Capstone Project

Appliance Energy Prediction

Appliance Energy Prediction

In this time of global uncertainty one thing is clear the world needs energy -- and in increasing quantities to support economic and social progress and build a better quality of life, in particular in developing countries. But even in today’s time there are many places especially in developing world where there are outages . These outages are primary because of excess load consumed by appliances at home . Heating and cooling appliances takes most power in house. In this project we will be analyzing the appliance usage in the house gathered via home sensors .All readings are taken at 10 mins intervals for 4.5 months . The goal is to predict energy consumption by appliances . In the age of smart homes , ability to predict energy consumption can not only save money for end user but can also help in generating money for user by giving excess energy back to Grid (in case of solar panels usage).

Problem Statement:

- We should predict Appliance energy consumption for a house based on factors like temperature, humidity & pressure . In order to achieve this, we need to develop a supervised learning model using regression algorithms. Regression algorithms are used as data consist of continuous features and there are no identification of appliances in dataset.

Dataset Information:

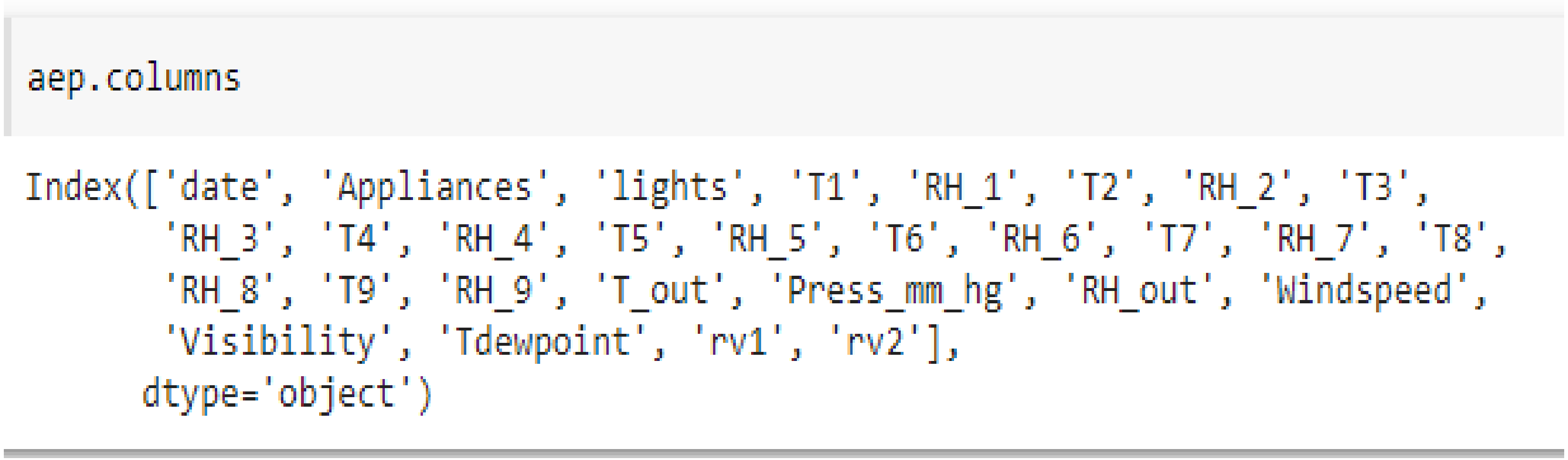
- The data set is at 10 min for about 4.5 months. The house temperature and humidity conditions were monitored with a ZigBee wireless sensor network. Each wireless node transmitted the temperature and humidity conditions around 3.3 min. Then, the wireless data was averaged for 10 minutes’ periods. The energy data was logged every 10 minutes with m-bus energy meters. Weather from the nearest airport weather station (Chèvres Airport, Belgium) was downloaded from a public data set from Reliable Prognosis (rp5.ru), and merged together with the experimental data sets using the date and time column. Two random variables have been included in the data set for testing the regression models and to filter out non predictive attributes (parameters). The dataset has 19375 instances and 29 attributes including predictors and target variable. The training data provided by author contains 14803 instances and testing data contains 4932 instances.

Attribute Information

1. date time year-month-day hour:minute:second
2. Appliances, energy use in Wh
3. lights, energy use of light fixtures in the house in Wh
4. T1, Temperature in kitchen area, in Celsius
5. RH_1, Humidity in kitchen area, in %
6. T2, Temperature in living room area, in Celsius
7. RH_2, Humidity in living room area, in %
8. T3, Temperature in laundry room area
9. RH_3, Humidity in laundry room area, in %
10. T4, Temperature in office room, in Celsius
11. RH_4, Humidity in office room, in %
12. T5, Temperature in bathroom, in Celsius
13. RH_5, Humidity in bathroom, in %
14. T6, Temperature outside the building (north side), in Celsius
15. RH_6, Humidity outside the building (north side), in %
16. T7, Temperature in ironing room , in Celsius
17. RH_7, Humidity in ironing room, in %
18. T8, Temperature in teenager room 2, in Celsius
19. RH_8, Humidity in teenager room 2, in %
20. T9, Temperature in parents room, in Celsius
21. RH_9, Humidity in parents room, in %
22. To, Temperature outside (from Chievres weather station), in Celsius
23. Pressure (from Chievres weather station), in mm Hg
24. RH_out, Humidity outside (from Chievres weather station), in %
25. Wind speed (from Chievres weather station), in m/s
26. Visibility (from Chievres weather station), in km
27. Tdewpoint (from Chievres weather station), Â°C
28. rv1, Random variable 1, nondimensional
29. rv2, Random variable 2, nondimensional

Where indicated, hourly data (then interpolated) from the nearest airport weather station (Chievres Airport, Belgium) was downloaded from a public data set from Reliable Prognosis, rp5.ru. Permission was obtained from Reliable Prognosis for the distribution of the 4.5 months of weather data.

