study

July 20, 2021

1 Solar Power Generation Data: Study

1.1 Solar power generation and sensor data for two power plants.

Description This data has been gathered at two solar power plants in India over a 34 day period. It has two pairs of files - each pair has one power generation dataset and one sensor readings dataset. The power generation datasets are gathered at the inverter level - each inverter has multiple lines of solar panels attached to it. The sensor data is gathered at a plant level - single array of sensors optimally placed at the plant.

There are a few areas of concern at the solar power plant -

- Can we predict the power generation for next couple of days? - this allows for better grid management
- Can we identify generation profiles?
- Can we identify the need for panel cleaning/maintenance?
- Can we identify faulty or suboptimally performing equipment?

Link to source

[1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

[2]:
```
dfGen01 = pd.read_csv("/home/zau/Desktop/UFRN_S/TAP3_2021.1/data/source/
 ↪Plant_1_Generation_Data.csv", index_col="SOURCE_KEY")
dfGen02 = pd.read_csv("/home/zau/Desktop/UFRN_S/TAP3_2021.1/data/source/
 ↪Plant_2_Generation_Data.csv", index_col="SOURCE_KEY")
```

1.2 Removing the Plant_ID column (contains the same value throughout the dataset)

[3]:
```
del dfGen01['PLANT_ID']
del dfGen02['PLANT_ID']
```

1.3 Column conversion from DateTime string to DateTime type

[4]:
```
dfGen01['DATE_TIME'] = pd.to_datetime(dfGen01['DATE_TIME'], format='%d-%m-%Y %H:
↪%M')
dfGen02['DATE_TIME'] = pd.to_datetime(dfGen02['DATE_TIME'], format='%Y-%m-%d %H:
↪%M:%S')
```

1.4 Separation of the date and time in different columns

[5]:
```
dfGen01['DATE'] = dfGen01['DATE_TIME'].dt.date
dfGen01['TIME'] = dfGen01['DATE_TIME'].dt.time

dfGen02['DATE'] = dfGen02['DATE_TIME'].dt.date
dfGen02['TIME'] = dfGen02['DATE_TIME'].dt.time
```

2 Initial verification of data structure after conversion

[6]:
```
dfGen01.info(), dfGen01.columns
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 68778 entries, 1BY6WEcLGh8j5v7 to zVJPv84UY57bAof
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   DATE_TIME    68778 non-null  datetime64[ns]
 1   DC_POWER     68778 non-null  float64
 2   AC_POWER     68778 non-null  float64
 3   DAILY_YIELD  68778 non-null  float64
 4   TOTAL_YIELD  68778 non-null  float64
 5   DATE         68778 non-null  object
 6   TIME         68778 non-null  object
dtypes: datetime64[ns](1), float64(4), object(2)
memory usage: 4.2+ MB
```

[6]:
```
(None,
 Index(['DATE_TIME', 'DC_POWER', 'AC_POWER', 'DAILY_YIELD', 'TOTAL_YIELD',
        'DATE', 'TIME'],
       dtype='object'))
```

[7]:
```
dfGen02.info(), dfGen02.columns
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 67698 entries, 4UPUqMRk7TRMgml to xoJJ8DcxJEcupym
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   DATE_TIME    67698 non-null  datetime64[ns]
 1   DC_POWER     67698 non-null  float64
```

```
 2   AC_POWER     67698 non-null  float64
 3   DAILY_YIELD  67698 non-null  float64
 4   TOTAL_YIELD  67698 non-null  float64
 5   DATE         67698 non-null  object
 6   TIME         67698 non-null  object
dtypes: datetime64[ns](1), float64(4), object(2)
memory usage: 4.1+ MB
```

[7]:
```
(None,
 Index(['DATE_TIME', 'DC_POWER', 'AC_POWER', 'DAILY_YIELD', 'TOTAL_YIELD',
        'DATE', 'TIME'],
       dtype='object'))
```

3 Absolute frequency check of data received by sensors

It is noticeable that there is inconsistency in the data

[8]:
```
dfGen01.index.value_counts()
```

[8]:
```
bvBOhCH3iADSZry    3155
1BY6WEcLGh8j5v7    3154
7JYdWkrLSPkdwr4    3133
VHMLBKoKgIrUVDU    3133
ZnxXDlPa8U1GXgE    3130
ih0vzX44oOqAx2f    3130
z9Y9gH1T5YWrNuG    3126
wCURE6d3bPkepu2    3126
uHbuxQJl8lW7ozc    3125
pkci93gMrogZuBj    3125
iCRJl6heRkivqQ3    3125
rGa61gmuvPhdLxV    3124
sjndEbLyjtCKgGv    3124
McdE0feGgRqW7Ca    3124
zVJPv84UY57bAof    3124
ZoEaEvLYb1n2sOq    3123
1IF53ai7Xc0U56Y    3119
adLQvlD726eNBSB    3119
zBIq5rxdHJRwDNY    3119
WRmjgnKYAwPKWDb    3118
3PZuoBAID5Wc2HD    3118
YxYtjZvoooNbGkE    3104
Name: SOURCE_KEY, dtype: int64
```

