SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

MBA – IT

Specialization – Data Analytics

Section-'A'

A

Project Report

On

Analyzing the solar power generation and predicting the error with Machine Learning algorithms using Python as a tool.

Suraj Shah (23030141083)

CERTIFICATE

This report has been prepared based on my work. Where other published and unpublished source materials have been used, these have been acknowledged. I , Suraj Shah, a MBA IT student of Symbiosis Institute Of Computer Studies and Research, Prn no, 23030141083 have completed the project and submitted on the behalf of the data mining project using machine learning project.

Student Name: Suraj Shah(23030141083)

Date of Submission: 27 Feb 2024

Signature:

ACKNOWLEDGEMENT

It is a matter of profound privilege and pleasure to extend my sense of respect and deepest gratitude to our guide Assistant Professor Dr. Amol D. Vibhute under whose precise guidance and gracious encouragement we had the privilege to work.

We would like to avail this opportunity to thank **Symbiosis Institute Of Computer Studies And Research,** for facilitating such a congenial environment in our department and also his unending encouragement throughout.

We would also like to thank the many people in my college, faculty members, and supporting staff, for always being helpful over the years.

Finally, we would like to express our deepest gratefulness to our parents for their continuous moral support and encouragement. Their love accompanies us wherever we go.

Table of Contents

Abstract

A data mining approach is considered here for a better knowledge discovery process. Solar power generation data is collected every 15 minutes from two plants in India, which includes the values of power generation and sensor data related to weather in the span of 34 days from 15/05/2020 to 17/06/2020. Power generation data is in terms of DC power generation, AC power generation, Daily Yield, and Total Yield whereas weather data consists of Ambient temperature, module temperature, and Irradiation. Utilizing the dataset, an exploratory data analysis is performed to understand the power generation of the plant by using the statistical methods, data visualization, data understanding, data preprocessing, discovery of knowledge.

The data is analysed using the machine learning techniques to predict the future value and calculate the error for better prediction. Here, linear regression and decision tree is used for determining the dependency between parameters which directly impacts on power generation. Python along with the libraries like pandas, matplotlib, sklearn for data mining approach. Pandas is used for importing the data set in python and then it went for data preprocessing, and data understanding. Matplotlib is used for analysing the datasets through different graphs which provide facts for supporting the problem statement.

Finally, all the results were analyzed and discussion was made through the continuous involvement of facts and figures providing a better conclusion for our problem statement.

Chapter 1: Introduction

Solar power plant is the clean, green, and efficient source of electricity. Solar power generation data is collected every 15 minutes from two plants in India, which includes the values of power generation and sensor data related to weather in the span of 34 days from 15/05/2020 to 17/06/2020. Power generation data is in terms of DC power generation, AC power generation, Daily Yield, and Total Yield whereas weather data consists of Ambient temperature, module temperature, and Irradiation.

1. Problem Statement: Solar power plant generates a lot of data in terms of generation and sensor data. However, neither of these data is utilized and analyzed for predicting future outcomes and even increasing its efficiency. So, a dataset is taken from Kaggle which has attributes related to generation and sensor data which is further explored using data mining approaches along with machine learning for the prediction of future values. An exploratory data analysis provides descriptive statistics which again is very useful to know the peak hours. Using a machine algorithm, all the parameters will be analyzed in terms of efficiency by segregating the input parameters and output parameters of the solar power plant.

2. Solution: Data mining is performed using different libraries in Python which helps to solve the analysis of the following:

    - Conversion of AC to DC power: Inverter efficiency.
    - Identifying the Peak hours and maximum power generation time and different statistics.
    - Identifying the losses, and process to recover it.
    - Analyzing the dependencies of different parameters.
    - Calculating the future prediction along with the range of error for better decision-making.
    - Data visualization using graphs for better insights and problem identification.

