Machine Learning with Applications in Python Final Group Project

By:
Mansi Jain
Miao Xi

01

Motivation, Data Introduction

02

EDA

03

ML Models

04

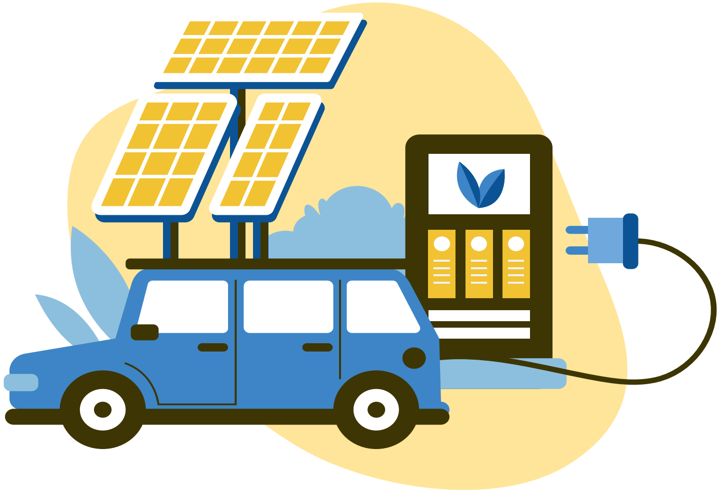
Conclusion & Learning

01

Motivation, data introduction

Motivation

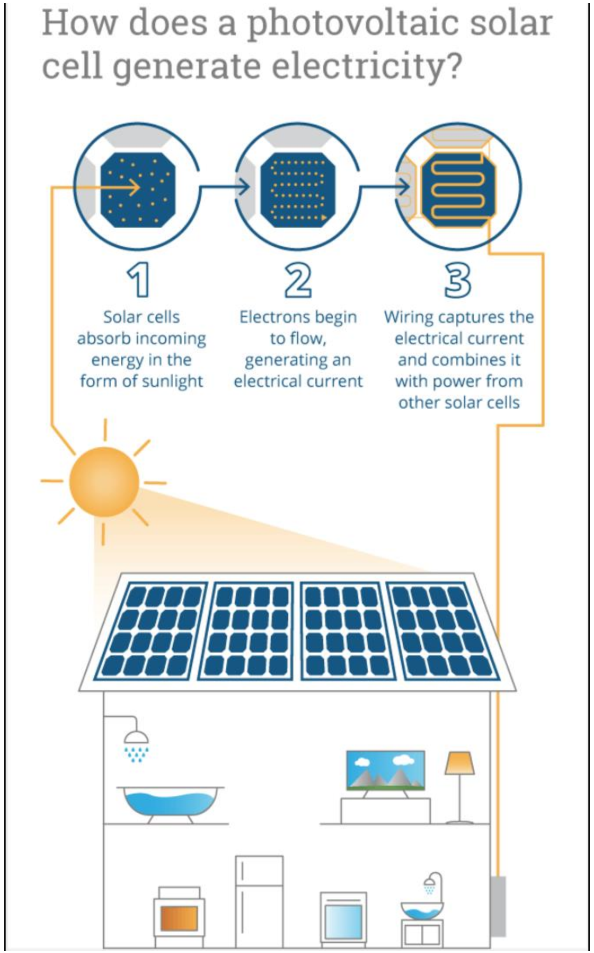
- Energy production in several developing countries often falls short of energy requirements which results in frequent power cuts
- Fossil fuel is limited, so it is important to consider clean sources such as solar to meet the energy demands in the future
- Current motto in USA solar industry – “install it and forget it”
- This study sheds light on how we can optimize energy generation in solar plants



Solar power generation

The photovoltaic process:

- The silicon photovoltaic solar cell absorbs solar radiation
- When the sun's rays interact with the silicon cell, electrons begin to move, creating a flow of electric current
- Wires capture and feed this direct current (DC) electricity to a solar inverter to be converted to alternating current (AC) electricity



Research questions

We will address following three research questions in our project:

1. Can we identify the faulty equipment in the plant?
2. Can we identify panels that may need cleaning and maintenance?
3. Can we predict the output of power for future days that would help better grid management?

Dataset description

Power Generation Data

	DATE_TIME	PLANT_ID	SOURCE_KEY	DC_POWER	AC_POWER	DAILY_YIELD	TOTAL_YIELD
0	2020-05-15	4136001	4UPUqMRk7TRMgml	0.0	0.0	9425.000000	2.429011e+06
1	2020-05-15	4136001	81aHJ1q11NBPMrL	0.0	0.0	0.000000	1.215279e+09
2	2020-05-15	4136001	9kRcWv60rDACzjR	0.0	0.0	3075.333333	2.247720e+09
3	2020-05-15	4136001	Et9kgGMDl729KT4	0.0	0.0	269.933333	1.704250e+06
4	2020-05-15	4136001	IQ2d7wF4YD8zU1Q	0.0	0.0	3177.000000	1.994153e+07

Weather Sensor Data

	DATE_TIME	PLANT_ID	SOURCE_KEY	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION
0	2020-05-15 00:00:00	4136001	iq8k7ZNt4Mwm3w0	27.004764	25.060789	0.0
1	2020-05-15 00:15:00	4136001	iq8k7ZNt4Mwm3w0	26.880811	24.421869	0.0
2	2020-05-15 00:30:00	4136001	iq8k7ZNt4Mwm3w0	26.682055	24.427290	0.0
3	2020-05-15 00:45:00	4136001	iq8k7ZNt4Mwm3w0	26.500589	24.420678	0.0
4	2020-05-15 01:00:00	4136001	iq8k7ZNt4Mwm3w0	26.596148	25.088210	0.0

Dataset description cont.

Merged power generation data with weather data to study the effect of weather on inverter outputs

Merged Data

	DATE_TIME	PLANT_ID	Inverter_ID	DC_POWER	AC_POWER	DAILY_YIELD	TOTAL_YIELD	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION
0	2020-05-15 00:00:00	4136001	4UPUqMRk7TRMgml	0.0	0.0	9425.000000	2.429011e+06	27.004764	25.060789	0.0
1	2020-05-15 00:00:00	4136001	81aHJ1q11NBPMrL	0.0	0.0	0.000000	1.215279e+09	27.004764	25.060789	0.0
2	2020-05-15 00:00:00	4136001	9kRcWv60rDACzjR	0.0	0.0	3075.333333	2.247720e+09	27.004764	25.060789	0.0
3	2020-05-15 00:00:00	4136001	Et9kgGMDl729KT4	0.0	0.0	269.933333	1.704250e+06	27.004764	25.060789	0.0
4	2020-05-15 00:00:00	4136001	IQ2d7wF4YD8zU1Q	0.0	0.0	3177.000000	1.994153e+07	27.004764	25.060789	0.0
...	...	...	...	...	...	...	...	...	...	...
67693	2020-06-17 23:45:00	4136001	q49J1IKaHRwDQnt	0.0	0.0	4157.000000	5.207580e+05	23.202871	22.535908	0.0
67694	2020-06-17 23:45:00	4136001	rrq4fwE8jgrTyWY	0.0	0.0	3931.000000	1.211314e+08	23.202871	22.535908	0.0
67695	2020-06-17 23:45:00	4136001	vOuJvMaM2sgwLmb	0.0	0.0	4322.000000	2.427691e+06	23.202871	22.535908	0.0
67696	2020-06-17 23:45:00	4136001	xMbIugepa2P7lBB	0.0	0.0	4218.000000	1.068964e+08	23.202871	22.535908	0.0
67697	2020-06-17 23:45:00	4136001	xoJJ8DcxJEcupym	0.0	0.0	4316.000000	2.093357e+08	23.202871	22.535908	0.0

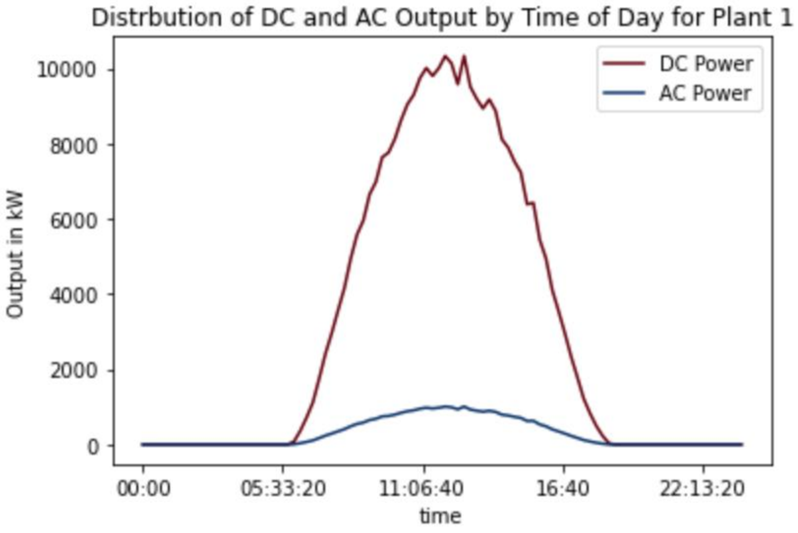
67698 rows × 10 columns

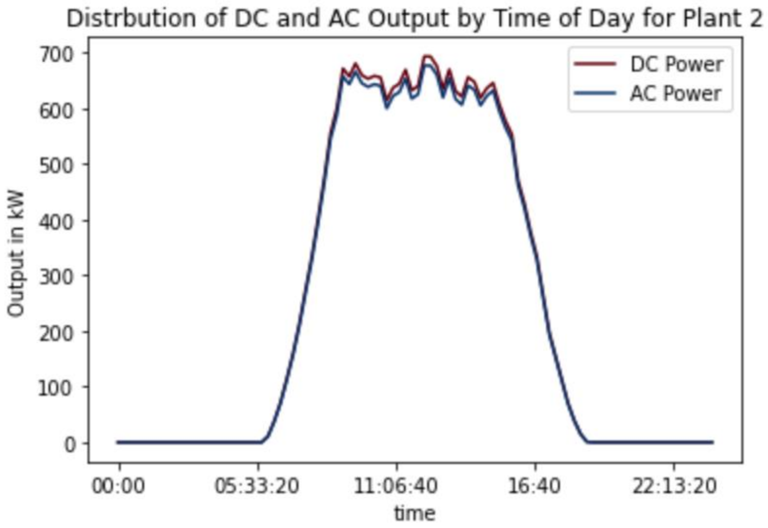
02

Exploratory Data Analysis

Issue with plant#1 data

- In plant 1 data, DC to AC power conversion rate is extremely low (less than 10% vs. 90% for industry benchmark)
- Plant 1 data is excluded from our study



