Predicting Solar Array Output Using Weather Sensor Data

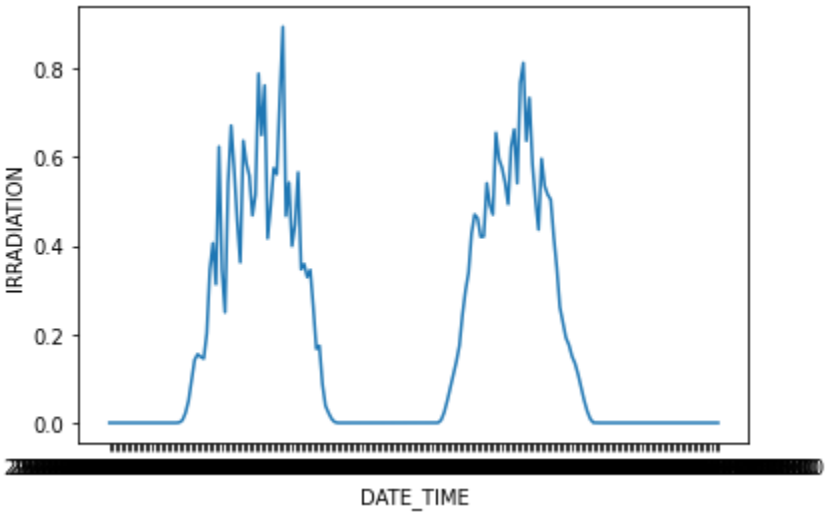
Maximilian Piech

1. Abstract

With the recent push for renewable energy sources, solar energy is one that is very promising due to rapidly falling costs and increasing availability. In this paper I will explore how to predict the power output of a solar array based on weather data collected from sensors throughout each day. By accurately predicting the output of a solar array, a grid operator could optimize the timing and sources of supplemental power and further minimize dependency on non-renewable sources. A linear regression machine learning model will be used to create a correlation between weather data such as irradiation and power output. The best model, the Gradient Boosting Regressor, produced a mean average error of just under 300w compared to over 1000w for the least accurate models for a 1200w array.

2. Introduction

Predicting solar output is a relatively new area of interest. Currently, the grid is powered nearly entirely by fossil fuels, and grid operators decide when to fire up different plants to meet demand. Fortunately, today solar power is becoming increasingly common due to its competitive cost. However, solar output cannot be ramped up on demand like gas-fired peaker turbines, so it is useful to predict the power output of solar panels in the future using weather data collected from sensors that can be forecasted ahead of time. With these predictions, grid operators can decide whether other sources of energy need to be used. I utilize the power of regression and machine learning to predict solar array output using weather forecast data.



3. Background

Significant work has been done to improve the efficiency of solar panels. Today, commodity solar panels usually have efficiencies around 15 to 17 percent[1]. The highest end off-the-shelf solar panels have efficiencies around 23 percent[2]. Solar panel efficiency varies with construction. Monocrystalline solar panels are more efficient, but also more expensive than polycrystalline solar panels. Solar panels with front-facing busbars have a lower efficiency than those with rear-facing busbars.

On the software side, other solar output models have been built that take into account latitude, longitude, altitude, month, humidity, wind speed, visibility, pressure, and cloud ceiling. These are important because these variables can be used to more accurately predict the weather in the future. Latitude and longitude can be used to predict the output of solar panels across the world because solar irradiation varies widely.

4. Dataset

I am using a dataset of power output from a 1200w solar farm over the course of a month, all of which is numerical. I used a 3:1 split between my training and testing data. Data preprocessing was not necessary because there were no null values.

