Indian Institute of Technology Hyderabad

Energy Forecasting of Home Appliances

S Suprabath Reddy

Pradeep Yemula

Abstract: As the popularity of home automation and the cost of electricity grow around the world, energy conservation has become a higher priority for many consumers. With a number of smart meter devices available for your home, you can now measure and record overall household power draw, and then with the output of a machine learning model, accurately predict individual appliance behavior simply by analyzing meter data. To provide a comprehensive overview of the current research and to identify challenges, this project conducts an application-oriented review of smart meter data analytics. We will review the different machine learning techniques and their performance.

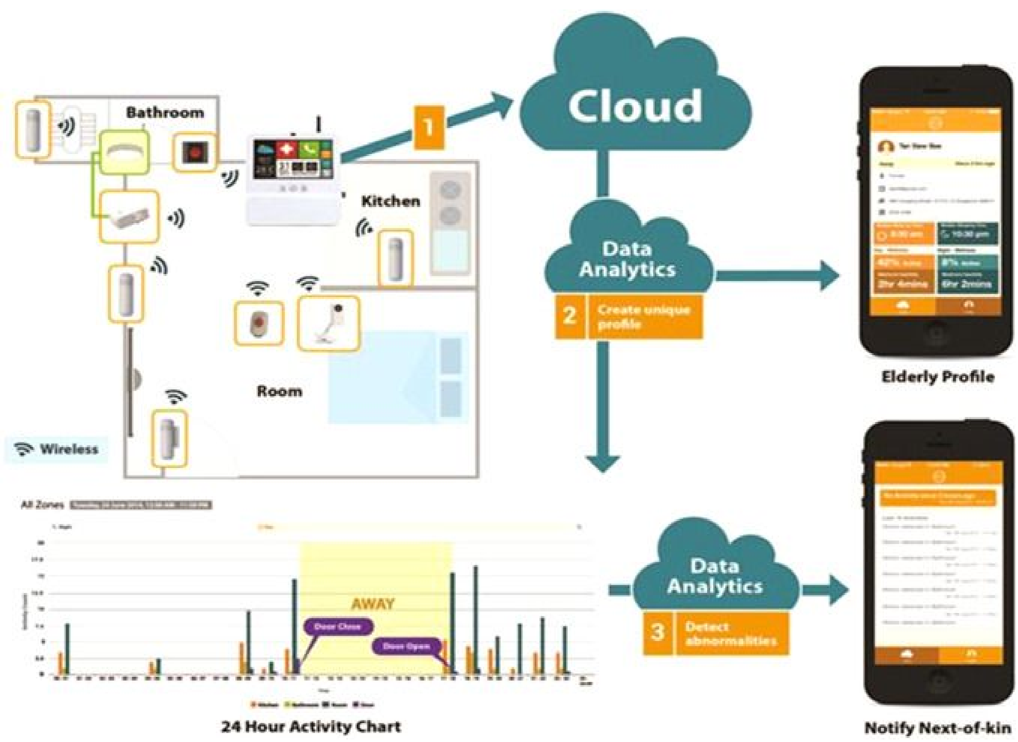
1. Introduction:

The modern power management systems face different challenges to manage the power in an efficient manner.The major drawback associated with power management systems is mostly their manual operation. For making the power management systems more efficient and reliable, the automation of these systems is very compulsory. The energy management systems are made fully automatic with the introduction of the concept of smart grids.The major components of the power management system, including sensors, actuators, and other components are integrated into the smart grid for bringing enhancements in the overall functionalities of power management systems.

With the increase in dependency on electrical appliances, the consumer doesn't know why his electricity bill is being too high than expected. An obvious reason for this being that most of the appliances are still consuming power even though it's not required. The provider of electricity also must manually go to every home for checking the final bill. The answer for each of these problems is to keep a track of the consumers electricity consumption.

