Machine Learning: Solar Energy Output

This manuscript (permalink) was automatically generated from Kylierb2/SolarEnergy@d93ce0a on December 7, 2020.

Authors

- **Kylie Burkett**
  · Kylierb2 · burkett_kylie
  Department of Civil & Environmental Engineering; University of Illinois at Urbana-Champaign

- **Allisa Hastie**
  · ahastie2 · AllisaG_Hastie
  Department of Civil & Environmental Engineering; University of Illinois at Urbana-Champaign

Abstract

1. Introduction

Solar generation is an up and coming alternative energy resource. On the campus of the University of Illinois at Urbana-Champaign researchers have created solar farms to try and reduce the University's carbon dioxide emissions. Solar farm 1.0 is the first UIUC solar farm; it has been operational since December of 2015. It is 20.8 acres of land and produces around 7,200 megawatt-hours of electricity annually. Surprisingly, this is only about 2% of the total megawatt-hours the university requires annually. In order to produce more the university recently published its plans to start work on "Solar Farm 2.0". Solar farm 2.0 will be around 54 acres and produce as much as 20 thousand megawatt-hours annually; approximately 6% of university demand annually. Solar generation is important to "balancing the grid" and the more one is able to predict this output the more efficient energy usage will be. Knowing when solar energy will "run out" is largely a part of being able to utilize the maximum amount of solar energy generation. The goal of this project is to use machine learning techniques learned in class and publicly available data in order to predict the daily energy output from the UIUC solar farms.

2. Literature Review

In order to better understand the current research on predicting solar generation, a literature review was conducted. Our goal in performing this review is to gain a more comprehensive understanding of modern techniques in machine learning particularly as it pertains to our application. We chose to exmine two papers, the first is a review paper that covers this field of study in general, the second examines work done by a specific team to predict solar output from publically available weather forecasts.

2.1 Machine learning methods for solar radiation forecasting: A Review - Cyril Voyant Et Al.

The journal paper explores other journals that have conducted research on predicting solar generation. This mass review found that most people who are doing research on solar generation are using artificial neural networks in order to predict outputs. This method is effective, however, the authors found that regression tree methods are actually performing with better results. Using his information the model for this project will use both neural networks and regression trees to see how they perform against each other.

2.2 Predicting Solar Generation from Weather Forecasts Using Machine Learning - Sharma, N.; Sharma, P.; Irwin, D.; and Shenoy, P

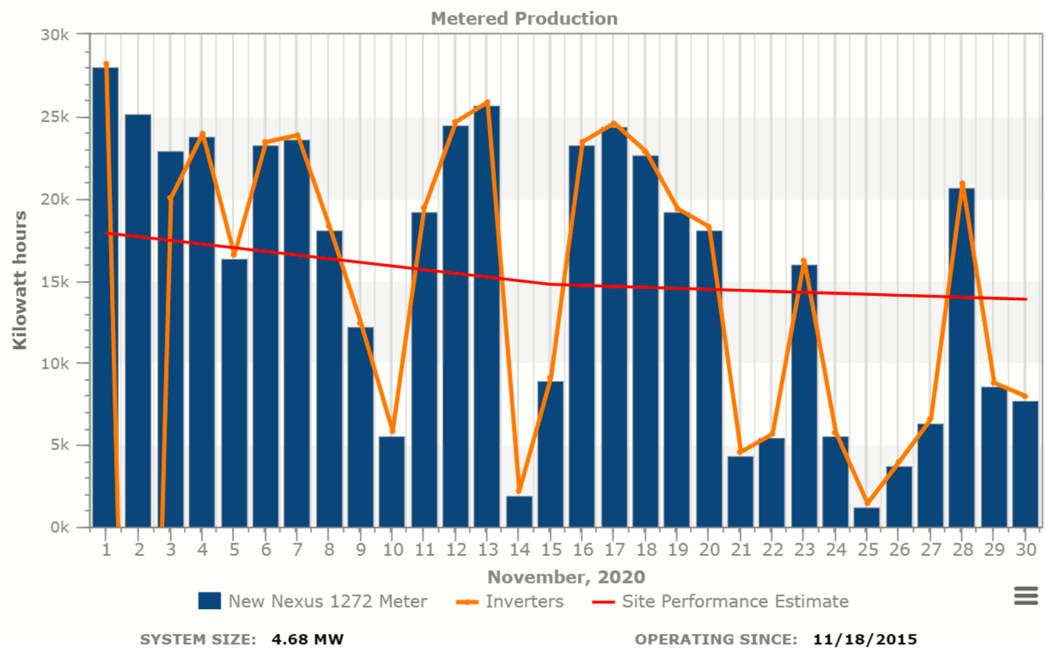
The next journal was a case study from a research group using national weather forecast data. The biggest issue they found was uncontrollable variability in weather patterns. This is expected in this type of research because the weather is a natural phenomenon that one can only predict toa certain extent. This group exclusively used support vector machines to resample the datasets. Support vector machines take in a large amount of data and then resample them into different smaller datasets in order to easily compare and analyze them. The most interesting aspect of this research was the use of datasets although they did not correlate well with test data. This is something that the group will keep in mind when making their own models.

