Capstone Project

Machine Learning Engineer
Nanodegree

Mazen Ahmed

August 27, 2018

Appliances Energy Prediction

I. Definition

a) Project Overview:

prediction of energy consumption in a home. Nowadays, smart-homes are increasing, and in some countries energy problem is rising, therefore we need to monitor our consumption which will help both governments and individuals.

b) Problem Statement:

Predict Energy consumption of appliances in a home based on weather condition inside and outside the home.

c) Metrics:

as Regression Problem I used R2 score or the coefficient of determination, which is “the proportion of the variance in the dependent variable that is predictable from the independent variable(s).”[1]

which defined mathematically as: $R^2=1-\frac{SS_{res}}{SS_{tot}}$ where, SS_{res} is the residuals sum of squares and SS_{tot} is the total sum of squares which can mathematically represents as:

$SS_{res}=\sum_i (y_i-f_i)^2=\sum_i e_i^2$, and $SS_{tot}=\sum_i (y_i-\bar{y})^2$ since,

$\bar{y}=\frac{1}{n}\sum_i^n y_i$ which is the mean.

II. Analysis

a) Data Exploration:

the dataset contains 19735 records and 29 attributes are listed below where the ‘Appliance’ is the target variable:

-date time year-month-day hour:minute:second

-Appliances, energy use in Wh

-lights, energy use of light fixtures in the house in Wh

-T1, Temperature in kitchen area, in Celsius

-RH_1, Humidity in kitchen area, in %

-T2, Temperature in living room area, in Celsius

-RH_2, Humidity in living room area, in %

-T3, Temperature in laundry room area

-RH_3, Humidity in laundry room area, in %

-T4, Temperature in office room, in Celsius

-RH_4, Humidity in office room, in %

-T5, Temperature in bathroom, in Celsius

-RH_5, Humidity in bathroom, in %

-T6, Temperature outside the building (north side), in Celsius

-RH_6, Humidity outside the building (north side), in %

-T7, Temperature in ironing room , in Celsius

-RH_7, Humidity in ironing room, in %

-T8, Temperature in teenager room 2, in Celsius

-RH_8, Humidity in teenager room 2, in %-T9, Temperature in parents room, in Celsius

-RH_9, Humidity in parents room, in %

-To, Temperature outside (from Chievres weather station), in Celsius

-Pressure (from Chievres weather station), in mm Hg

-RH_out, Humidity outside (from Chievres weather station), in %

-Wind speed (from Chievres weather station), in m/s

-Visibility (from Chievres weather station), in km

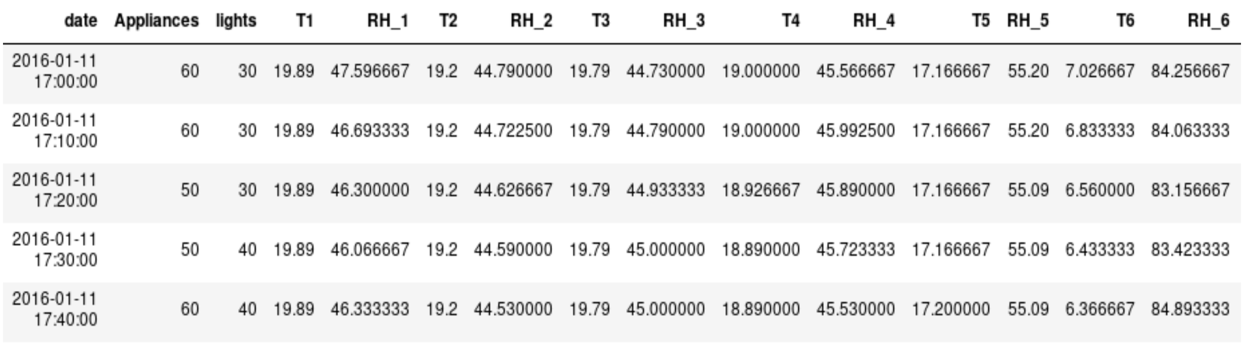
-Tdewpoint (from Chievres weather station), Â°C