Chapter 2: Literature Review

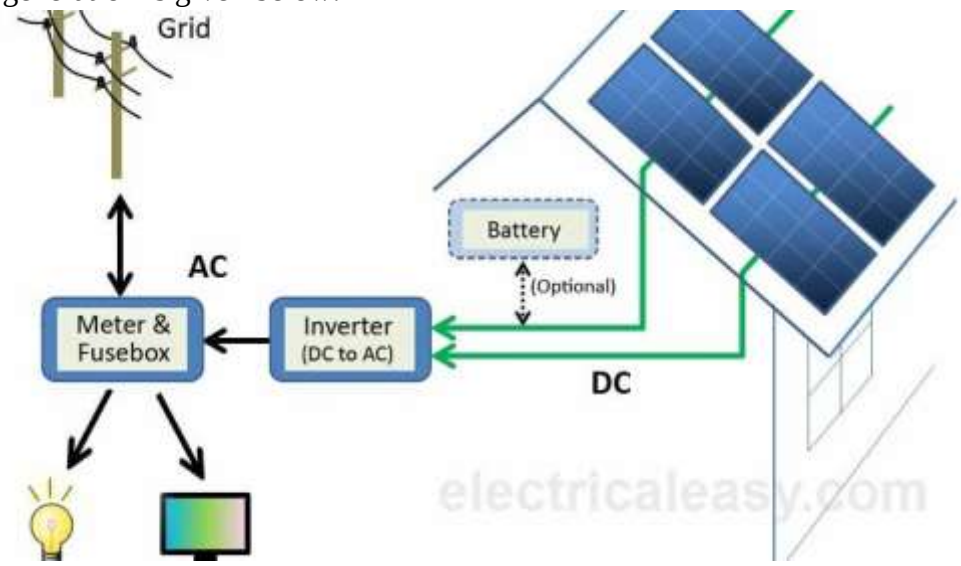
Solar energy has long been recognized as one of the most plentiful and cleanest energy supplies available to humans. With the benefits and advantages of solar energy, several countries throughout the world are on track to achieve success with energy generation utilizing solar systems.

According to the Indian Renewable Energy Development Agency Limited (IREDA), India has an abundance of solar energy that can generate 5,000 trillion kilowatts (kW) of clean electricity. Furthermore, most portions of India receive 300 bright days per year, with solar insolation ranging from 4 to 7 kWh per square meter.

Photovoltaic sun Power has emerged as the best source of green energy in recent years, particularly in countries like India that receive a lot of sun insolation. With the continuing development of efficient PV modules, battery storage, and smart grids, power generation through PV solar plants has gained speed and has a highly bright future.

The solar power plant is sometimes called a photovoltaic (PV) power plant. It is a large-scale photovoltaic facility designed to generate electricity from solar radiation. The solar power plant generates electricity by converting solar energy. As a result, it is a traditional power plant. Solar energy may be utilized directly to generate electricity using solar PV panels. For a better understanding, an explanation of Solar power plants with the integration of the attributes in our dataset is explained below:

a) Solar power generation: Solar power generation is the conversion of sun rays to power through photovoltaic cells. The energy further can be stored in the form of battery storage, thermal storage, and grid supply. The schematic representation of Solar power generation is given below:



b) AC and DC power: Electric supply produce current in two forms one is alternating current and other is direct current. The alternating current transfers periodically in a direction in the form sinusoidal waveform whereas direct current (DC) transfers in the one direction. From the concept explained above, AC power generates the AC current which transfer periodically in the form of sinusoidal wave in a direction periodically and DC power generates the DC current which transfers in the one direction, sources are battery, solar panel.

c) Photovoltaic system: A photovoltaic system is the combination of solar panel in array connected with the inverter and other electrical and mechanical hardware which produces the electricity using the sunlight.

d) Inverter: It is an electrical application which has a power range in terms of power rating and is used to convert direct current from solar panels to alternating current which can be used in our daily works. Inverter has the potential to withstand any electrical failure and is responsible for providing all the sensor data since it is connected to cloud storage.

e) Module: An individual solar cell can produce electricity up to 1 or 2 watts when connected with different cells in a series a module is formed which is also said to be the Solar panel.

f) Daily Yield: The power rating developed by the solar power plant daily according to the capacity of the power plant is said to be the daily yield. The daily yield depends upon the irradiation produced daily according to the weather.

g) Total Yield: The energy gathered from the solar plant when compared with its power can be termed as total yield. The efficiency of the plant can be defined as the power generated by actual power rating including the scenarios like dust, air, and shade in the solar module.

h) Sensors: Sensors are data collectors, and electrical devices which is used to collect different types of data from the surroundings like solar module temperature, ambient temperature through temperature sensor, and wind speed through wind sensor. Here, the weather dataset is collected through different sensors.

i) Irradiation: It is the process by which solar panels are exposed to radiation and moving particles (sun-emitted photons) resulting in ionization. Simply, Irradiance is the solar generation per unit area measured in W/m2.

j) PSH (full sun hours). The number of PSH for the day is the number of hours during which power at a rate of 1kW/m2 would provide an equal amount of energy to the total energy for that day. The words peak sunlight hours and peak sunshine hours can also be used. Irradiation is the total amount of radiant solar energy received per unit area over a specific time, such as a day, month, or year. Insolation is another word for irradiation. Peak sun hours (kWh/m2/day) are a measurement of daily insolation, which is the quantity of solar energy that falls on the surface over time. Irradiance is the solar radiation that strikes a surface at any given time.

Different challenges faced in solar power plants are grid power inefficiencies, environmental challenges, solar panel efficiency, design of solar power plant, shortage of labor, land problems, panel cleanliness, limited technical guidance, and lack of proper weather analysis for building structure.