3. Data Collection and Analysis

This section will discuss the different types of data the group collected, cleaning said data, and the analyses performed on the datasets.

3.1 Raw Data Collection

3.1.1 Solar Data



Solar data were taken from the UIUC Solar Dashboard on a daily timestep.

3.1.2 Daily Weather Observations

Illinois Climate Network (ICN)
January 2015
Monthly Summary For Champaign

DAY	MAX WIND SPEED MPH	AVG WIND SPEED MPH	DIR AVG WIND o	TOTAL SOLAR RAD MJ/M*M	MAX AIR TEMP oF	MIN AIR TEMP oF	AVG AIR TEMP oF	MAX REL HUM %	MIN REL HUM %	AVG DEW POINT oF	TOTAL PRECIP IN	TOTAL EVAP IN	MAX 4" SOIL TEMP UNDER SOD oF	MIN 4" SOIL TEMP UNDER SOD oF	AVG 4" SOIL TEMP UNDER SOD oF	MAX 8" SOIL TEMP UNDER SOD oF	MIN 8" SOIL TEMP UNDER SOD oF	AVG 8" SOIL TEMP UNDER SOD oF	MAX 4" SOIL TEMP UNDER BARE SOIL oF	MIN 4" SOIL TEMP UNDER BARE SOIL oF	AVG 4" SOIL TEMP UNDER BARE SOIL oF
1	22.4	6.9	244.9	8.9	33.0	16.0	23.8	81.5	44.2	14.5	0.00	0.03	36.4	35.9	36.1	35.5	34.4	34.9	31.2	28.6	30.1
2	8.3	2.6	204.2	7.3	39.3	21.3	30.4	95.6	53.2	23.6	0.03	0.03	37.9	36.0	37.6	35.3	34.0	34.6	31.4	29.3	30.5
3	9.2	2.8	186.4	1.1	40.0	32.9	35.9	99.3	95.5	35.5	0.72	0.00	38.1	37.7	37.9	35.3	33.9	34.5	31.8	31.4	31.7
4	28.3	9.1	286.3	2.2	34.8	9.0	24.0	99.5	70.1	20.8	0.02	0.01	M	M	M	35.3	34.1	34.6	31.9	31.7	31.8
5	18.0	5.1	239.1	6.5	13.4	2.1	8.6	91.8	61.8	2.2	0.26	0.01	M	M	M	35.2	34.3	34.8	32.3	31.9	32.1
6	20.6	5.9	211.0	8.2	13.9	8.8	11.8	92.0	75.1	8.4	0.00	0.02	M	M	M	35.4	34.2	34.7	32.3	32.3	32.3
7	21.7	8.0	313.0	10.4	13.0	-6.1	2.2	88.6	58.8	-5.2	0.02	0.02	35.0	34.5	34.8	35.1	34.1	34.6	32.4	32.3	32.3
8	34.6	8.9	223.1	4.3	24.4	-7.2	6.6	83.1	59.7	-0.4	0.01	0.02	34.5	34.1	34.3	35.0	33.6	34.4	32.4	32.2	32.3
9	25.0	8.9	287.2	10.4	16.4	1.2	7.6	87.5	57.2	0.0	0.00	0.02	34.7	34.4	34.6	34.6	33.6	34.1	32.3	32.2	32.3
10	18.4	5.8	186.7	11.8	24.9	-6.3	12.9	89.6	53.2	4.3	0.00	0.03	34.6	34.0	34.3	34.4	33.3	33.9	32.2	32.0	32.1
11	9.2	3.4	191.2	2.3	32.9	23.4	29.8	98.8	60.9	23.6	0.22	0.02	35.0	34.6	34.8	34.5	33.3	33.8	32.3	32.1	32.2
12	17.4	6.8	165.3	8.3	24.0	11.7	22.8	84.6	72.0	19.3	0.00	0.02	35.2	35.1	35.1	34.3	33.5	33.9	32.3	32.3	32.3
13	13.7	5.1	45.9	9.9	17.9	5.2	11.3	87.8	71.7	6.6	0.00	0.02	35.2	34.9	35.1	34.5	33.6	34.0	32.3	32.3	32.3
14	8.2	1.6	186.3	5.5	19.7	-0.1	11.1	93.1	70.3	6.9	0.00	0.01	34.9	34.6	34.8	34.4	33.3	33.9	32.4	32.3	32.3
15	13.6	4.5	209.6	10.9	37.5	9.1	24.8	95.1	65.2	19.8	0.01	0.03	35.0	34.5	34.8	34.2	33.1	33.6	32.4	32.2	32.3
16	7.8	2.8	209.9	11.4	39.5	21.6	30.3	95.5	67.6	26.9	0.00	0.04	35.1	34.8	35.0	34.3	33.1	33.6	32.4	32.3	32.4
17	18.2	6.8	207.5	8.6	48.4	31.4	38.3	91.8	67.3	33.1	0.00	0.04	35.2	35.0	35.1	34.3	33.1	33.6	32.5	32.3	32.4
18	23.7	7.1	262.1	10.8	44.2	26.1	36.0	96.5	64.0	31.0	0.00	0.04	35.2	35.2	35.2	34.8	33.0	33.7	32.4	32.2	32.3
19	7.7	1.7	120.2	10.6	45.7	24.2	35.3	96.7	66.4	31.1	0.01	0.04	35.3	35.2	35.2	34.3	33.2	33.8	32.5	32.3	32.4
20	14.9	3.7	260.3	9.3	45.0	62.7	36.5	81.9	30.3	33.0	0.00	0.04	35.3	35.2	35.3	34.7	33.3	33.9	35.8	32.4	33.7
21	20.7	6.1	262.1	4.3	37.3	28.6	34.1	98.8	77.0	31.3	0.00	0.02	35.7	35.3	35.4	35.1	33.6	34.1	35.9	33.4	34.5
22	7.3	2.1	174.7	3.9	37.4	31.7	33.9	97.2	68.1	29.0	0.00	0.02	39.1	35.7	37.1	36.0	33.9	35.0	38.2	34.2	35.9

