Power Consumption Analysis

(Reference: https://ieeexplore.ieee.org/document/8322199)

S Suprabath Reddy
Pradeep Yemula

Abstract: The increasing demand for energy is one of the biggest reasons behind the integration of solar energy into the electric grids or networks. To ensure the efficient use of energy PV systems it becomes important to forecast information reliably. The accurate prediction of solar irradiance variation can enhance the quality of service. This integration of solar energy and accurate prediction can help in better planning and distribution of energy. In this project, we are going to use different machine learning techniques and compare for better prediction of solar panels data

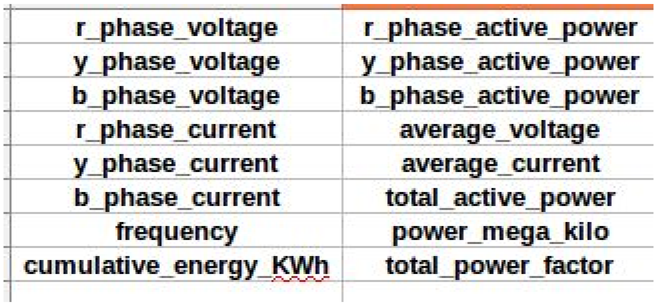
1. Introduction:

Load forecasts have been widely used by the electric power industry. Power distribution companies rely on short- and long-term forecasts at the feeder level to support operations and planning processes, while retail electricity providers make pricing, procurement and hedging decision largely based on the forecasted load of their customers.

Owing to its recent advance, machine learning has spawned a large collection of solar forecasting works. In particular, machine learning is currently one of the most popular approaches for hourly solar forecasting. We are going to use different techniques like SVR, artificial neural network and Long Short Term Memory Network for forecasting solar data in this project.

2. **Project Description:**

2.1 Dataset Information: I have taken the data of IIT Hyderabad Solar panels of academic blocks A and C. I have used one of them to train the data and other for checking the accuracy of model. The data set is at every 12sec for about 8 days between 24th March to 31st March. We have used following variables in the data for prediction -



r_phase_voltage	r_phase_active_power
y_phase_voltage	y_phase_active_power
b_phase_voltage	b_phase_active_power
r_phase_current	average_voltage
y_phase_current	average_current
b_phase_current	total_active_power
frequency	power_mega_kilo
cumulative_energy_KWh	total_power_factor

2.2 Process Description: Initially we analysed all the variables and found out which are the variables which are going to play a major role in prediction. After which we plotted average voltage and current because those are the variable which plays a major role in the total_active_power. It was found that average_voltage changes periodically. After that we plotted total_active_power against day and also with time and found that there is trend every day but the values are changing every day(I mean the bias part). Since it looks like a data which can be predicted using time series analysis we have used Long Short Term Memory network for prediction along with the ANN and SVR. The machine learning techniques have been taken from the paper which was shared by sir.