-rv1, Random variable 1, nondimensional

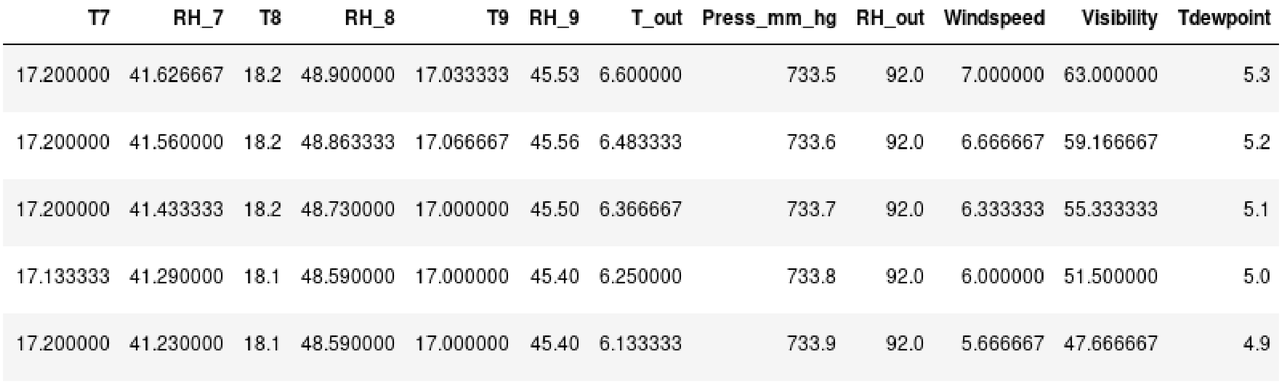
-rv2, Random variable 2, nondimensional

dataset:
https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction

- **Sample from the dataset:**

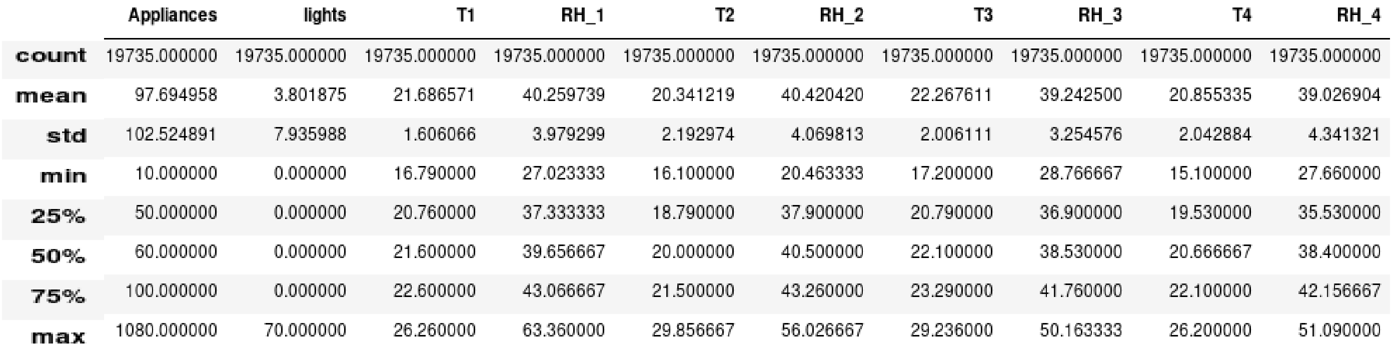


date	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	RH_4	T5	RH_5	T6	RH_6
2016-01-11 17:00:00	60	30	19.89	47.596667	19.2	44.790000	19.79	44.730000	19.000000	45.566667	17.166667	55.20	7.026667	84.256667
2016-01-11 17:10:00	60	30	19.89	46.693333	19.2	44.722500	19.79	44.790000	19.000000	45.992500	17.166667	55.20	6.833333	84.063333
2016-01-11 17:20:00	50	30	19.89	46.300000	19.2	44.626667	19.79	44.933333	18.926667	45.890000	17.166667	55.09	6.560000	83.156667
2016-01-11 17:30:00	50	40	19.89	46.066667	19.2	44.590000	19.79	45.000000	18.890000	45.723333	17.166667	55.09	6.433333	83.423333
2016-01-11 17:40:00	60	40	19.89	46.333333	19.2	44.530000	19.79	45.000000	18.890000	45.530000	17.200000	55.09	6.366667	84.893333

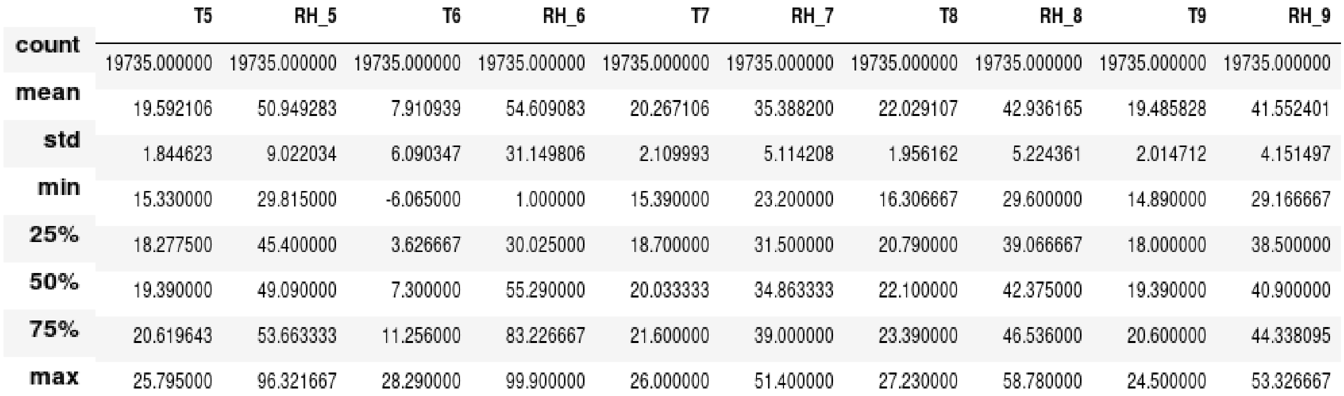


T7	RH_7	T8	RH_8	T9	RH_9	T_out	Press_mm_hg	RH_out	Windspeed	Visibility	Tdewpoint
17.200000	41.626667	18.2	48.900000	17.033333	45.53	6.600000	733.5	92.0	7.000000	63.000000	5.3
17.200000	41.560000	18.2	48.863333	17.066667	45.56	6.483333	733.6	92.0	6.666667	59.166667	5.2
17.200000	41.433333	18.2	48.730000	17.000000	45.50	6.366667	733.7	92.0	6.333333	55.333333	5.1
17.133333	41.290000	18.1	48.590000	17.000000	45.40	6.250000	733.8	92.0	6.000000	51.500000	5.0
17.200000	41.230000	18.1	48.590000	17.000000	45.40	6.133333	733.9	92.0	5.666667	47.666667	4.9