The present undertaking **measuring electricity consumption of appliances and monitoring them using IoT and machine learning** addresses the issues of consumers and also focuses on the conservation of natural resources. As very few work have been developed in this area, major challenge is to develop meaningful and performant feature extraction on top of which the machine learning algorithms could be developed.

The values of electricity consumed by individual appliances at our home are retrieved on the mobile application which is taken from the cloud with the help of the WIFI module. These values are taken from the phase wire are categorized into different appliances by using the Machine Learning Algorithms. The application first trains for the existing appliances and then constantly shows the values of the electricity consumed. Finally, after studying the usage patterns of the devices the application turns off a device by user permission if it’s currently not in use.



2. **Project Description:**

2.1. Dataset Information: I have taken the data from UCI Machine Learning Repository which has a use collection of open source data. I have taken appliances energy prediction dataset where the data has been collected using different sensors. The data set is at 10 min for about 4.5 months. The house temperature and humidity conditions were monitored with a wireless sensor network. Then, the wireless data was averaged for 10 minutes periods. Weather from the nearest airport weather station and merged together with the experimental data sets using the date and time column.

NAME	DESCRIPTION	UNIT
	Features	
T1	Kitchen Temperature	°C
T2	Living Room Temperature	°C
T3	Laundry Room Temperature	°C
T4	Office Temperature	°C
T5	Bathroom Temperature	°C
T6	Temperature outside Building (North)	°C
T7	Ironing Room Temperature	°C
T8	Teenager Room Temperature	°C
T9	Parents Room Temperature	°C
T_out	Outside Temperature (Weather Station)	°C
T_dewpoint	Dewpoint Temperature (Weather Station)	°C
RH_1	Kitchen Humidity	%
RH_2	Living Room Humidity	%
RH_3	Laundry Room Humidity	%
RH_4	Office Humidity	%
RH_5	Bathroom Humidity	%
RH_6	Humidity outside Building (North)	%
RH_7	Ironing Room Humidity	%
RH_8	Teenager Room Humidity	%
RH_9	Parents Room Humidity	%
RH_out	Outside Humidity (Weather Station)	%
Pressure	Outside Pressure (Weather Station)	mm Hg
Wind speed	Outside Windspeed (Weather Station)	m/s
Visibility	Visibility (Weather Station)	km

2.2. <u>Process Description:</u> Initially, I analysed all the variables in the dataset and how they are varying with the time. Also, how the appliances energy consumption changing with time and trends which it is following. After which I have done feature extraction to get better results.

I have tried to find out different correlated variables so that there won't be any duplication for a better performance. We have found out that (T6,T_out) and (R3,RH4) are correlated variables and we have removed T6 and RH4 from dataset for analysing using various machine learning techniques.

We have applied different machine learning techniques like Linear Regression, Ridge Regression, Extra Trees Regressor and deep neural networks after extraction of needed features. Deep neural networks has performed far better than all these regression techniques even better than time series model techniques.

3. **Procedure:**

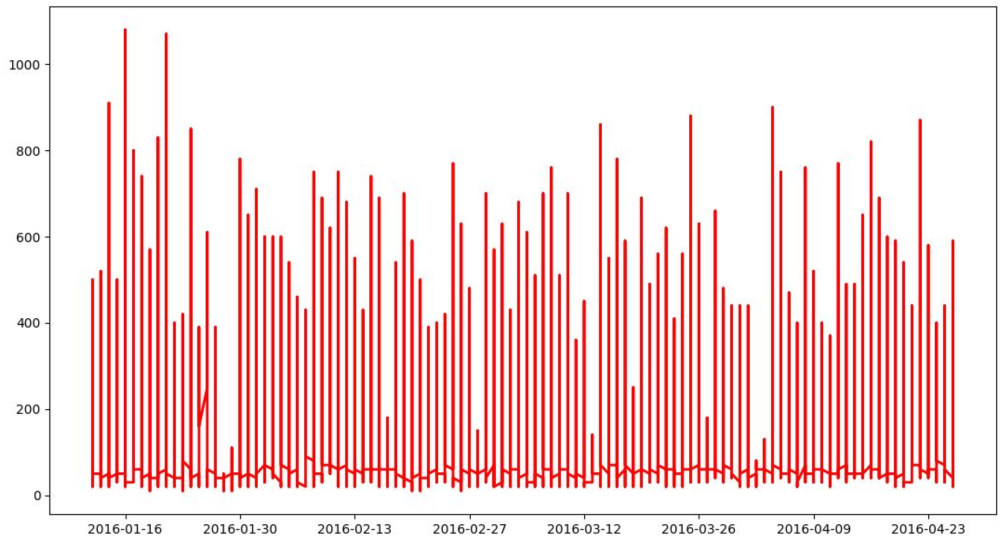
- Calling all the required libraries. Also printing total no.of training data and also no.of variables in dataset

