STAT 5000

Statistical Methods and Applications I

Spring 2023

Project Report

Group Information	
Student Name	Student ID
NITISH VENKATESH, SEPTANKULAM RAMAKRISHNAN	110574396
ASHUTOSH KALSI	110210166

Project Title	AN ANALYSIS PROJECT BASED ON THE APPLIANCE-ENERGY USAGE OF INTERNET-OF-THINGS (IOT) DEVICES
Date Submitted	10-May-2023

Contents

Overall context for project

The Internet of Things (IoT) technology has been developed for directing and maintaining the environment in smart homes in real-time. It is crucial to predict future energy demand for optimization, schedule regular maintenance, promote future development for IoT devices. Establishing a perfect prediction of energy consumption at the building's level is vital and significant to efficiently managing the consumed energy by utilizing a strong predictive model. Low forecast accuracy is one of several reasons why energy consumption and prediction models have failed to advance. Thus, the focus of our project is to analyze and predict the energy usage of appliances with the aid of IoT devices.

Problem Definition

A deep analysis of the energy usage of appliances measured by IoT devices. By carrying out a thorough examination of the energy consumption of appliances through the use of Internet of Things (IoT) devices, one can obtain significant insights into the way energy is being consumed within a building. This analysis can also highlight areas where improvements can be made to decrease energy wastage and decrease expenses. The data needed for this analysis can be gathered from an IoT network set up within the building. The network devices can measure energy usage at specific intervals, such as hourly or minutely, and then send this data to a central database or cloud platform.

Project Motivation

There are several factors that motivated us for this project. Key factors are as follows:

1. IoT devices are becoming ubiquitous.
2. Applications include but are not limited to smart homes, smart grids, automation, etc.
3. Any electronic device consumes electricity. Therefore, it is imperative to monitor energy usage of appliances using IoT devices for various reasons mentioned previously.

Project Methodology

Our plan of Approach are as follows:

Data Analysis, EDA & Data Visualization

Feature importance and feature engineering

Algorithm selection

Conclusion

Data Source

This dataset was obtained from Amazon Hackaday. The dataset is recorded in a 10 minute interval for 4.5 months. Dataset includes information from sensors obtained from a wireless network, on temperature and humidity. Each wireless node transmitted the temperature and humidity conditions around 3.3 min. Then, the wireless data was averaged for 10 minutes. The energy data was logged every 10 minutes with m-bus energy meters. Weather from the nearest airport weather station (Chievres 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).

Data Analyses & EDA

Our first step is to have a look at the data before diving into analysis. Printing the first five rows of the data gives us a glimpse. The report contains the transposed version for your perusal.

	1	2	3	4	5
date	2016-01-11 17:00:00	2016-01-11 17:10:00	2016-01-11 17:20:00	2016-01-11 17:40:00	2016-01-11 17:50:00
Appliances	60	60	50	60	50
lights	30	30	30	40	40
T1	19.89	19.89	19.89	19.89	19.89
RH_1	47.59667	46.69333	46.30000	46.33333	46.02667
T2	19.2	19.2	19.2	19.2	19.2
RH_2	44.79000	44.72250	44.62667	44.53000	44.50000
T3	19.79	19.79	19.79	19.79	19.79
RH_3	44.73000	44.79000	44.93333	45.00000	44.93333
T4	19.00000	19.00000	18.92667	18.89000	18.89000
RH_4	45.56667	45.99250	45.89000	45.53000	45.73000
T5	17.16667	17.16667	17.16667	17.20000	17.13333
RH_5	55.20	55.20	55.09	55.09	55.03
T6	7.026667	6.833333	6.560000	6.366667	6.300000
RH_6	84.25667	84.06333	83.15667	84.89333	85.76667
T7	17.20000	17.20000	17.20000	17.20000	17.13333
RH_7	41.62667	41.56000	41.43333	41.23000	41.26000
T8	18.2	18.2	18.2	18.1	18.1
RH_8	48.90000	48.86333	48.73000	48.59000	48.59000
T9	17.03333	17.06667	17.00000	17.00000	17.00000
RH_9	45.53	45.56	45.50	45.40	45.29
T_out	6.600000	6.483333	6.366667	6.133333	6.016667
Press_mm_hg	733.5	733.6	733.7	733.9	734.0
RH_out	92	92	92	92	92
Windspeed	7.000000	6.666667	6.333333	5.666667	5.333333
Visibility	63.00000	59.16667	55.33333	47.66667	43.83333
Tdewpoint	5.3	5.2	5.1	4.9	4.8
rv1	13.27543	18.60619	28.64267	10.08410	44.91948
rv2	13.27543	18.60619	28.64267	10.08410	44.91948
NSM	61200	61800	62400	63600	64200
WeekStatus	Weekday	Weekday	Weekday	Weekday	Weekday
Day_of_week	Monday	Monday	Monday	Monday	Monday