Chapter 3: Dataset Used

Solar power generation data is collected every 15 minutes from two plants in India, which includes the values of power generation and sensor data related to weather in the span of 34 days from 15/05/2020 to 17/06/2020. Power generation data is in terms of DC power generation, AC power generation, Daily Yield, and Total Yield whereas weather data consists of Ambient temperature, module temperature, and Irradiation.

The dataset consists of information from two plants where data is collected every 15 minutes in two categories one is a generation report, and the other is a weather report. The attributes related to each dataset are explained below:

1. Generation dataset:
    a) Date and timestamp: The date and timestamp represent the observations collected every 15-minute intervals.
    b) Plant ID: This ID represents the plant number and is assigned to identify the plant number.
    c) Source key: The Source Key is defined as the inverter ID for the respective inverter.
    d) AC power: The amount of AC power converted by the Inverter every 15-minute interval. The unit of power is KW.
    e) DC power: The amount of DC power generated by the Inverter every 15-minute interval. The unit of power is KW.
    f) Daily Yield: It is defined as the sum of cumulative power generated on that day and then.
    g) Total Yield: It is defined as the total yield of the inverter at that time.
2. Weather dataset:
    a) Date and Timestamp: The date and timestamp represent the observations collected every 15-minute intervals.
    b) Plant ID: This ID represents the plant number and is assigned to identify the plant number.
    c) Source key: The Source Key is defined as the inverter ID for the respective inverter.
    d) Ambient temperature: It is the temperature defining the surrounding temperature of the Solar plant.
    e) Module temperature: It is defined as the temperature of the module of the solar plant.
    f) Irradiation: It is defined as the process where radiation from the sun is used to generate high energy in the plant. Simply, it is the level of radiation all over the plant on that day.

Chapter 4: Methods Used

1. Importing Libraries in Python:

   Different libraries like pandas for data manipulation , matplotlib for generating graphs, sklearn for performing the machine learning algorithms.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.ensemble import RandomForestRegressor

```

2. Data Understanding:

   The code explain the use of attributes like shape providing the columns and rows of the dataset.

```
# Load the datasets
generation_data =
pd.read_csv('C:\\Users\\ZACOBE\\Desktop\\MBA(IT)\\semester 2nd\\Data
Warehouse\\Solar data\\Plant_1_Generation_Data.csv')
weather_data =
pd.read_csv('C:\\Users\\ZACOBE\\Desktop\\MBA(IT)\\semester 2nd\\Data
Warehouse\\Solar data\Plant_1_Weather_Sensor_Data.csv')

print('The first five rows of the datasets are:',
generation_data.head)
print('-----------------------------------------------------------')
print('The first five rows of the datasets are:', weather_data.head)
print('-----------------------------------------------------------')
print('Number of samples(rows) of
Plant_1_Generation_Data:',generation_data.shape[0])
print('Number of features(Columns) of
Plant_1_Generation_Data:',generation_data.shape[1])
print('-----------------------------------------------------------')
print('Number of samples(rows) of
Plant_2_Generation_Data:',weather_data.shape[0])
print('Number of features(columns) of
Plant_2_Generation_Data:',weather_data.shape[1])
print('-----------------------------------------------------------')
```

3. Data Preprocessing:

   The code is used for analyzing the information of the dataset and finding the missing values using the function shown below in the code.

```
print('-----------------------------------------------------------')
print('The percentage of missing value on Generation_data is :' ,
generation_data.isna().sum().sum()/(generation_data.shape[0]*generatio
n_data.shape[1]) *100 , '%')
print('The percentage of missing value on Weather_data is :' ,
```

```
weather_data.isna().sum().sum()/(weather_data.shape[0]*weather_data.sh
ape[1]) *100 , '%')
print('-----------------------------------------------------------')
print('The information of the generation dataset:')
print(generation_data.info())
print('-----------------------------------------------------------')
print('The information of weather dataset:')
print(weather_data.info())
```

4. Exploratory Data Analysis:

The code represents the basic statistics of the dataset by using the function of describe.

```
print('-----------------------------------------------------------')
print('Number of unique SOURCE_KEY values in generation_data
:',len(generation_data.SOURCE_KEY.unique()))
print('Number of unique SOURCE_KEY values in weather_data
:',len(weather_data.SOURCE_KEY.unique()))
print('-----------------------------------------------------------')
# Convert the 'DateTime' column to a datetime object
generation_data['DATE_TIME'] =
pd.to_datetime(generation_data['DATE_TIME'], format="%d-%m-%Y %H:%M")
weather_data['DATE_TIME'] = pd.to_datetime(weather_data["DATE_TIME"])
# Basic Statistics of data
print(generation_data.describe())
print('-----------------------------------------------------------')
print(weather_data.describe())
print('-----------------------------------------------------------')
```

5. Data Visualization:

   Using different graphs in matplotlib like bar plot, line graph, and pie chart for data visualization.