The dataset included:

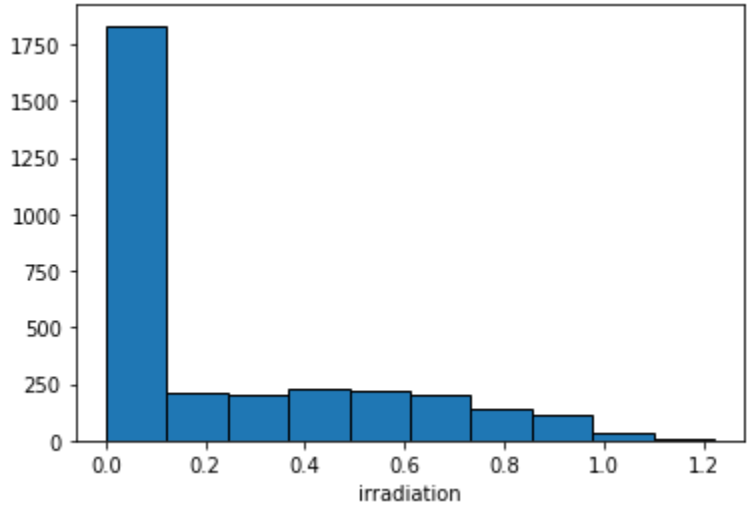
- Date and time
- DC Power
- AC Power
- Daily Yield
- Total Yield
- Ambient Temperature
- Module Temperature
- Irradiation (strength of sunlight)

DC Power is the power leaving the solar panels and entering the inverters. AC Power is the power leaving the inverters and entering the grid. The dataset includes power output data and weather data for two separate solar farms. I will only be working with data from the first solar farm. The chart below is a histogram of

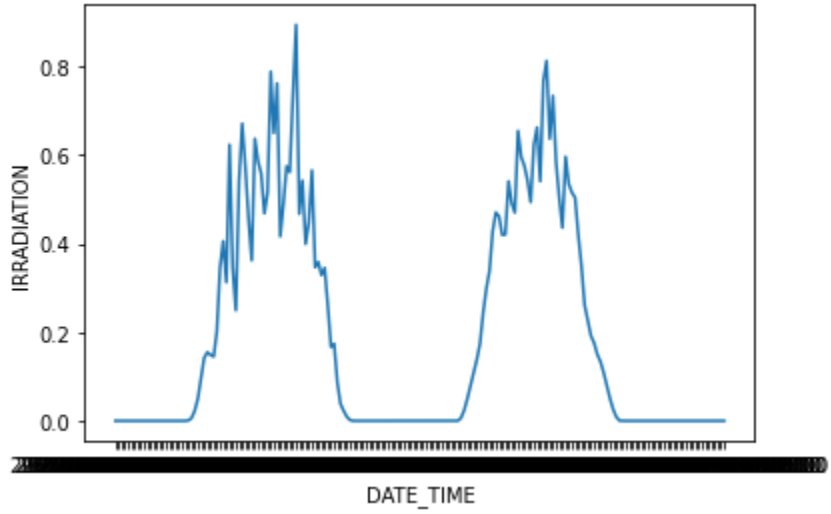
[1] https://secure.sharp.eu/cps/rde/xbcr/documents/documents/Marketing/Datasheet/1711_NDAH325-330W_Poly_Datasheet_EN.pdf

[2] https://www.enfsolar.com/pv/cell-datasheet/1740

irradiation from the weather data with the number of datapoints on the X axis.