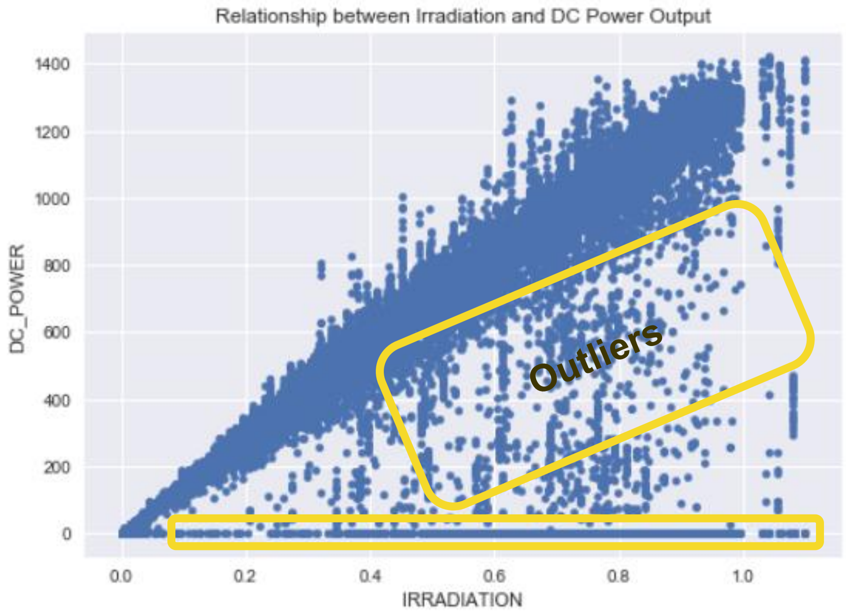


Correlation matrix

Strong linear correlation between irradiation and power output

	DC_POWER	AC_POWER	DAILY_YIELD	TOTAL_YIELD	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION	month	hour
DC_POWER	1.000000	0.999997	0.005593	0.004528	0.563232	0.749676	0.780978	-0.080431	0.027817
AC_POWER	0.999997	1.000000	0.005395	0.004533	0.563324	0.749604	0.780851	-0.080248	0.027842
DAILY_YIELD	0.005593	0.005395	1.000000	-0.068472	0.321785	0.046787	-0.107987	-0.040094	0.596705
TOTAL_YIELD	0.004528	0.004533	-0.068472	1.000000	0.002774	-0.004646	-0.006720	-0.032167	-0.003695
AMBIENT_TEMPERATURE	0.563232	0.563324	0.321785	0.002774	1.000000	0.848976	0.671998	-0.355423	0.360336
MODULE_TEMPERATURE	0.749676	0.749604	0.046787	-0.004646	0.848976	1.000000	0.947057	-0.183838	0.150493
IRRADIATION	0.780978	0.780851	-0.107987	-0.006720	0.671998	0.947057	1.000000	-0.092924	0.021706
month	-0.080431	-0.080248	-0.040094	-0.032167	-0.355423	-0.183838	-0.092924	1.000000	-0.005384
hour	0.027817	0.027842	0.596705	-0.003695	0.360336	0.150493	0.021706	-0.005384	1.000000

Outlier detection

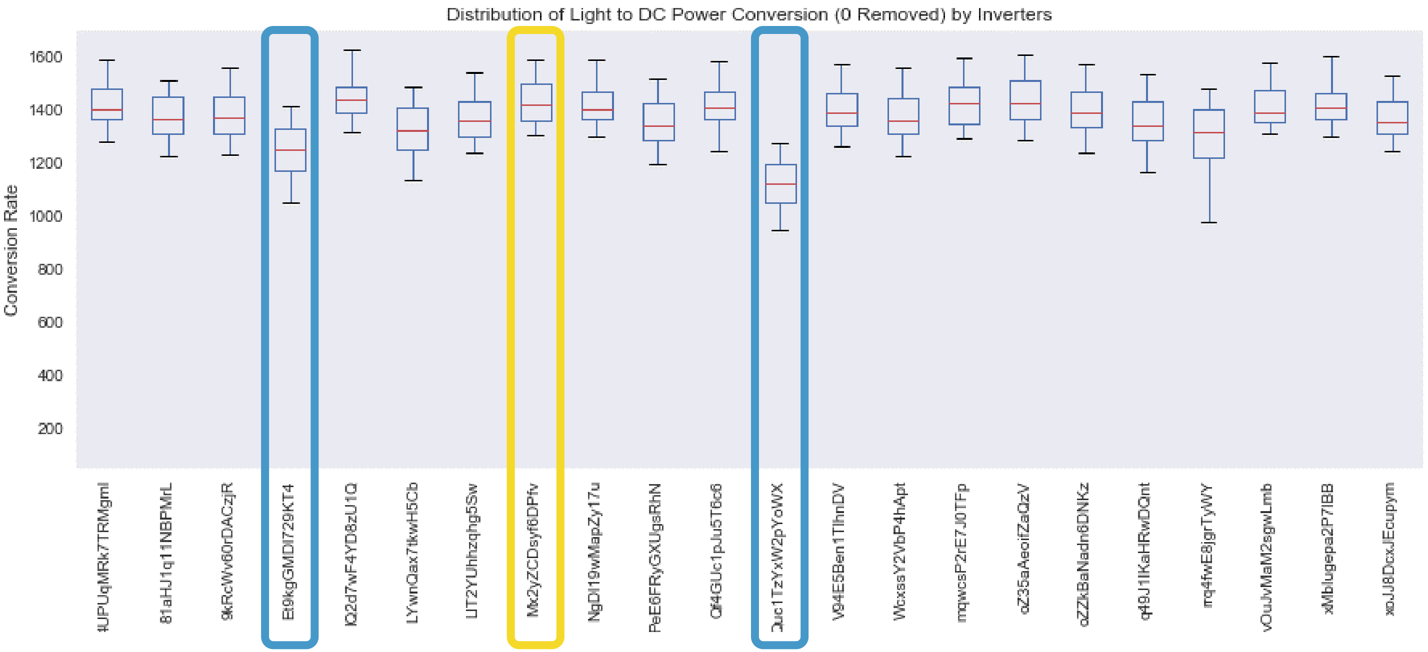


- Relationship between irradiation and DC output is mostly linear, however, for some cases, when irradiation is high the DC output remains 0 or deviate from the linear pattern

- We suspect that this could be caused by malfunctioning of some equipment or some performing sub-optimally compared to others

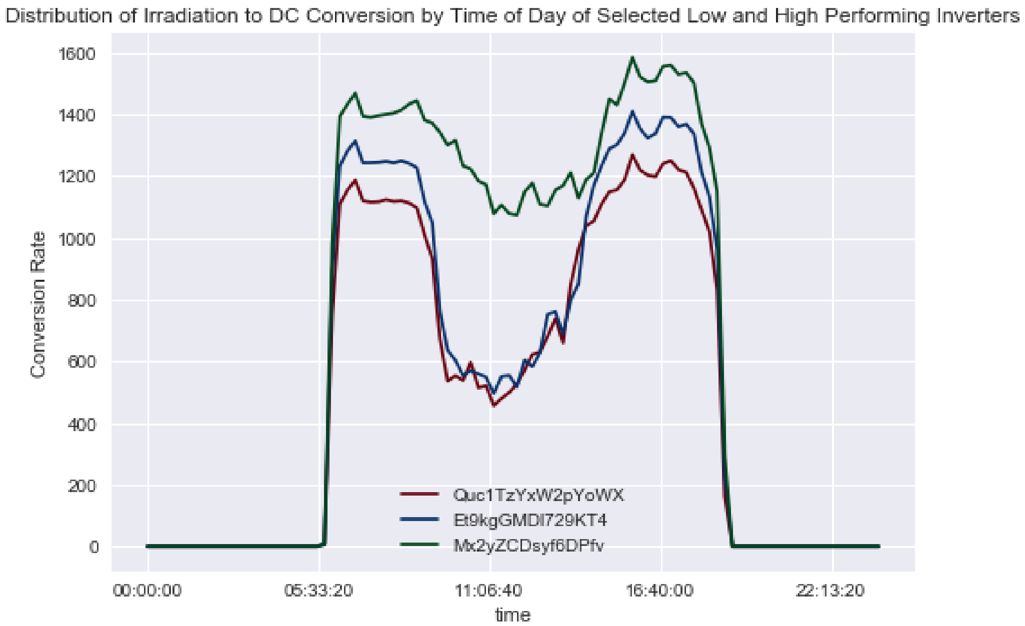
Performance by inverters

- Use conversion rate by DC output / irradiation to measure the performance of inverters
- Identify inverters with high and low efficiencies



Performance by inverters

- Irradiation to DC conversion rate differs between high and low performing inverters
- Conversion rate gap especially large around noon time where irradiation tend to be strongest
- This could be credited to some inverters require more maintenance / repair than others