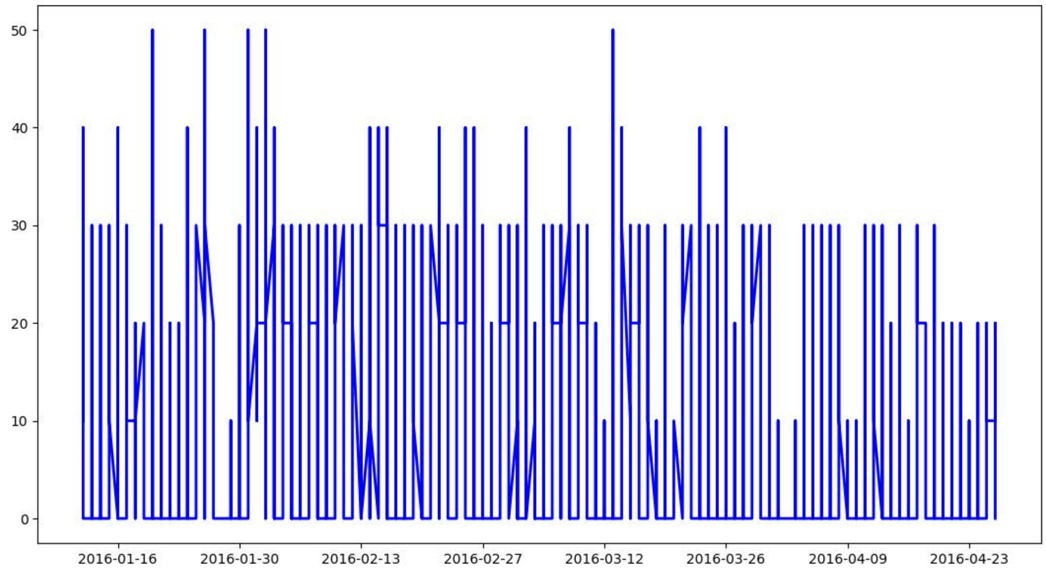
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn import linear_model


train = pd.read_csv("training_data.csv")
test = pd.read_csv("testing_data.csv")

f = open('result.csv', 'w')
f.write("Machine Learning Technique,Mean Absolute Error\n")

print("Number of instances in dataset = {}".format(train.shape[0]))
print("Total number of columns = {}".format(train.columns.shape[0]))
print("Column wise count of null values:-")
print(train.isnull().sum())
```

```
Number of instances in dataset = 15264
Total number of columns = 27
```

- We are checking how the appliances energy and lights energy change with the date in all the 5 months.