```
# Plotting the bar graph
plt.figure(figsize=(6, 8))  # Adjust the figure size as needed
plt.bar(generation_data['DATE_TIME'], generation_data['DC_POWER'],
color='red')

# Adding labels and title
plt.xlabel('Date and Time')
plt.ylabel('DC Power')
plt.title('DC Power vs Date and Time')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Show plot
plt.tight_layout()
plt.show()
# plotting the pie chart using source key and DC_Power of every
inverter
power_by_inverter =
generation_data.groupby('SOURCE_KEY')['DC_POWER'].sum()
plt.figure(figsize=(8, 6))  # Adjust figure size if needed
plt.pie(power_by_inverter, labels=power_by_inverter.index,
autopct='%1.1f%%', startangle=140)
plt.title('DC Power Generated by Inverter ID')
plt.axis('equal')
plt.show()
```

```
#creating a line plot to show case the trend s of Ambient and module
temperature over time
plt.figure(figsize=(10, 6))  # Adjust figure size if needed
plt.plot(generation_data['DATE_TIME'], generation_data['AC_POWER'],
label='AC Power')
plt.plot(generation_data['DATE_TIME'], generation_data['DC_POWER'],
label='DC Power')

# Adding labels and title
plt.xlabel('Date')
plt.ylabel('Power')
plt.title('AC vs DC Power Over Time')
plt.legend()

# Show plot
plt.grid(True)  # Add grid for better readability
plt.tight_layout()  # Adjust layout
plt.show()
#creating a line plot to show case the trend s of AC power and DC
power
plt.figure(figsize=(10, 6))  # Adjust figure size if needed
plt.plot(weather_data['DATE_TIME'],
weather_data['AMBIENT_TEMPERATURE'], label='AMBIENT_TEMPERATURE')
plt.plot(weather_data['DATE_TIME'],
weather_data['MODULE_TEMPERATURE'], label='MODULE_TEMPERATURE')

# Adding labels and title
plt.xlabel('Date')
plt.ylabel('Temperature')
plt.title('Ambient vs Module Temperature Over Time')
plt.legend()

# Show plot
plt.grid(True)  # Add grid for better readability
plt.tight_layout()  # Adjust layout
plt.show()
#Merging the generation and weather data together
Main_data = pd.merge(generation_data.drop(columns=['PLANT_ID']),
weather_data.drop(columns=['PLANT_ID', 'SOURCE_KEY']), on='DATE_TIME')
```

6. Machine learning algorithms:

   Machine learning algorithms like linear regression and random forest regression are done for analyzing the solar power dependencies among the parameters present in the dataset.

```
#Merging the generation and weather data together
Main_data = pd.merge(generation_data.drop(columns=['PLANT_ID']),
weather_data.drop(columns=['PLANT_ID', 'SOURCE_KEY']), on='DATE_TIME')
print('Linear Regression')
X = Main_data[['DAILY_YIELD', 'TOTAL_YIELD',
'AMBIENT_TEMPERATURE','MODULE_TEMPERATURE', 'IRRADIATION']]
y = Main_data['AC_POWER']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=.2, random_state=42)

lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

lr_score = lr_model.score(X_test, y_test)
```

```
print(f" Linear Regression Score is {lr_score*100:.4f} %")


y_pred_lr = lr_model.predict(X_test)
R2_Score_lr = round(r2_score(y_pred_lr,y_test) * 100, 2)

print("R2 Score : ",R2_Score_lr,"%")
print('------Random Forest Regression--------')

rfr = RandomForestRegressor()
rfr.fit(X_train,y_train)
y_pred_rfr = rfr.predict(X_test)
R2_Score_rfr = round(r2_score(y_pred_rfr,y_test) * 100, 2)
print("R2 Score : ",R2_Score_rfr,"%")
print('------Predicting the future value-------')
prediction = rfr.predict(X_test)
print(prediction)
print('------cross checking the actual and predicted value------')
cross_checking = pd.DataFrame({'Actual' : y_test , 'Predicted' :
prediction})
print(cross_checking.head())
print('------cross checking error between the actual and predicted
value-----')
cross_checking['Error'] = cross_checking['Actual'] -
cross_checking['Predicted']
print(cross_checking.head(20))
```

Chapter 5: Result and Discussion

5.1 Figures, Charts and Tables

1. The result of generation dataset is shown below for knowing the first five dataset and last five datasets which helps to identify the instances in a dataset helping in a proper data understanding.