03

Machine Learning Models

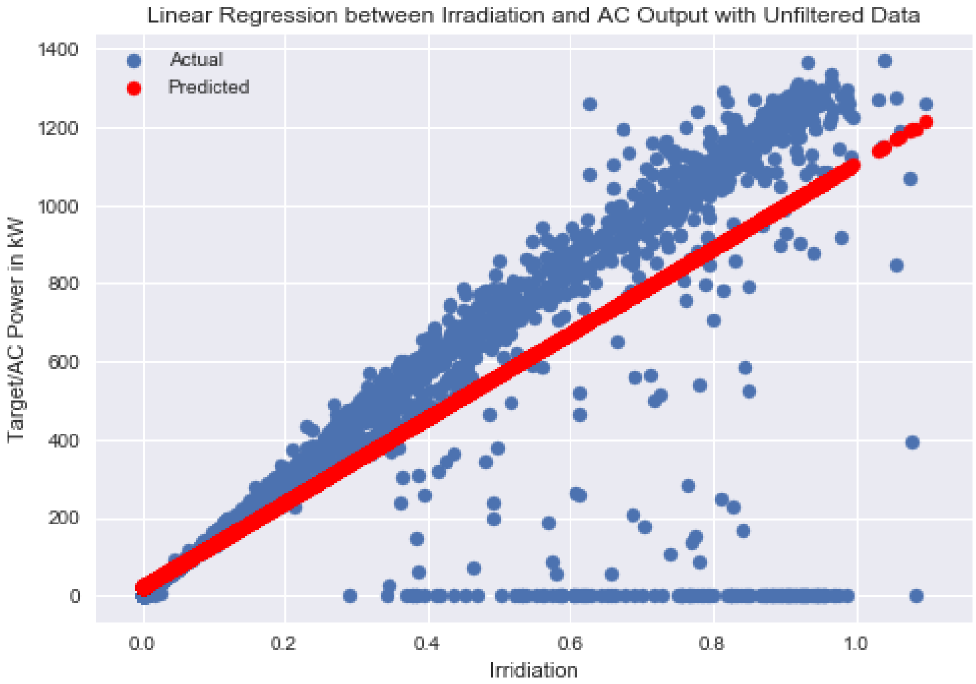
Overview of methodologies

Research Questions	Methodologies
1. Identify faulty inverters 2. Identify inverters that require maintenance	1. Filter data to exclude power output outliers 2. Build ML models **(Linear Regression, Regression Tree, SVR, Random Forest)** based on filtered data to predict power output then identify the best model serving as the 'golden standard' 3. Compare actual output with the 'golden standard' and develop labeling technique to label faculty or require maintenance
3. Forecast future output	• Time series forecast **(FB Prophet, VARMA)** • I.I.D. manipulation on time series data

Filter power output outliers

Demonstrate with **one** inverter

- Outliers distorted the linear relationship between irradiation and AC output
- ‘Golden standard’ model is impossible to built without any filtering



Filter power output outliers

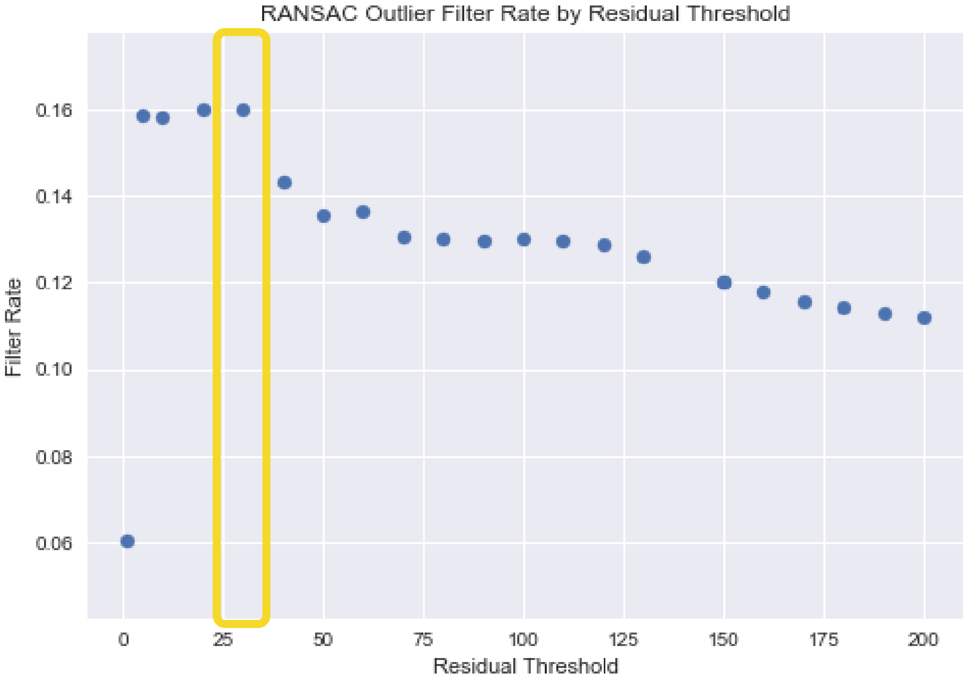
Demonstrate with **one** inverter

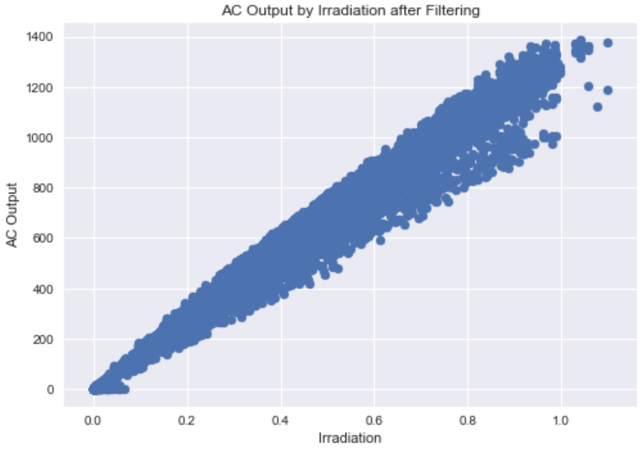
- Robust regression **(RANSAC)** fit through data points representing the linear relationship
- Calculate distance for every data point to the predicted line
- 1.5 IQR rule applied on the distances to distinguish inliers **(the yellow area)** and outliers

Filter power output outliers

Filter the entire dataset

Tune hyperparameter for RANSAC to obtain the strictest filtering (remove **16%**)





Before

After

Building ML models

Model inputs

Input Data

	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION	month	day_of_month	hour	minute	day_of_week
17	27.004764	25.060789	0.000000	5	15	0	0	4
39	26.880811	24.421869	0.000000	5	15	0	15	4
61	26.682055	24.427290	0.000000	5	15	0	30	4
83	26.500589	24.420678	0.000000	5	15	0	45	4
105	26.596148	25.088210	0.000000	5	15	1	0	4
127	26.512740	25.317970	0.000000	5	15	1	15	4
149	26.494339	25.217193	0.000000	5	15	1	30	4

Target

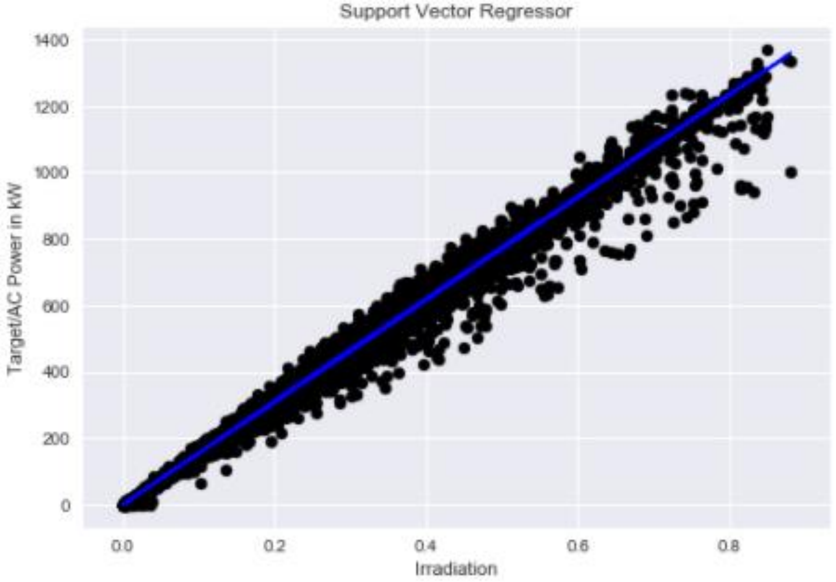
	AC_POWER
17	0.000000
39	0.000000
61	0.000000
83	0.000000
105	0.000000
127	0.000000
149	0.000000