The dataframe df_train and df_test's dimensions are then found as shown below.

finding the dimensions of the train and test data

```
1 dim(df_train)
```

```
14803  32
```

```
1 dim(df_test)
```

```
4932  32
```

combining train and test data

```
1 df = rbind(df_train,df_test)
```

```
1 dim(df)
```

```
19735  32
```

In addition to printing the data, the next step is to find the data types. In addition to getting a glimpse of the data types, it also allows us to preview data, and serves as a foundation for data analysis/ pre-processing.

```
'data.frame':   19735 obs. of  32 variables:
 $ date       : POSIXct, format: "2016-01-11 17:10:00" "2016-01-11 17:20:00" ...
 $ Appliances : int  60 50 60 50 60 60 70 430 100 80 ...
 $ lights     : int  30 30 40 40 50 40 40 50 10 30 ...
 $ T1         : num  19.9 19.9 19.9 19.9 19.9 ...
 $ RH_1       : num  46.7 46.3 46.3 46 45.6 ...
 $ T2         : num  19.2 19.2 19.2 19.2 19.2 ...
 $ RH_2       : num  44.7 44.6 44.5 44.5 44.5 ...
 $ T3         : num  19.8 19.8 19.8 19.8 19.7 ...
 $ RH_3       : num  44.8 44.9 45 44.9 44.9 ...
 $ T4         : num  19 18.9 18.9 18.9 18.9 ...
 $ RH_4       : num  46 45.9 45.5 45.7 45.9 ...
 $ T5         : num  17.2 17.2 17.2 17.1 17.1 ...
 $ RH_5       : num  55.2 55.1 55.1 55 54.9 ...
 $ T6         : num  6.83 6.56 6.37 6.3 6.19 ...
 $ RH_6       : num  84.1 83.2 84.9 85.8 86.4 ...
 $ T7         : num  17.2 17.2 17.2 17.1 17.1 ...
 $ RH_7       : num  41.6 41.4 41.2 41.3 41.2 ...
 $ T8         : num  18.2 18.2 18.1 18.1 18.1 ...
 $ RH_8       : num  48.9 48.7 48.6 48.6 48.6 ...
 $ T9         : num  17.1 17 17 17 17 ...
 $ RH_9       : num  45.6 45.5 45.4 45.3 45.3 ...
 $ T_out      : num  6.48 6.37 6.13 6.02 5.92 ...
 $ Press_mm_hg: num  734 734 734 734 734 ...
 $ RH_out     : num  92 92 92 92 91.8 ...
 $ Windspeed  : num  6.67 6.33 5.67 5.33 5.17 ...
 $ Visibility : num  59.2 55.3 47.7 43.8 40 ...
 $ Tdewpoint  : num  5.2 5.1 4.9 4.8 4.68 ...
 $ rv1        : num  18.6 28.6 10.1 44.9 33 ...
 $ rv2        : num  18.6 28.6 10.1 44.9 33 ...
 $ NSM        : int  61800 62400 63600 64200 65400 66000 66600 68400 69600 72000 ...
 $ WeekStatus : Factor w/ 2 levels "Weekday","Weekend": 1 1 1 1 1 1 1 1 1 1 ...
 $ Day_of_week: Factor w/ 7 levels "Friday","Monday",..: 2 2 2 2 2 2 2 2 2 2 ...
```

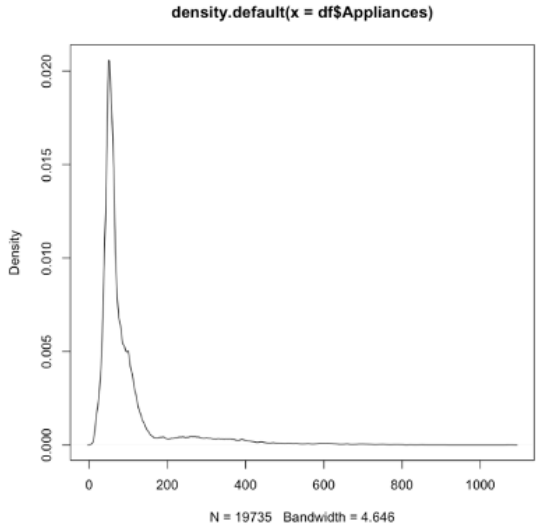
```
'date'  'Appliances'  'lights'  'T1'  'RH_1'  'T2'  'RH_2'  'T3'  'RH_3'  'T4'  'RH_4'  'T5'  'RH_5'  'T6'  'RH_6'  'T7'  'RH_7'  'T8'  'RH_8'  'T9'
'RH_9'  'T_out'  'Press_mm_hg'  'RH_out'  'Windspeed'  'Visibility'  'Tdewpoint'  'rv1'  'rv2'  'NSM'  'WeekStatus'  'Day_of_week'
```