```
The first five rows of the datasets are: <bound method NDFrame.head of
0      15-05-2020 00:00   4135001  ...       0.000    6259559.0
1      15-05-2020 00:00   4135001  ...       0.000    6183645.0
2      15-05-2020 00:00   4135001  ...       0.000    6987759.0
3      15-05-2020 00:00   4135001  ...       0.000    7602960.0
4      15-05-2020 00:00   4135001  ...       0.000    7158964.0
...                 ...       ...  ...         ...          ...
68773  17-06-2020 23:45   4135001  ...    5967.000    7287002.0
68774  17-06-2020 23:45   4135001  ...    5147.625    7028601.0
68775  17-06-2020 23:45   4135001  ...    5819.000    7251204.0
68776  17-06-2020 23:45   4135001  ...    5817.000    6583369.0
68777  17-06-2020 23:45   4135001  ...    5910.000    7363272.0
```

2. The result of weather dataset is shown below for knowing the first five dataset and last five datasets which helps to identify the instances in a dataset helping in a proper data understanding.

```
------------------------------------------------------------
The first five rows of the datasets are: <bound method NDFrame.head of
0      5/15/2020 0:00   4135001  ...          22.857507          0.0
1      5/15/2020 0:15   4135001  ...          22.761668          0.0
2      5/15/2020 0:30   4135001  ...          22.592306          0.0
3      5/15/2020 0:45   4135001  ...          22.360852          0.0
4      5/15/2020 1:00   4135001  ...          22.165423          0.0
...                 ...       ...  ...                ...          ...
3177  6/17/2020 22:45   4135001  ...          21.480377          0.0
3178  6/17/2020 23:00   4135001  ...          21.389024          0.0
3179  6/17/2020 23:15   4135001  ...          20.709211          0.0
3180  6/17/2020 23:30   4135001  ...          20.734963          0.0
3181  6/17/2020 23:45   4135001  ...          20.427972          0.0

[3182 rows x 6 columns]>
------------------------------------------------------------
```

3. The dataset number of rows and columns is analyzed by using the code and output is obtained in the form of:

```
Number of samples(rows) of Plant_1_Generation_Data: 68778
Number of features(Columns) of Plant_1_Generation_Data: 7
------------------------------------------------------------
Number of samples(rows) of Plant_2_Generation_Data: 3182
Number of features(columns) of Plant_2_Generation_Data: 6
------------------------------------------------------------
```

4. The missing values is analyzed for better data cleaning purpose. The dataset has no missing values which directly help to predict better result.

```
------------------------------------------------------------
The percentage of missing value on Generation_data is : 0.0 %
The percentage of missing value on Weather_data is : 0.0 %
------------------------------------------------------------
```

5. The information of the generation dataset is shown below as an output were each dataset information is shown below:

```
------------------------------------------------------------
The information of the generation dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 68778 entries, 0 to 68777
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   DATE_TIME    68778 non-null  object
 1   PLANT_ID     68778 non-null  int64
 2   SOURCE_KEY   68778 non-null  object
 3   DC_POWER     68778 non-null  float64
 4   AC_POWER     68778 non-null  float64
 5   DAILY_YIELD  68778 non-null  float64
 6   TOTAL_YIELD  68778 non-null  float64
dtypes: float64(4), int64(1), object(2)
memory usage: 3.7+ MB
None
------------------------------------------------------------
```

6. The information of the weather dataset is shown below as an output were each dataset information is shown below:

```
------------------------------------------------------------
The information of weather dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3182 entries, 0 to 3181
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   DATE_TIME            3182 non-null   object
 1   PLANT_ID             3182 non-null   int64
 2   SOURCE_KEY           3182 non-null   object
 3   AMBIENT_TEMPERATURE  3182 non-null   float64
 4   MODULE_TEMPERATURE   3182 non-null   float64
 5   IRRADIATION          3182 non-null   float64
dtypes: float64(3), int64(1), object(2)
memory usage: 149.3+ KB
None
------------------------------------------------------------
```

7. The number of inverters from generation dataset and weather dataset is shown below where we came to know about the inverter. The basic statistic of dataset is shown below which is obtained from the dataset explaining, the mean, median, mode and count of the values present in the dataset.

```
------------------------------------------------------------
Number of unique SOURCE_KEY values in generation_data : 22
Number of unique SOURCE_KEY values in weather_data : 1
------------------------------------------------------------
                           DATE_TIME   PLANT_ID  ...   DAILY_YIELD   TOTAL_YIELD
count                          68778    68778.0  ...  68778.000000  6.877800e+04
mean   2020-06-01 08:02:49.458256896  4135001.0  ...   3295.968737  6.978712e+06
min              2020-05-15 00:00:00  4135001.0  ...      0.000000  6.183645e+06
25%              2020-05-24 00:45:00  4135001.0  ...      0.000000  6.512003e+06
50%              2020-06-01 14:30:00  4135001.0  ...   2658.714286  7.146685e+06
75%              2020-06-09 20:00:00  4135001.0  ...   6274.000000  7.268706e+06
max              2020-06-17 23:45:00  4135001.0  ...   9163.000000  7.846821e+06
std                              NaN        0.0  ...   3145.178309  4.162720e+05

[8 rows x 6 columns]
------------------------------------------------------------
```

8. The number of inverters from weather dataset and weather dataset is shown below where we came to know about the inverter. The basic statistic of dataset is shown below which is obtained from the dataset explaining, the mean, median, mode and count of the values present in the dataset.