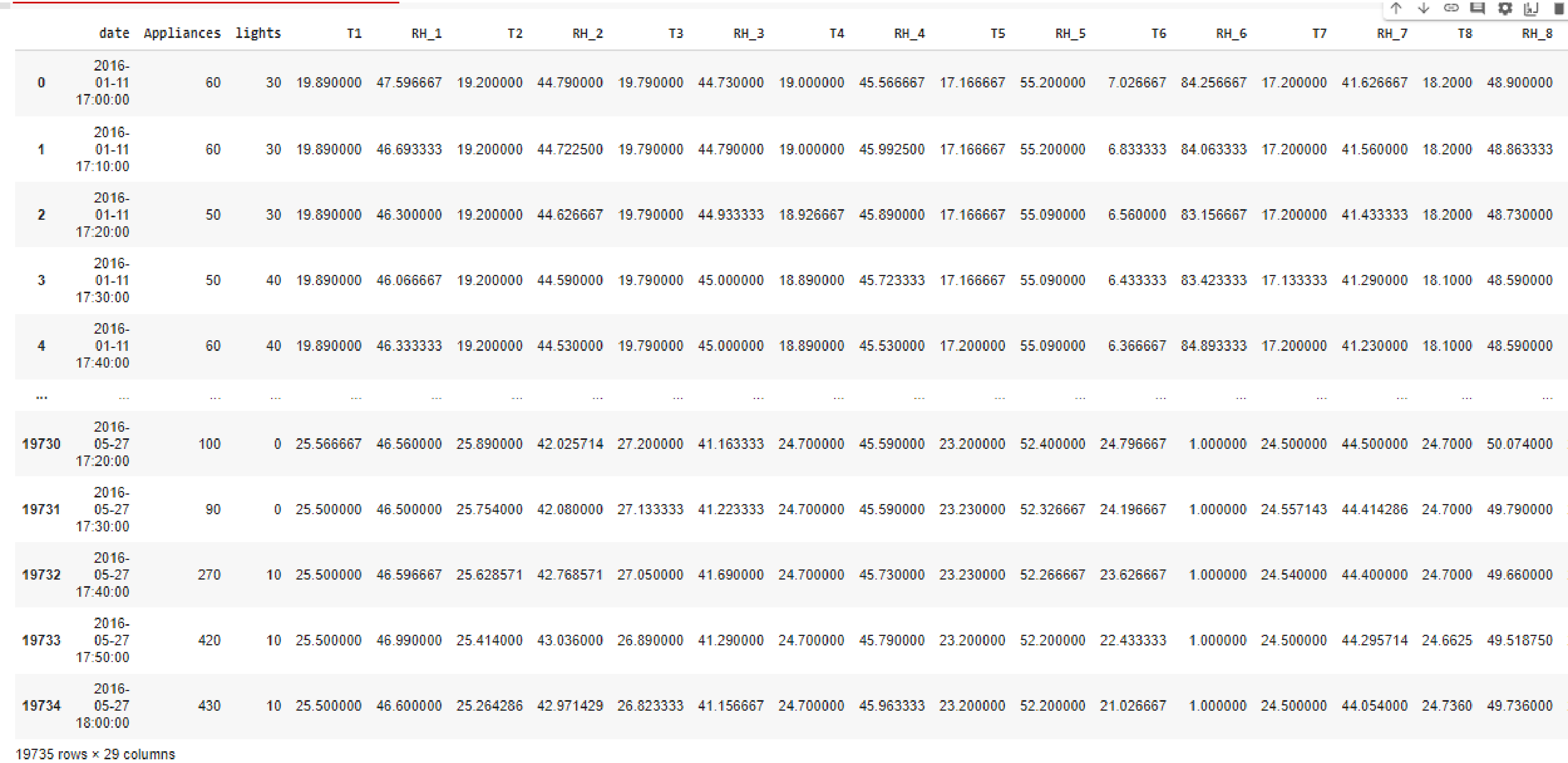
Columns Name:



```
aep.columns
```

```
Index(['date', 'Appliances', 'lights', 'T1', 'RH_1', 'T2', 'RH_2', 'T3',
       'RH_3', 'T4', 'RH_4', 'T5', 'RH_5', 'T6', 'RH_6', 'T7', 'RH_7', 'T8',
       'RH_8', 'T9', 'RH_9', 'T_out', 'Press_mm_hg', 'RH_out', 'Windspeed',
       'Visibility', 'Tdewpoint', 'rv1', 'rv2'],
      dtype='object')
```