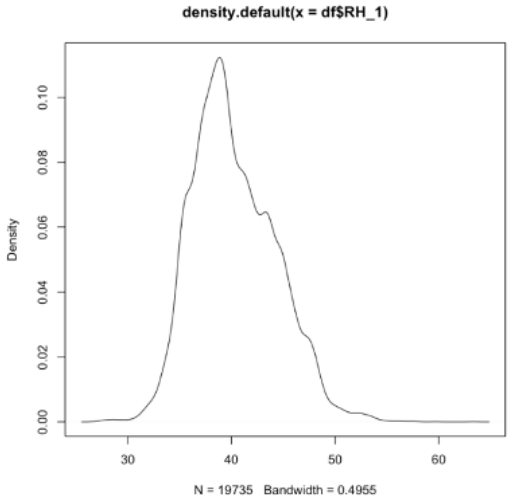
The summary function allows us to obtain a quick summary of the dataframe in consideration. It also conveys the properties and the distribution of a dataframe. This is considered to be essential before EDA.

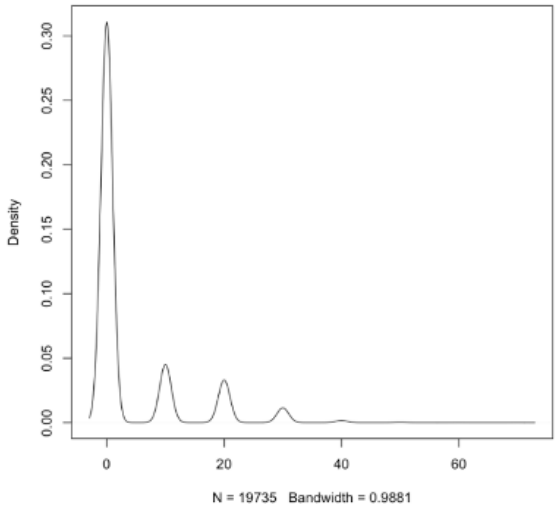
```
summary(df)
```

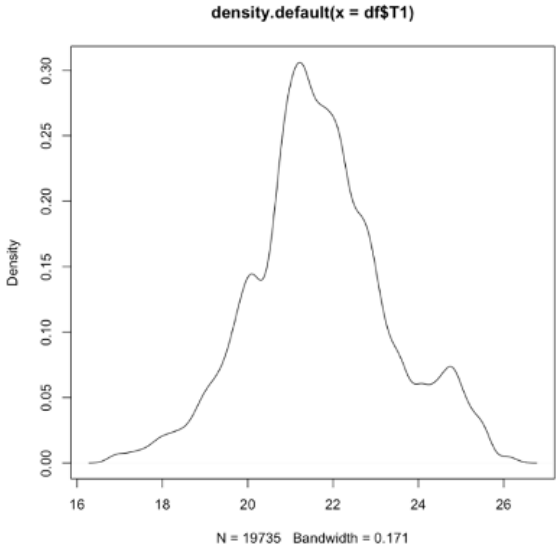
```
     date                       Appliances          lights       
Min.   :2016-01-11 17:00:00   Min.   :  10.00   Min.   : 0.000  
1st Qu.:2016-02-14 23:15:00   1st Qu.:  50.00   1st Qu.: 0.000  
Median :2016-03-20 05:30:00   Median :  60.00   Median : 0.000  
Mean   :2016-03-20 05:30:00   Mean   :  97.69   Mean   : 3.802  
3rd Qu.:2016-04-23 11:45:00   3rd Qu.: 100.00   3rd Qu.: 0.000  
Max.   :2016-05-27 18:00:00   Max.   :1080.00   Max.   :70.000  

       T1              RH_1             T2              RH_2       
Min.   :16.79   Min.   :27.02   Min.   :16.10   Min.   :20.46  
1st Qu.:20.76   1st Qu.:37.33   1st Qu.:18.79   1st Qu.:37.90  
Median :21.60   Median :39.66   Median :20.00   Median :40.50  
Mean   :21.69   Mean   :40.26   Mean   :20.34   Mean   :40.42  
3rd Qu.:22.60   3rd Qu.:43.07   3rd Qu.:21.50   3rd Qu.:43.26  
Max.   :26.26   Max.   :63.36   Max.   :29.86   Max.   :56.03  

       T3              RH_3             T4              RH_4       
Min.   :17.20   Min.   :28.77   Min.   :15.10   Min.   :27.66  
1st Qu.:20.79   1st Qu.:36.90   1st Qu.:19.53   1st Qu.:35.53  
Median :22.10   Median :38.53   Median :20.67   Median :38.40  
Mean   :22.27   Mean   :39.24   Mean   :20.86   Mean   :39.03  
3rd Qu.:23.29   3rd Qu.:41.76   3rd Qu.:22.10   3rd Qu.:42.16  
Max.   :29.24   Max.   :50.16   Max.   :26.20   Max.   :51.09  

       T5              RH_5             T6              RH_6       
Min.   :15.33   Min.   :29.82   Min.   :-6.065   Min.   : 1.00  
1st Qu.:18.28   1st Qu.:45.40   1st Qu.: 3.627   1st Qu.:30.02  
Median :19.39   Median :49.09   Median : 7.300   Median :55.29  
Mean   :19.59   Mean   :50.95   Mean   : 7.911   Mean   :54.61  
3rd Qu.:20.62   3rd Qu.:53.66   3rd Qu.:11.256   3rd Qu.:83.23  