```
train['date'] = pd.to_datetime(train['date'])
train['just_date'] = train['date'].dt.date

plt.plot(train['just_date'],train['Appliances'],color = 'red', linewidth=2, linestyle="-" )
plt.show()
plt.plot(train['just_date'],train['lights'],color = 'blue', linewidth=2, linestyle="-" )
plt.show()

train = train.drop(["just_date","date","lights"], axis=1)
```





- Describing all the temperature variables measured by sensor

```
train[temp_cols].describe()
```

	T1	T2	T3	T4	T5	T6	T7	T8	T9
count	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000
mean	21.152644	19.645445	21.575486	20.153001	18.958487	6.152260	19.564421	21.474501	18.767674
std	1.225767	1.520170	1.512331	1.530607	1.311563	4.711561	1.597267	1.717919	1.445634
min	16.790000	16.100000	17.200000	15.100000	15.330000	-6.065000	15.390000	16.306667	14.890000
25%	20.463333	18.600000	20.500000	19.200000	18.000000	2.863333	18.500000	20.390000	17.790000
50%	21.290000	19.566667	21.633333	20.290000	18.890000	6.090000	19.533333	21.700000	18.600000
75%	22.033333	20.633333	22.600000	21.230000	20.000000	9.390000	20.890000	22.730000	20.050000
max	24.100000	24.600000	27.600000	23.760000	22.967778	21.290000	23.566667	25.200000	23.840000

- Describing all the humidity variables measure by sensors

