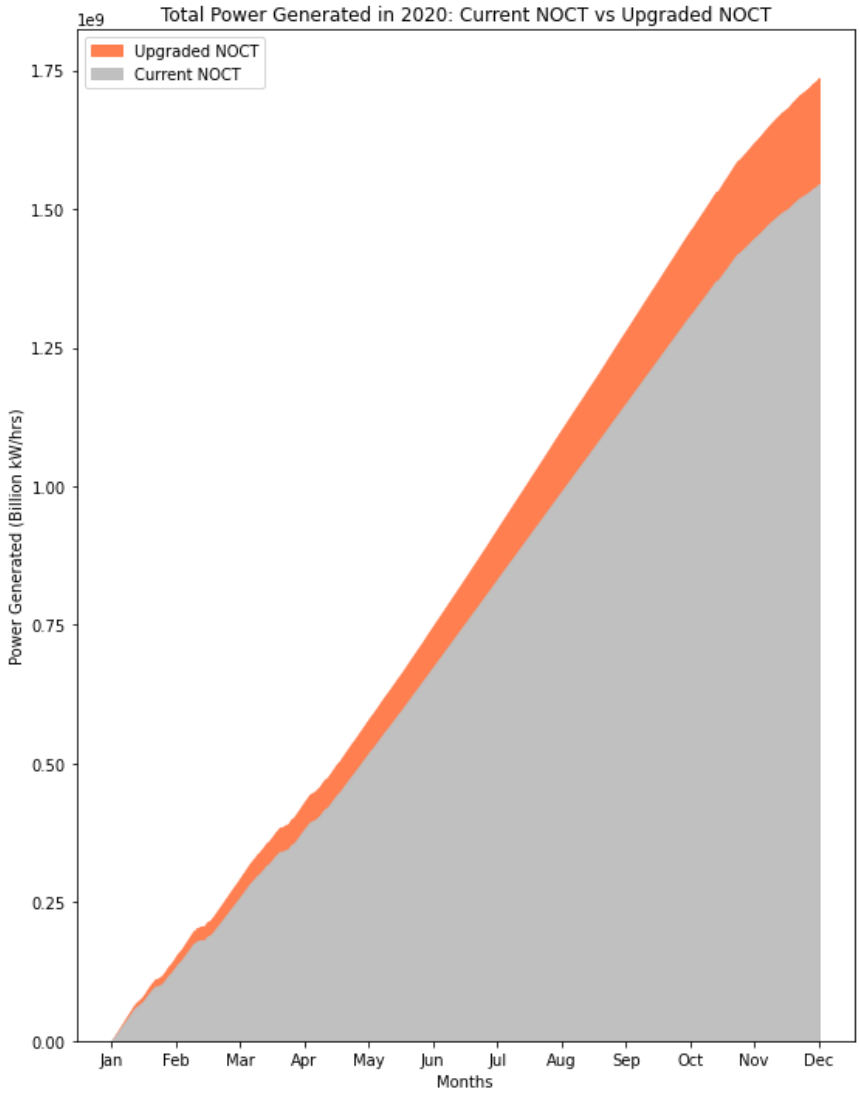
Conclusion

After all this analysis and coding, one might think “is this all worth the effort?” Is it really important to use machine learning when making complex predictions like this. The answer is yes, but let me show you why. Let’s say that instead of creating a model we decided to just multiply the output in the original 34 days by (365/34) to get a yearly prediction, how far off would that be? If we took a simple sum like that - the predicted output would be off by over 200,000,000 kW/hrs.



Machine learning is not just interesting in theory but very useful and important for problems like this.

One more thing I wanted to figure out was how much more the power plant would generate if it upgraded their solar arrays to have a NOCT value of 48 (industry average in 2021). Turns out, making that upgrade would be very beneficial for the power plant.



This upgrade would allow the plant to generate 191 Million kW/hrs of power - which translates to $6.3 Million in additional revenue. With smart investment strategies that revenue could have gone on to pay more salaries, improve the plant's other important facilities, or expand the power plant to produce even more power. Upgrading an entire power plant is no easy task, but I think with what we now know about solar efficiency and the potential it has - upgrading seems like the best course of action.