Max.   :25.80   Max.   :96.32   Max.   :28.290   Max.   :99.90  

       T7              RH_7             T8              RH_8       
Min.   :15.39   Min.   :23.20   Min.   :16.31   Min.   :29.60  
1st Qu.:18.70   1st Qu.:31.50   1st Qu.:20.79   1st Qu.:39.07  
Median :20.03   Median :34.86   Median :22.10   Median :42.38  
Mean   :20.27   Mean   :35.39   Mean   :22.03   Mean   :42.94  
3rd Qu.:21.60   3rd Qu.:39.00   3rd Qu.:23.39   3rd Qu.:46.54  
Max.   :26.00   Max.   :51.40   Max.   :27.23   Max.   :58.78  


       T9              RH_9            T_out          Press_mm_hg     
Min.   :14.89   Min.   :29.17   Min.   :-5.000   Min.   :729.3  
1st Qu.:18.00   1st Qu.:38.50   1st Qu.: 3.667   1st Qu.:750.9  
Median :19.39   Median :40.90   Median : 6.917   Median :756.1  
Mean   :19.49   Mean   :41.55   Mean   : 7.412   Mean   :755.5  
3rd Qu.:20.60   3rd Qu.:44.34   3rd Qu.:10.408   3rd Qu.:760.9  
Max.   :24.50   Max.   :53.33   Max.   :26.100   Max.   :772.3  

     RH_out         Windspeed        Visibility      Tdewpoint     
Min.   : 24.00   Min.   : 0.000   Min.   : 1.00   Min.   :-6.600  
1st Qu.: 70.33   1st Qu.: 2.000   1st Qu.:29.00   1st Qu.: 0.900  
Median : 83.67   Median : 3.667   Median :40.00   Median : 3.433  
Mean   : 79.75   Mean   : 4.040   Mean   :38.33   Mean   : 3.761  
3rd Qu.: 91.67   3rd Qu.: 5.500   3rd Qu.:40.00   3rd Qu.: 6.567  
Max.   :100.00   Max.   :14.000   Max.   :66.00   Max.   :15.500  

      rv1                rv2                NSM          WeekStatus    
Min.   : 0.00532   Min.   : 0.00532   Min.   :    0   Weekday:14263  
1st Qu.:12.49789   1st Qu.:12.49789   1st Qu.:21600   Weekend: 5472  
Median :24.89765   Median :24.89765   Median :43200                  
Mean   :24.98803   Mean   :24.98803   Mean   :42907                  
3rd Qu.:37.58377   3rd Qu.:37.58377   3rd Qu.:64200                  
Max.   :49.99653   Max.   :49.99653   Max.   :85800                  

    Day_of_week  
Friday   :2845  
Monday   :2778  
Saturday :2736  
Sunday   :2736  
Thursday :2880  
Tuesday  :2880  
Wednesday:2880
```