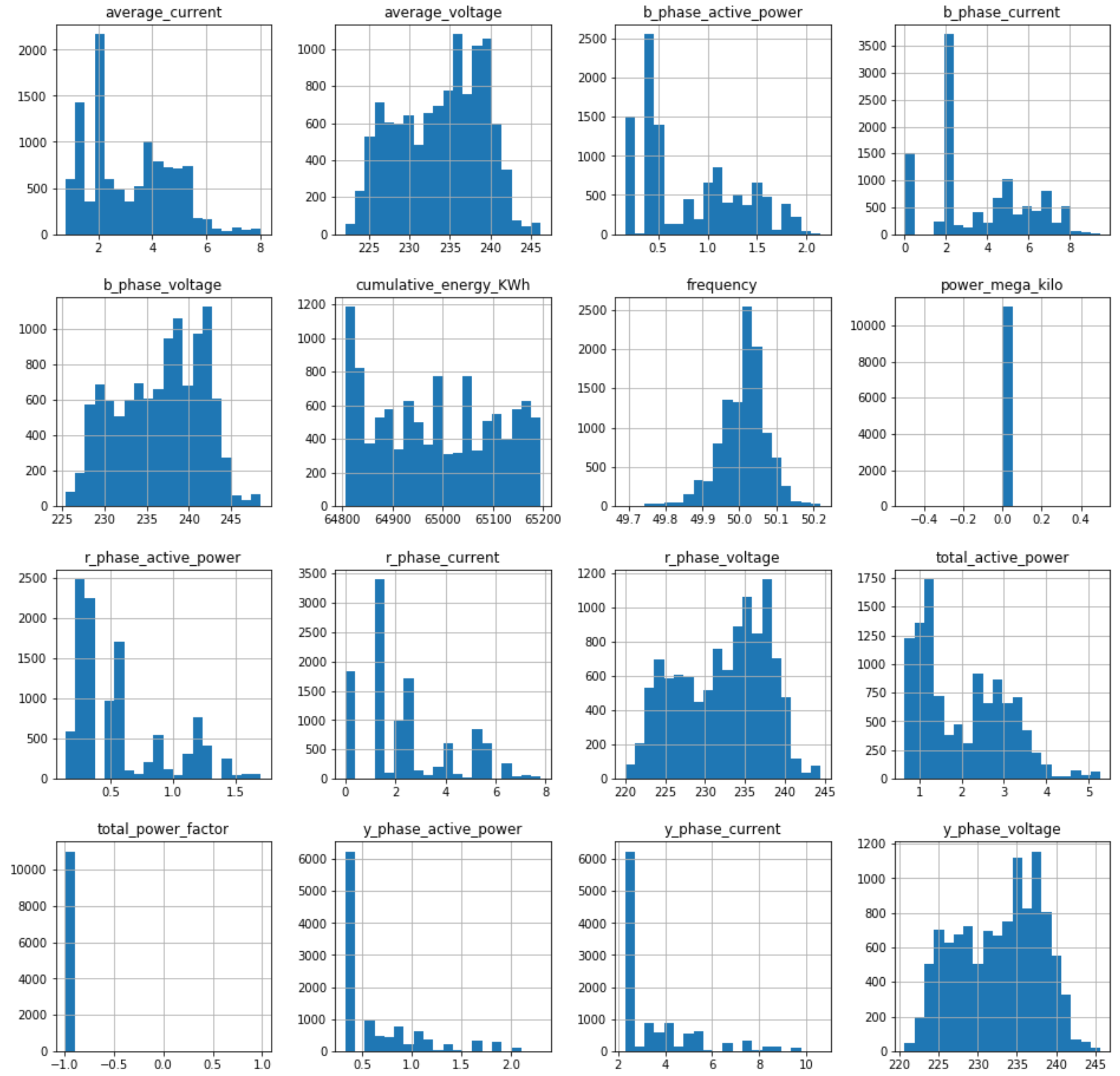
3. **Procedure:** Complete code is in the folder shared with you.

- Histograms of each variable in dataset

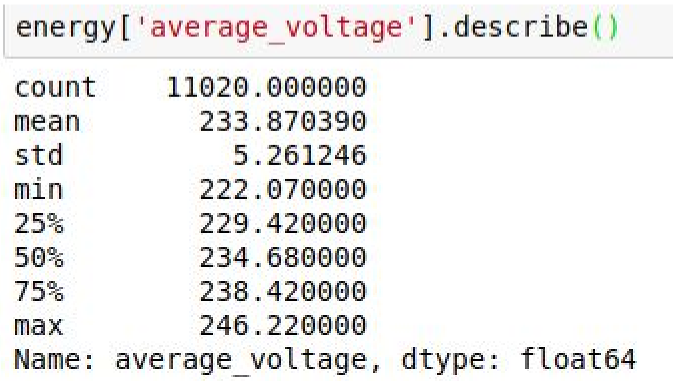
```
energy = pd.read_csv("IIT_A_Emergency_Panel_11_4.csv")

histograms = energy.hist(figsize=(16, 16), bins=20)

plt.show()
```



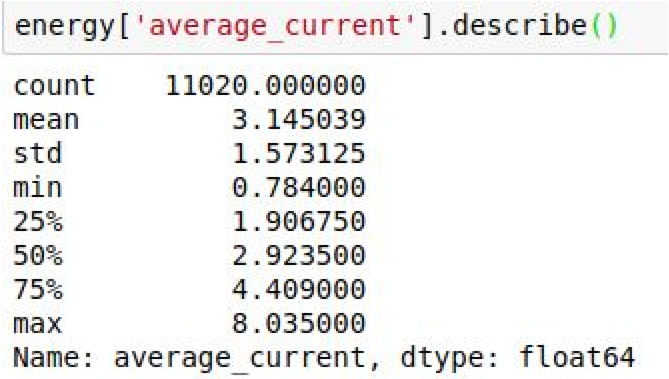
- Describing statistically average_voltage



```
energy['average_voltage'].describe()
```

```
count    11020.000000
mean       233.870390
std          5.261246
min        222.070000
25%        229.420000
50%        234.680000
75%        238.420000
max        246.220000
Name: average_voltage, dtype: float64
```