We utilized daily wether observations from the Illinois State Water Survey.

3.1.3 Monthly Solar Radiation

We disaggregated average monthly solar radiation values from the Illinois State Water Survey.

3.2 Exploratory Analysis

This exploratory data analysis will compile weather and solar data into a single cohesive data frame and examine the data themselves to better understand the characteristics of the features and labels and better understand their relationship with each other. Towards these goals, we will use a variety of statistical methods and graphical and tabular representations to identify trends and relationships.

3.2.1 Clean and compile weather data

The weather data that was obtained was in a .txt format, which can be difficult to work with. In order to make this file a CSV to analyze with our other CSV files. First, the .txt file needed to be called into the kaggle workspace. Once we were able to read all the files in, we had to remove blank rows, metadata, and any column titles that were poorly formatted we renamed. Following these actions, we were able to take a closer look into the dataset. In order to properly match up columns, the "Unnamed: 0" column was renamed to date to fit in with our other dataset. Lastly, the dataset's incomplete data frames were dropped, columns without proper names were renamed, and then the smaller data frames were compiled into a larger data frame and the index reset. This allowed us to see a cleaner view and understand what this dataset actually is portraying. Following this, it was clear that the weather and solar data should be combined in order to analyze and explore all aspects of the data.

3.2.2 Clean and combine weather and soalr data

In order to read these two CSV files, they needed to be merged on the date column. Upon further inspection, it was observed that some cells in the data frame had the character string "M" attached to the end of the number. This would construe data and our analysis so we removed the M with an empty character. Following this, the dropping of NaN cells had to be performed to make sure the dataset was as complete as possible. There were also columns that were in as an "object" type. This was converted to be a "float" type. Once the index was reset we were able to set up a dictionary of the average daily solar radiation for each month in champaign. This was the data gathers from the IL State Water Survey. One last final touch was to make sure we had no empty cells and that our data frame was organized by the date. Following these actions, we continued to the exploratory data analysis.