```
train[rho_cols].describe()
```

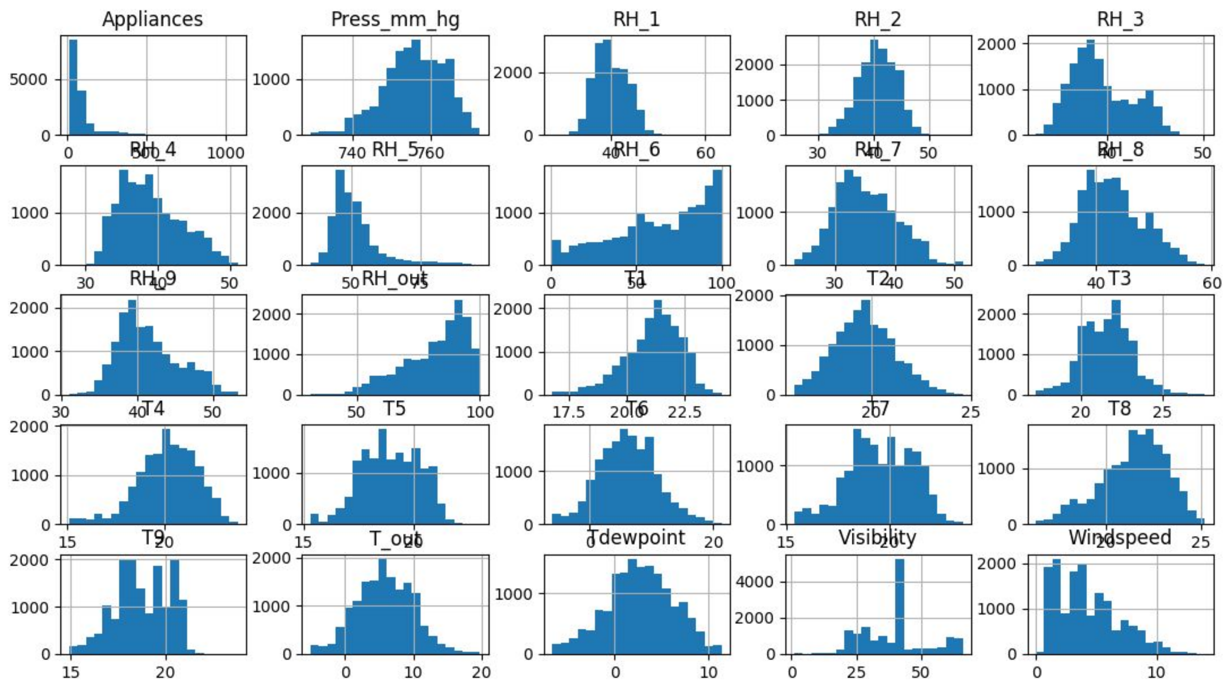
	RH_1	RH_2	RH_3	RH_4	RH_5	RH_6	RH_7	RH_8	RH_9
count	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000
mean	40.147432	40.648161	39.443947	39.011256	51.414795	63.654182	35.152151	42.917032	41.555932
std	3.549504	3.367031	3.159183	4.258235	8.955720	27.641212	5.011035	5.246468	3.994293
min	27.733333	25.763333	32.626667	27.660000	35.363333	1.000000	23.200000	29.600000	31.033333
25%	37.500000	38.392500	37.100000	35.566667	45.790000	44.696667	31.426667	39.036500	38.590000
50%	39.666667	40.693333	38.790000	38.400000	49.183611	68.026667	34.590000	42.431389	40.805714