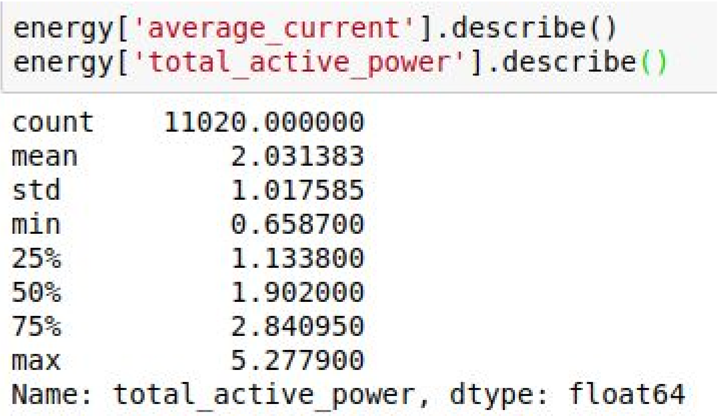
- Describing statistically average current



```
energy['average_current'].describe()
```

```
count    11020.000000
mean         3.145039
std          1.573125
min          0.784000
25%          1.906750
50%          2.923500
75%          4.409000
max          8.035000
Name: average_current, dtype: float64
```

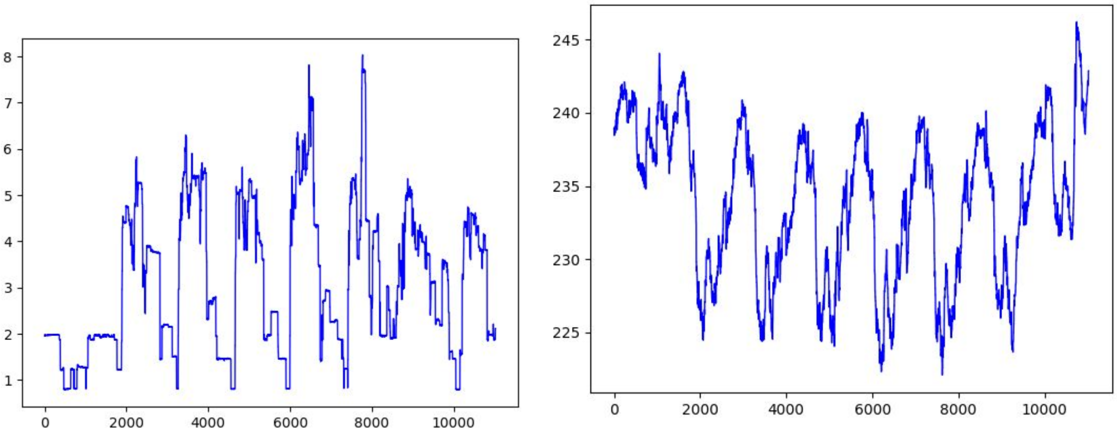
- Describing statistically average active power

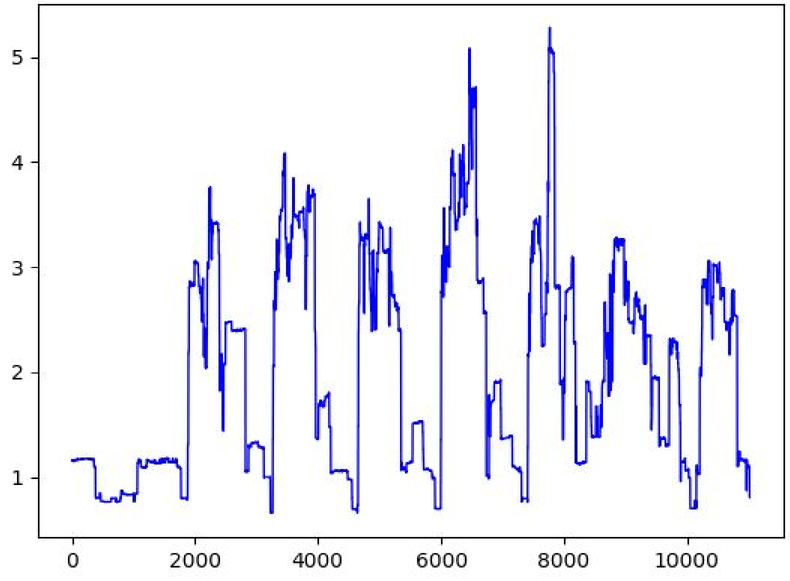


```
energy['average_current'].describe()
energy['total_active_power'].describe()
```