Support vector machine

trained on filtered data & tuned hyperparameters

Hyperparameter tuning with GridSearchcv

C / Kernel	0.1	1	5
linear			Best model
rbf			
poly			



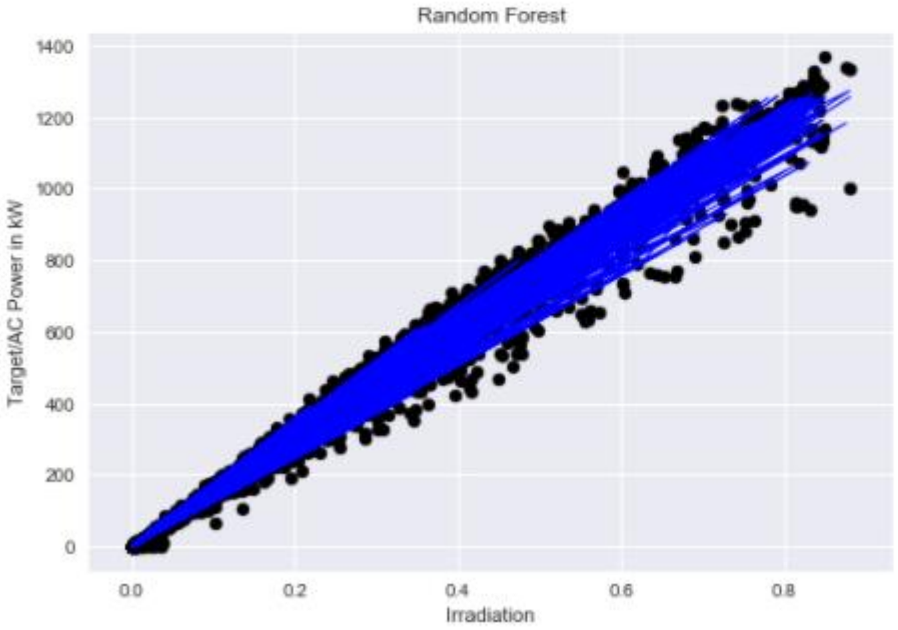
Model: SVR

RMSE: 26.87

R2: 0.9918

Random forest

trained on filtered data & tuned hyperparameters



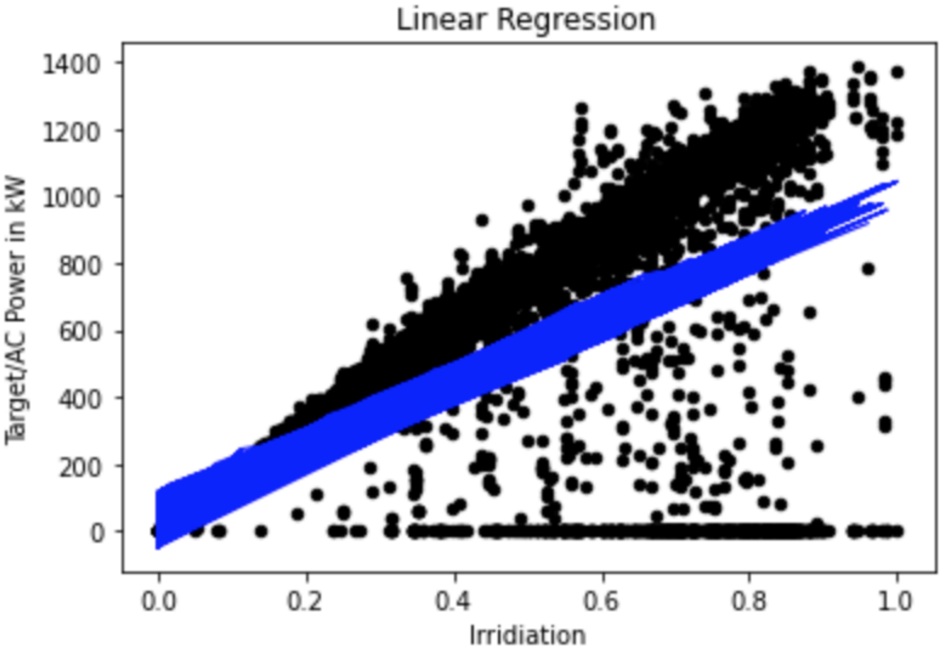
Model: Random forest

RMSE: 21.46

R2: 0.9948

Linear regression model

trained on unfiltered data



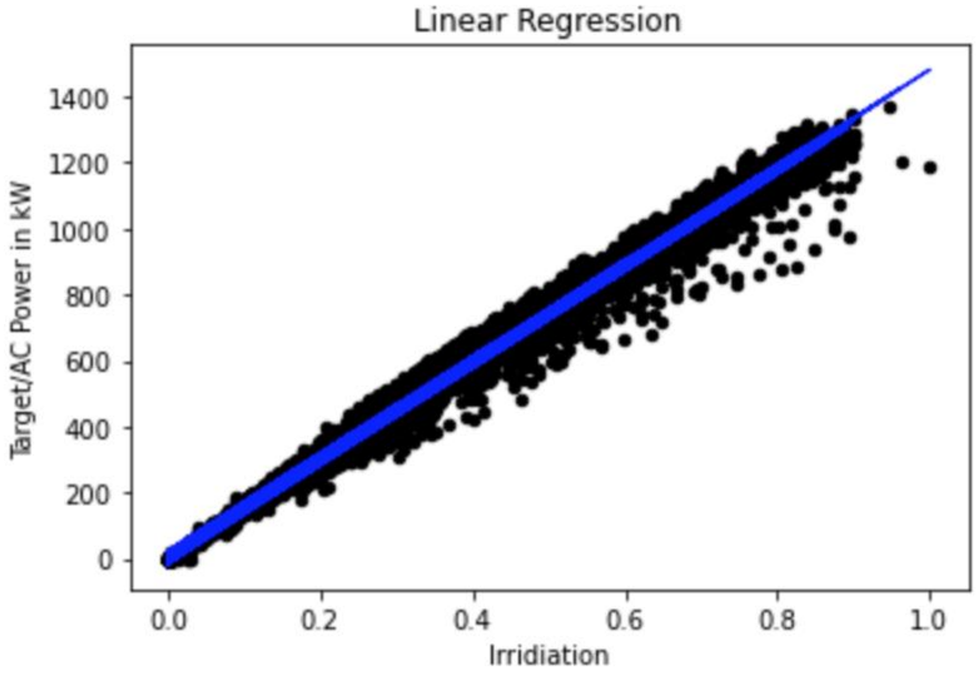
Model: Linear Regression

RMSE: 223.83

R2: 0.62

Linear regression model

trained on filtered data



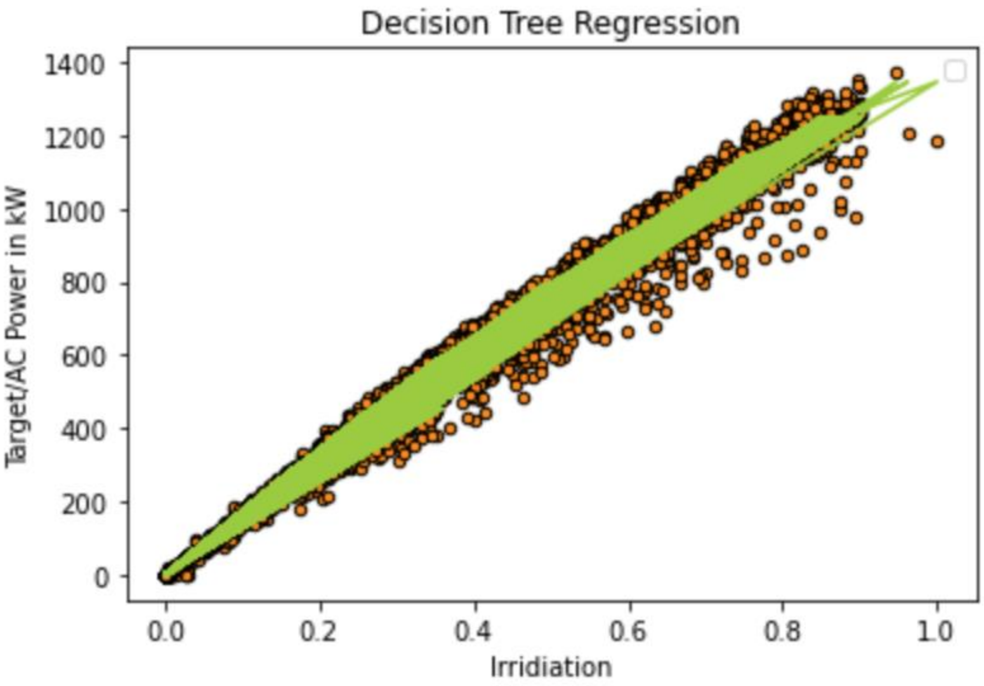
Model: Linear Regression

RMSE: 29.54

R2: 0.993

Basic regression tree model

trained on filtered data

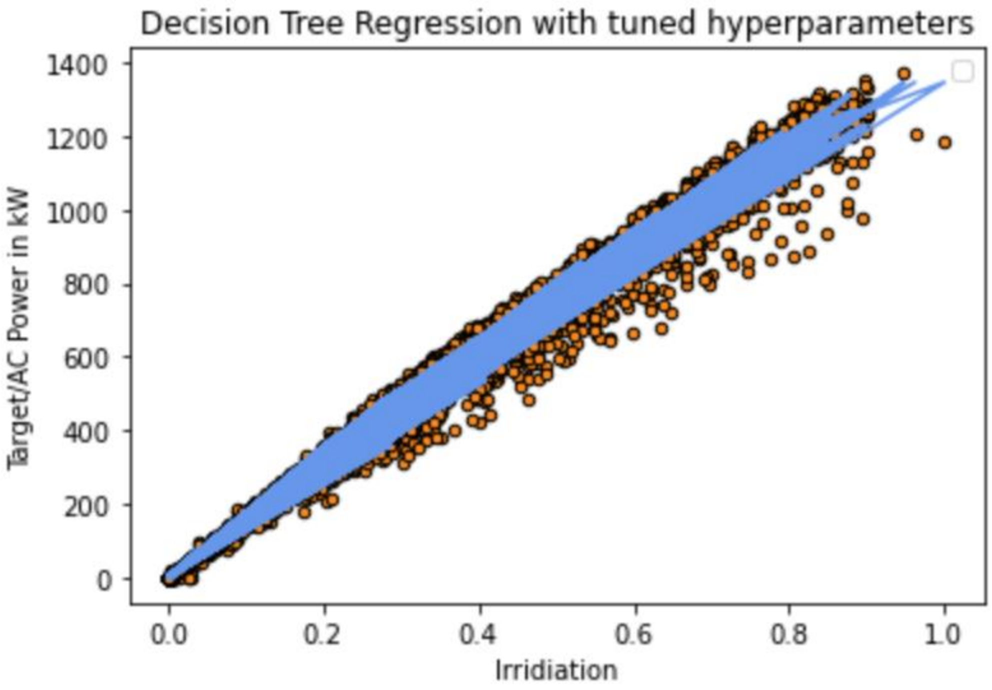


Model: Regression Tree

RMSE: 26.36

R2: 0.9944

Regression tree model
trained on filtered data & tuned hyperparameters



Model: Regression Tree Best

RMSE: 24.14

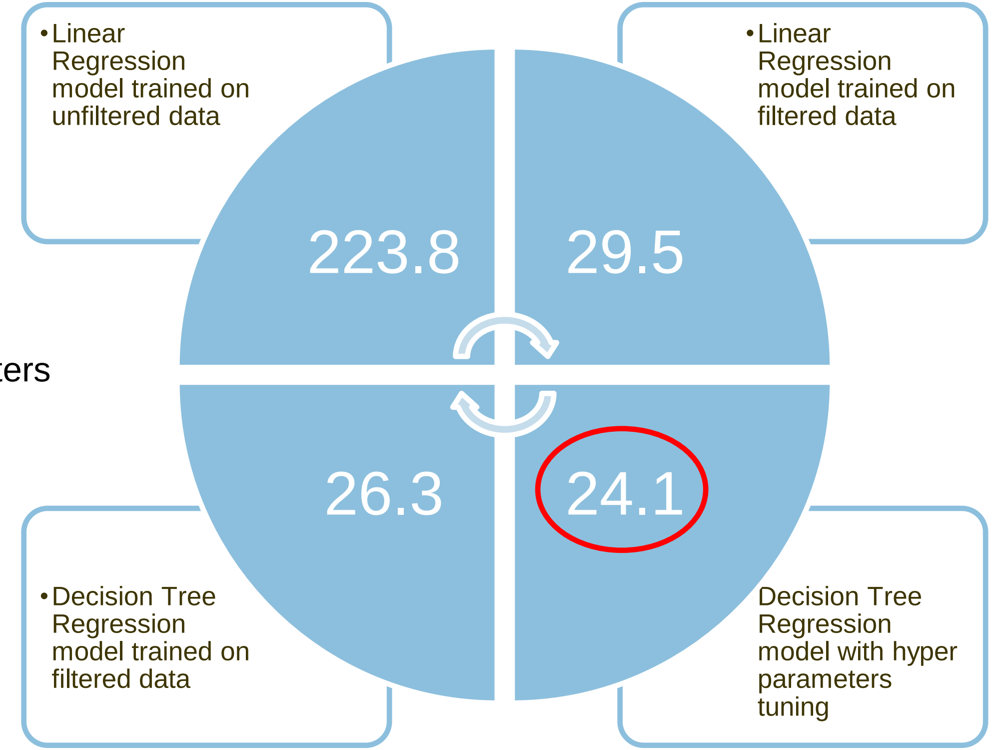
R2: 0.9954

Comparing the results of 4 models

RMSE

- Linear Regression
  - Unfiltered Data
  - Filtered Data

- Decision Tree Regression
  - Filtered Data
  - Filtered Data and tuned hyperparameters



Decision Tree Regression model performed better.