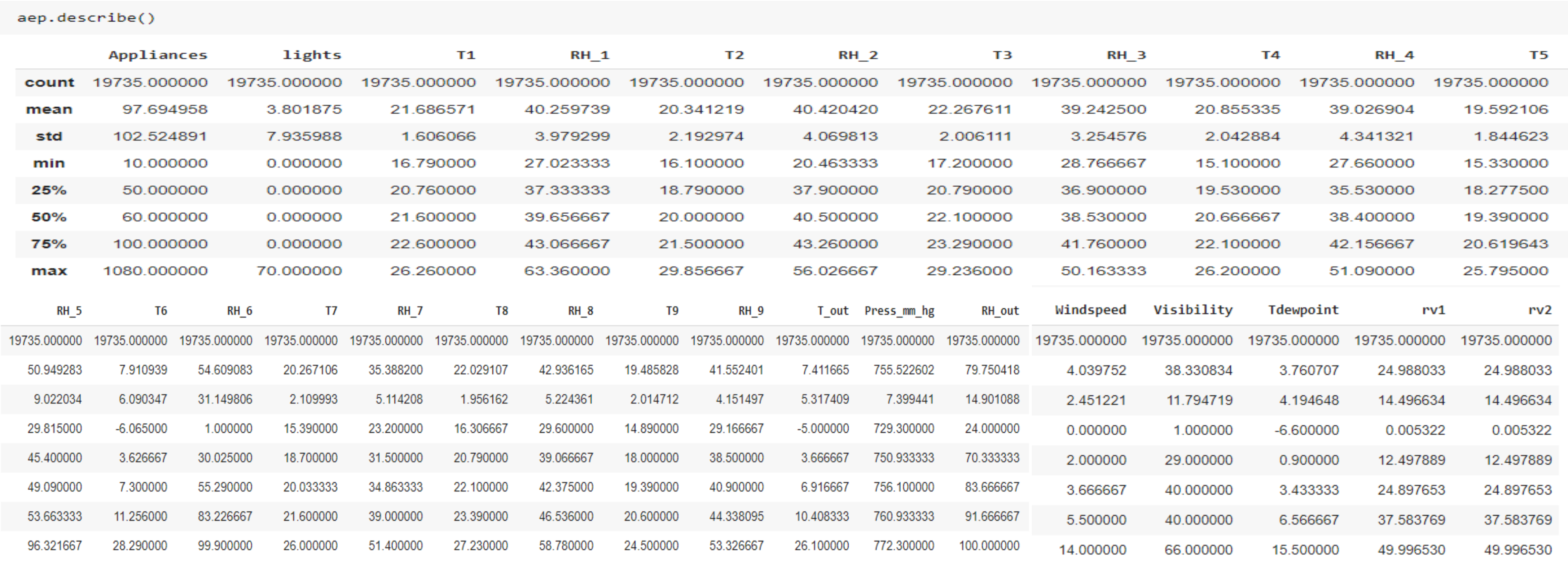
Sample Data



	date	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	RH_4	T5	RH_5	T6	RH_6	T7	RH_7	T8	RH_8
0	2016-01-11 17:00:00	60	30	19.890000	47.596667	19.200000	44.790000	19.790000	44.730000	19.000000	45.566667	17.166667	55.200000	7.026667	84.256667	17.200000	41.626667	18.2000	48.900000
1	2016-01-11 17:10:00	60	30	19.890000	46.693333	19.200000	44.722500	19.790000	44.790000	19.000000	45.992500	17.166667	55.200000	6.833333	84.063333	17.200000	41.560000	18.2000	48.863333
2	2016-01-11 17:20:00	50	30	19.890000	46.300000	19.200000	44.626667	19.790000	44.933333	18.926667	45.890000	17.166667	55.090000	6.560000	83.156667	17.200000	41.433333	18.2000	48.730000
3	2016-01-11 17:30:00	50	40	19.890000	46.066667	19.200000	44.590000	19.790000	45.000000	18.890000	45.723333	17.166667	55.090000	6.433333	83.423333	17.133333	41.290000	18.1000	48.590000
4	2016-01-11 17:40:00	60	40	19.890000	46.333333	19.200000	44.530000	19.790000	45.000000	18.890000	45.530000	17.200000	55.090000	6.366667	84.893333	17.200000	41.230000	18.1000	48.590000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19730	2016-05-27 17:20:00	100	0	25.566667	46.560000	25.890000	42.025714	27.200000	41.163333	24.700000	45.590000	23.200000	52.400000	24.796667	1.000000	24.500000	44.500000	24.7000	50.074000
19731	2016-05-27 17:30:00	90	0	25.500000	46.500000	25.754000	42.080000	27.133333	41.223333	24.700000	45.590000	23.230000	52.326667	24.196667	1.000000	24.557143	44.414286	24.7000	49.790000
19732	2016-05-27 17:40:00	270	10	25.500000	46.596667	25.628571	42.768571	27.050000	41.690000	24.700000	45.730000	23.230000	52.266667	23.626667	1.000000	24.540000	44.400000	24.7000	49.660000
19733	2016-05-27 17:50:00	420	10	25.500000	46.990000	25.414000	43.036000	26.890000	41.290000	24.700000	45.790000	23.200000	52.200000	22.433333	1.000000	24.500000	44.295714	24.6625	49.518750
19734	2016-05-27 18:00:00	430	10	25.500000	46.600000	25.264286	42.971429	26.823333	41.156667	24.700000	45.963333	23.200000	52.200000	21.026667	1.000000	24.500000	44.054000	24.7360	49.736000

19735 rows × 29 columns

Describe All about the data:



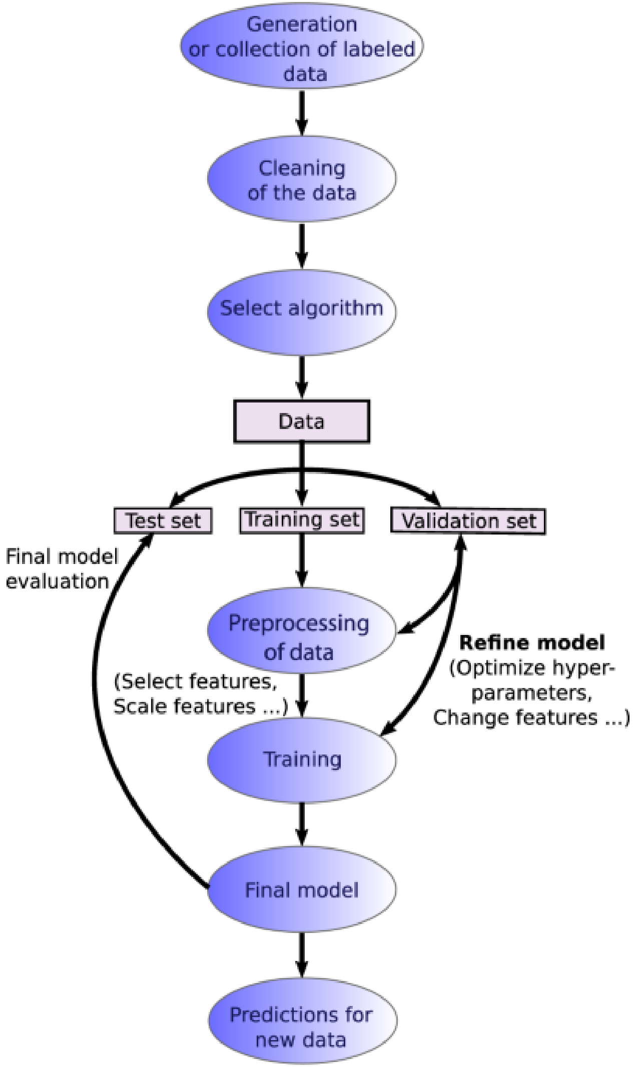
```
aep.describe()
```

	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	RH_4	T5
count	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000
mean	97.694958	3.801875	21.686571	40.259739	20.341219	40.420420	22.267611	39.242500	20.855335	39.026904	19.592106
std	102.524891	7.935988	1.606066	3.979299	2.192974	4.069813	2.006111	3.254576	2.042884	4.341321	1.844623
min	10.000000	0.000000	16.790000	27.023333	16.100000	20.463333	17.200000	28.766667	15.100000	27.660000	15.330000
25%	50.000000	0.000000	20.760000	37.333333	18.790000	37.900000	20.790000	36.900000	19.530000	35.530000	18.277500
50%	60.000000	0.000000	21.600000	39.656667	20.000000	40.500000	22.100000	38.530000	20.666667	38.400000	19.390000
75%	100.000000	0.000000	22.600000	43.066667	21.500000	43.260000	23.290000	41.760000	22.100000	42.156667	20.619643
max	1080.000000	70.000000	26.260000	63.360000	29.856667	56.026667	29.236000	50.163333	26.200000	51.090000	25.795000

RH_5	T6	RH_6	T7	RH_7	T8	RH_8	T9	RH_9	T_out	Press_mm_hg	RH_out	Windspeed	Visibility	Tdewpoint	rv1	rv2
19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000
50.949283	7.910939	54.609083	20.267106	35.388200	22.029107	42.936165	19.485828	41.552401	7.411665	755.522602	79.750418	4.039752	38.330834	3.760707	24.988033	24.988033
9.022034	6.090347	31.149806	2.109993	5.114208	1.956162	5.224361	2.014712	4.151497	5.317409	7.399441	14.901088	2.451221	11.794719	4.194648	14.496634	14.496634
29.815000	-6.065000	1.000000	15.390000	23.200000	16.306667	29.600000	14.890000	29.166667	-5.000000	729.300000	24.000000	0.000000	1.000000	-6.600000	0.005322	0.005322
45.400000	3.626667	30.025000	18.700000	31.500000	20.790000	39.066667	18.000000	38.500000	3.666667	750.933333	70.333333	2.000000	29.000000	0.900000	12.497889	12.497889
49.090000	7.300000	55.290000	20.033333	34.863333	22.100000	42.375000	19.390000	40.900000	6.916667	756.100000	83.666667	3.666667	40.000000	3.433333	24.897653	24.897653
53.663333	11.256000	83.226667	21.600000	39.000000	23.390000	46.536000	20.600000	44.338095	10.408333	760.933333	91.666667	5.500000	40.000000	6.566667	37.583769	37.583769
96.321667	28.290000	99.900000	26.000000	51.400000	27.230000	58.780000	24.500000	53.326667	26.100000	772.300000	100.000000	14.000000	66.000000	15.500000	49.996530	49.996530

- From describe method we find out count, mean, std, min & max etc. as you can see above the data.

FLOW CHART:


Generation or collection of labeled data
Cleaning of the data
Select algorithm
Data
Test set
Training set
Validation set
Final model evaluation
Preprocessing of data
(Select features, Scale features ...)
Refine model
(Optimize hyper-parameters, Change features ...)
Training
Final model
Predictions for new data

Data Preprocessing & Implementation

- **Data processing-1:** In first part we have to remove unnecessary features. Since there were many column with all null values.
- **Data processing-2:** we have manually go through each features select from part 1, and encoded the numerical features.
- **EDA:** In this part we do some exploratory data analysis (EDA) on the features selected in part-1 and part-2 to see the trend.
- **Split the data:** we have to split the data into two parts train and test.
- **Create the model:** Finally, in the last part but not the last part we creates models and function, and import some libraries it's not the easy task. Its also an iterative process. We show how to start with simple models and then add complexity for better performance.

Key observation:

1. Date column is only used for understanding the consumption vs date time behavior and given this is not a time series problem it was removed .
2. Light column was also removed as the are the reading of submeter and we are not focusing on appliance specific reading
3. Number of Independent variables at this stage – 26
4. Number of Dependent variable at this stage – 1
5. Total number of rows – 19735
6. The data set will be split 80-20 % between train & test.
7. Total # of rows in training set – 15788
8. Total # of rows in test set – 3947
9. All the features have numerical values. There are no categorical or ordinal features.
10. Number of missing values & null values = 0

Solution Statement :

Regression is used for problems like this . Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor). The regression methods used are:

1. Linear Regression : In linear regression we wish to fit a function in this form $\hat{Y} = \beta0+\beta1X1+\beta2X2+\beta3X3$ where X is the vector of features and $\beta0$, $\beta1$, $\beta2$, $\beta3$ are the coefficients we wish to learn. It updates β at every step by reducing the loss function as much as possible. Once we reach the minimum point of the loss function we can say that we completed the iterative process and learned the parameters.

2. Ridge regression : Regularized machine learning model, in which model’s loss function contains another element that should be minimized as well. $L = \sum(\hat{Y}i– Yi)$ ^2 $+ \lambda\sum \beta2$. The second element sums over squared β values and multiplies it by another parameter λ. The reason for doing that is to “punish” the loss function for high values of the coefficients β

3. Lasso regression : Lasso is another extension built on regularized linear regression . The loss function of Lasso is in the form: $L = \sum(\hat{Y}i– Yi)$ ^2 $+ \lambda\sum |\beta|$. The only difference from Ridge regression is that the regularization term is in absolute value.