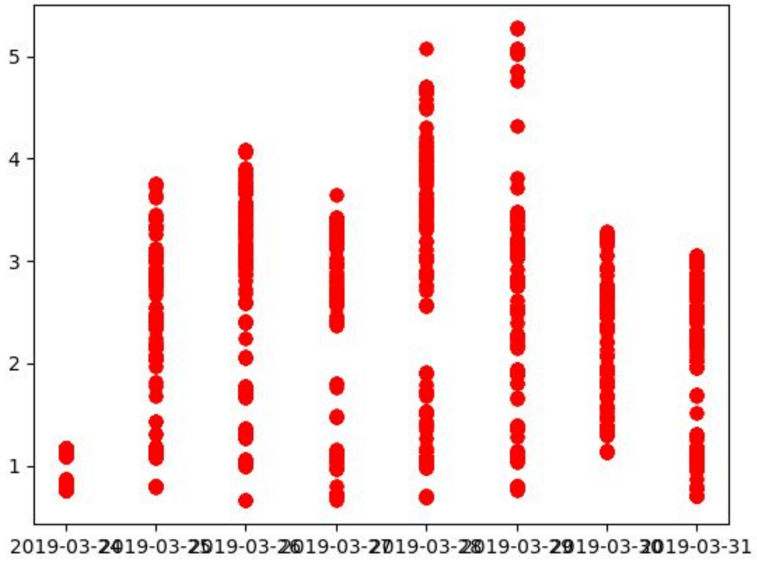
```
count    11020.000000
mean         2.031383
std          1.017585
min          0.658700
25%          1.133800
50%          1.902000
75%          2.840950
max          5.277900
Name: total_active_power, dtype: float64
```

- Plots to find out how average voltage, average current and average active power changing with time

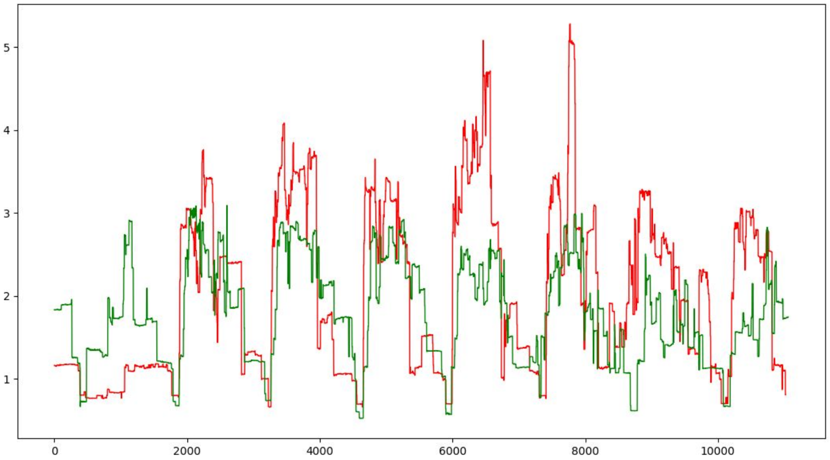


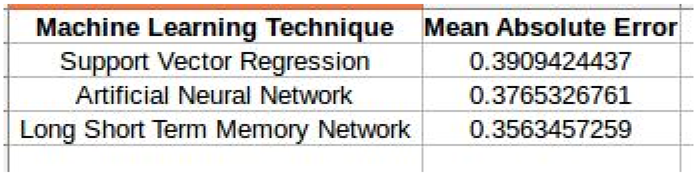


- How total active power changing with the day and within a day.



- Final Result -





Machine Learning Technique	Mean Absolute Error
Support Vector Regression	0.3909424437
Artificial Neural Network	0.3765326761
Long Short Term Memory Network	0.3563457259

4. **Problems faced:** The major problems I faced were -
   - The major problem I faced was lack of knowledge of many techniques which were there in research paper. It took lot of my efforts to know this techniques and implement it. Some of the techniques were not even supported by my laptop because it needs a better processor
   - Also, it was very difficult to predict because margin of error was very small and values were low.
   - There was also lot of bad data which I was not able to remove because of which prediction were not fully accurate

5. **Learnings/Conclusion:** I have learnt a lot from this project. It was my first time experience with the data where there is a seasonal trend. It was great to learn new machine learning techniques for data which is kind of periodic. There were various techniques I followed but the best technique remains to be a LSTM which I cam to know through the research project shared with me. It could be done in much better way with modification of it with much better facilities and techniques.

Appendix: We only need python 3.5 (https://www.python.org/downloads/release/python-350/) for running the code. We also need to install various libraries of python like numpy, scipy which is needed for running the code. Also, need to install keras and tensorflow (https://www.tensorflow.org/install) for running the deep learning techniques code.

Apart from that,, you need to install a Jupyter Notebook to see python code in .ipynb files which has pictures of all outputs. (https://jupyter.readthedocs.io/en/latest/install.html)