Capstone Project

Solar Energy Forecasting and Performance Modeling

Ruth Caswell Smith

February 25, 2021

Problem Context

Solar energy is radiant light and heat from the Sun that is harnessed using a range of ever-evolving technologies such as solar heating, photovoltaics, solar thermal energy, solar architecture, molten salt power plants and artificial photosynthesis.

It is an essential source of renewable energy, and its technologies are broadly characterized as either passive solar or active solar depending on how they capture and distribute solar energy or convert it into solar power. Active solar techniques include the use of large ground-based photovoltaic systems such as the one used for this project.

The large magnitude of solar energy available makes it a highly appealing source of electricity. The United Nations Development Programme in its 2000 World Energy Assessment found that the annual potential of solar energy was several times larger than the total world energy consumption[1].

Being able to accurately detect (and/or predict) poor performance as well as forecast energy output is an essential tool that can help solar installations operate effectively and return on capital investment.

[1] Wikipedia

Data Source

The data is from a 1 MWatt installation of 5000 panels in the mid-Atlantic region. There are two inverters, each connected to 2500 panels and each producing a maximum of 500 Watts of power.

Data is available for six years at 15-minute increments, with each year being contained in a single CSV file.

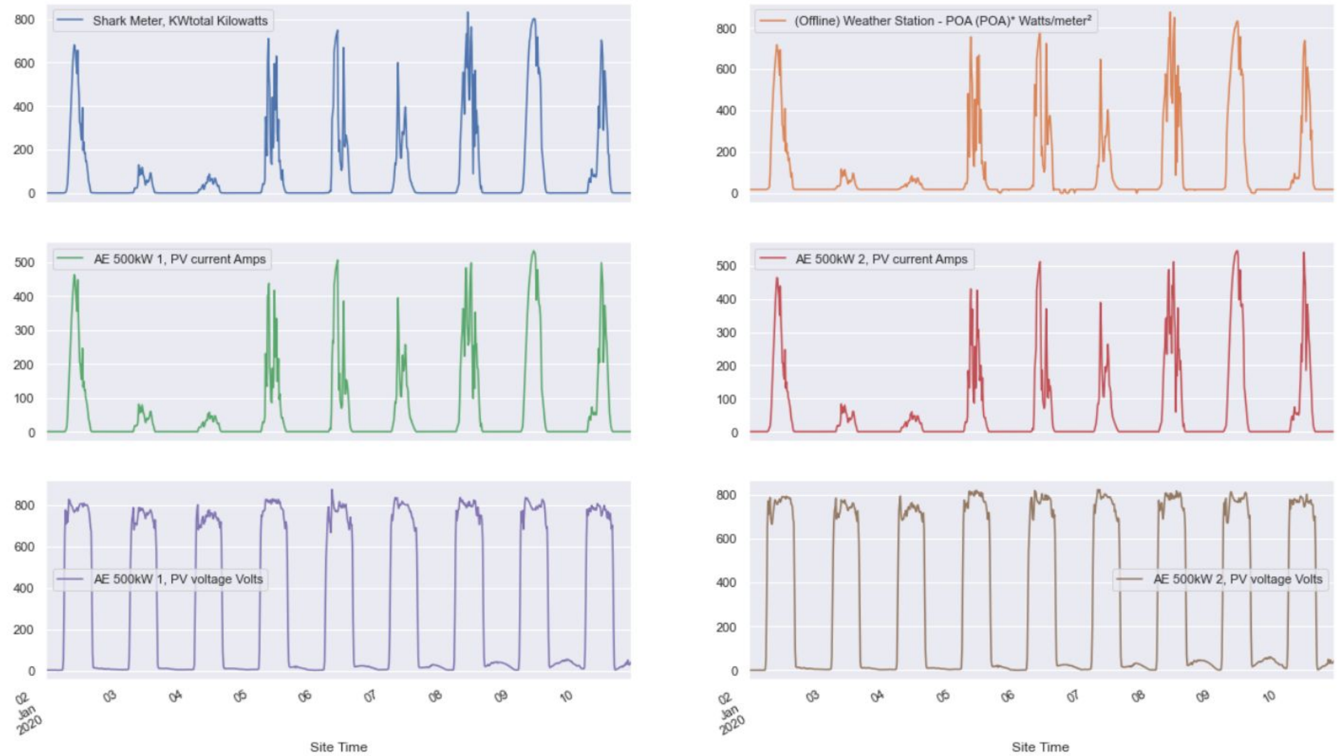
The dataset consists of the following attributes:

- Timestamp
- Shark Meter, KWtotal Kilowatts
- AE 500kW 1, AC Power Kilowatts
- AE 500kW 2, AC Power Kilowatts
- (Offline) Weather Station - POA (POA)* Watts/meter^2
- Weather Station (POA) (SO31456) (POA)* Watts/meter^2
- RECx31 Weather Station, Module Temp Degrees Celsius
- RECx31 Weather Station, Ambient Temp Degrees Celsius
- Weather Station (POA) (SO31456), CabF Degrees Celsius
- (Offline) Weather Station - POA, CabF Degrees Celsius
- AE 500kW 1, PV current Amps
- AE 500kW 2, PV current Amps
- AE 500kW 1, PV voltage Volts
- AE 500kW 2, PV voltage Volts

The output of the Shark Meter (total power generated) should be equivalent to the sum of the power output of the two inverters.