Evaluation Metrics :

The regression metrics used as standards to measure regression models are

1. Mean Absolute Error:

$MSE = \frac{1}{n}\Sigma(y - \hat{y})^2$ where, y = actual value

$\hat{y}$ = predicted value

2. Root Mean Squared Error (RMSE):

$$RMsE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}$$

3. MAE (Mean Absolute Error):

$$MAE = \frac{1}{n}\Sigma|y - \hat{y}|$$

MAPE (Mean Absolute Percentage Error):

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_{\mathrm{i}} - \hat{y}_i}{y_i}\right|$$

4. R2 (R – Squared):

$$R^2 = 1 - \frac{MSE(model)}{MSE(baseline)}$$

5. Adjusted R2:

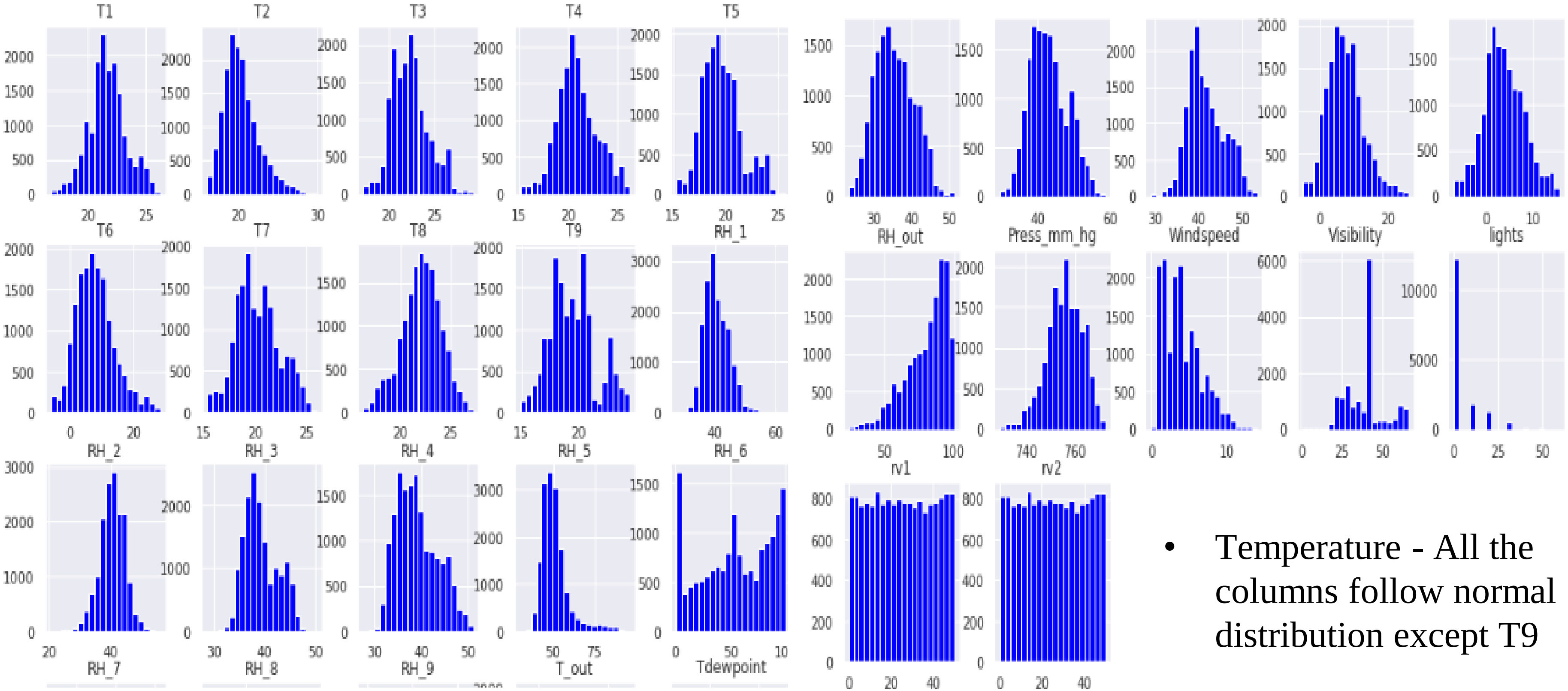
$$R_a^2 = 1 - \left[\left(\frac{n-1}{n-k-1}\right) \times (1 - R^2)\right]$$

Project design:

The steps to be followed are mentioned below:

1. Data Visualization : Visual plots to detect the correlation between different independent variables and between independent and dependent variables . Ranges and other statistical data can also be verified
2. Pre Processing : In this process we will be organizing and tidying up the data, removing what is no longer needed, replacing what is missing and standardizing the format across all the data collected.
3. Feature Engineering : Find all the features which impacts the models and reduce the number of features if possible using PCA
4. Choosing a Model : Check all the applicable models and select the one which provides best metrics .
5. Hyperparameter Tuning : Find best possible combination of selected algorithm in order to maximize the performance using Grid Search
6. Prediction : Using Test set predict the dependent variable and check accuracy

