

Question : What is the predicted energy output of the solar panels over the next week/month/year?

Dataset

More information about the dataset you can find [here](#).

Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
```

Data Loading

```
# power generation datasets
g1 =
pd.read_csv("/kaggle/input/solar-power-generation-data/Plant_1_Generation_Data.csv")
g1.head()
g2 =
pd.read_csv("/kaggle/input/solar-power-generation-data/Plant_2_Generation_Data.csv")
g2.head()
```

	DATE_TIME	PLANT_ID	SOURCE_KEY	DC_POWER	AC_POWER
0	2020-05-15 00:00:00	4136001	4UPUqMRk7TRMgm1	0.0	0.0
1	2020-05-15 00:00:00	4136001	81aHJ1q11NBPMrL	0.0	0.0
2	2020-05-15 00:00:00	4136001	9kRcWv60rDACzjR	0.0	0.0
3	2020-05-15 00:00:00	4136001	Et9kgGMD1729KT4	0.0	0.0
4	2020-05-15 00:00:00	4136001	IQ2d7wF4YD8zU1Q	0.0	0.0

	DAILY_YIELD	TOTAL_YIELD
0	9425.000000	2.429011e+06
1	0.000000	1.215279e+09
2	3075.333333	2.247720e+09
3	269.933333	1.704250e+06
4	3177.000000	1.994153e+07

```
# sensor datasets
s1 =
pd.read_csv("/kaggle/input/solar-power-generation-data/Plant_1_Weather_Sensor_Data.csv")
s1.head()
s2 =
pd.read_csv("/kaggle/input/solar-power-generation-data/Plant_2_Weather_Sensor_Data.csv")
s2.head()
```

	DATE_TIME	PLANT_ID	SOURCE_KEY	AMBIENT_TEMPERATURE
0	2020-05-15 00:00:00	4136001	iq8k7ZNt4Mwm3w0	27.004764
1	2020-05-15 00:15:00	4136001	iq8k7ZNt4Mwm3w0	26.880811
2	2020-05-15 00:30:00	4136001	iq8k7ZNt4Mwm3w0	26.682055
3	2020-05-15 00:45:00	4136001	iq8k7ZNt4Mwm3w0	26.500589
4	2020-05-15 01:00:00	4136001	iq8k7ZNt4Mwm3w0	26.596148

	MODULE_TEMPERATURE	IRRADIATION
0	25.060789	0.0
1	24.421869	0.0
2	24.427290	0.0
3	24.420678	0.0
4	25.088210	0.0

```
# power generation datasets
def check(data):
    l = []
    columns = data.columns
    for col in columns:
        instances = data[col].count()
        dtypes = data[col].dtype
        unique = data[col].nunique()
        sum_null = data[col].isnull().sum()
        duplicates = data.duplicated().sum()
        l.append([col, dtypes, instances, unique, sum_null,
duplicates])
    data_check = pd.DataFrame(l, columns=["column", "dtype",
"instances", "unique", "sum_null", "duplicates"])
    return data_check

check1 = check(g1)
check2 = check(g2)

output = pd.concat([check1, check2], axis=1)
```

output

```
# g1.info()
# print("\n")
# g1.describe()
```

```
# g2.info()
# print("\n")
# g2.describe()
```

	column	dtype	instances	unique	sum_null	duplicates
column \						
0	DATE_TIME	object	68778	3158	0	0
DATE_TIME						
1	PLANT_ID	int64	68778	1	0	0
PLANT_ID						
2	SOURCE_KEY	object	68778	22	0	0
SOURCE_KEY						
3	DC_POWER	float64	68778	32909	0	0
DC_POWER						
4	AC_POWER	float64	68778	32686	0	0
AC_POWER						
5	DAILY_YIELD	float64	68778	29900	0	0
DAILY_YIELD						
6	TOTAL_YIELD	float64	68778	37267	0	0
TOTAL_YIELD						

	dtype	instances	unique	sum_null	duplicates
0	object	67698	3259	0	0
1	int64	67698	1	0	0
2	object	67698	22	0	0
3	float64	67698	30825	0	0
4	float64	67698	30783	0	0
5	float64	67698	30490	0	0
6	float64	67698	33115	0	0

```
# sensor datasets
```

```
def check(data):
    l = []
    columns = data.columns
    for col in columns:
        instances = data[col].count()
        dtypes = data[col].dtype
        unique = data[col].nunique()
        sum_null = data[col].isnull().sum()
        duplicates = data.duplicated().sum()
        l.append([col, dtypes, instances, unique, sum_null,
duplicates])
    data_check = pd.DataFrame(l, columns=["column", "dtype",
"instances", "unique", "sum_null", "duplicates"])
```