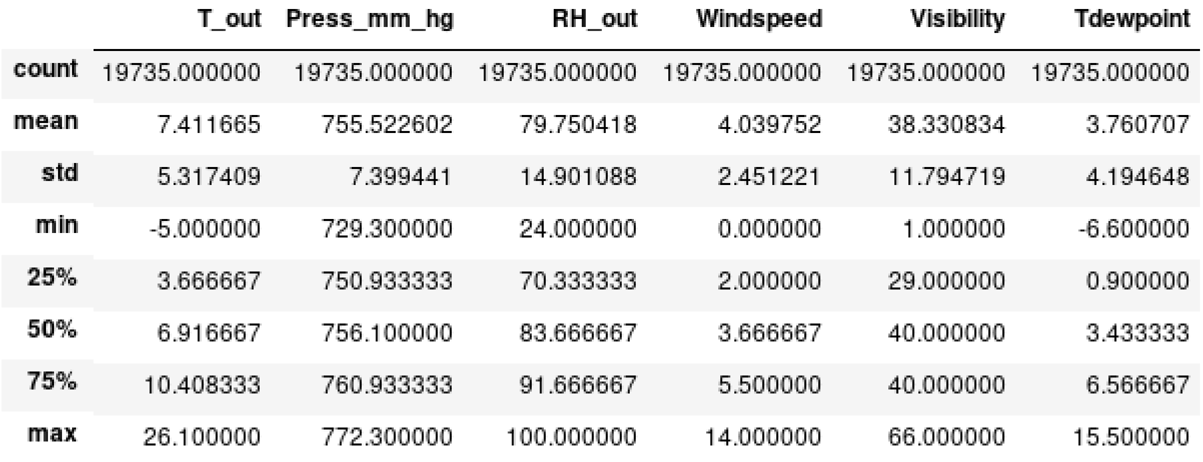
- **Statistic Information:**



	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	RH_4
count	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
std	102.524891	7.935988	1.606066	3.979299	2.192974	4.069813	2.006111	3.254576	2.042884	4.341321
min	10.000000	0.000000	16.790000	27.023333	16.100000	20.463333	17.200000	28.766667	15.100000	27.660000
25%	50.000000	0.000000	20.760000	37.333333	18.790000	37.900000	20.790000	36.900000	19.530000	35.530000
50%	60.000000	0.000000	21.600000	39.656667	20.000000	40.500000	22.100000	38.530000	20.666667	38.400000
75%	100.000000	0.000000	22.600000	43.066667	21.500000	43.260000	23.290000	41.760000	22.100000	42.156667
max	1080.000000	70.000000	26.260000	63.360000	29.856667	56.026667	29.236000	50.163333	26.200000	51.090000



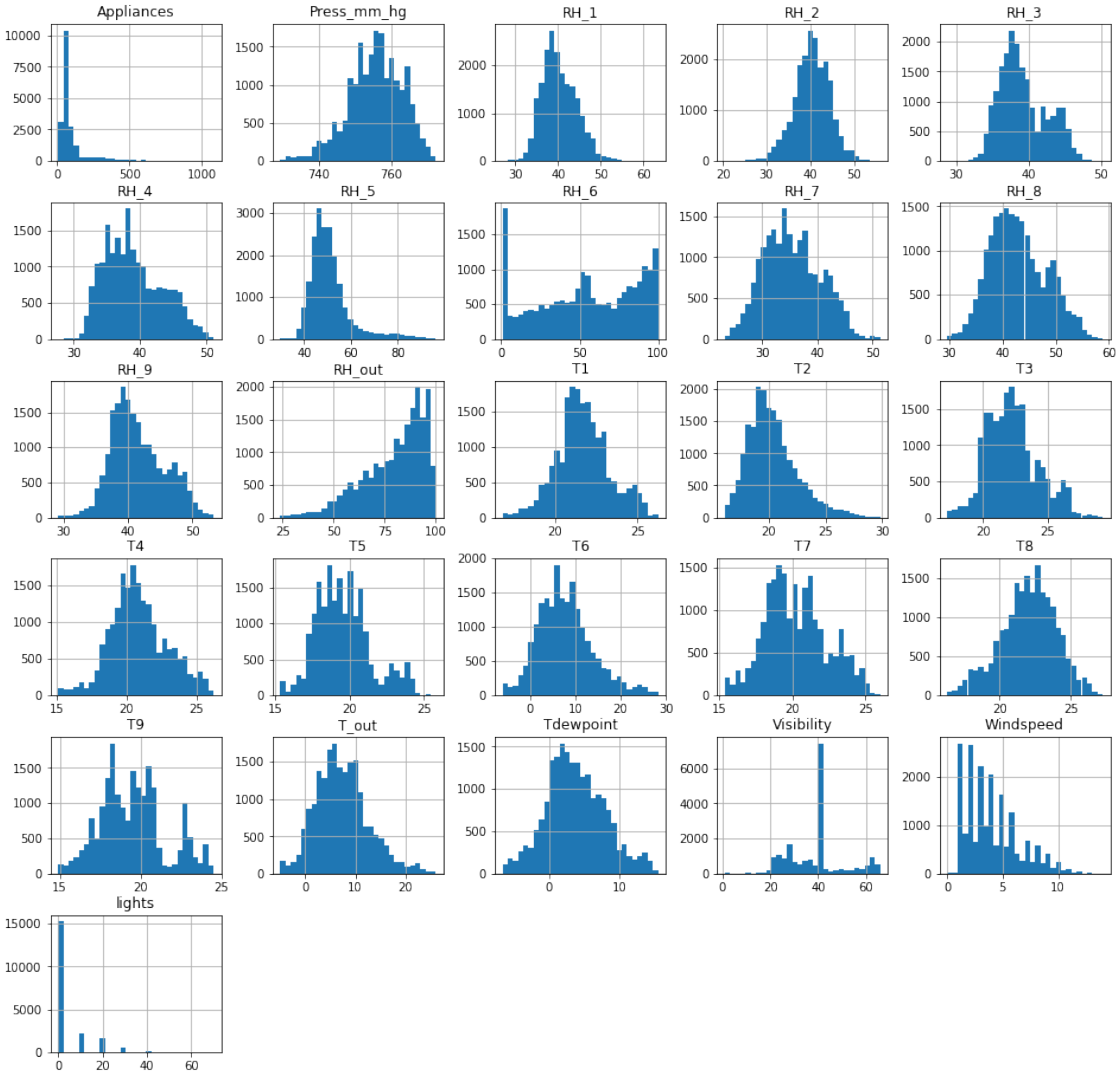
	T5	RH_5	T6	RH_6	T7	RH_7	T8	RH_8	T9	RH_9
count	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000
mean	19.592106	50.949283	7.910939	54.609083	20.267106	35.388200	22.029107	42.936165	19.485828	41.552401
std	1.844623	9.022034	6.090347	31.149806	2.109993	5.114208	1.956162	5.224361	2.014712	4.151497
min	15.330000	29.815000	-6.065000	1.000000	15.390000	23.200000	16.306667	29.600000	14.890000	29.166667
25%	18.277500	45.400000	3.626667	30.025000	18.700000	31.500000	20.790000	39.066667	18.000000	38.500000
50%	19.390000	49.090000	7.300000	55.290000	20.033333	34.863333	22.100000	42.375000	19.390000	40.900000
75%	20.619643	53.663333	11.256000	83.226667	21.600000	39.000000	23.390000	46.536000	20.600000	44.338095
max	25.795000	96.321667	28.290000	99.900000	26.000000	51.400000	27.230000	58.780000	24.500000	53.326667



