Synopsis

The given data was for a solar PV plant. The data was a time series data with power and weat her information from `2017-10-01 00:00:00 to 2019-09-30 23:00:00 local time.`

1. Data Pre-processing-

- The datafile named ‘power_actuals’ was having a large number of null values and the columns named ‘ghi’ and ‘gti’ were having a large number of null values as well as zero values were there.
- As the ‘ghi’ and ‘gti’ are both location based data and the plant location is unknown to us, the best strategy was to drop these columns altogether.
- The power generation of the PV model is noted every 15 mins so there are total 96 observations per day.
- In weather_actuals dataset there were a total 13619 entries with a time span of 1 entry per hour. So 24 entries per day.
- However both the datasets were for the same period of time and hence I used **down sampling of the power data from minutes to hours**.
- Also, the weather dataset was having some missing dates and thus I used **Forward Filling with mean method to impute.**

2. Feature Engineering-

- Firstly I separated the weather data into object and non object types. So that it is easy to work with.
- There were some features that we had to drop initially and most of them were duplicate datetimes and raw indexes.
- For the categorical features in the object data I used frequency encoding to map the datapoints into a smaller and relevant scale.
- After feature encoding the graphs were useful for the purpose of finding out the category that was repeating the most in the dataset.
- From these plots ‘clear day’ were the most repeated category.
- There were some values in the dataset which were given a value of **-9999**. These were nothing but the NaN values, most probably produced by the IoT sensors. These were also taken care of in the code.
- After doing all these feature engineering the correlation matrix showed that only **17** features from the weather dataset were relevant.
- Also after cleaning and looking at the correlation matrix it was clear that there a very low probability of snow and precipitation at the plant site. **That’s why I drew a conclusion that the solar plat might be present in dessert type conditions.**

3. Train-Test-Validation Split-

- I used-
  train_data=df2.loc['2017-10-01':'2019-07-30']
  val_data=df2.loc['2019-08-01':'2019-08-31']
  test_data=df2.loc['2019-09-01':'2019-09-30']
- Thus the splitting was done with respect to datetime index.

4. Feature Scaling-

- Standard Scaler is used for scaling the data.
- As we know it gives us a range from -1 to 1.
- I would like to recall here that frequency encoding worked best in this scenario because the features were not increased and hence dimensionality remained same while reducing the range as well.

5. LSTM Modelling-

- I have used stacked LSTM inside RNN architecture for the purpose of building this NN
- The activation function used for the model is ReLU
- Dense layers were also added in the architecture.
- The initial model was having 3 stacked layers with 124 neurons for first 2 layers and 64 for the third layer. With dense layers set to 0.

6. Model Comparison after Tuning-

- After building the model I did hyperparameter tuning. The best model which precisely followed the test dataset was with the result table ‘power_pred1”.
- This can be seen from the graph as well.
- I have forecasted the power by using the weather forecast for 27 days of October 2019.