[9]:
```
dfGen02.index.value_counts()
```

[9]:
```
xoJJ8DcxJEcupym    3259
WcxssY2VbP4hApt    3259
```

```
9kRcWv60rDACzjR    3259
vOuJvMaM2sgwLmb    3259
rrq4fwE8jgrTyWY    3259
LYwnQax7tkwH5Cb    3259
LlT2YUhhzqhg5Sw    3259
q49J1IKaHRwDQnt    3259
oZZkBaNadn6DNKz    3259
PeE6FRyGXUgsRhN    3259
81aHJ1q11NBPMrL    3259
V94E5Ben1TlhnDV    3259
oZ35aAeoifZaQzV    3195
4UPUqMRk7TRMgml    3195
Qf4GUc1pJu5T6c6    3195
Mx2yZCDsyf6DPfv    3195
Et9kgGMDl729KT4    3195
Quc1TzYxW2pYoWX    3195
mqwcsP2rE7J0TFp    2355
NgDl19wMapZy17u    2355
IQ2d7wF4YD8zU1Q    2355
xMbIugepa2P71BB    2355
Name: SOURCE_KEY, dtype: int64
```

4 Initial DataSet View

[10]:
```
dfGen01
```

[10]:
```
                           DATE_TIME  DC_POWER  AC_POWER  DAILY_YIELD  \
SOURCE_KEY
1BY6WEcLGh8j5v7 2020-05-15 00:00:00       0.0       0.0        0.000
1IF53ai7Xc0U56Y 2020-05-15 00:00:00       0.0       0.0        0.000
3PZuoBAID5Wc2HD 2020-05-15 00:00:00       0.0       0.0        0.000
7JYdWkrLSPkdwr4 2020-05-15 00:00:00       0.0       0.0        0.000
McdE0feGgRqW7Ca 2020-05-15 00:00:00       0.0       0.0        0.000
…                               …         …         …            …
uHbuxQJl8lW7ozc 2020-06-17 23:45:00       0.0       0.0     5967.000
wCURE6d3bPkepu2 2020-06-17 23:45:00       0.0       0.0     5147.625
z9Y9gH1T5YWrNuG 2020-06-17 23:45:00       0.0       0.0     5819.000
zBIq5rxdHJRwDNY 2020-06-17 23:45:00       0.0       0.0     5817.000
zVJPv84UY57bAof 2020-06-17 23:45:00       0.0       0.0     5910.000

                 TOTAL_YIELD        DATE      TIME
SOURCE_KEY
1BY6WEcLGh8j5v7    6259559.0  2020-05-15  00:00:00
1IF53ai7Xc0U56Y    6183645.0  2020-05-15  00:00:00
3PZuoBAID5Wc2HD    6987759.0  2020-05-15  00:00:00
7JYdWkrLSPkdwr4    7602960.0  2020-05-15  00:00:00
```

```
McdE0feGgRqW7Ca    7158964.0  2020-05-15  00:00:00
…                        …         …         …
uHbuxQJl8lW7ozc    7287002.0  2020-06-17  23:45:00
wCURE6d3bPkepu2    7028601.0  2020-06-17  23:45:00
z9Y9gH1T5YWrNuG    7251204.0  2020-06-17  23:45:00
zBIq5rxdHJRwDNY    6583369.0  2020-06-17  23:45:00
zVJPv84UY57bAof    7363272.0  2020-06-17  23:45:00

[68778 rows x 7 columns]
```

[11]:
```
dfGen02
```

[11]:
```
                          DATE_TIME  DC_POWER  AC_POWER  DAILY_YIELD  \
SOURCE_KEY
4UPUqMRk7TRMgml 2020-05-15 00:00:00       0.0       0.0  9425.000000
81aHJ1q11NBPMrL 2020-05-15 00:00:00       0.0       0.0     0.000000
9kRcWv60rDACzjR 2020-05-15 00:00:00       0.0       0.0  3075.333333
Et9kgGMDl729KT4 2020-05-15 00:00:00       0.0       0.0   269.933333
IQ2d7wF4YD8zU1Q 2020-05-15 00:00:00       0.0       0.0  3177.000000
…                               …       …         …           …
q49J1IKaHRwDQnt 2020-06-17 23:45:00       0.0       0.0  4157.000000
rrq4fwE8jgrTyWY 2020-06-17 23:45:00       0.0       0.0  3931.000000
vOuJvMaM2sgwLmb 2020-06-17 23:45:00       0.0       0.0  4322.000000
xMbIugepa2P7lBB 2020-06-17 23:45:00       0.0       0.0  4218.000000
xoJJ8DcxJEcupym 2020-06-17 23:45:00       0.0       0.0  4316.000000

                  TOTAL_YIELD        DATE      TIME
SOURCE_KEY
4UPUqMRk7TRMgml  2.429011e+06  2020-05-15  00:00:00
81aHJ1q11NBPMrL  1.215279e+09  2020-05-15  00:00:00
9kRcWv60rDACzjR  2.247720e+09  2020-05-15  00:00:00
Et9kgGMDl729KT4  1.704250e+06  2020-05-15  00:00:00
IQ2d7wF4YD8zU1Q  1.994153e+07  2020-05-15  00:00:00
…                         …         …         …
q49J1IKaHRwDQnt  5.207580e+05  2020-06-17  23:45:00
rrq4fwE8jgrTyWY  1.211314e+08  2020-06-17  23:45:00
vOuJvMaM2sgwLmb  2.427691e+06  2020-06-17  23:45:00
xMbIugepa2P7lBB  1.068964e+08  2020-06-17  23:45:00
xoJJ8DcxJEcupym  2.093357e+08  2020-06-17  23:45:00

[67698 rows x 7 columns]
```

4.1 Add Weekday column

[12]:
```
dfGen01["WEEKDAY"] = dfGen01['DATE_TIME'].dt.dayofweek
dfGen02["WEEKDAY"] = dfGen02['DATE_TIME'].dt.dayofweek
```