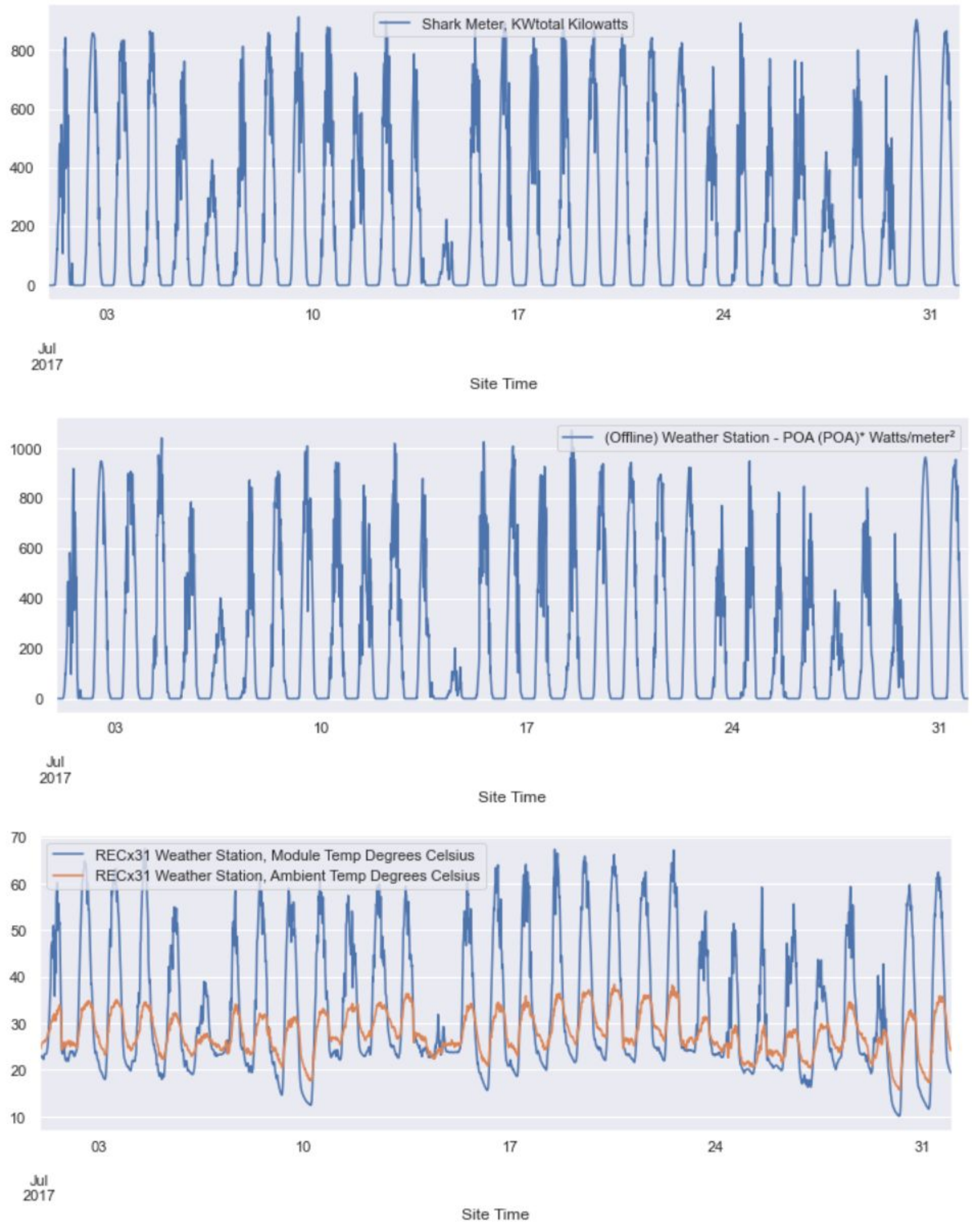
The offline POA measuring irradiance actually provided data throughout the six years, while the second POA measurement only went online recently, and so that column was dropped. The CabF temperature measurements also provided very limited data and in addition had many extreme outliers, so those two fields were dropped as well.

The current times the voltage should be equivalent to the power generated for each inverter. The panels operate such that the maximum voltage is somewhat independent of the irradiance, but the current closely tracks the irradiance, and the power generated therefore also closely tracks the irradiance. For this reason, irradiance and temperature were chosen as an exogenous variable when doing forecasting and classification, but current and voltage were not included so as not to introduce high collinearity among features.



The maximum power is reached consistently at all times of the year, but it is reached for more hours during the summer months. For this reason, seasonality was easier to see by calculating energy output as KWH as opposed to looking at power in KWatts.   Energy output was calculated by integrating power over an hourly or daily time period.

It can be seen in this figure that power output tracks both irradiance and temperature closely.



Missing Values

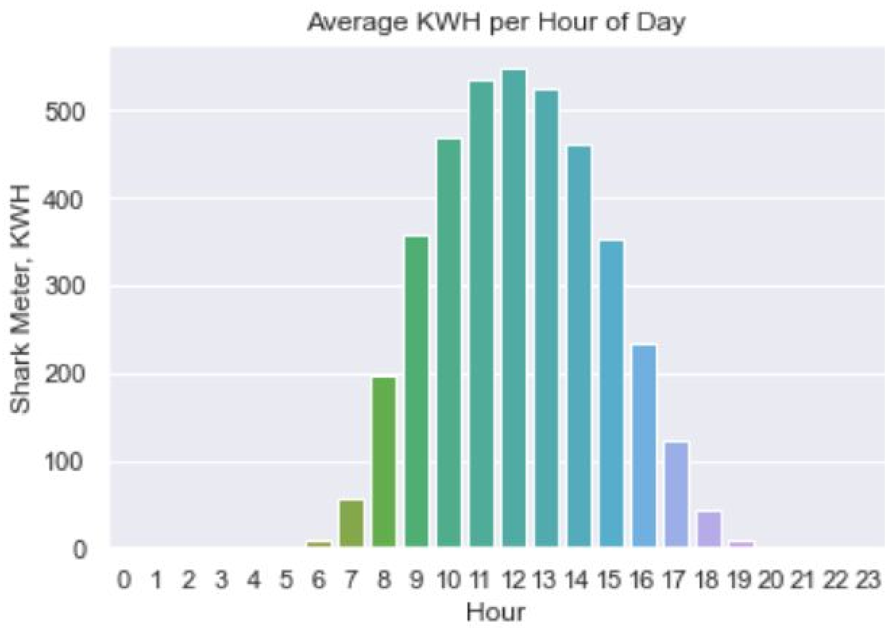
The data contained both a fair amount of missing data. Two approaches were taken to missing data.

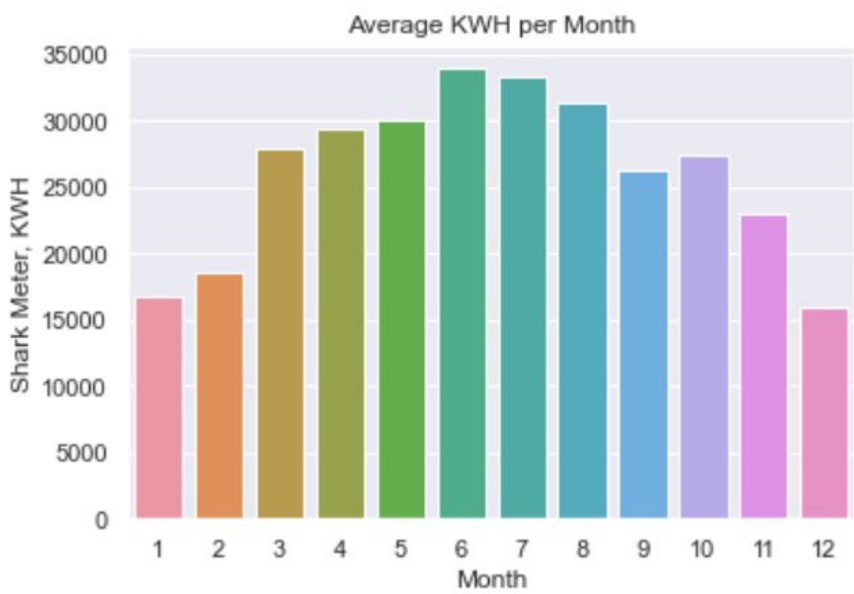
For forecasting, it was ideal to fill in missing data so that our future forecast was not affected by long outages or periods of missing data. For this reason, the missing data in the Shark Meter feature was filled with the sum of the two inverters if it appeared to be malfunctioning. In cases where one inverter was offline for a long period of time, the shark meter was filled with twice the value of the operating inverter. In this case, the Shark Meter provided a somewhat idealized production value to be used for forecasting trend data.