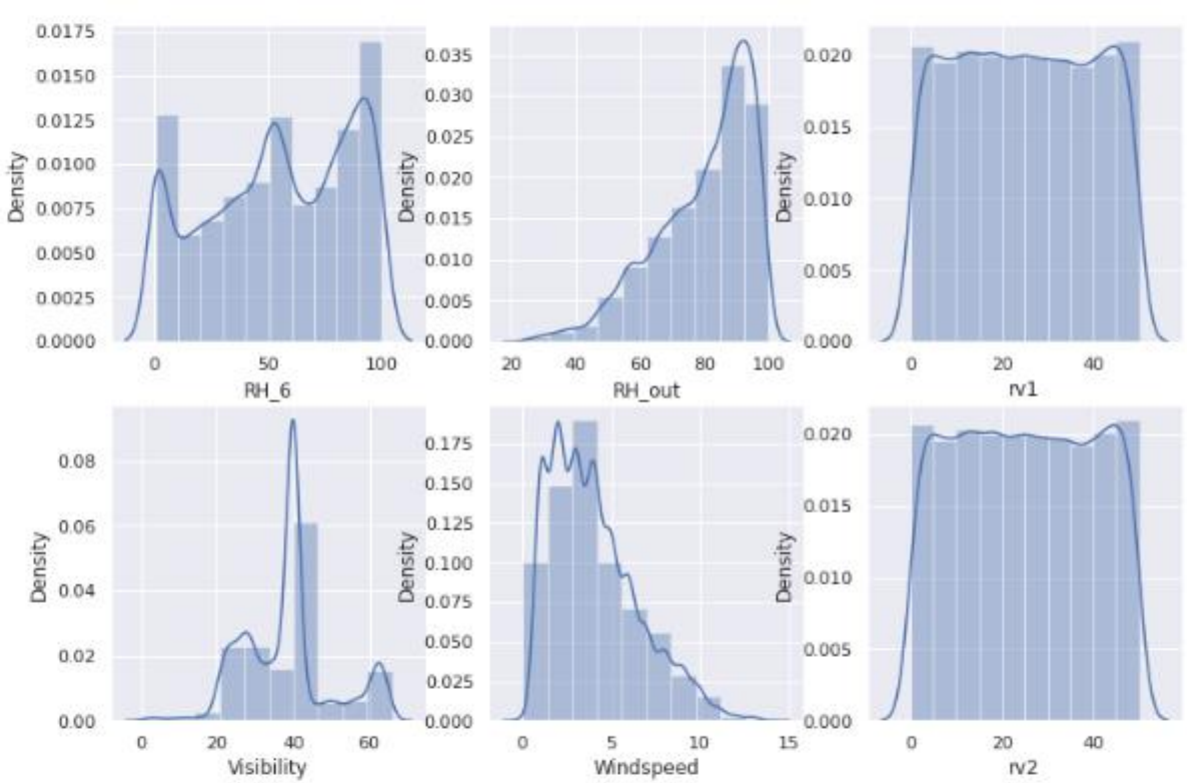
Independent variable



- Temperature - All the columns follow normal distribution except T9

- Visibility - This column is negatively skewed
- Windspeed - This column is positively skewed

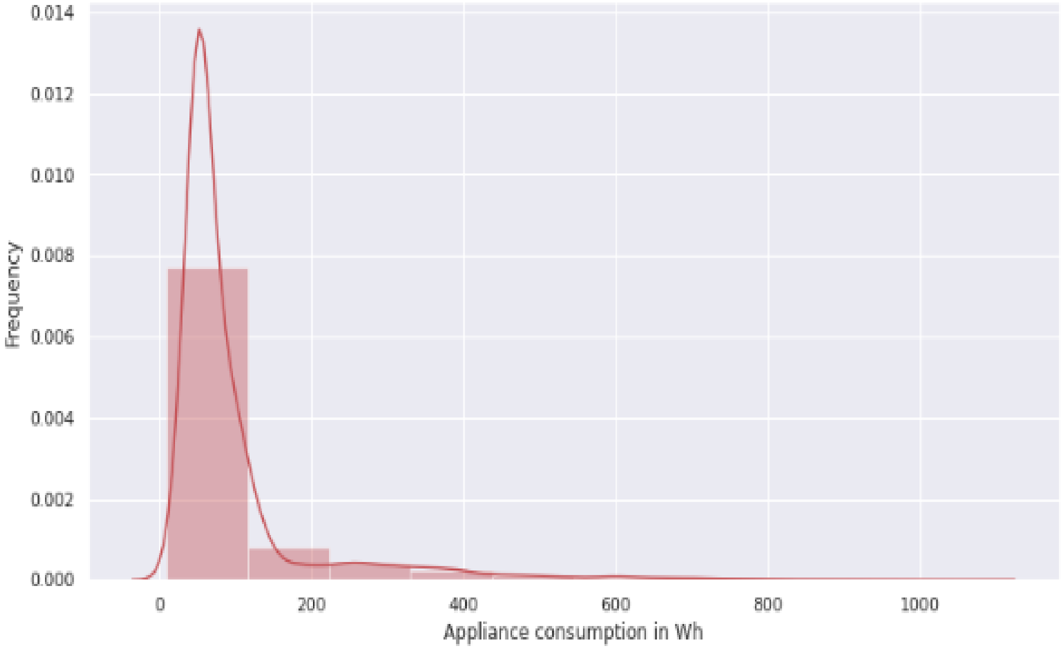
Independent variable



All graph not follow normal distribution RH_6, RH_out, rv1 , visibility, windspeed and rv2 primarily because these sensors are outside the house

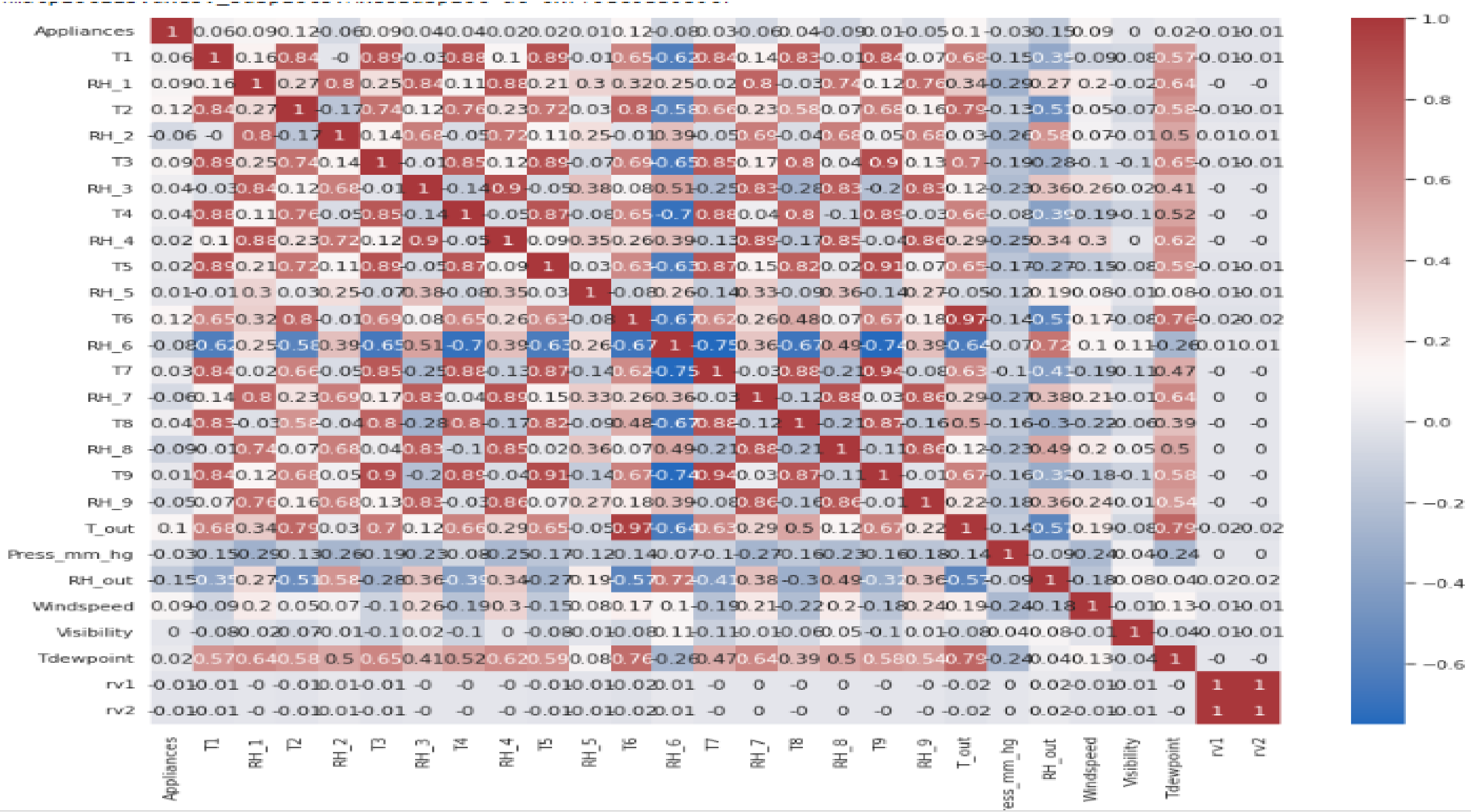
This graph which is not showing normal distribution so we have removed this variable from the train data sets.