```
                           DATE_TIME  ...  IRRADIATION
count                           3182  ...  3182.000000
mean   2020-06-01 05:52:22.080452608  ...     0.228313
min              2020-05-15 00:00:00  ...     0.000000
25%              2020-05-23 22:48:45  ...     0.000000
50%              2020-06-01 09:52:30  ...     0.024653
75%              2020-06-09 16:56:15  ...     0.449588
max              2020-06-17 23:45:00  ...     1.221652
std                              NaN  ...     0.300836

[8 rows x 5 columns]
```

9. The machine learning algorithm is used here to training and testing the data. All the parameters which are responsible for the power generation are kept as a X and the power generated as Y. Machine learning algorithm is used here for predicting the values which directly resembles the efficiency of the plant.

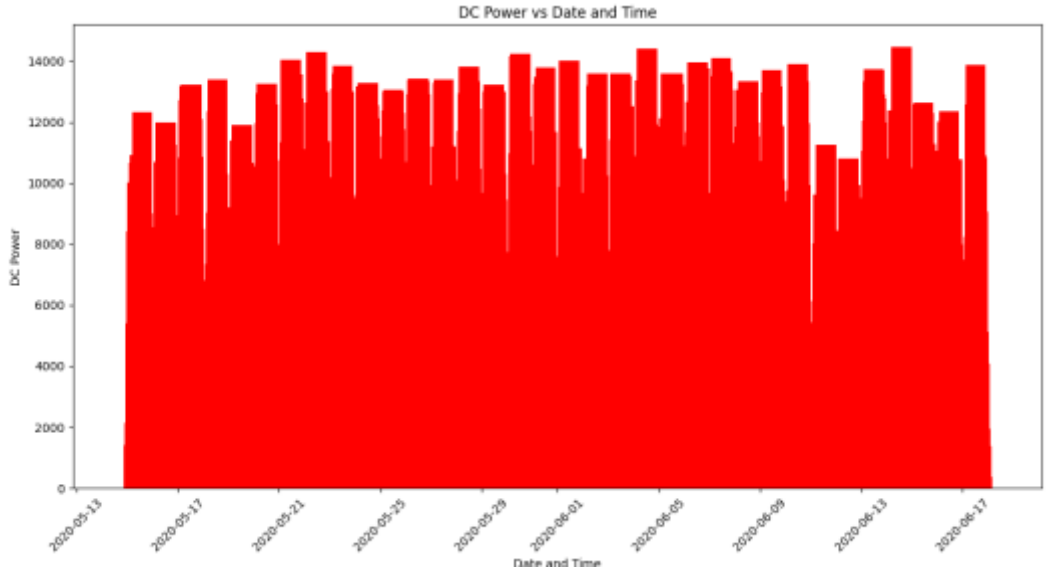
```
Linear Regression
 Linear Regression Score is 98.0130 %
R2 Score :  97.96 %
------Random Forest Regression--------
R2 Score :  99.08 %
------Predicting the future value-------
[1.00133589e+03 1.10625000e-01 5.63137089e+02 ... 0.00000000e+00
 0.00000000e+00 1.06930611e+03]
------cross checking the actual and predicted value------
           Actual    Predicted
48910  873.237500  1001.335893
45151    0.000000     0.110625
17954  613.785714   563.137089
51959    0.000000     0.000000
53225   98.628571    94.809804
```

10. Since, the R2 score is good and near to one we can notice that the predicted value will also be good and unique resulting in the minimization of error. So, the 20 values are shown below in the form of actual, predicted, and error. The error represents the probability of the weather conditions.

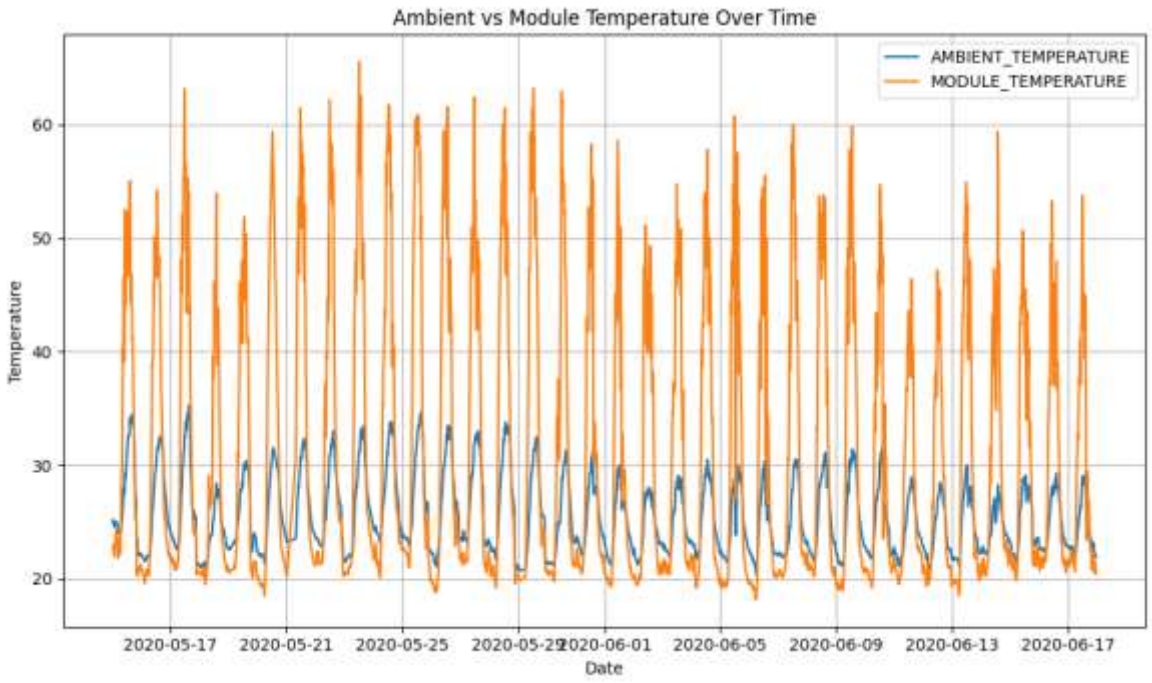
```
------cross checking error between the actual and predicted value-----