3.2.3 Examine and visualize training data

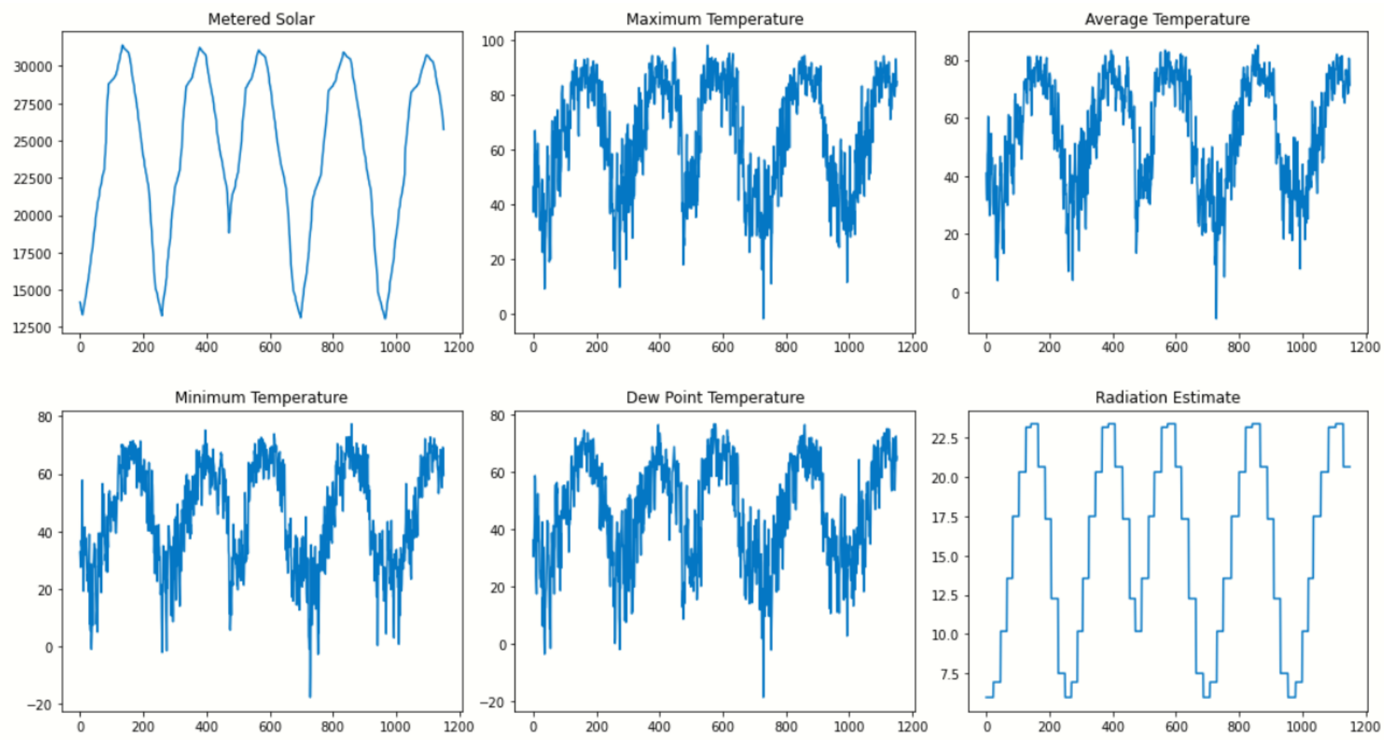
These measurements are all taken from historic weather observations, but when predicting solar output we will only be able to use features that we can predict a day ahead or use the previous day's observed measurements to predict the next day's features. The goal of our examination is to view basic attributes of each column in the data frame, observe any seasonal changes, view the shape of each column distribution, and see what columns are most correlated. Once we find the most correlated we will compare those columns with the solar output.

3.2.3.1 Observations

From the basic statistics of the dataset, it was observed that there are large standard deviations for most columns. This is most likely due to the seasonal variation.

Insert histograms here

From the observed histograms, one will notice the variance in distribution type. The most normally distributed histograms are wind direction (dir_wind) and minimum humidity (min_hum). The more logarithmic distributions are average wind (avg_wind), maximum humidity (max_hum), and total precipitation (tot_precip). One will also notice that certain distributions such as the soil temperature measurements favor extremes in their histograms. Upon analysis of the correlation coefficients between each column in the dataset, it was observed that solar radiation is strongly correlated. This was expected, however, this cannot be predicted ahead of time. Additionally, there was a strong correlation between temperatures and the dew point, both commonly used to predict weather forecasts. Below in figure one will see the time series graphs for the most correlated variables.