Dependent variable consumption graph:



- Appliance - This column is positively skewed , most the values are around mean 100 Wh . There are outliers in this column
- 75% of Appliance consumption is less than 100 Wh . With the maximum consumption of 1080 Wh , there will be outliers in this column and there are small number of cases where consumption is very high

Correlation by heatmap



Observations based on correlation plot:

1.Temperature - All the temperature variables from T1-T9 and T_out have positive correlation with the target Appliances . For the indoor temperatures, the correlations are high as expected, since the ventilation is driven by the RH, rv unit and minimizes air temperature differences between rooms. Four columns have a high degree of correlation with T9 –T3,T5,T7,T8 also T6 & T_Out has high correlation (both temperatures from outside) . Hence T6 & T9 can be removed from training set as information provided by them can be provided by other fields.

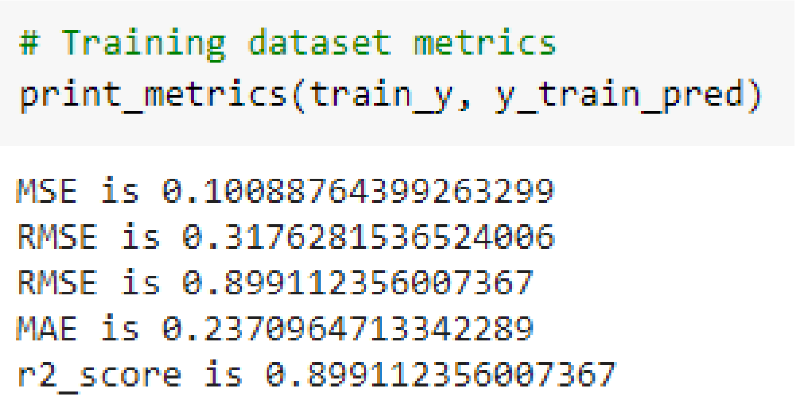
2.Weather attributes - Visibility, Tdewpoint, Press_mm_hg have low correlation values

3.Humidity - There are no significantly high correlation cases (> 0.9) for humidity sensors.

4.Random variables have no role to play

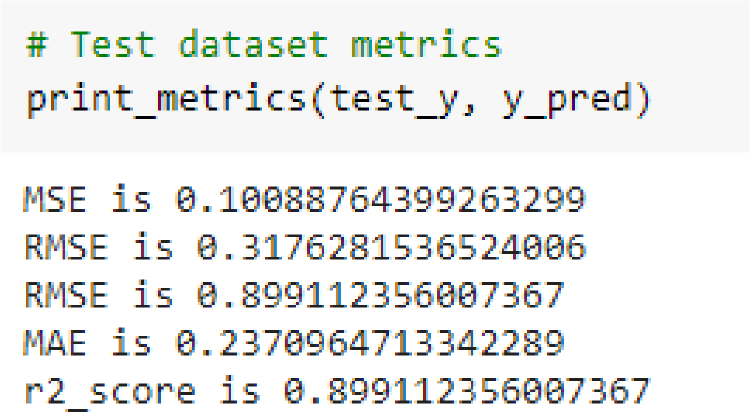
Result:

Linear regression:



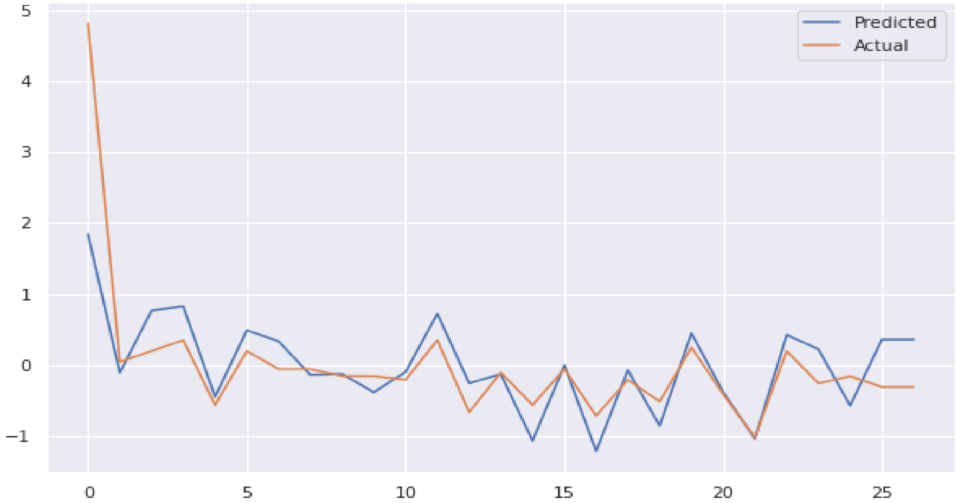
```
# Training dataset metrics
print_metrics(train_y, y_train_pred)
```

```
MSE is 0.10088764399263299
RMSE is 0.3176281536524006
RMSE is 0.899112356007367
MAE is 0.2370964713342289
r2_score is 0.899112356007367
```