Labelling records

with faulty or need maintenance status

STEP 1:
Predict AC Power for whole data set by the **best model** trained with **filtered data**

→

STEP 2:
Calculate gap between actual AC power and predicted AC power

→

STEP 3:
Updated status of each record:
Faulty Equipment
Need Maintenance (IQR rule)
Working Fine

IQR for gaps between actual and predicted AC output (kW)

Q25 : -0.53
Q75 : 0.62
IQR : 1.14
Lower Cutoff : -2.24 (Q25 – 1.5 * IQR)

Labelling records

with faulty or need maintenance status

```python
# Function to update status of each record as faulty, working fine or need maintenance
def update_status(lower):
        for i in range(len(df_predict_updated)):
            gap = df_predict_updated.iloc[i]["Gap"]
            AC_Power = df_predict_updated.iloc[i]["AC_POWER"]
            AC_Power_Predicted = df_predict_updated.iloc[i]["AC_Power_Predicted"]
            Irridiation = df_predict_updated.iloc[i]["IRRADIATION"]
            DC_Power = df_predict_updated.iloc[i]["DC_POWER"]
            if(Irridiation > 0):
                if((AC_Power == 0.0) & (AC_Power_Predicted > 0)):
                    if(DC_Power > 0.0):
                        df_predict_updated.at[i, 'Status'] = "Faulty Inverter"
                    else:
                        df_predict_updated.at[i, 'Status'] = "Faulty Equipment"
                elif(gap < lower):
                    df_predict_updated.at[i, 'Status'] = "Need Maintenance"
                else:
                    df_predict_updated.at[i, 'Status'] = "Working Fine"
            elif(Irridiation == 0.0):
                    df_predict_updated.at[i, 'Status'] = "Working Fine"
```

Results

For research question# 1 & 2

```
df_predict_updated.Status.value_counts()
```

Working Fine	: 52264
Need Maintenance	: 11464
Faulty Equipment	: 3970

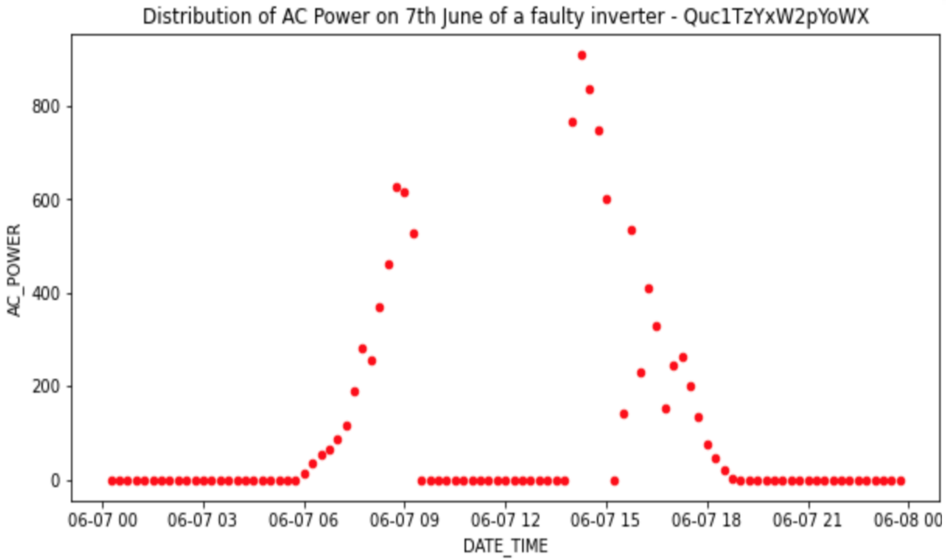
- The most faults were recorded on 2020-06-06 and 2020-06-07
- The most maintenance needs were recorded on 2020-06-07 and 2020-06-02
- The top 3 inverters with most faults or underperformance are Quc1TzYxW2pYoWX, Et9kgGMDl729KT4 and rrq4fwE8jgrTyWY

Faulty/sub-optimally performing equipment

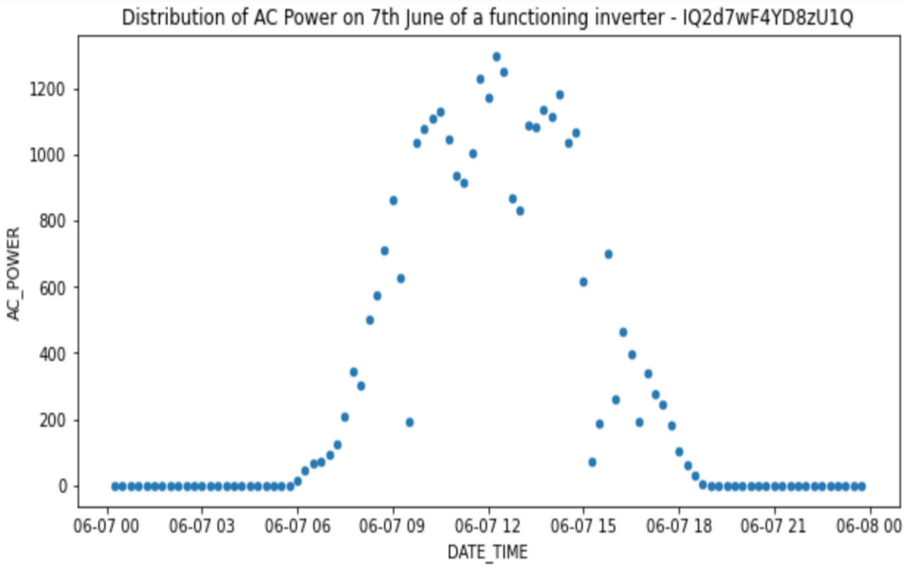
```
print("The most faults were recorded on {} and {} "
      .format(df_predict_updated[df_predict_updated["Status"] == "Faulty Equipment"]["Date"].value_counts().index[0],
              df_predict_updated[df_predict_updated["Status"] == "Faulty Equipment"]["Date"].value_counts().index[1]))
```

```
The most faults were recorded on 2020-06-06 and 2020-06-07
```

Faulty



Normal

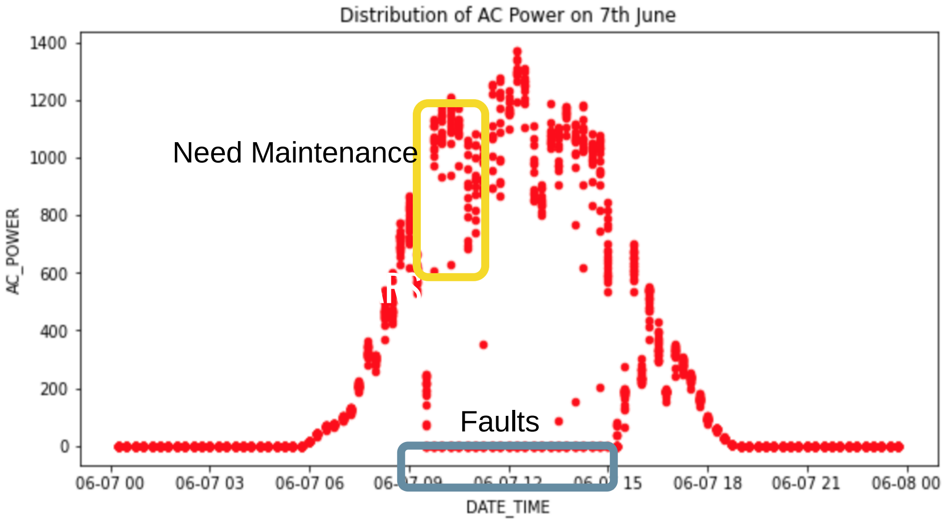


Equipment needing maintenance

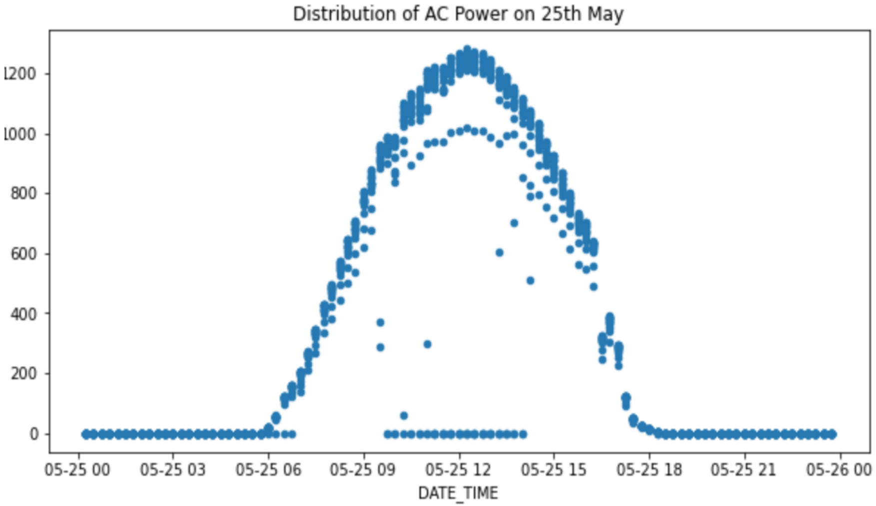
```
print("The most maintenance needs were recorded on {} and {} "
      .format(df_predict_updated[df_predict_updated["Status"]== "Need Maintenance"]["Date"].value_counts().index[0],
            df_predict_updated[df_predict_updated["Status"]== "Need Maintenance"]["Date"].value_counts().index[1]))
```

```
The most maintenance needs were recorded on 2020-06-07 and 2020-06-02
```