For poor performance detection and classification, missing data in the inverters was replaced with 0 values because it was helpful to know that the inverter was not functioning.

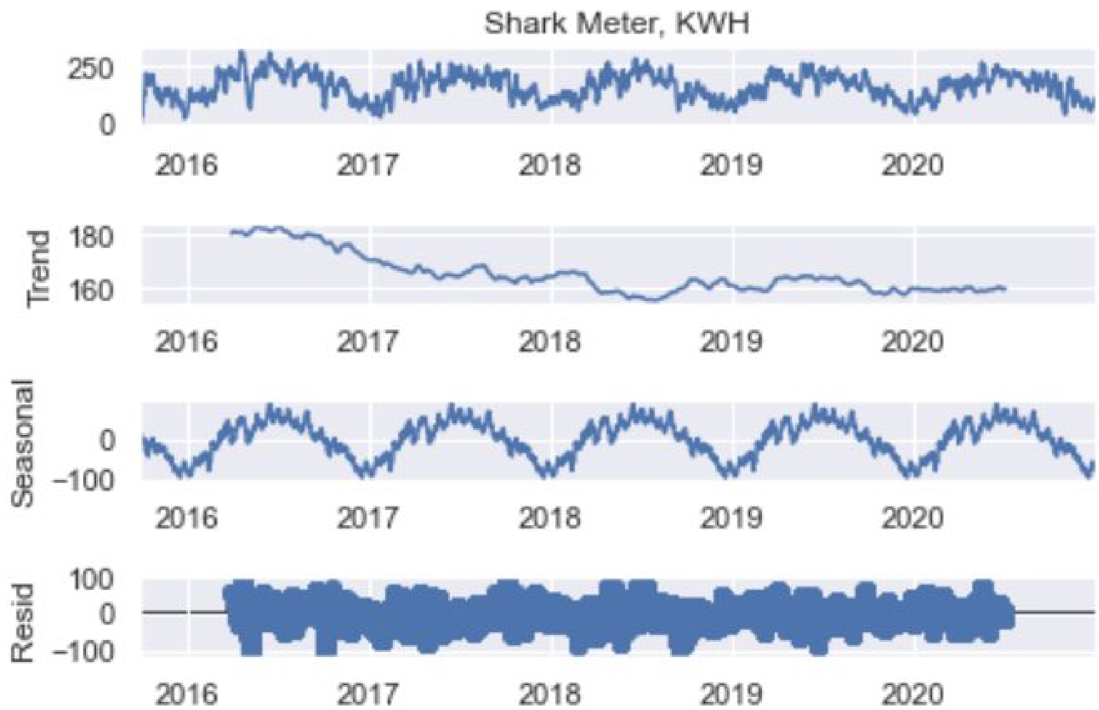
Seasonality

There is strong daily and annual seasonality to the data, as can be seen from the following charts.

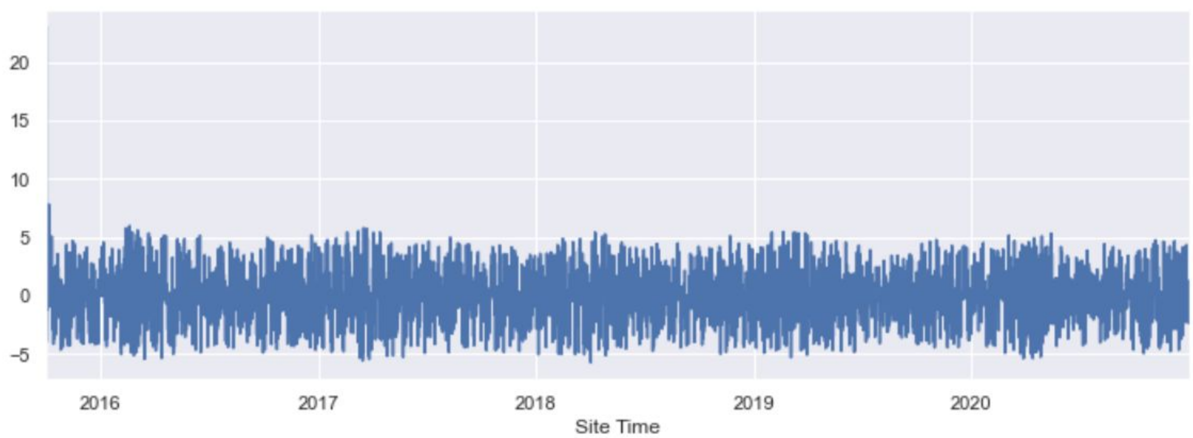




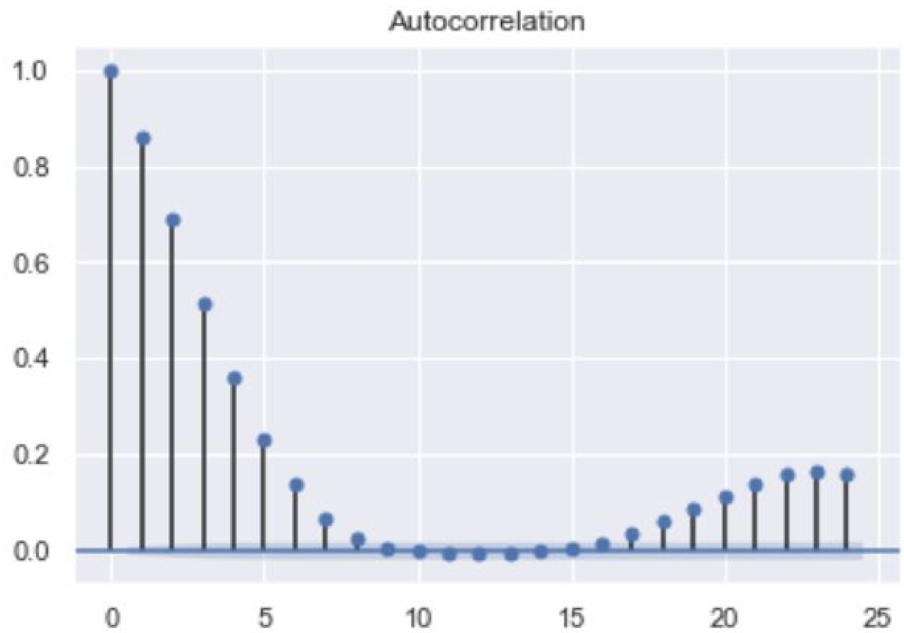
A seasonal decomposition on the hourly energy output showed both strong seasonality and a downward trend over the six years, illustrating the degradation of the panels. A rolling 7-day average was used for this analysis in order to smooth out the very noisy data.