	T_out	Press_mm_hg	RH_out	Windspeed	Visibility	Tdewpoint
count	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000
mean	7.411665	755.522602	79.750418	4.039752	38.330834	3.760707
std	5.317409	7.399441	14.901088	2.451221	11.794719	4.194648
min	-5.000000	729.300000	24.000000	0.000000	1.000000	-6.600000
25%	3.666667	750.933333	70.333333	2.000000	29.000000	0.900000
50%	6.916667	756.100000	83.666667	3.666667	40.000000	3.433333
75%	10.408333	760.933333	91.666667	5.500000	40.000000	6.566667
max	26.100000	772.300000	100.000000	14.000000	66.000000	15.500000

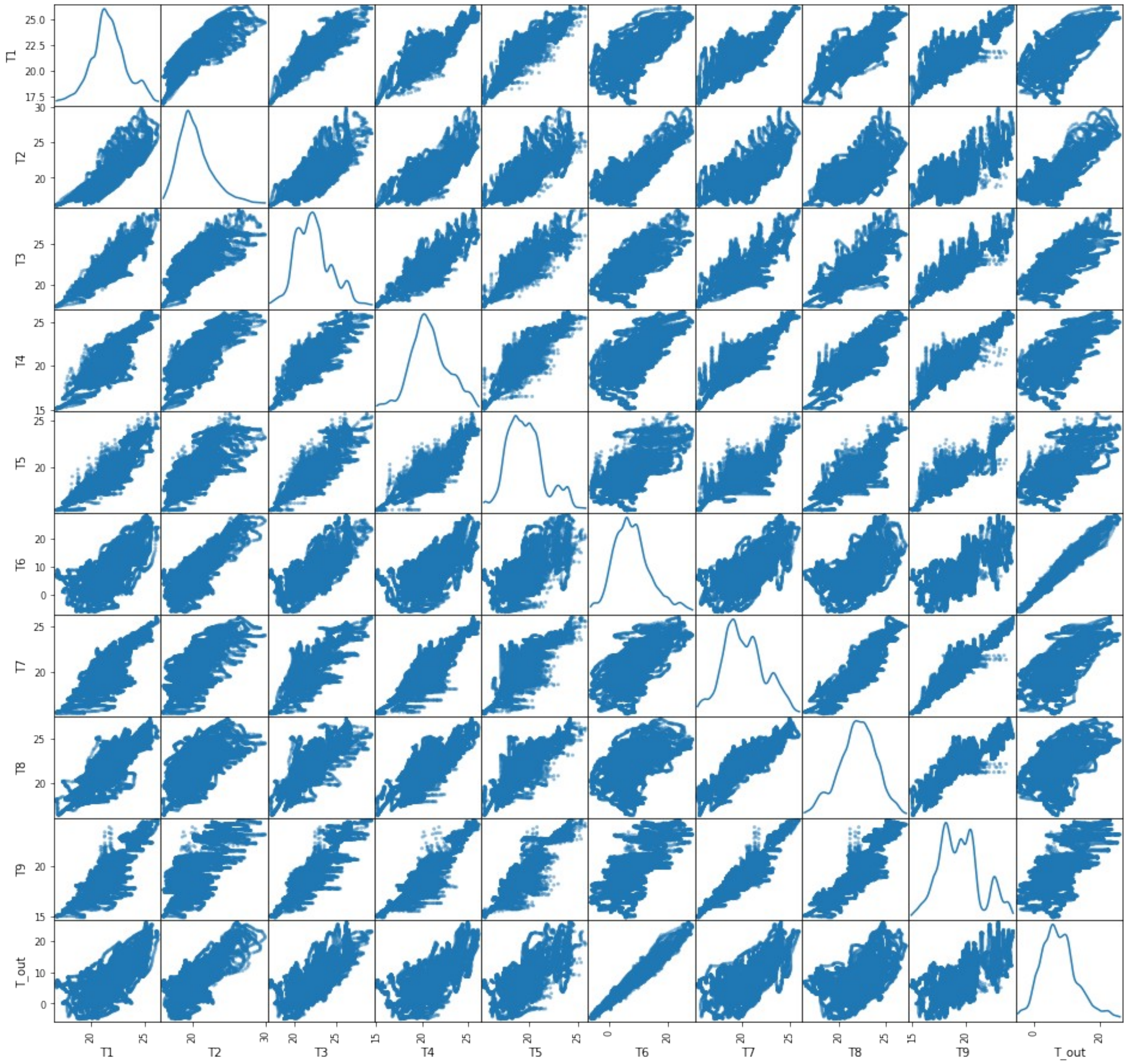
b) Exploratory Visualization:

- **Histograms for all the data:**



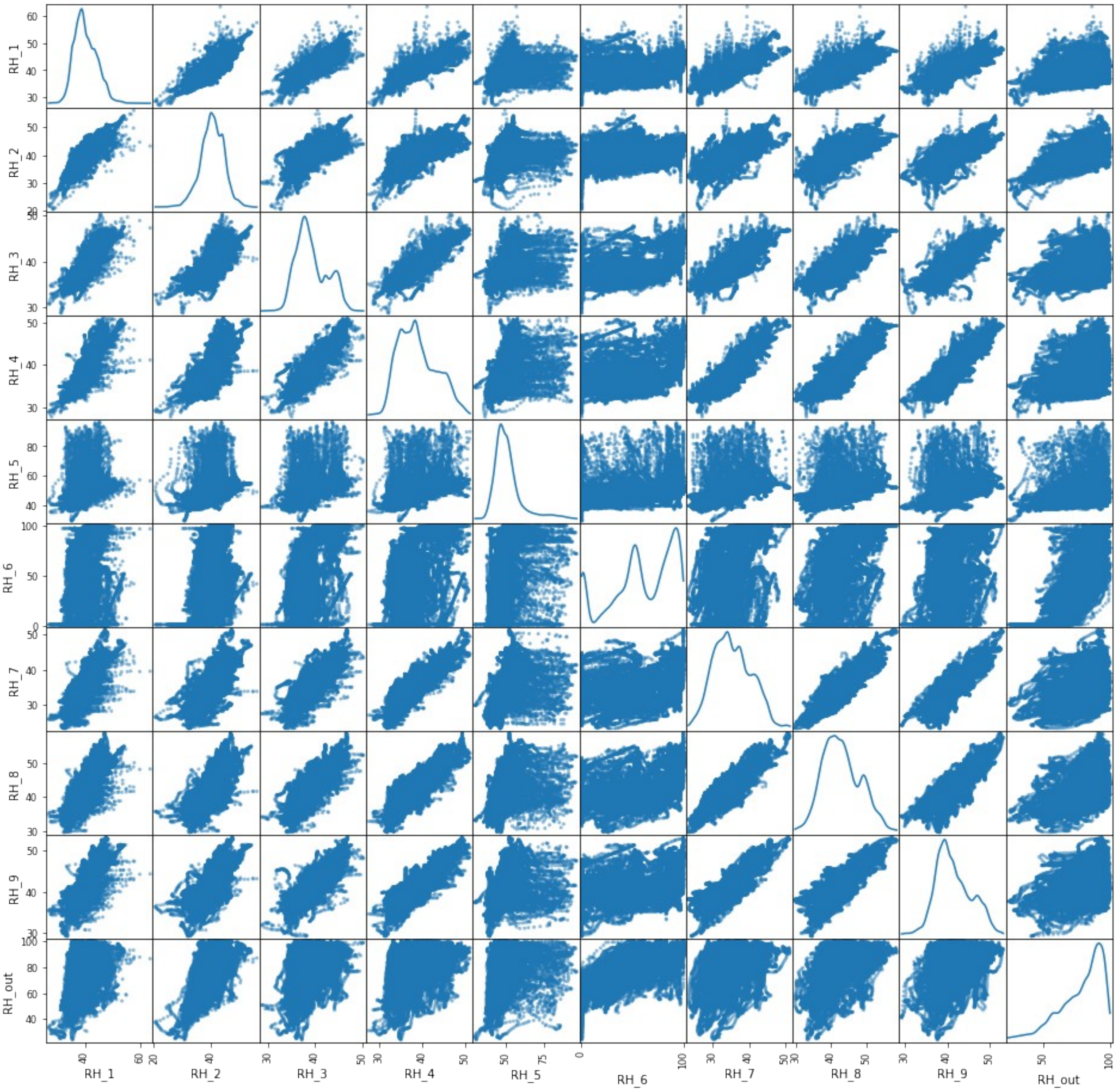
From the distributions above we can’t see a linear relation between the target variable(Appliances) and the other features.