Seasonal patterns are clear in our key weather features.

From the time-series graphs, one can observe the seasonal variability in the graphs. The seasonal trends are apparent, yet not perfect. There is still quite a bit of noise in these observations. The next thing to look at will be the scatter plots of the 4 most correlated features with solar output. Once this is completed a regression analysis will be run to see how each feature affects the variation in solar output.

insert correlation graphs here

The results of the linear regressions of these plots were quite interesting. It was observed that the average daily solar radiation in Champaign explains about 93% of the variation in solar output, maximum daily temperature explains 54%, minimum daily temperature explains 53%, average daily temperature explains 56%, and dew temperature explains 45%. Although these values are all around 50% the goal is to use all of them and increase the predictive value as a whole.

3.2.4 Key take-aways

The Illinois State Water Survey's estimate of daily solar output in Champaign is a very good predictor of solar output, daily temperature indicators and dew point are also good indicators of solar output and can be predicted ahead of time. These features will be useful in model development. Day-ahead predictions of radiation and evaporation are not easily obtainable, realistically it would be difficult to use these to develop day-ahead prediction of solar output. Features that are commonly included in weather predictions (temperature and humidity) may be suitable predictors of solar output. Correlation between solar output and minimum humidity is positive and a linear regression fits the data with an R-squared of 0.62. Maximum daily temperature is positively correlated with solar output and when a linear regression model is developed, that model accounts for approximately 33% of variation in the data. The correlation between average daily temperature and solar output is weaker but still positive. The model created for this project will utilize five features: radiation estimation (rad_est), average temperature (avg_temp), dew point (dew), minimum temperature (min_temp), and maximum temperature (max_temp). From the EDA these seem to be the most promising at achieving an accurate prediction model.

4. Model Development

In order to predict daily solar output, two different models will be used: neural network and random forest regression. In order to assess the accuracy of the model, a persistence model will be used for comparison. The means squared error, R-squared, and correlation will determine which model performs best.

4.1 Neural Network

```
x=training_data[["rad_est", "avg_temp", "dew", "min_temp", "max_temp"]]
y=np.ravel(training_data["Site Performance Estimate"])
scaler=MinMaxScaler()
x=scaler.fit_transform(x)

train_ds=tf.data.Dataset.from_tensor_slices((x, y)).batch(batch_size=150)
epochs=500

model=tf.keras.Sequential()

model.add(tf.keras.layers.Dropout(0.2))

model.add(tf.keras.layers.Dense(units=128, input_shape=(5,)))

model.add(tf.keras.layers.Dense(units=256, activation="relu"))

model.add(tf.keras.layers.Dense(units=256, activation="relu"))

model.add(tf.keras.layers.Dense(1))

model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.01,
decay=0.01/epochs), loss="mse" )


history=model.fit(train_ds.shuffle(10), epochs=epochs)
```

This neural network will utilize five features from the training dataset: radiation estimation (rad_est), average temperature (avg_temp), dew point (dew), minimum temperature (min_temp), and maximum temperature (max_temp). The hidden layers of this network are made up of 256 nodes each and both use the Rectified Linear Unit activation (relu). This is a common activation function in neural networks because it avoids the vanishing gradient problem that may occur with other activations. We also include a dropout layer to avoid overfitting the training data. This dropout layer randomly sets node weights to 0 at a rate of 0.2 for each step in the learning process.

4.2 Random Forest Regression

```
x = df_train[["rad_est", "avg_temp",  "dew", "min_temp", "max_temp"]]
y = np.ravel(df_train["Site Performance Estimate"])

scaler=MinMaxScaler()
x = scaler.fit_transform(x)



train_x, test_x, train_y, test_y = train_test_split(x, y, test_size =.2,
random_state = 42)

regr = RandomForestRegressor(n_estimators = 5000,
                              criterion= 'mse',
                              max_depth=100,
                              random_state=0,
                              min_samples_split = 2,
                              min_samples_leaf=1,
                              max_features = 'auto',
                             bootstrap = True)

regr.fit(train_x, train_y)
```

The second model that was developed is the random forest regression model. This model is interesting because unlike a neural network it takes the datasets and creates multiple decision trees in order to find the best-fit prediction. Random forest models are less likely to overfit because of this. For this project, the same features will be used in order to compare the two models more accurately. These features are as follows: radiation estimation (rad_est), average temperature (avg_temp), dew point (dew), minimum temperature (min_temp), and maximum temperature (max_temp). An interesting hyperparameter that was used is the “bootstrap” function. Bootstrapping is a way of resampling. This model implemented bootstrap = True which will use subsets of the data to form decision trees. If bootstrap = false was utilized then the entire dataset would have been used to form trees. Interestingly, better results were obtained using “True” over “False” which is seldom the case in literature reviews.