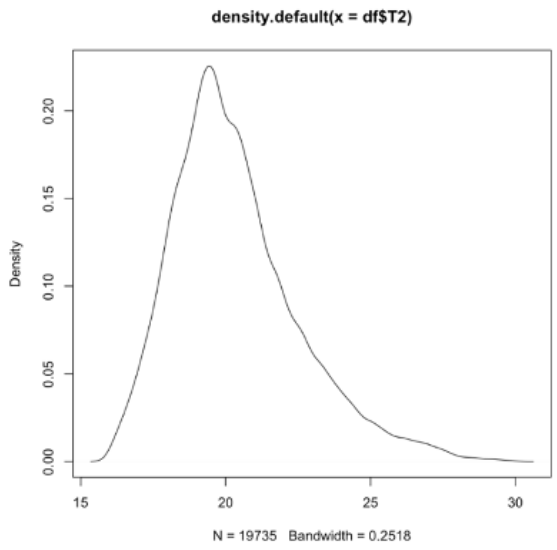
The first in Exploratory Data Analysis (EDA) is to plot density function plots for each feature present in the dataframe. The density plot allows one to describe the relative likelihood of each possible value present in the dataframe. This allows us to visualize the features, granting a pictorial perspective. It is observed that the "appliances" and "lights feature" are long-tailed. Other than a T7, T9, RH_6, a curve somewhat resembling a bell shape is observed.