- **Scatter Plot For Temperatures:**



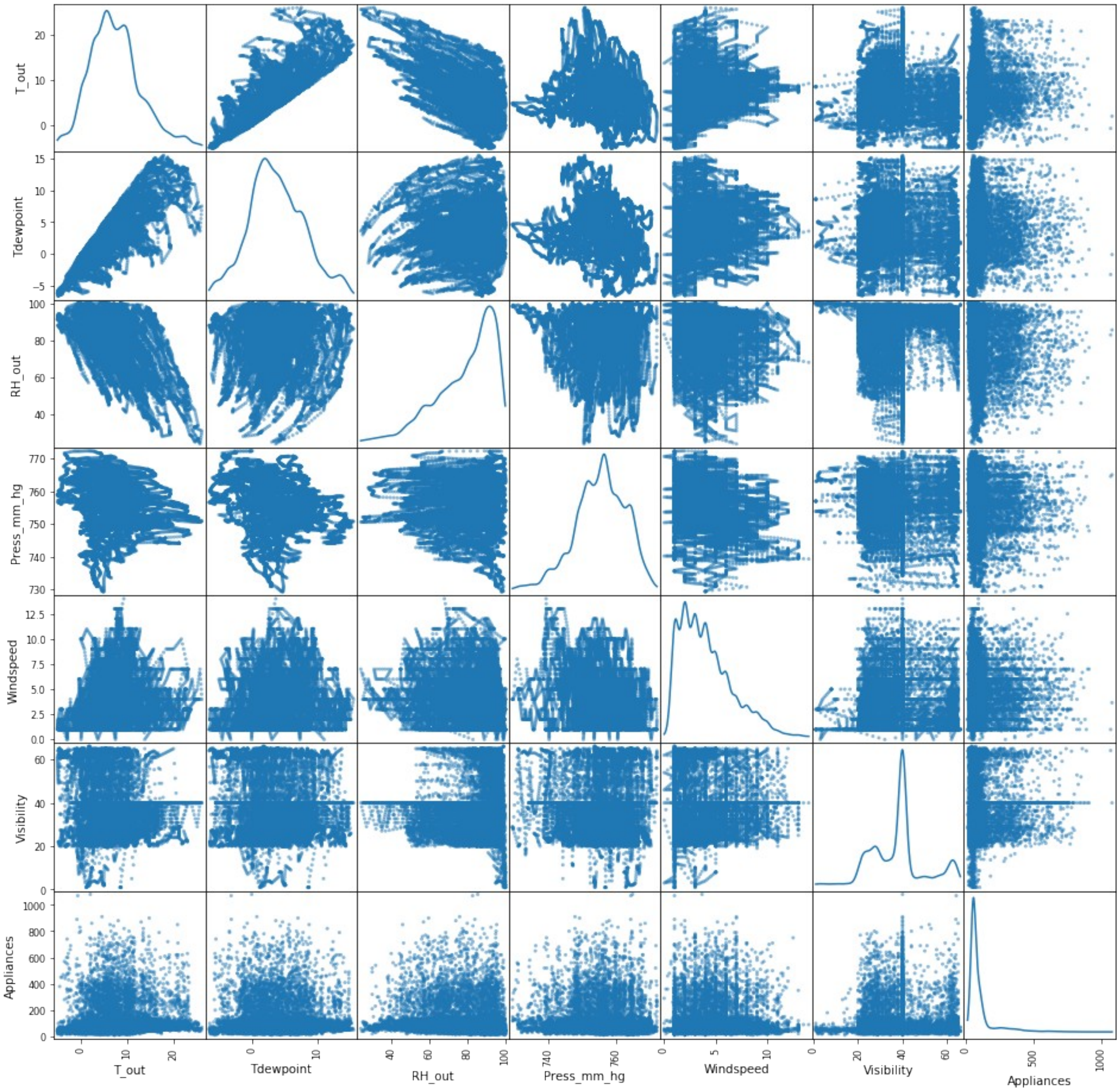
- From the plots above, we can see linear correlations between T1&T3, T6&T_out ,and T7&T9

- **Scatter Plot Humidity:**



- From the plots above we can see that there's no any linear correlation.

- **Scatter Plot Weather Information:**



- From the plots above we can see that there's no any linear correlation.

c) Algorithms and Techniques:

I will try out those Techniques:

1. **Linear Regression:** The overall idea of regression is to examine two things: (1) does a set of predictor variables do a good job in predicting an outcome (dependent) variable? (2) Which variables in particular are significant predictors of the outcome variable, and in what way do they–indicated by the magnitude and sign of the beta estimates–impact the outcome variable? The simplest form of the regression equation with one dependent and one independent variable is defined by the formula y = c + b*x, where y = estimated dependent variable score, c = constant, b = regression coefficient, and x = score on the independent variable.[2]

2. **Logistic Regression:** is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). [3]

3. **Random Forest Regression:** is a versatile machine learning method capable of performing both regression and classification tasks. It also undertakes dimensional reduction methods, treats missing values, outlier values and other essentials steps of the data, and does a fairly good job. It is a type of ensemble learning method, where a group of weak models combine to form a powerful model.[4]

4. **SVM:** In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well.[5]

for all these techniques I will input all the features without rv1 and rv2 then I will take the technique that will have the highest r2 score to tuning it.

d) Benchmark:

The benchmark model is Gradient Boosting

- **Scoring observation:**
  - R2 score on testing data is 25.08%.
  - Score on train data is 34.20%.