75%	42.790000	43.200000	41.860000	41.966667	54.000000	87.866667	38.530000	46.360000	44.090000
max	63.360000	56.026667	50.163333	51.090000	96.321667	99.900000	51.400000	58.780000	53.326667

- Describing all the external weather variables

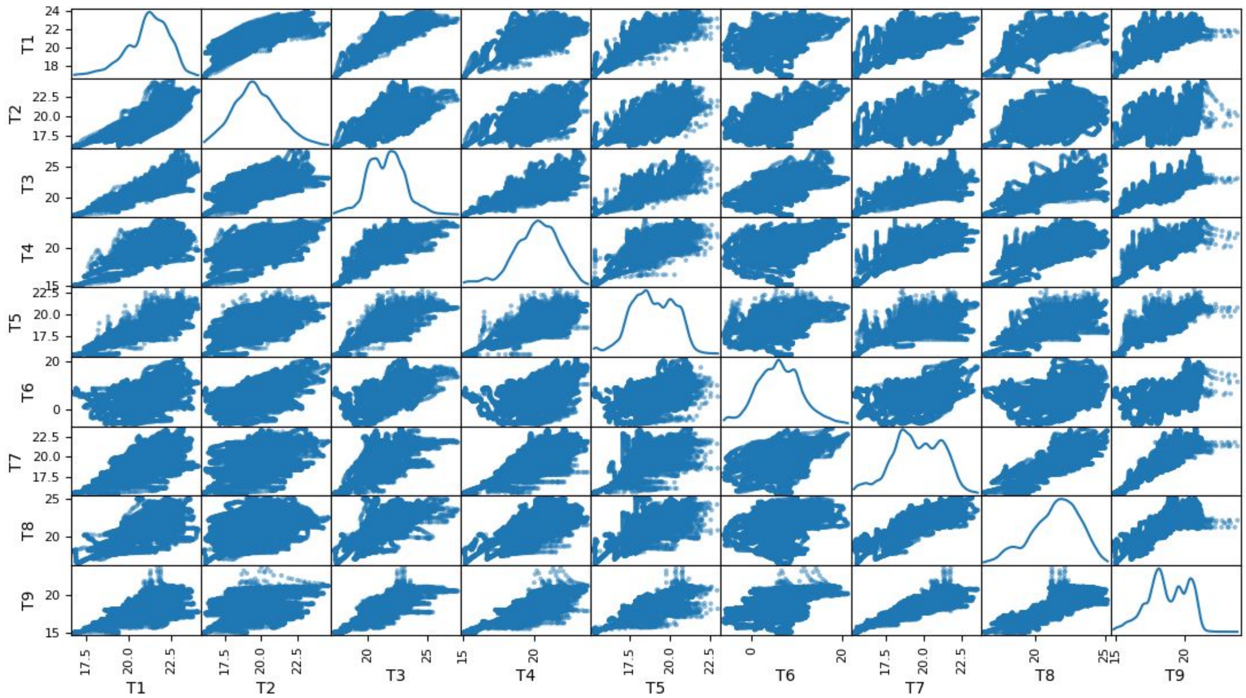
```
train[weather_cols].describe()
```

	T_out	Tdewpoint	RH_out	Press_mm_hg	Windspeed	Visibility
count	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000	15264.000000
mean	5.870869	2.744176	81.826815	755.479809	4.290881	38.687303
std	4.171170	3.557876	12.939027	7.828836	2.575784	12.519245
min	-5.000000	-6.600000	31.000000	729.300000	0.000000	1.000000
25%	2.933333	0.400000	73.166667	750.533333	2.000000	29.000000