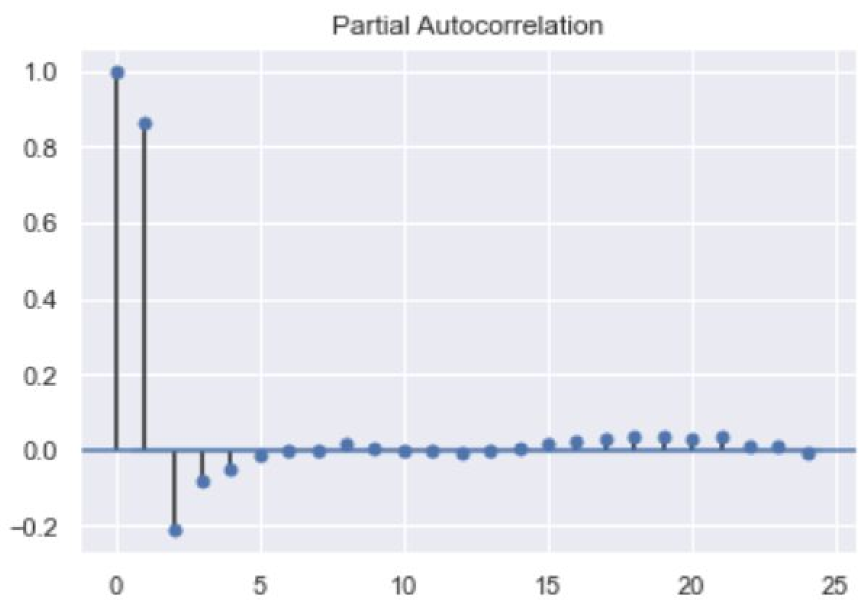


By differencing the energy output by one day, the time series was made stationary and an Augmented Dickey-Fuller test confirmed as much.



The ACF and PACF of this differenced time series indicated that an appropriate model would be an AR(2) model, since the PACF dropped off significantly after two lags and the ACF decreased gradually. Note the auto-correlation with hours of the day on the ACF.





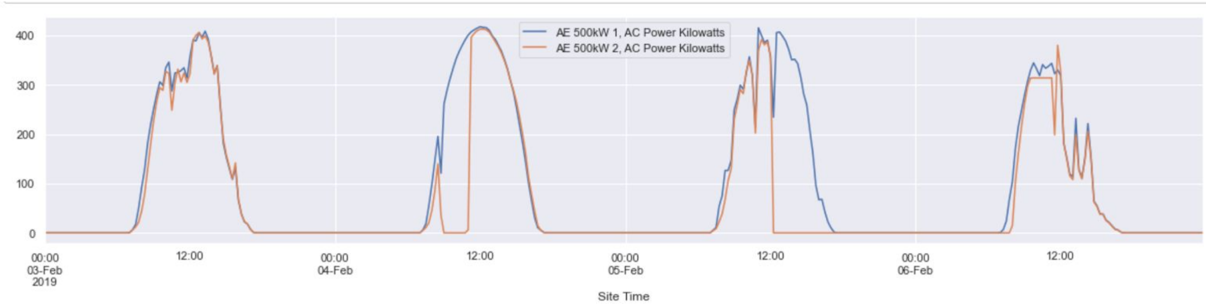
Analysis of Poor Performance

One of the questions of this project was whether poor performance of the inverters could be forecast. However, given that none of the actual data was labeled with outage information, poor performance had to be inferred from other information.

After exploring several different feature variables and the relationship between them, it was determined that the most intuitive way of determining poor performance was to look at whether the inverters were producing the same power. Each inverter should have been behaving identically given they were hooked up to the same number of panels and were experiencing the same irradiance, temperatures, and other environmental conditions.

For this reason, the difference between the inverters was calculated. Then, for each inverter separately, a time series was generated which was the amount that each inverter was lagging the other inverter. This time series was zero if that inverter was generating the same or more power than the other inverter, but nonzero if that inverter was lagging the other inverter. This indicated how poorly the inverter was performing.

Below is an example of several days where the second inverter was lagging the first. For these days, the time series for Inverter1 would be zero for these days, but the time series for the second inverter would be non-zero for portions of the second, third, and fourth day.