5. Results

In order to compare our models we developed two figures to observe. The first being a graph of our predicted results and the actual results. This graph displays the overlap of the data and how accurate our model was at predicting the correct values.

insert line graphs here

As you can see the random forest regression produced a more accurate graph. The more defined line in blue is the actual results and the orange more noisy line is the prediction the model produced. For the most part, the two lines follow the same path, which is a good sign. The prediction doesn’t do as well with the extreme values which are due to overfitting in the neural network. In the random tree regression the extreme values are a bit more accurate, but at the very tip are still off. The next graph type we looked at was the correlation betwene our actual data and the model’s predicted data.

insert correlation graphs

A truly correlated graph will be a tight exact line. Both of our models produced a linearly increasing line, which shows correlation. The graph for random forest is much more tight than the neural network, this is because our random forest regression is more accurate. Further inspection of the data allows us to compare the root means squared, mean absolute error, and the R-squared values for each model.

Figure 1: We can visually exmine the fit between our **neural network** prediction and the raw data over our entire training dataseet

Figure 2: We can visually exmine the fit between our **random forest** prediction and the raw data over our entire training dataseet

Model Type	RMSE	MAE	R-squared
Neural Network	1486	1557	0.93
Random Forest	824	634	0.97
Persistence Model	184	135	0.99

As you can see in the table above the random forest regression out performs the neural network. It is noted that the mean absolute error and root mean squared error are both lower than the neural network results. This is the first indication that the model is out performing the neural network. The R-squared value is quite impressive at 0.97. This means that the predicted values of the model explain about 97% of the variance. A high r-squared value indicates that the model values fit the actual values very well, a result one hopes for when predicting future values. Obviously, the more correlated the actual values are to the predicted values the more accurate the model is. However, one cannot only rely on correlation graphs because as seen in the neural network the correlation was still high even though the r-squared was 0.93.

6. Conclusion

The ability to predict solar output will be very important in the future due to the growing use of alternative energy sources. This project aimed to find an accurate and reliable way to predict this output. The results proved that using a random forest regression model may be the best choice. While many groups seem to be utilizing neural networks it may be time for them to switch over to random tree regressions. The findings from this project were that random forest regression out performs neural networks in the realm of solar energy output predictions. Due to the complexity of random forest and the ability to avoid overfitting, this model seems to be promising.

Although the model was fairly accurate there are still some changes we would make for future research. First off, hourly weather prediction datasets would allow us to predict hourly solar radiation outputs. This would lead us to be able to balance the grid more accurately and result in optimal energy use. Current operators have to determine when to shut off solar energy and switch back to conventional energy sources. If they have a more accurate prediction of when solar generation will “run out” then one could optimize the energy used. Next, in order to improve our model we would like to use a cloud cover database and solar radiation estimates. The cloud database would improve our models because you can get a better idea of how much total radiation is actually being caught on the solar panels. If there is too much cloud coverage then one will observe less radiation and more radiation if there is less cloud coverage. Having actual numbers for solar radiation estimates would be effective in improving the model, however this task is very time consuming and difficult to produce. One would need to utilize large equations such as the Penman Monteith equation. Lastly, we would want to spend more time tuning the hyperparameters for the random tree regression model. This would involve setting up the code to view ranges of hyperparameters instead of just one singular variable. The goal of tuning hyperparameters is that we could get the parameter as close to the optimal one as possible, resulting in a more accurate model.

With the ever growing need for alternative energy sources, the ability to predict renewable energy output such as solar will be vital to the future of energy efficiency and production. This project aimed to utilize public data and machine learning techniques to predict the solar output of the UIUC solar farm. With high r-squared values and high correlation to actual values our model has great potential. With future research and fine tuning we expect our model to be more accurate and successful at predicting solar energy outputs.

References