III. Methodology

a) Data Preprocessing:

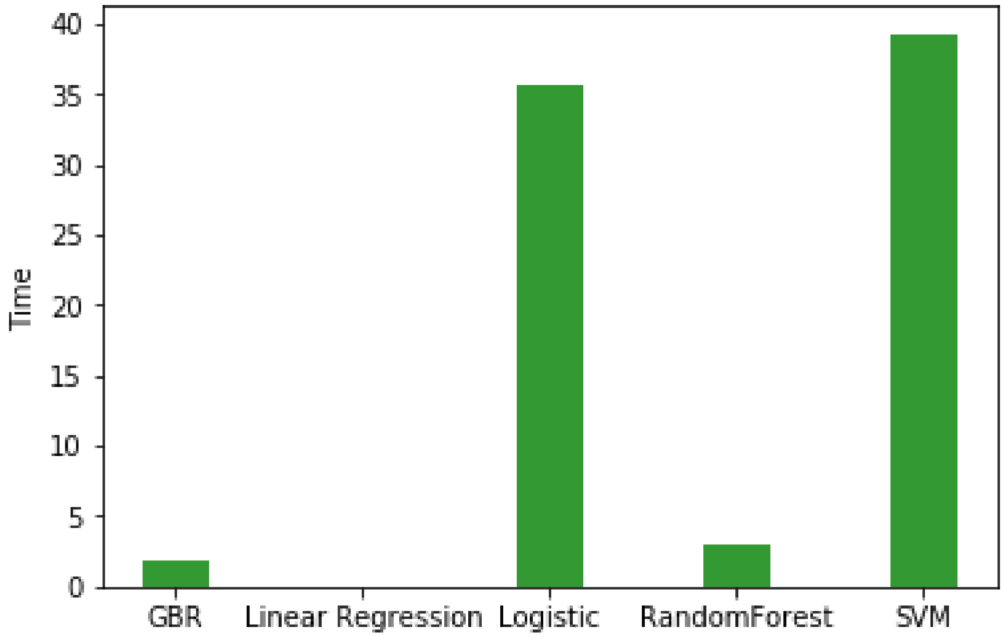
as a preprocessing part i’ve drop out Date, rv1, rv2, and the target variable (Appliance) from the DataFrame and I did split to the dataset into training and testing sets.

b) Implementation:

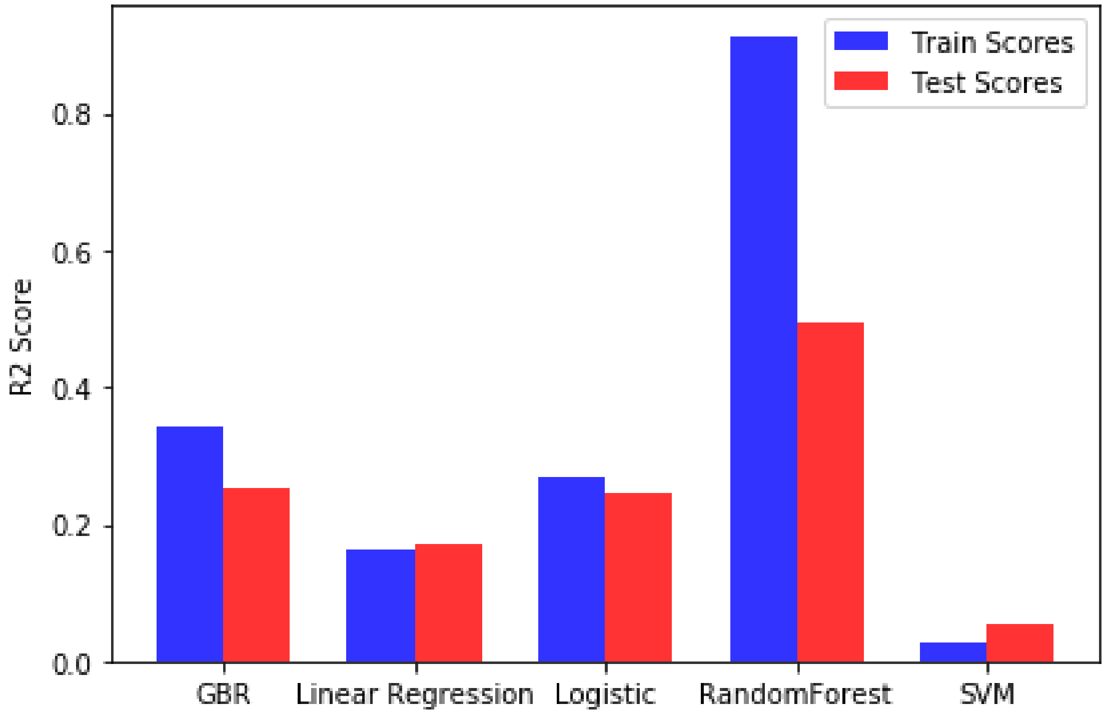
i’ve applied every algorithm listed in Algorithms and technique Section and I calculated the R2 score and the elapsed time.

	R2 Score on training	**R2 Score on testing**	**Training time**
Gradient Boosting (Benchmark)	0.3420	0.2508	1.8748
Linear Regression	0.1632	0.1695	0.0094
Logistic Regression	0.2677	0.2447	35.5616
Random Forest	0.9133	0.4950	3.0382
SVM	0.0278	0.0548	39.3045

- **Elapsed Training Time Visualization:**



- **R2 Score Visualization:**



C) Refinement:

I tried to tune the following parameters of Random Forest Regression:

1. **n_estimators:** The number of trees in the forest.[6]

2. **max_features:** The number of features to consider when looking for the best split.[6]

3. **max_depth:** The maximum depth of the tree.[6]

model before tuning, the R2 Score on test data is 49.5%. Now, after tuning, R2 Score on test data is 58.61%. the performance increased by 9.11%.