Need Maintenance



Normal



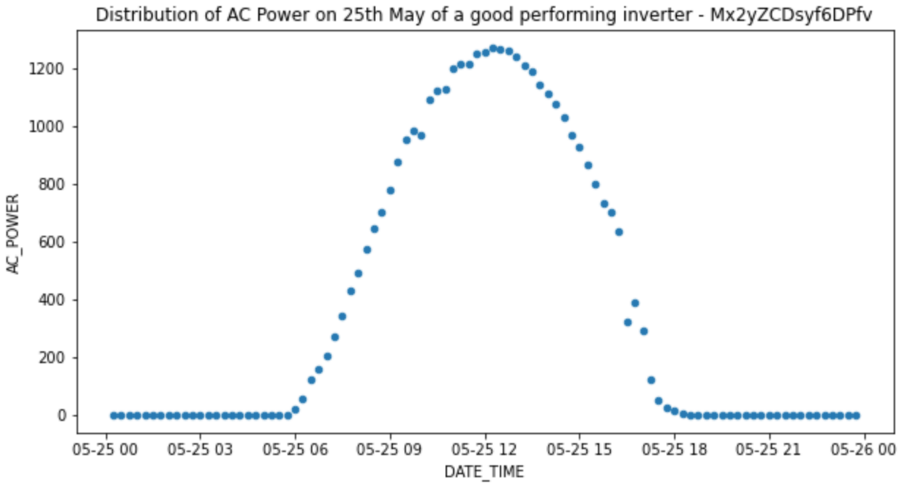
On June 7th the variance of AC output among inverters are large contrast to May 25th

Equipment needing maintenance

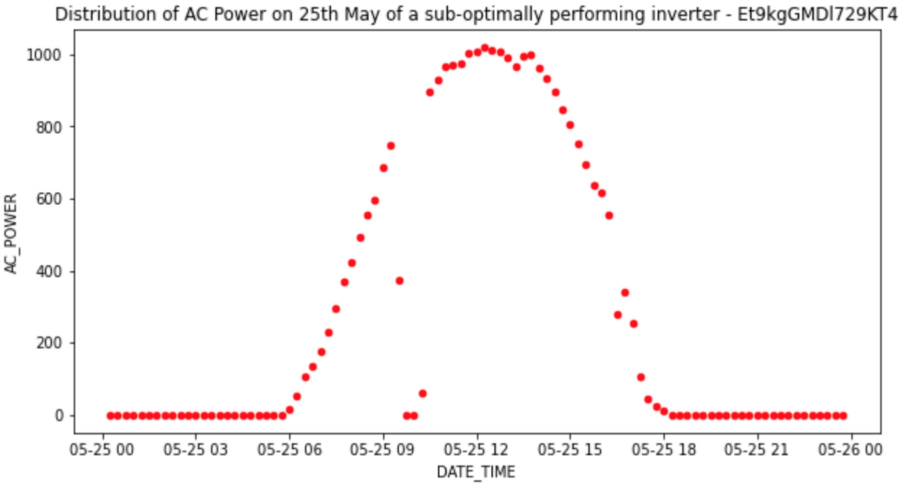
```
print("The 3 inverters with most faults or underperformance are {}, {} and {} "
      .format(df_predict_updated[((df_predict_updated['Status']== "Faulty Equipment") |
            (df_predict_updated['Status']== "Need Maintenance"))]["Inverter_ID"].value_counts().index[0],
             df_predict_updated[((df_predict_updated['Status']== "Faulty Equipment") |
            (df_predict_updated['Status']== "Need Maintenance"))]["Inverter_ID"].value_counts().index[1],
             df_predict_updated[((df_predict_updated['Status']== "Faulty Equipment") |
            (df_predict_updated['Status']== "Need Maintenance"))]["Inverter_ID"].value_counts().index[2]))
```

```
The 3 inverters with most faults or underperformance are Quc1TzYxW2pYoWX, Et9kgGMDl729KT4 and rrq4fwE8jgrTyWY
```