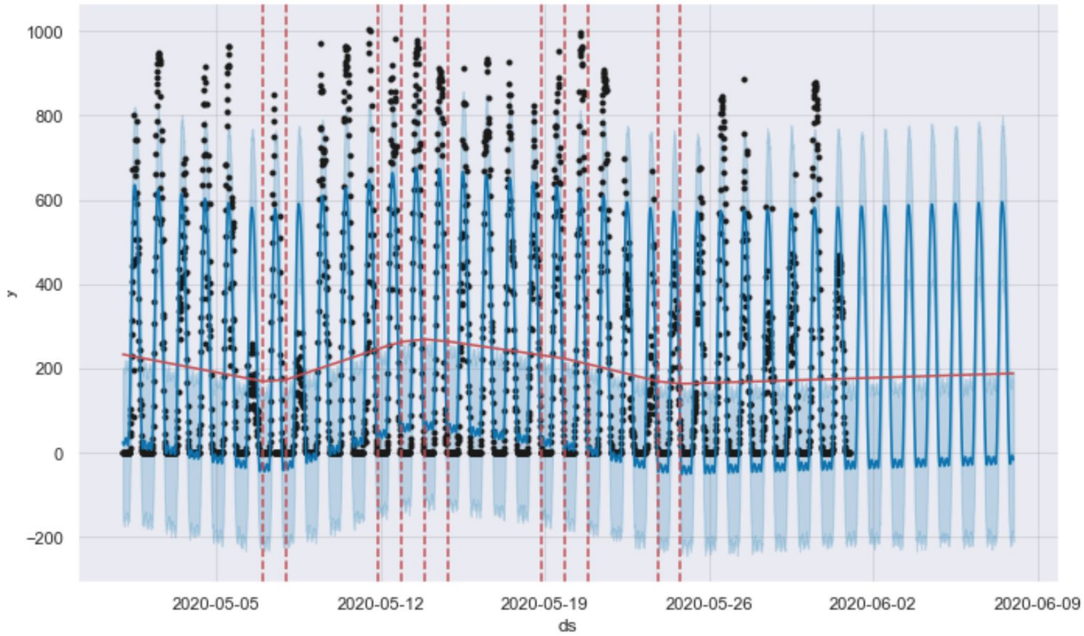


Time-Series Forecasting with Facebook Prophet

Forecasting Energy Output

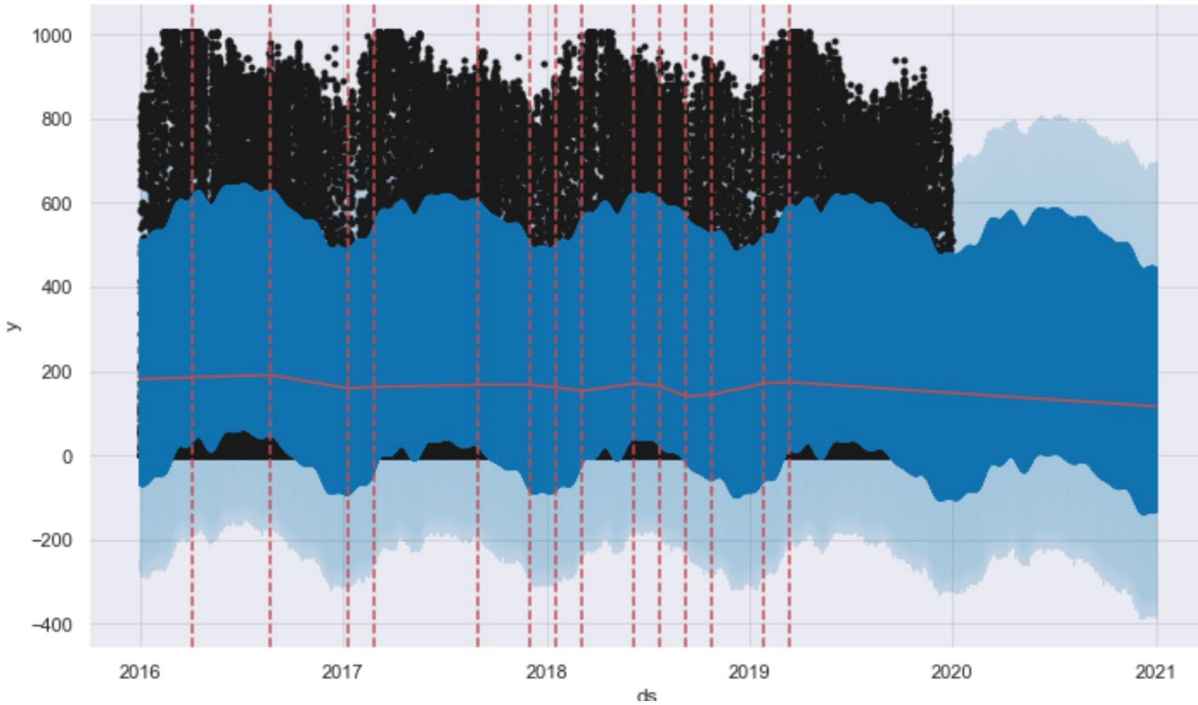
Time series forecasting was conducted using Facebook Prophet. Two types of forecasting were done, a simple forecast of energy output for the entire solar array based on the shark energy historical data (with missing values filled as detailed earlier).

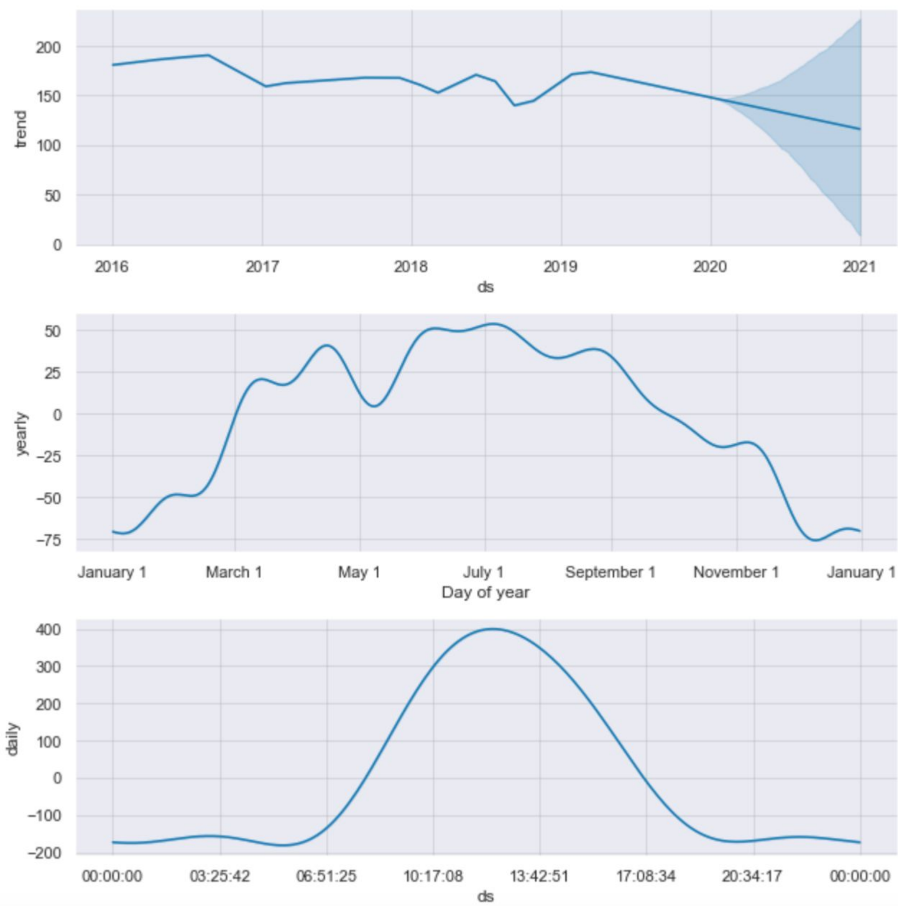
Below is an example of a forecast on 15-minute data based on two weeks of historical data.



In this plot, actual data is black dots, forecast data is blue (with confidence intervals light blue), the trend line is red, and inflection points for the trend line are vertical lines. It is clear that Prophet is averaging out the noise in the data, but it is certainly capturing the daily seasonality.