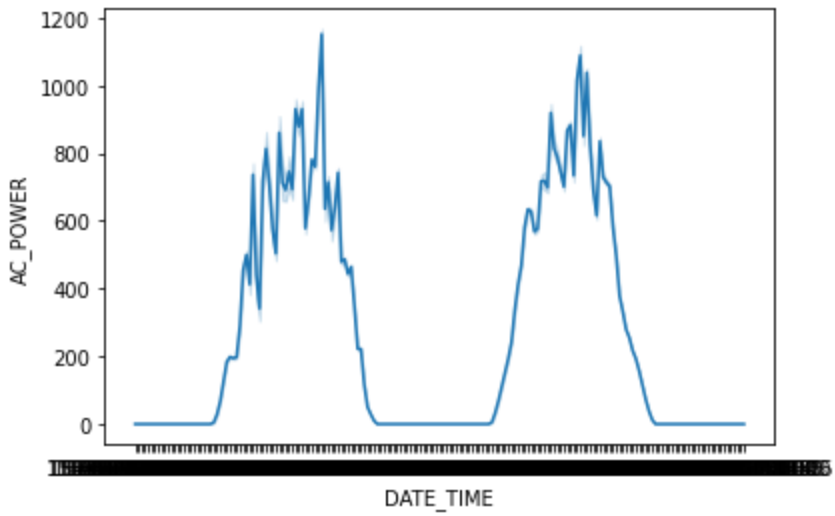


There are many data points with a value of near 0. This dataset includes a reading taken every 15 minutes, so many of the readings were taken at night.



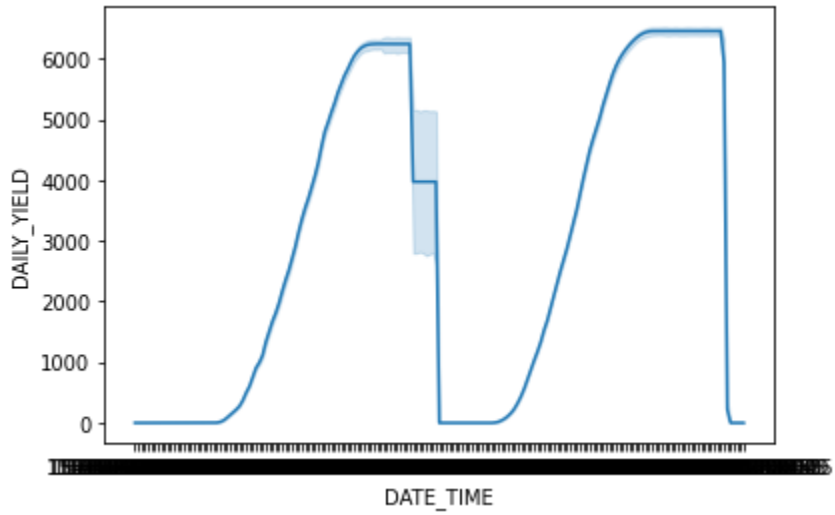
Two day period

In this two day period, it can be seen that there are two high areas of irradiation during sunny hours. During this time, the AC Output of this array follows nearly an identical pattern.



Two day period

This graph shows the total power generated over the course of each day.



Two day period

5. Methodology/Models

I used multiple SKLearn regression models. I split the dataset into 75 percent training data and 25 percent testing data. I used these columns as features:

- Date/Time
- Ambient Temperature
- Module Temperature
- Irradiation
- AC Power

The following sections contain the models that I tried:

5.1 Linear Regression

Linear Regression finds a line of best fit between the input data and output data, and runs test input data through the equation of the line of best fit. The mean absolute error was 466w.

5.2 k-Nearest Neighbors

k-Nearest Neighbors categorizes a datapoint based on the categories of the data points closest to it.. I got the best results with 12 for k and 1 for p. The mean absolute error was 1056w.

5.3 SGD Regressor

A linear model that minimizes regularized empirical loss with SGD. I got best results with ‘squared error’ for loss, 0.001 for alpha, and ‘optimal’ for learning rate. The mean absolute error was 405w.

5.4 Decision Tree Regressor

Decision trees are easy to visualize. I got best results with ‘absolute error’ for criterion, ‘random’ for splitter, and 0.001 for ccp alpha. The mean absolute error was 370w.

5.5 MLP Regressor

This is a neural network. I got best results with [10, 10, 10, 10] for hidden layer size, 0.00001 for alpha, ‘adaptive’ for learning rate, and 500 for max iterations. The mean absolute error was 1026w.

5.6 AdaBoost Regressor

Uses an original regressor and then additional adjusted regressors. I got the best results with 75 for ‘n-estimator’, 1 for learning rate, and ‘square’ for loss. The mean absolute error was 404w.