Underperform



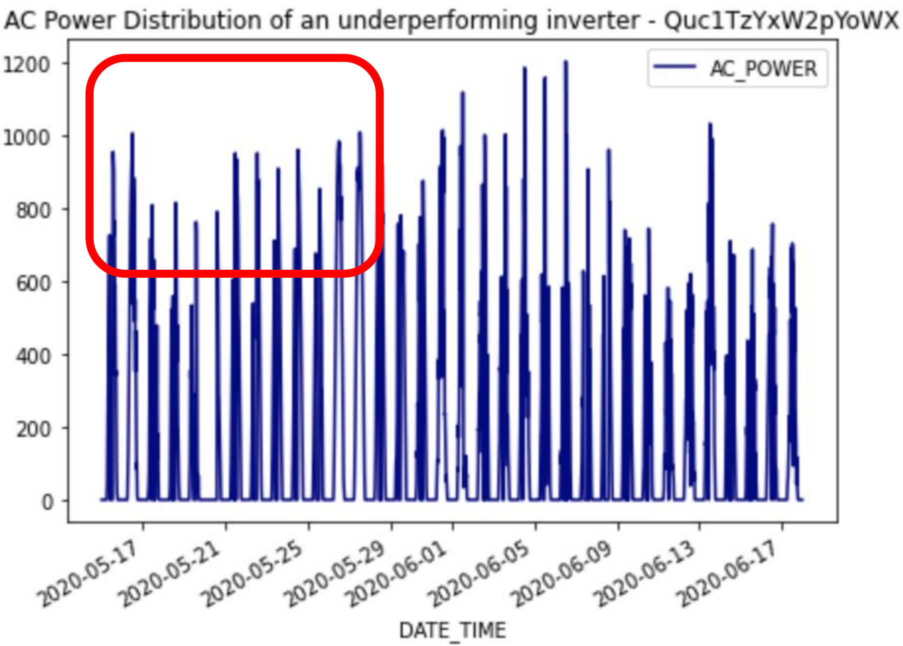
Normal



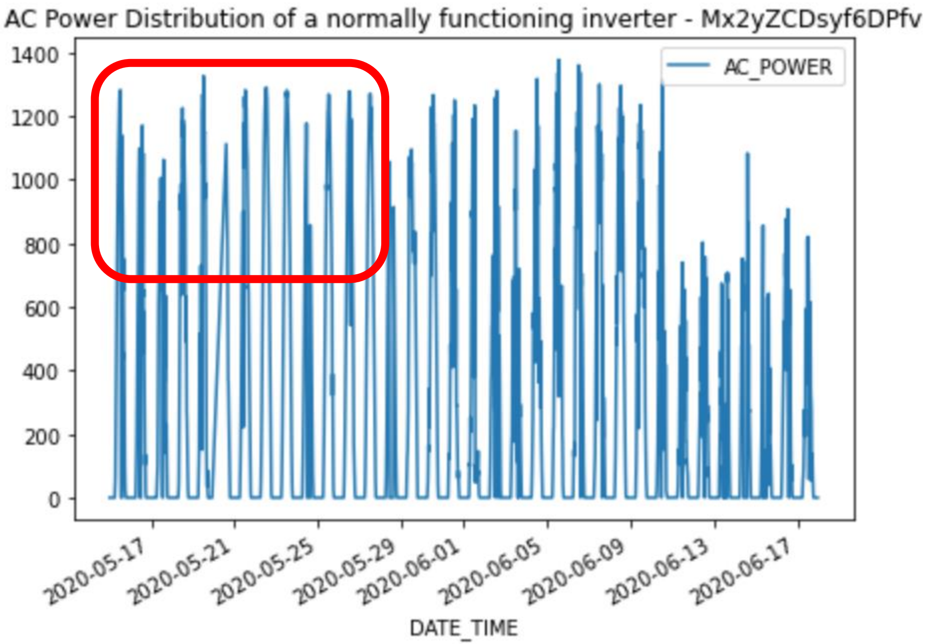
Equipment needing maintenance

Comparing performances of an underperforming and a normally performing inverter

Underperforming



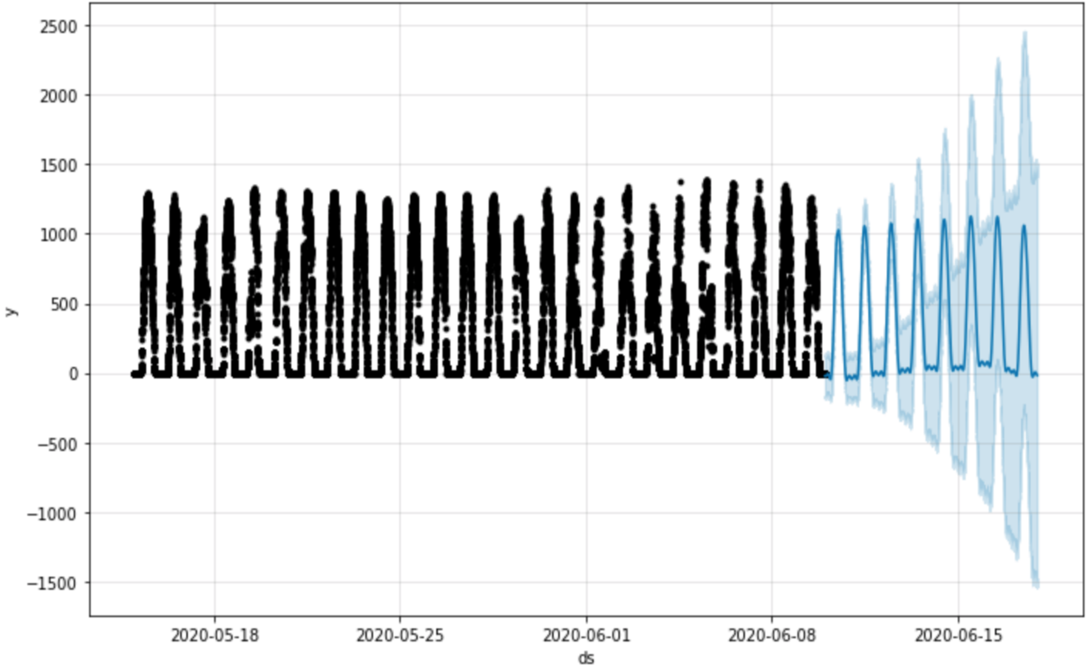
Normal



Research question# 3 – univariate time series model

Facebook Prophet

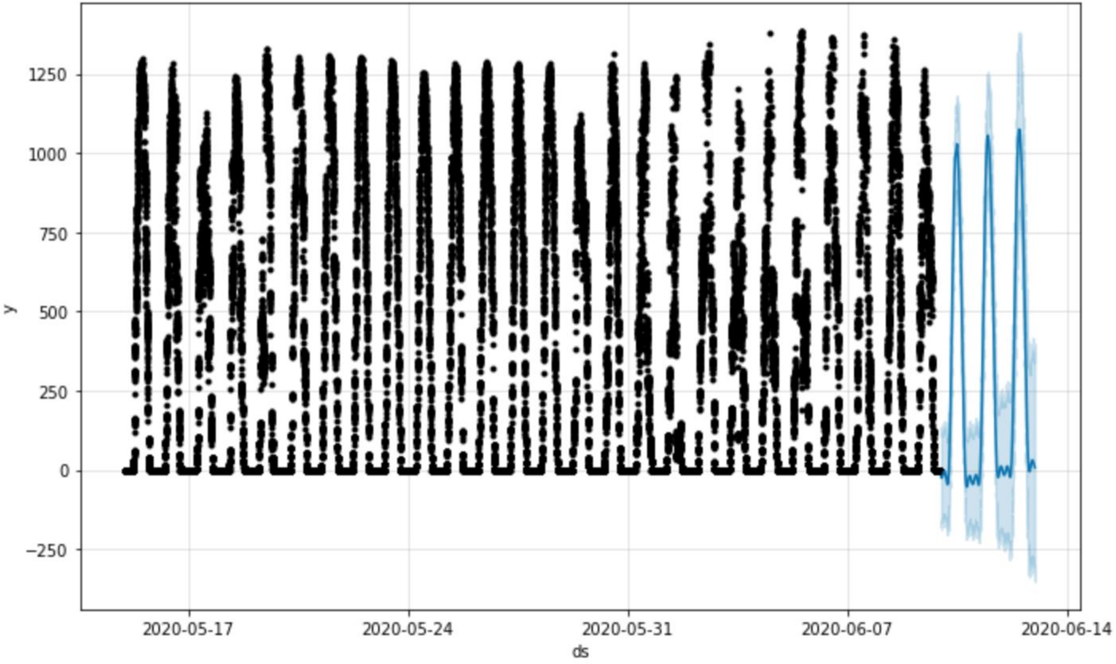
```
df_train = df_input[df_input['DATE_TIME'] < '2020-06-10']
df_predict = df_input[df_input['DATE_TIME'] >= '2020-06-10']
```



Predicting 8 days ahead

- **MAE:** 328.502
- Same performance for filtered and unfiltered data

Research question# 3 – univariate time series model
Facebook Prophet

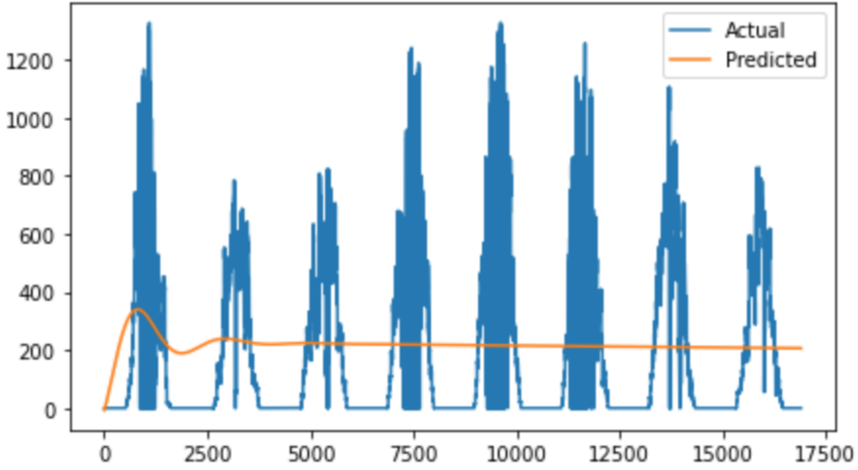


Predicting 3 days ahead

- **MAE:** 321.192
- Same performance for filtered and unfiltered data

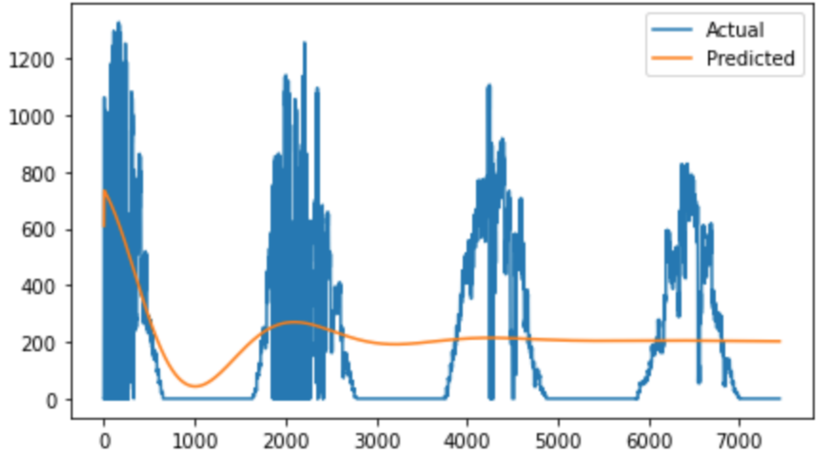
Research question# 3 – Multivariate time series model
VARMA

MAE: 234.443



Predicting 8 days ahead

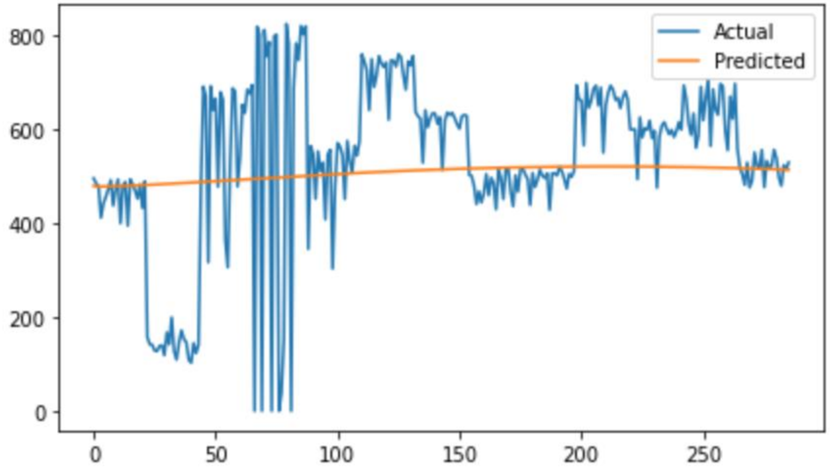
MAE: 217.592



Predicting 4 days ahead

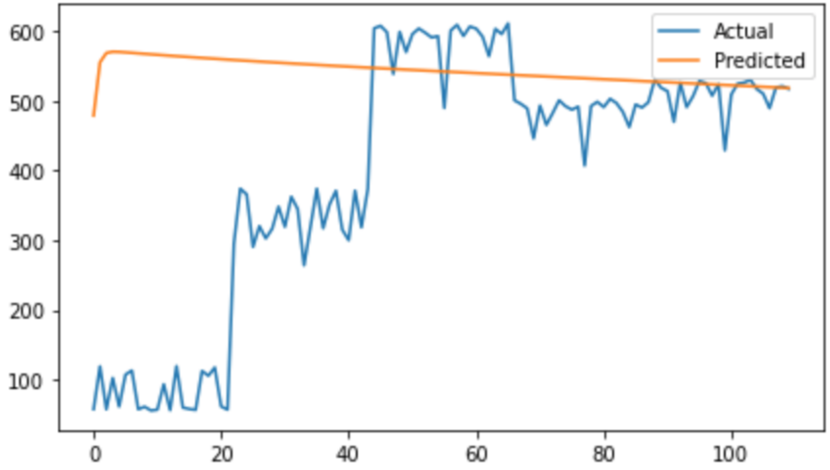
Research question# 3 – Multivariate time series model

MAE: 127.551



Predicting 3 hours ahead

MAE: 163.637



Predicting 1 hour ahead

Research question# 3 – I.I.D approach

Data manipulation: convert time series data into I.I.D format (dummy data for demonstration)

Time	Irradiation	AC output
T day 15:00	1.0	1000
T day 15:15	0.9	920
T day 15:30	1.1	1115
T day 15:45	0.8	788
T day 16:00	0.7	713

Time	AC output	T-1 Irradiation	T-1 AC output	T- 2 Irradiation	T-2 AC output
T day 15:00	1000	1.1	1100	0.9	923
T day 15:15	920	1.0	996	0.9	931
T day 15:30	1115	1.0	1002	0.8	774
T day 15:45	788	0.8	832	0.6	610
T day 16:00	713	0.9	903	0.7	721