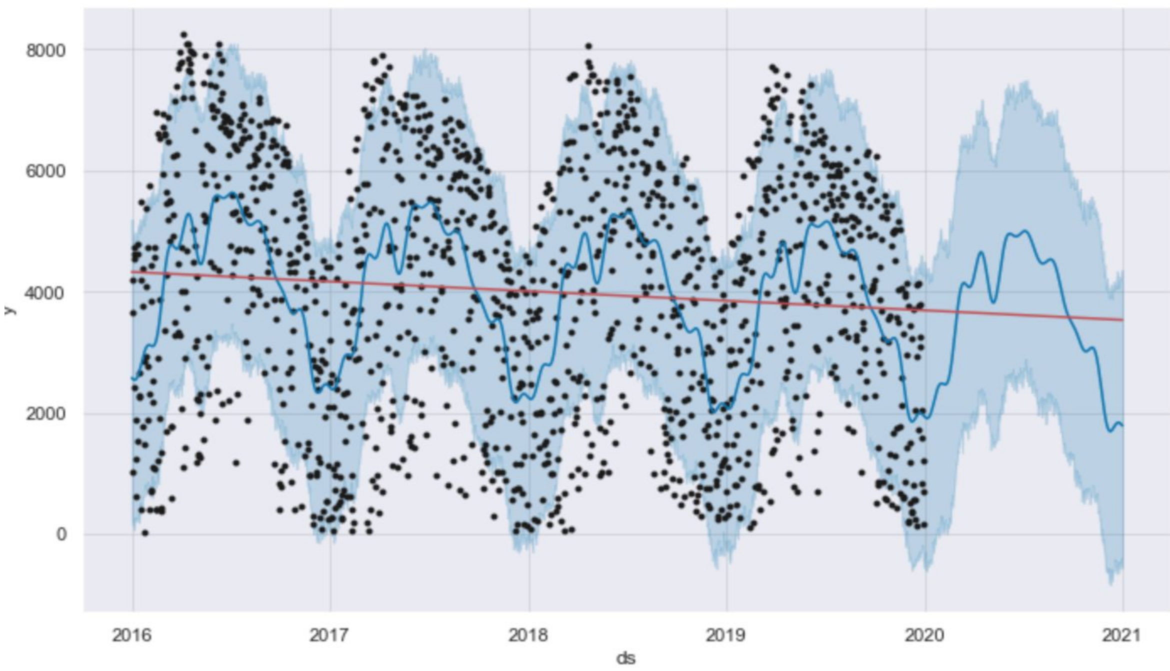
Following this a yearly forecast for 2020 was performed based on historical data. Again, Prophet smoothed the data, but correctly captured the annual seasonality, and predicted an overall downward trend in the energy output. In addition, Prophet was very fast to fit and predict.


y
ds
1000
800
600
400
200
0
−200
−400
2016
2017
2018
2019
2020
2021


trend
ds
yearly
Day of year
January 1
March 1
May 1
July 1
September 1
November 1
January 1
daily
00:00:00
03:25:42
06:51:25
10:17:08
13:42:51
17:08:34
20:34:17
00:00:00

A time series forecast of averaged daily data was also performed. This model was optimized using cross-validation and hyper-parameter tuning. The two hyper-parameters that were tuned were `changepoint_prior_scale`, which controls the flexibility of the trend, and and `seasonality_prior_scale`, which controls the flexibility of the seasonality. The metric that was used for scoring was the RMSE. Interestingly, the optimized parameters produced an RMSE that was very close to the default settings.

Below is the optimized prediction.



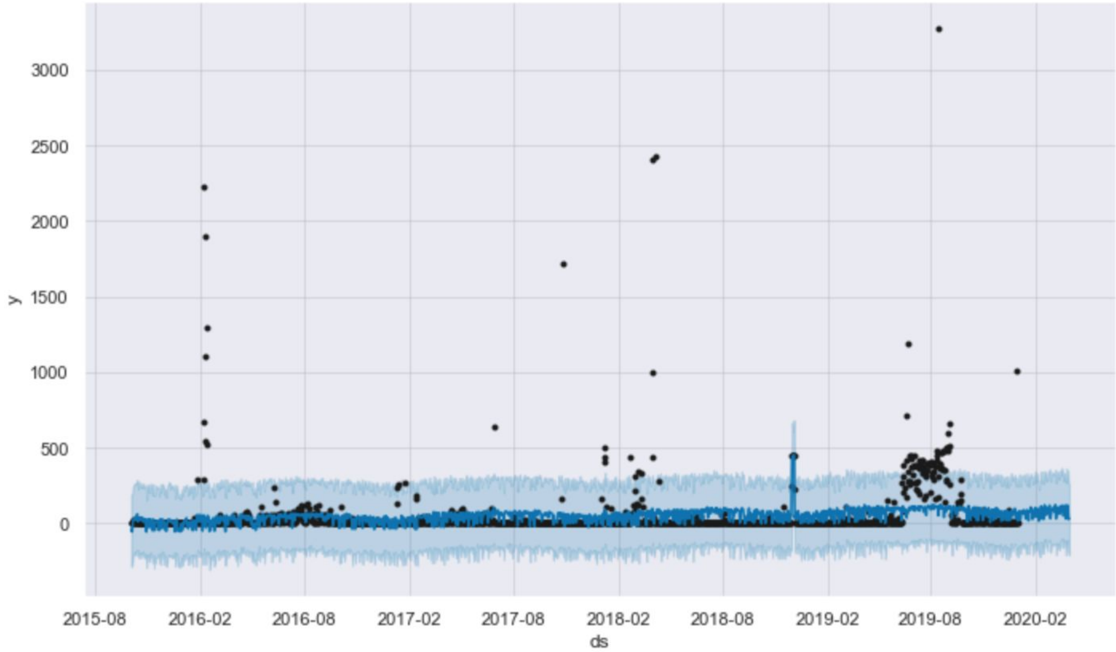
Forecasting Poor Performance

The second type of forecasting that Prophet was used for was to see if poor performance from the inverters could be forecast based on exogenous data. For this forecast, the behavior of each inverter was forecast separately.

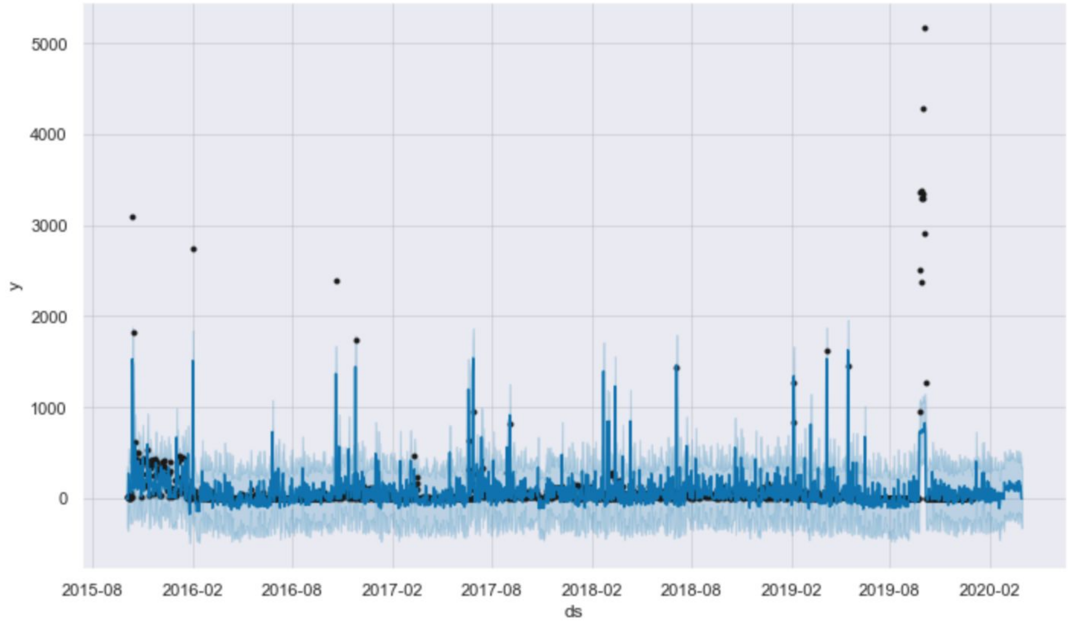
Exogenous variables for the time series forecasting were the previous day's maximum module and ambient temperatures and the previous day's maximum power lag from the other inverter.