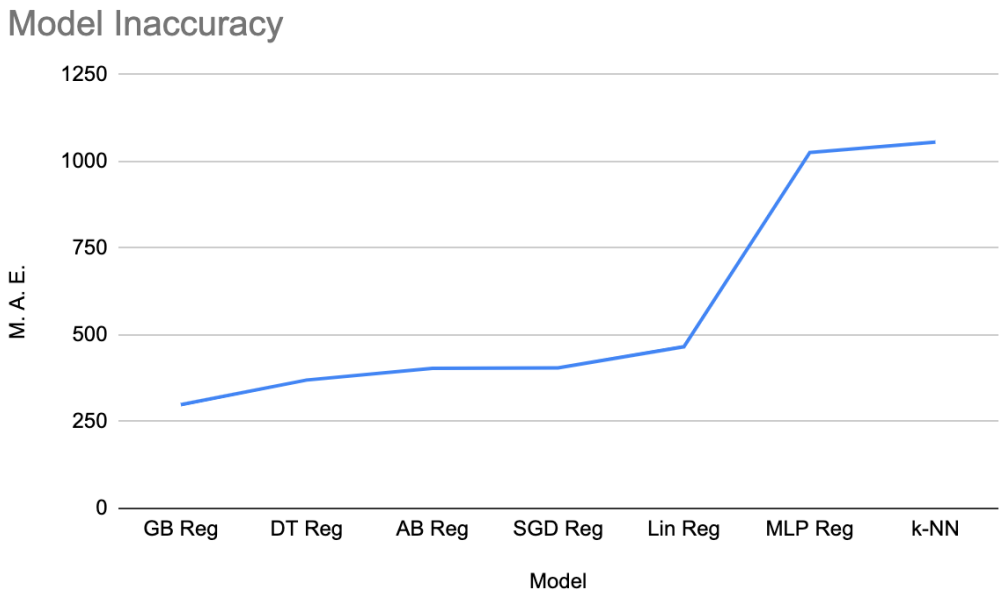
5.7 Gradient Boosting Regressor

Additive model in forward stage-wise fashion. I got the best results with 200 for ‘n-estimator’, 0.1 for learning rate, and ‘squared error’ for loss. The mean absolute error was 299w.

6. Results and Discussion

The linear regression model had a mean absolute error of about 466w. The k-nearest neighbors regression model had a significantly worse mean absolute error of about 1056w. The SGD Regressor lowered the mean absolute error to about 405w. The Decision Tree Regressor lowered the mean absolute error to about 370w. The MLP Regressor had poor performance with a mean absolute error of about 1026w. The AdaBoost Regressor had a mean absolute error of about 404w.

All of the regressors had relatively similar results with mean absolute errors between about 300w and 500w. The MLP Regressor and k-Nearest Neighbors regressor were both far worse with mean absolute errors of more than 1000w because they are really meant for classification, not regression.



7. Conclusion

I built different regression and classification models to predict the output of an array of solar panels for the coming day. This can be used to decide when and how much more energy from other sources is needed to meet demand. I tried 7 different models from SKLearn. The Gradient Boosting Regressor was the most accurate. As a next step, I could improve these results by building a more complex linear regression model that would include the latitude, longitude, time of the year, and location of the solar power plant. This would allow the model to be applicable anywhere in the world.

8. Acknowledgements

I would like to thank my project mentor Chris Mauck for providing support in the exploration of this subject.

9. References

Kannal, Ani. “Solar Power Generation Data.” Kaggle, https://www.kaggle.com/datasets/anikannal/solar-power-generation-data. Accessed 14 Sept. 2022.

"ND-AH325, ND-AH330." Sharp, https://secure.sharp.eu/cps/rde/xbcr/documents/documents/Marketing/Datasheet/1711_NDAH325-330W_Poly_Datasheet_EN.pdf. Accessed 14 Sept. 2022.

"SunPower | Maxeon Gen III | Solar Cell Datasheet | ENF Solar Cell Directory." ENF List of Solar Companies and Products - Including Solar Panel and Inverter PV Manufacturers, https://www.enfsolar.com/pv/cell-datasheet/1740. Accessed 14 Sept. 2022.