50%	5.800000	2.800000	85.333333	756.100000	4.000000	40.000000
75%	8.900000	5.266667	92.000000	761.600000	6.000000	40.000000
max	19.700000	11.400000	100.000000	772.300000	14.000000	66.000000

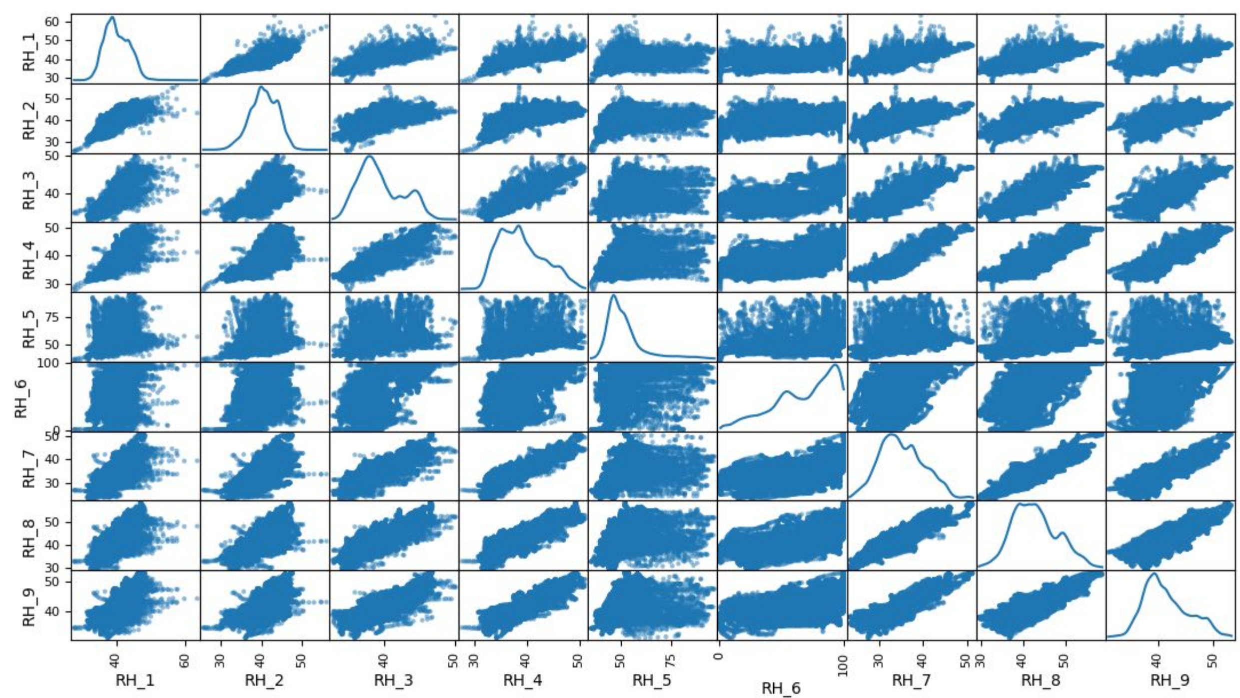
- Histogram all the columns in the dataset.



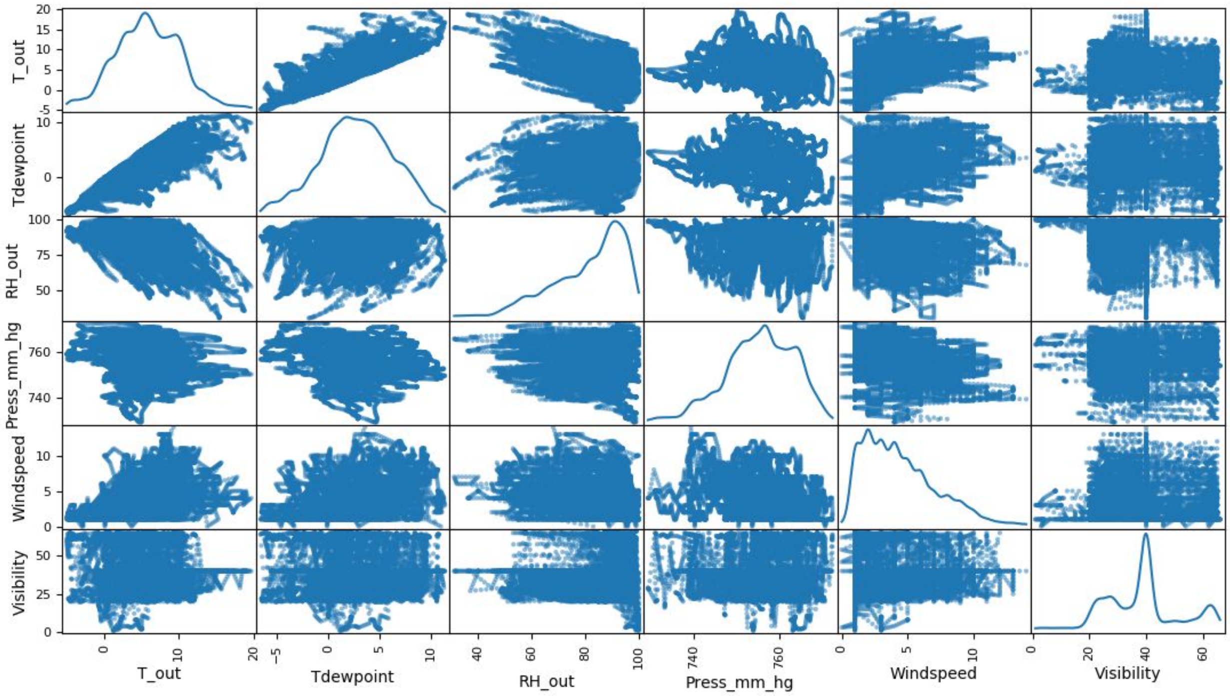
- Scattering plot of how temperature variables are correlated with each other.



● Scattering plot of how humidity variables are correlated with each other.

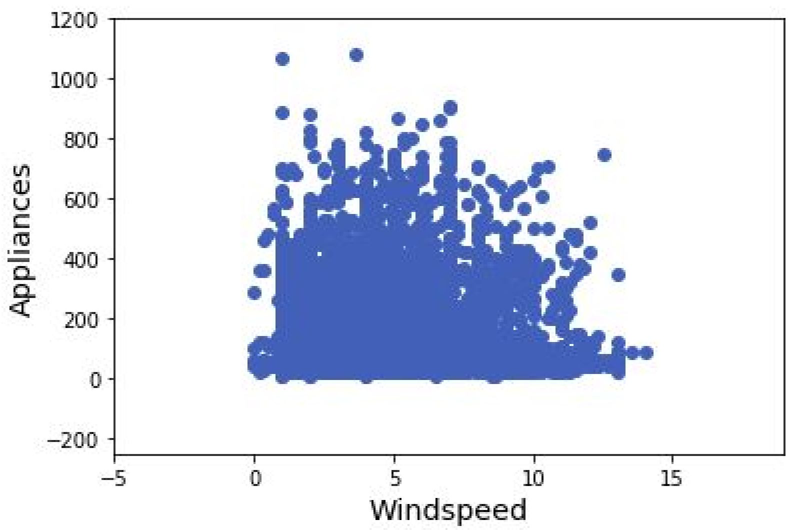


● Scattering plot of how external weather variables are correlated with each other.



- Plot between Appliances and Wind Speed to find out if there is any correlation