These forecasts were performed with cross-validation and hyper-parameter tuning as before.

The result for Inverter 1 is shown below. It is not very impressive, although it does capture one of the peaks.



However, for Inverter 2 it actually performs quite well, capturing most of the peaks.

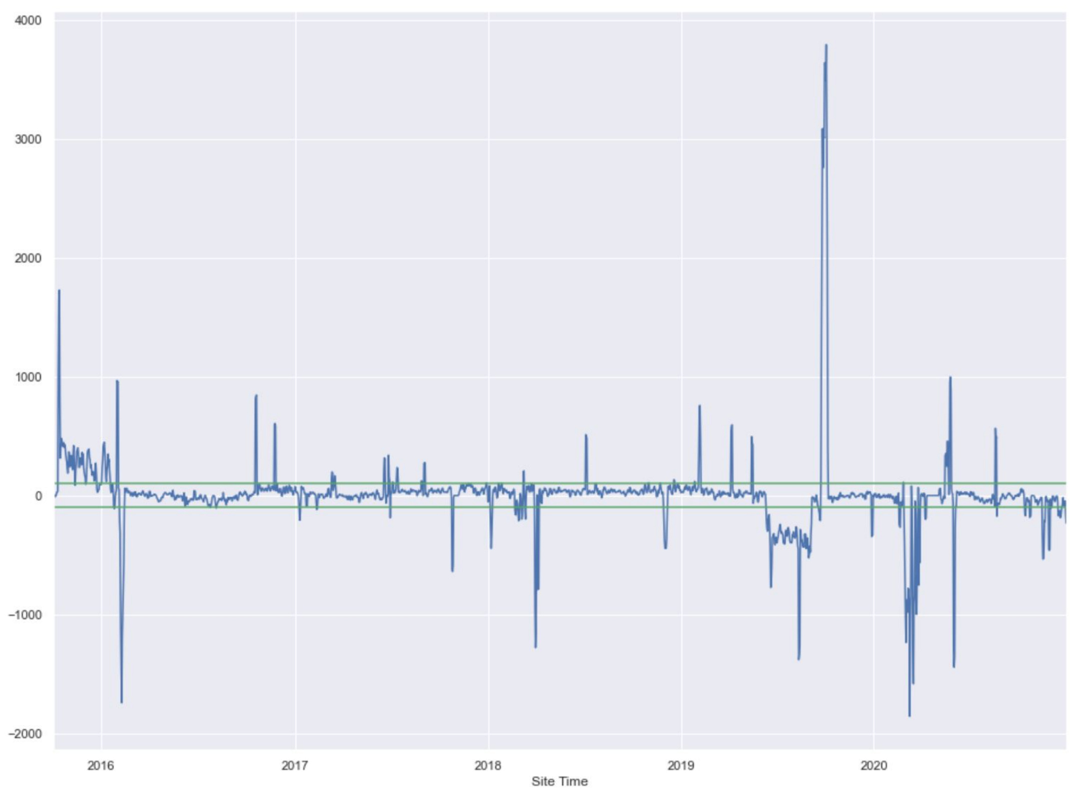


Classifier for Poor Performance

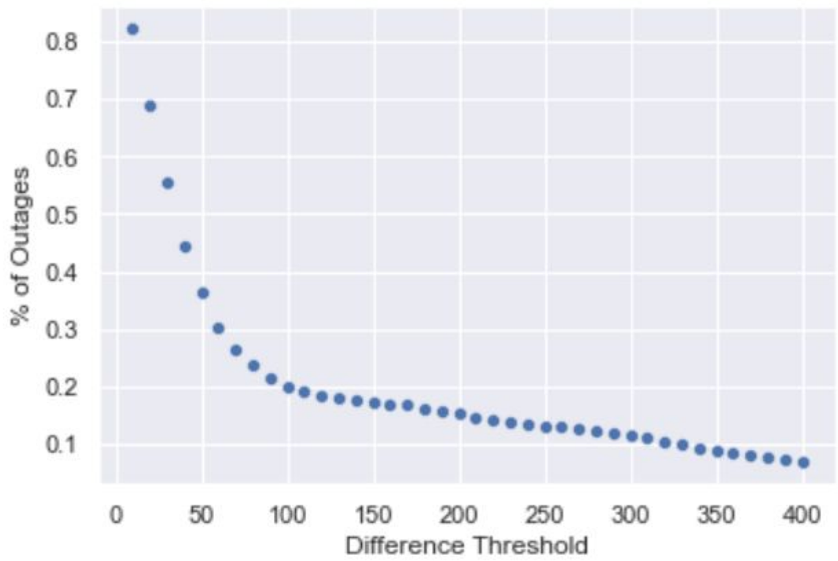
Poor performance was also framed as a binary classification problem. Again, since the data was not labeled, the first challenge was to determine which days were "bad" and which days were "good" based on the previously calculated time series of inverter performance.

In order to smooth the noisy data, it was aggregated on a daily basis and smoothed with a 3-day rolling average.

Several methods were tried to classify the days as "good" or "bad". Two clustering methods were attempted, KMeans and DBSCAN. KMeans did not perform particularly well, missing data points that seemed to be obvious bad days. DBSCAN showed more promise, since it is good at classifying noise, however it also missed some days that were obviously bad days. In the end, it made the most sense from visual inspection that the best way to classify 1's (bad days) and 0's (good days) was based on a simple threshold.



The threshold was determined by plotting the percentage of 1's based on the chosen threshold. As seen in the image below, there is an obvious elbow in this plot, and so a threshold of 100 was chosen. This resulted in approximately 20% of the days showing poor performance.



For our classifier each day was an instance, the target variable was our 1 or 0 as determined above, and our primary feature variable was the inverter lag. In addition, the following features were added:

- Inverter lag for each of the past seven days
- Max ambient temperature for each of the past seven days
- Max module temperature for each of the past seven days
- Energy output for each of the past seven days
- Max inverter lag for each of the past seven days
- Month of the year

With these features set, a random forest classifier and a support vector classifier were optimized using GridSearchCV with 5-fold cross validation and a scoring metric of recall (in order to minimize false negatives).