            Actual    Predicted       Error
48910   873.237500  1001.335893 -128.098393
45151     0.000000     0.110625   -0.110625
17954   613.785714   563.137089   50.648625
51959     0.000000     0.000000    0.000000
53225    98.628571    94.809804    3.818768
53135   945.028571   914.284768   30.743804
30305   643.385714   662.873780  -19.488065
34495   168.700000   172.563768   -3.863768
29694   372.071429   372.924750   -0.853321
68543     0.000000     0.000000    0.000000
59598  1018.150000  1006.015304   12.134696
25808     0.000000     0.000000    0.000000
6702    174.225000   165.967964    8.257036
27422    51.066667    51.792690   -0.726024
21654     0.000000     0.000000    0.000000
8607      0.000000     0.000000    0.000000
62640     0.000000     0.000000    0.000000
63568   580.942857   609.942429  -28.999571
42081   128.237500   142.600411  -14.362911
5031    897.150000   963.666964  -66.516964
```

5.2 Graphs

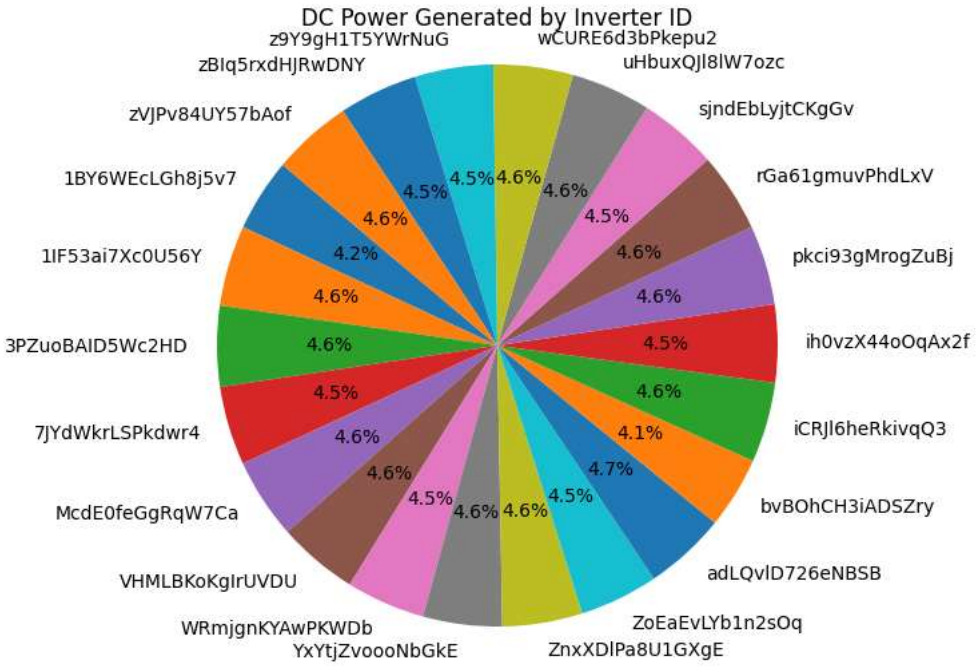
The graph shows the bar plot between the axes DC power and Date_Time. It helps to analyze the daily DC power generation of the plant. On further analyzing, the maximum power generation was on 2020-6-15.



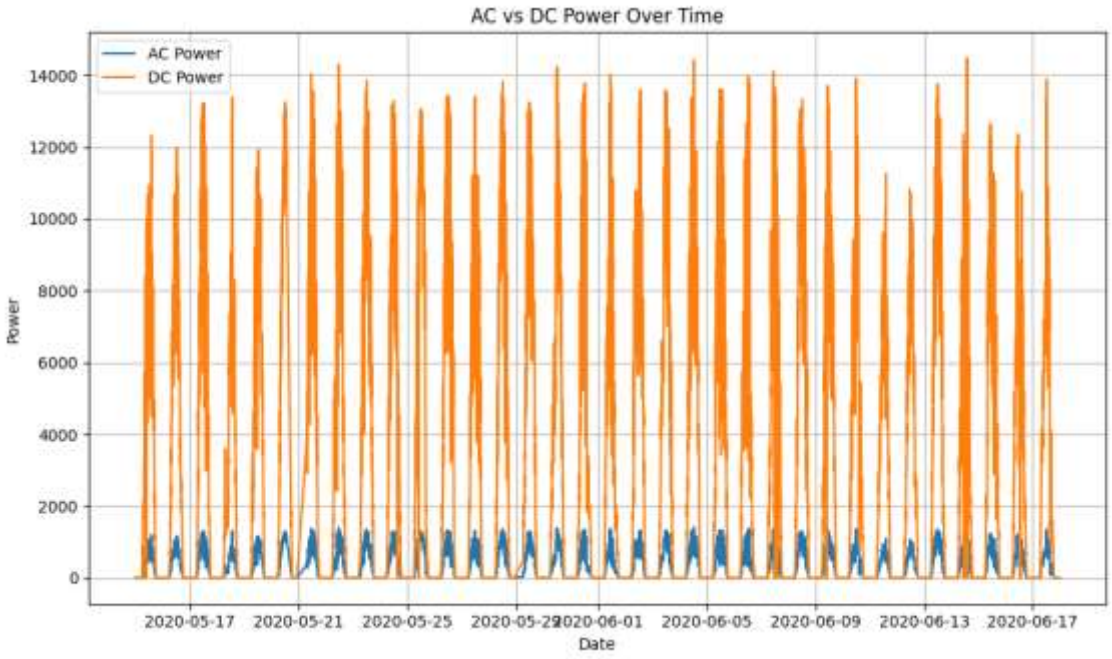
The graph presents below helps to analyze the difference in module and ambient temperature based on date. Ambient temperature is directly proportional to module temperature. If one rises, the other gets risen itself. This graph helps to analyze the dependencies of temperature in power generation. It also helps to analyze the defective module. If the cleaning is required in the module, it can also be verified from the graph. Hence, data visualization is helping to know the root cause of the problem.