IV. Results

a) Model Evaluation and Validation:

Features before Tuning:

1. n_estimators = 10
2. max_features = auto = n_features = 25
3. max_depth = None

Features After Tuning:

1. n_estimators = 400
2. max_features = log2
3. max_depth = 50

the optimized model is being able to generalize the data by 58.61%.

b) Justification:

	R2 Score on training data	R2 Score on testing data
Optimized Model	0.9434	0.5861
Benchmark Model	0.3420	0.2508
Performance increased by	60.14%	33.53%

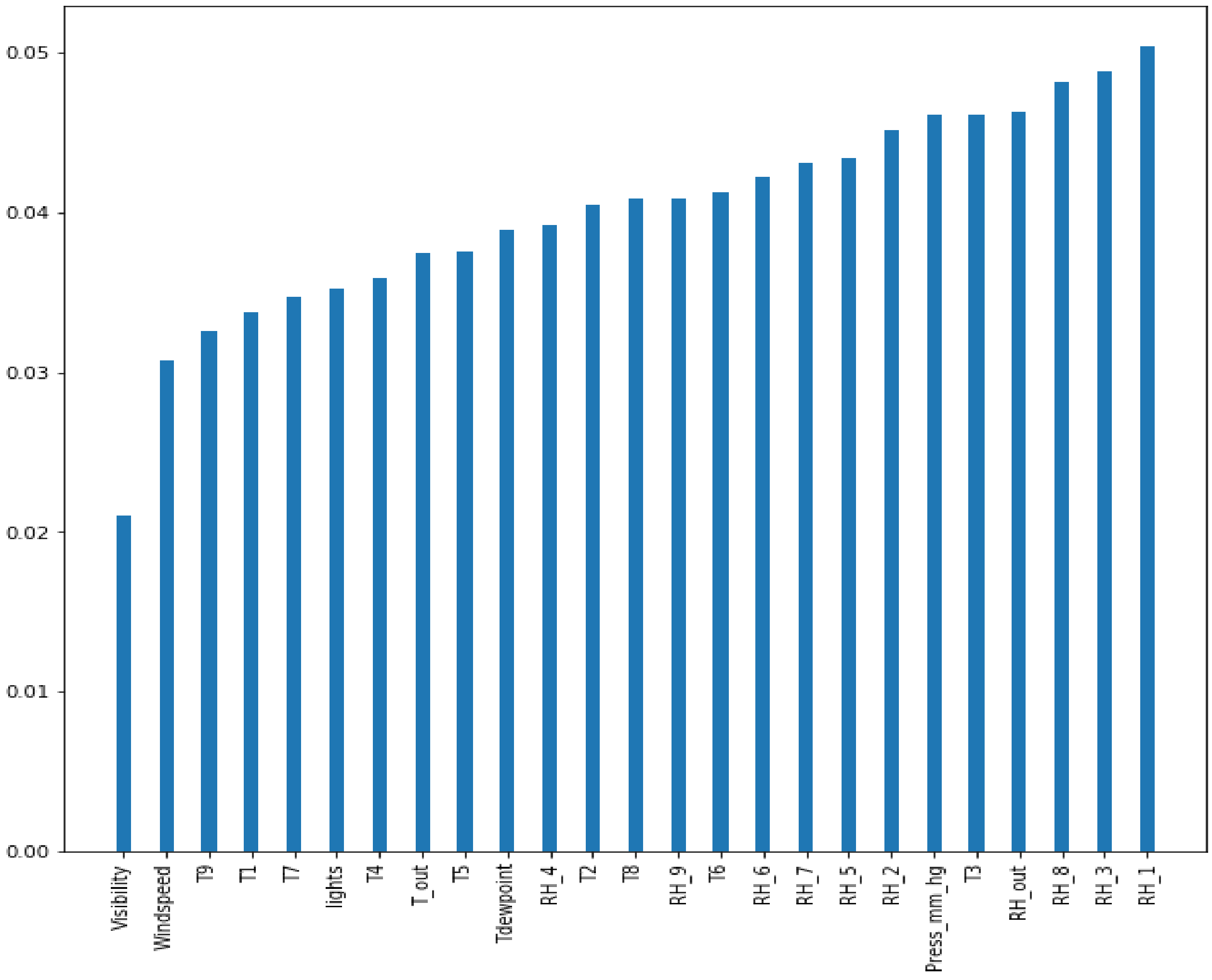
Therefore, the optimized model (final model) is doing better than the benchmark model.

V. Conclusion

a) Free-Form Visualization:

Visualization of the features that distinguish which features is more relevant than the others, to the Random Forest Algorithm.

From the below chart, I can conclude that the Kitchen Humidity, Teenager Room Humidity, and Laundry Room Humidity are more relevant to our target. on the contrary, the Visibility and Wind speed are irrelevant.



b) Reflection:

Firstly, I started with explore the dataset, secondly, I did some visualization to the data, thirdly, I did some preprocessing and make the benchmark model, fourthly, I tried out some algorithms on the data then I decided which model I would optimize, lastly I tuned the parameters by using Gridsearch.

The preprocessing and tuning the parameter were the hard parts, but choosing the Algorithms to be used was interesting.

The final model gained 58.61% score which is not too good however, this model need some improvements to be better, and yes it should be used in this problem domain.

c) Improvement:

trying Occan’s razor with minimum description length by dropping out irrelevant features visibility and Windspeed, or the correlated feature to reduce dimensionality.

Appendix:

[1] https://en.wikipedia.org/wiki/Coefficient_of_determination/

[2] http://www.statisticssolutions.com/what-is-linear-regression/

[3] https://www.medcalc.org/manual/logistic_regression.php

[4] https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/#nine

[5] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[6] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html