```
plt.xlabel("Windspeed", fontsize='x-large')
plt.ylabel("Appliances", fontsize='x-large')
plt.xlim(-5, train.Windspeed.max() + 5)
plt.ylim(-250, 1200)
plt.scatter(train["Windspeed"], train["Appliances"])
plt.show()
```



- Finding the percentage of appliances which are in range of 0-200 Wh because to find out whether high values are outliers or not.Also finding correlation coefficient between T7 and T9 because they seem to be correlated.

```
print("Percentage of dataset in range of 0-200 Wh")
print("{:.3f}%".format((train[train.Appliances <= 200]["Appliances"].count()*100.0) /train.shape[0]))
```

```
Percentage of dataset in range of 0-200 Wh
89.891%
```

```
from scipy.stats import pearsonr

#Calculate the coefficient and p-value between T7 and T9
corr_coef, p_val = pearsonr(train["T7"], train["T9"])
print("Correlation coefficient : {}".format(corr_coef))
print("p-value : {}".format(p_val))
```

```
Correlation coefficient : 0.8871912999102088
p-value : 0.0
```

- Finding out all the correlated pairs in the dataset

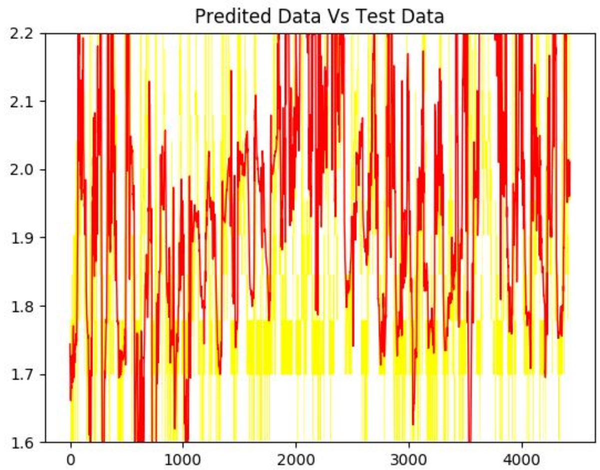
```
from itertools import combinations

for pair in combinations(train.columns, 2):
    col_1, col_2 = pair
    corr_coef, p_val = pearsonr(train[col_1], train[col_2])
    # Check for high correlation
    if corr_coef > 0.9 or corr_coef < -0.9:
        print("Column pair : {}, {}".format(*pair))
        print("Correlation coefficient : {}".format(corr_coef))
        print("p-value : {}".format(p_val))
```

```
Column pair : RH_3, RH_4
Correlation coefficient : 0.9025064570996248
p-value : 0.0
Column pair : T6, T_out
Correlation coefficient : 0.9722317157145699
p-value : 0.0
```

- After which I have run all the machine learning algorithms which are in the code file shared. I am attaching results of all techniques individually.

Machine Learning Technique	Mean Absolute Error
Linear Regression	7.1620058591
Ridge Regression	7.1620058591
Extra Trees Regressor	7.1491790723
Deep Neural Network	6.5456917995

- Final Result - Prediction data vs Test Data



4. **Problems faced:** The major problems I faced were -
    - It was very difficult to analyse the data because many of them being highly correlated and it was difficult to extract the features which are actually needed.
    - The major problem was lack of research work done in this area, it was difficult to get different machine learning techniques and their applications. So, it was most of the general techniques which was needed to apply for the data.
    - The other major problem I faced was applying different machine learning techniques since it was my first time practically applying on a practical data.
    - Apart from that, I was not able to predict the instance which has an high randomness and is an outlier.

5. **Learnings/Conclusion:** I have learnt a lot from this project. I have learnt how the IoT with machine learning could change the energy efficiency in future. It was great to keep machine learning techniques on energy consumption data because of very high randomness of data. It was great to see how data of energy or power behaves. It was also good to see the seasonal changes and trends in data with which we could save huge amount of energy. There were various techniques I followed but the best again remains to be deep learning techniques itself. But it could be done in much better way with modification of it but it is also important to come up with such techniques for energy data.

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)