The pie chart shown below shows the graphical representation of DC power generated by every inverter present in solar plant. From here, we can analyse this and can see that every inverter is making contribution in Dc power accumulation. It will help in analysing the further solar power problems, failure of the inverter, power loss, and increasing the inverter efficiency.



The graph present below explains about the daily AC and DC power generation in the context of date, here we can see conversion of AC to DC helps by identifying the power conversion rate; efficiency. The line plot helps to analyze the peak production in period and helps to identify the problems that occurred.



Chapter 6: Conclusion

Finally, after performing the data mining approach using the Python as a tool. The generation dataset and weather dataset were part of structured data which show case us a statistical analysis, data visualization, and prediction through Python libraries. The solar power plant was analyzed and solutions were obtained on the form of following:

- Conversion of AC to DC power: Inverter efficiency.
- The Peak hours and maximum power generation time and different statistics were identified.
- The losses were analyzed and root causes were obtained.
- Analyzing the dependencies of different parameters.
- Calculating the future prediction along with the range of error for better decision-making.
- Data visualization using graphs for better insights and problem identification.

Chapter 7: References

- https://www.kaggle.com/datasets/anikannal/solar-power-generation-data/data
- https://en.wikipedia.org/wiki/Solar_power
- https://www.ibm.com/topics/data-mining
- https://www.w3schools.com/python/python_ml_getting_started.asp
- https://www.miquido.com/blog/data-analytics-in-solar-energy/#:~:text=Solar%20energy%20data%20analysis%20allows,and%20improve%20overall%20energy%20production.