Target variable remains the same

Research question# 3 – I.I.D approach

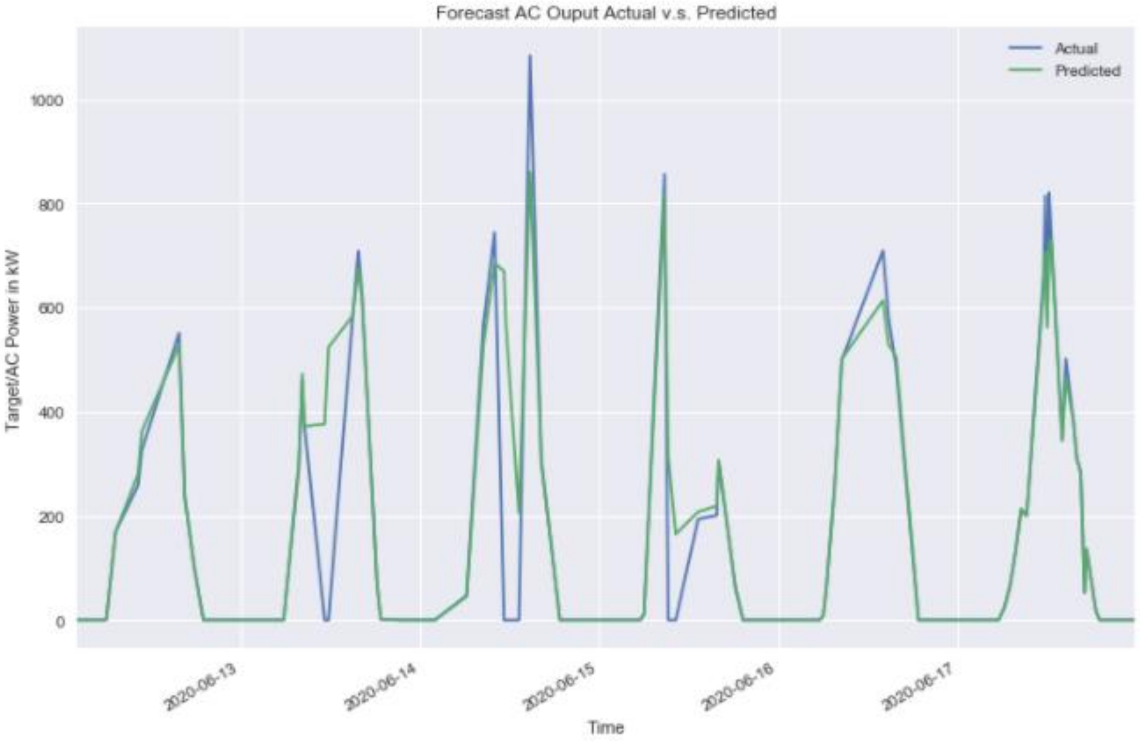
Data manipulation: convert time series data into I.I.D format (dummy data for demonstration)

Time	Irradiation	AC output
T day 15:00	1.0	1000
T day 15:15	0.9	920
T day 15:30	1.1	1115
T day 15:45	0.8	788
T day 16:00	0.7	713

Time	AC output	T-1 Irradiation	T-1 AC output	T- 2 Irradiation	T-2 AC output
T day 15:00	1000	1.1	1100	0.9	923
T day 15:15	920	1.0	996	0.9	931
T day 15:30	1115	1.0	1002	0.8	774
T day 15:45	788	0.8	832	0.6	610
T day 16:00	713	0.9	903	0.7	721

Target variable remains the same

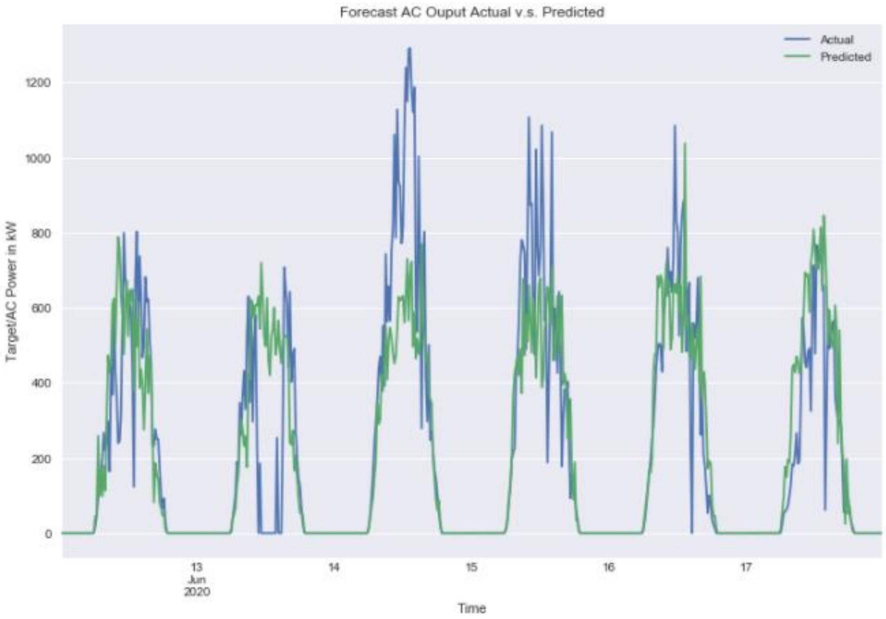
Research question# 3 – I.I.D approach



Predicting T based on T-1, T-2 and T-3 data

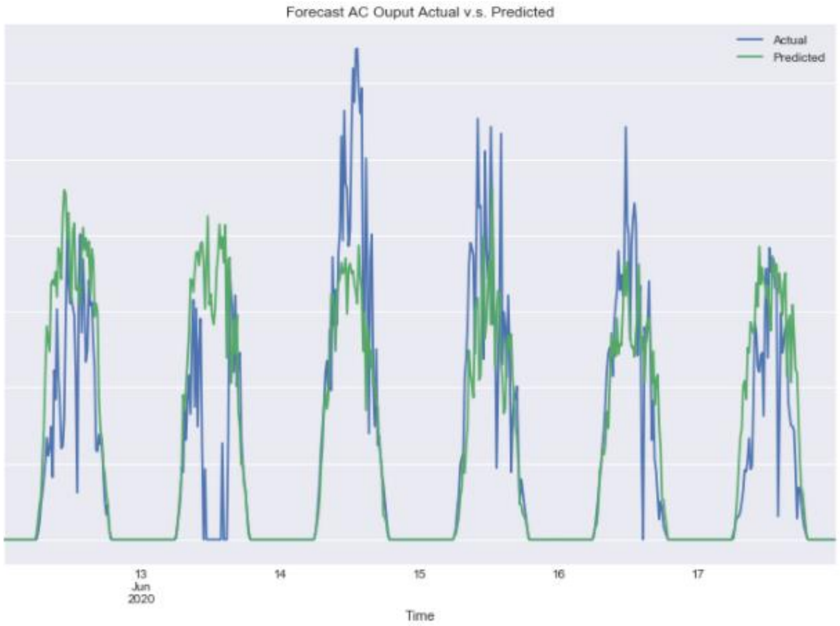
- **MAE:** 57.885
- Random Forest
- With hyperparameter tuning

MAE: 96.71



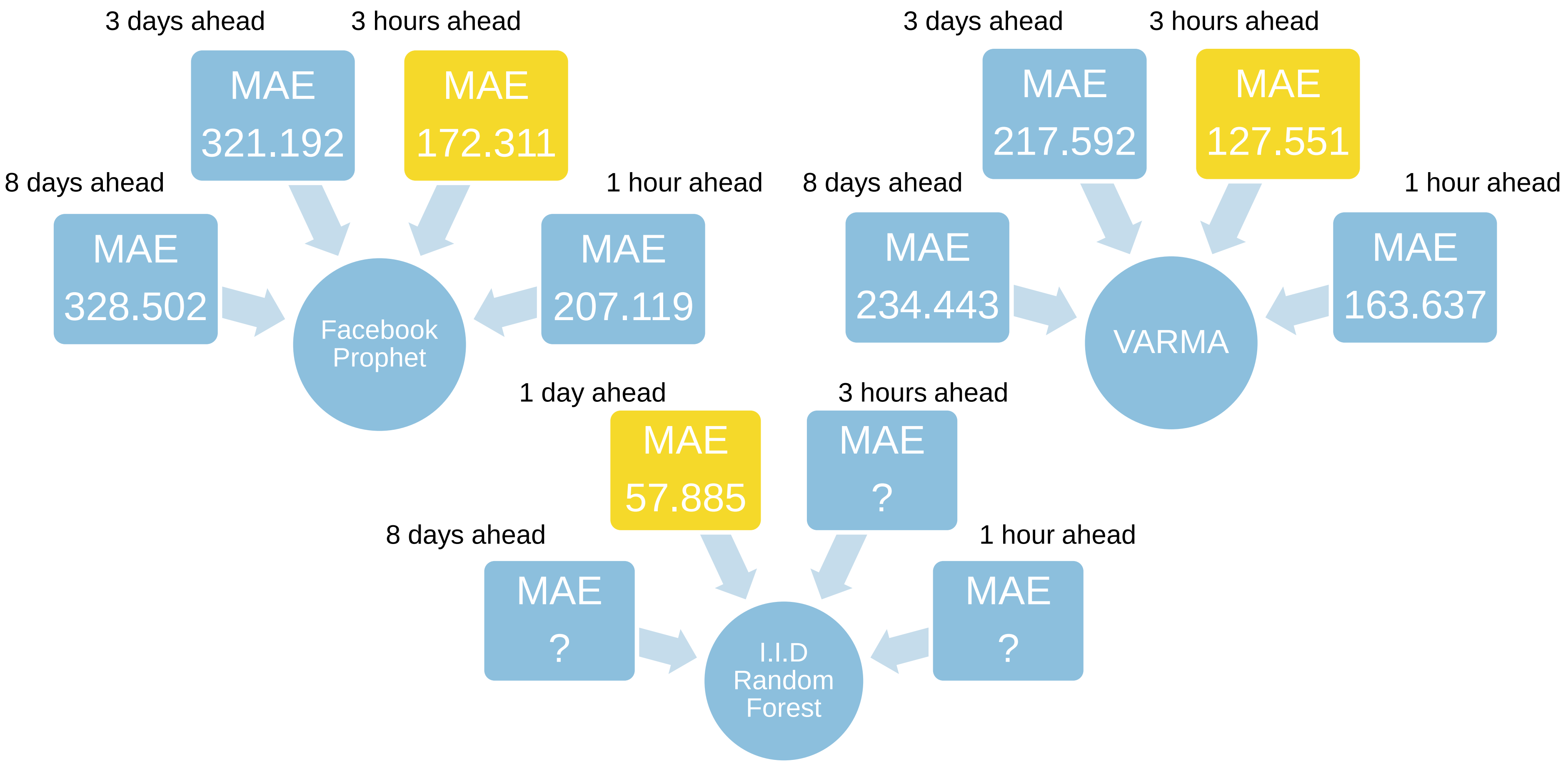
Predicting 1 day ahead

MAE: 111.59



Predicting 3 days ahead

Research question# 3 – Comparing between forecast methods



04

Conclusion and Learning

Conclusion

- ML models are built to identify faulty / underperforming / requiring maintenance inverter in a real-time manner

- Using I.I.D approach to tackle time series problem can be more powerful than dedicated time series models

Learnings

- Identify and excluding outliers are critical in ML practices

- ML could also be a powerful tool to identify outliers