A random forest classifier was first trained on Inverter 1 alone. The optimized results are below.

```
              precision    recall  f1-score   support

       False       0.97      1.00      0.99       342
        True       0.97      0.78      0.86        41

    accuracy                           0.97       383
   macro avg       0.97      0.89      0.93       383
weighted avg       0.97      0.97      0.97       383

Tuned Model Parameters: {'RF__criterion': 'gini', 'RF__max_depth': 7, 'RF__n_estimators': 100}

Confusion Matrix:
 [[341   1]
 [  9  32]]
```

Next, a classifier was trained on Inverter 2 alone.

```
              precision    recall  f1-score   support

       False       0.97      1.00      0.99       348
        True       1.00      0.71      0.83        35

    accuracy                           0.97       383
   macro avg       0.99      0.86      0.91       383
weighted avg       0.97      0.97      0.97       383

Tuned Model Parameters: {'RF__criterion': 'entropy', 'RF__max_depth': 7, 'RF__n_estimators': 100}

Confusion Matrix:
 [[348   0]
 [ 10  25]]
```

And finally, a classifier was trained on instances from both inverters combined.

```
              precision    recall  f1-score   support

       False       0.98      0.99      0.99       697
        True       0.93      0.76      0.84        68

    accuracy                           0.97       765
   macro avg       0.95      0.88      0.91       765
weighted avg       0.97      0.97      0.97       765

Tuned Model Parameters: {'RF__criterion': 'gini', 'RF__max_depth': 7, 'RF__n_estimators': 200}

Confusion Matrix:
 [[693   4]
 [ 16  52]]
```

All three classifiers performed similarly, and there is a benefit to having one classifier trained on both sets of data so that it does not overfit for a particular inverter. This final model was then trained on the train and test data combined using the optimal parameters.

Finally, the top ten most important features were returned from the classifier.

	importance
DIFF_2	0.281104
DIFF_1	0.250655
DIFF_3	0.083647
DIFF_4	0.065961
DIFF_5	0.049973
DIFF_6	0.049040
MAX_DIFF_1	0.025398
DIFF_7	0.020344
MAX_DIFF_2	0.018808
KWH_2	0.010573

It can be seen from this that the most important feature in predicting an outcome of 1 was whether or not the previous day or days were labeled with a 1. This is somewhat disappointing, as it doesn't really speak to the classifier finding other patterns in the data. However, the overall accuracy is really quite good, as well as the recall score.

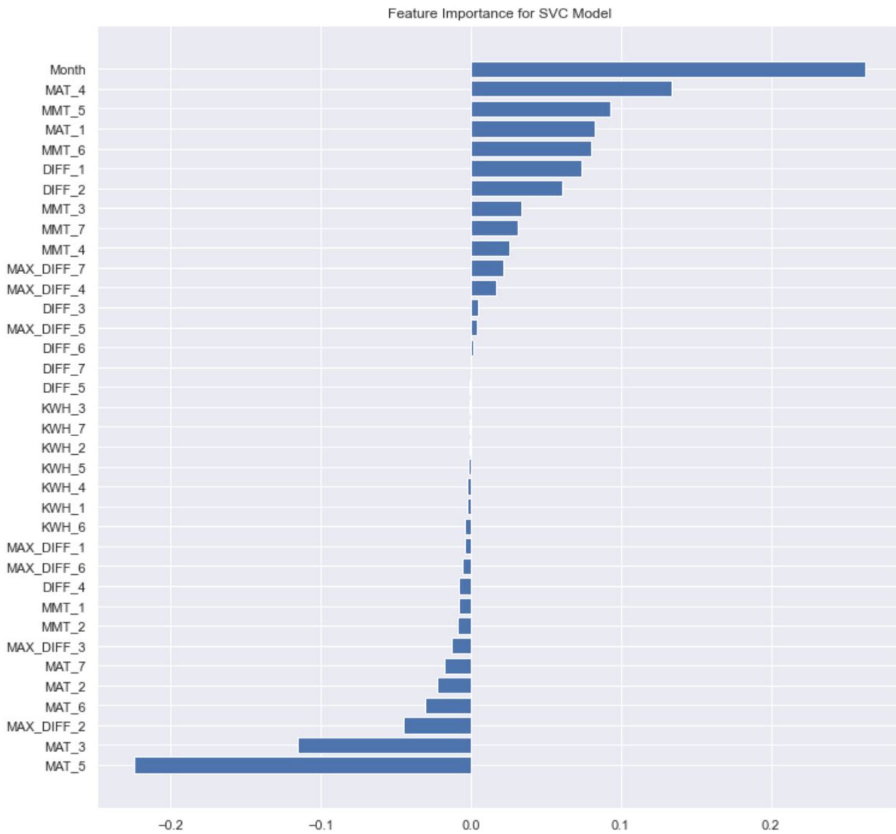
The same data was then used to train a Support Vector Classifier. This classifier actually performed better than the random forest classifier in terms of the recall score, and interestingly, it showed a dependence on features other than the previous days' value. So the SVC classifier shows some real promise.

```
              precision    recall  f1-score   support

       False       0.98      0.99      0.98       679
        True       0.91      0.83      0.87        86

    accuracy                           0.97       765
   macro avg       0.94      0.91      0.92       765
weighted avg       0.97      0.97      0.97       765


Confusion Matrix:
 [[672   7]
 [ 15  71]]
```



It should be noted that some other things were tried as well. For instance, the problem was set up as a multi-class problem, with a 1 being the first day of a series of bad days, and subsequent bad days being set to a 2. This resulted in equally accurate results, but with both the random forest classifier and the support vector classifier, it was very unlikely for a 1 to actually be forecast. This is most likely due to the imbalance in the dataset as only very few days were categorized with a 1 as opposed to a 2, but it is also due to the fact that predicting the first day of a string of bad days is quite difficult.

Conclusions and Final Thoughts

The forecasting of energy output using FB Prophet was useful in terms of predicting the trend line showing degradation of the panels. However, the data, even when aggregated to daily data, was so noisy that Prophet smoothed it quite a bit and did not do particularly well with capturing peaks and valleys.

The time series forecasting of inverter performance was interesting and showed some real promise. This could be further explored as a potential predictive algorithm if we had more data, both in terms of exogenous features, and actual outage information.

Similarly, the attempt to classify and predict poor performance showed mixed results. The accuracy and recall scores of the random forest classifiers was actually pretty good, but it's not clear that was due to much more than looking at the previous days and seeing whether those had poor performance. The support vector classifier actually showed much more promise for finding patterns in the data. Again, this could be further explored with additional data from the solar array, particularly information on actual outages.

Another avenue of further work could be structuring the entire problem as a regression problem instead of a classification problem. Given that the performance of the inverters is a continuous variable, it could be structured as such and might give interesting results.