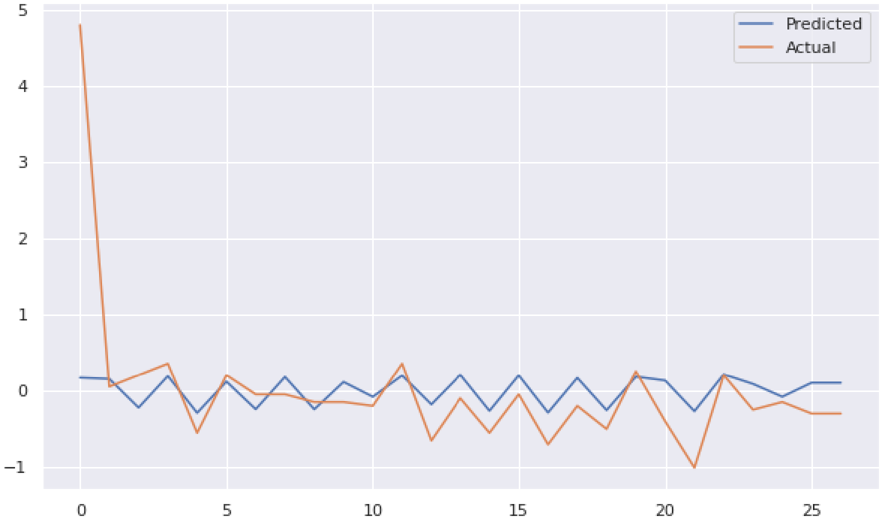


```
# Test dataset metrics
print_metrics(test_y, y_pred)
```

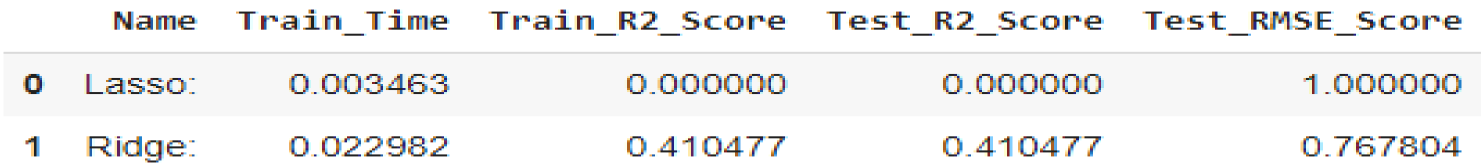
```
MSE is 0.10088764399263299
RMSE is 0.3176281536524006
RMSE is 0.899112356007367
MAE is 0.2370964713342289
r2_score is 0.899112356007367
```



Ridge algorithm result between predicted and actual.



Lasso algorithm result between predicted and actual.



	Name	Train_Time	Train_R2_Score	Test_R2_Score	Test_RMSE_Score
0	Lasso:	0.003463	0.000000	0.000000	1.000000
1	Ridge:	0.022982	0.410477	0.410477	0.767804

Conclusion:

- Temperature columns - Temperature inside the house varies between 14.89 Deg & 29.85 Deg , temperature outside (T6) varies between -6.06 Deg to 28.29 Deg . The reason for this variation is sensors are kept outside the house

- Humidify columns - Humidity inside house varies is between 20.60% to 63.36% with exception of RH_5 (Bathroom) and RH_6 (Outside house) which varies between 29.82% to 96.32% and 1% to 99.9% respectively.

- Appliances - 75% of Appliance consumption is less than 100 Wh . With the maximum consumption of 1080 Wh , there will be outliers in this column and there are small number of cases where consumption is very high

- Best results over test set are given by Ridge with R2 score of 0.41
- Least RMSE score is also by Ridge 0.76
- Lasso regularization over Linear regression was worst performing model

- The top 3 important features are humidity attributes, which leads to the conclusion that humidity affects power consumption more than temperature. Windspeed is least important as the speed of wind doesn’t affect power consumption inside the house. So controlling humidity inside the house may lead to energy savings.

- When predicting electricity consumption, it is necessary to determine an appropriate prediction method according to the expected Fore-casting results and characteristics of the prediction model.

- Here in this study we have predicted the result on the test data set with the supervised machine learning algorithm based on regression (Lasso and Ridge). We performed exploratory data analysis, pre-processing, and train-test split before training the model.

- We used various metrics to test the advantages of the proposed model: mean absolute error, mean absolute percent error, mean squared error and r2_score

Challenges & Learning gained during project

1. Feature scaling is very important for regressions models , I initially tried without it and the results were not good . On Kaggle this is suggested by all users.
2. Using seed value helped in reproducing results for algorithms . Without this value the results were different each time.
3. It is very important to check the intercorrelation between all the variables in order to remove the redundant features with high correlation values.
4. While scaling data , it is useful to maintain separate copies of data frame which can be created using index and column names of original data frame
5. The pipeline of adding algorithms should be easy to manage
6. Seaborn and pyplot are good libraries to plot various properties of data frame
7. For performing Exhaustive search or Random search in the hyperparameter space for tuning the model, always parallelize the process since there are a lot of models with different configurations to be fitted. (Set n_jobs parameter with the value -1 to utilize all CPUs)
8. One effective way to check the robustness of the model is to fit it on a reduced feature space in case of high dimensional data. Select the first ‘k’ (usually >= 3) key features for this task.

Reference:

1) https://www.almabetter.com/

2) https://www.wikipedia.org

3) https://www.kaggle.com/

4) https://github.com/