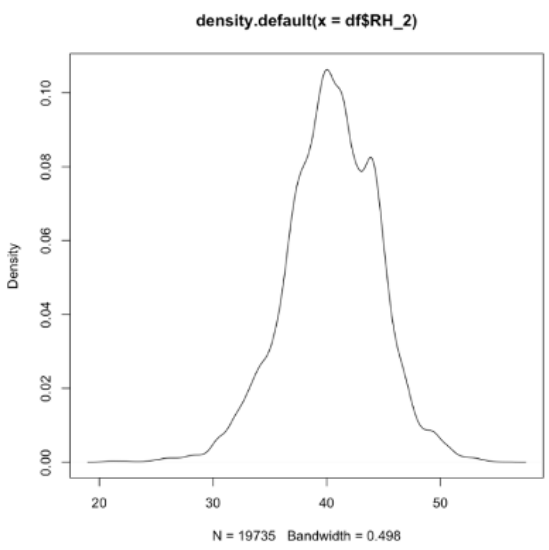


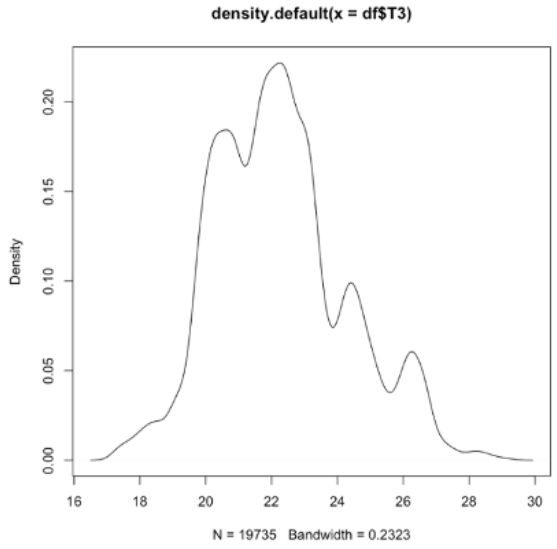


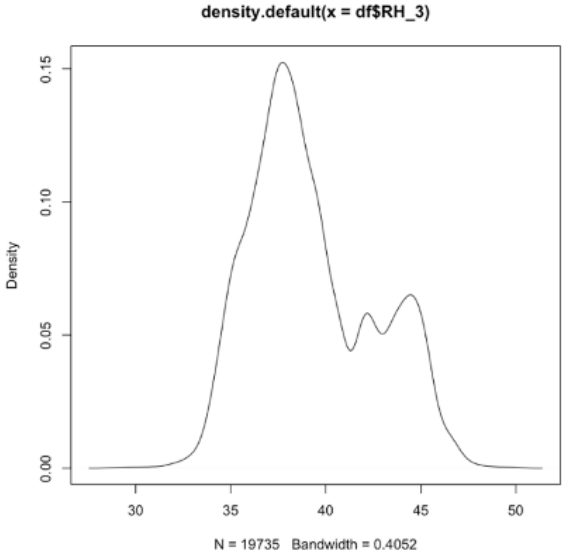


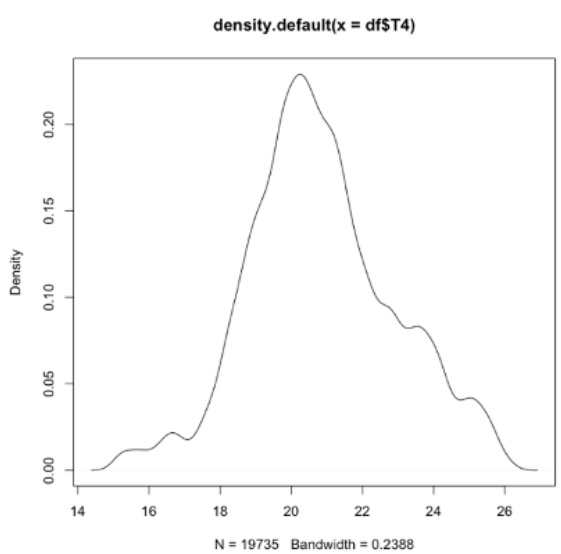


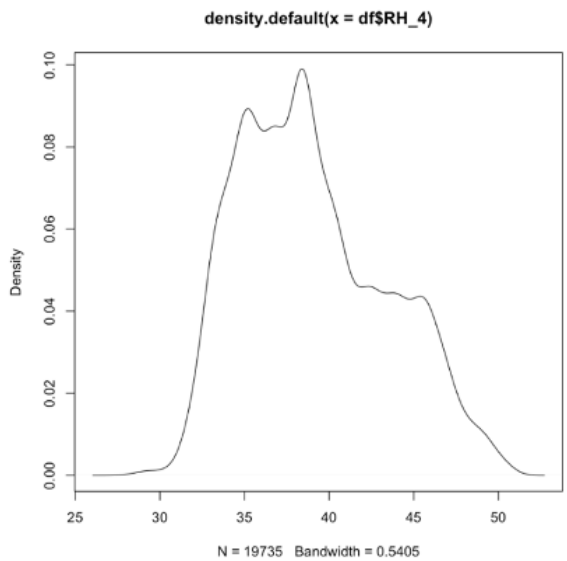

density.default(x = df$T2)
Density
N = 19735 Bandwidth = 0.2518

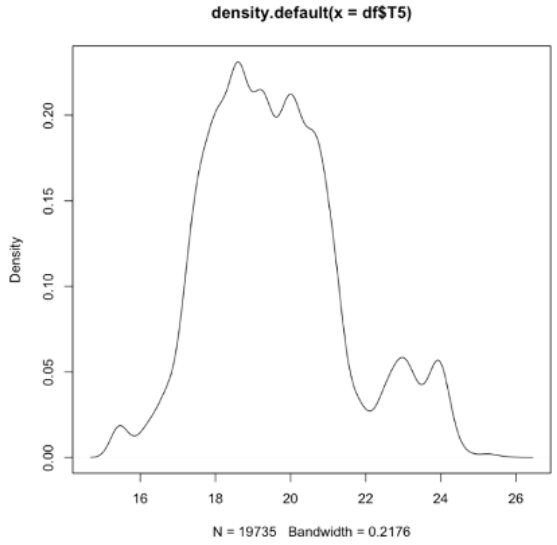

density.default(x = df$RH_2)
Density
N = 19735 Bandwidth = 0.498

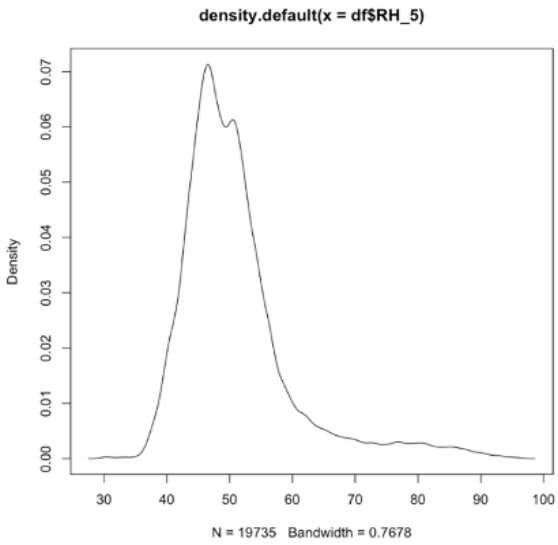

density.default(x = df$T3)
Density
N = 19735 Bandwidth = 0.2323

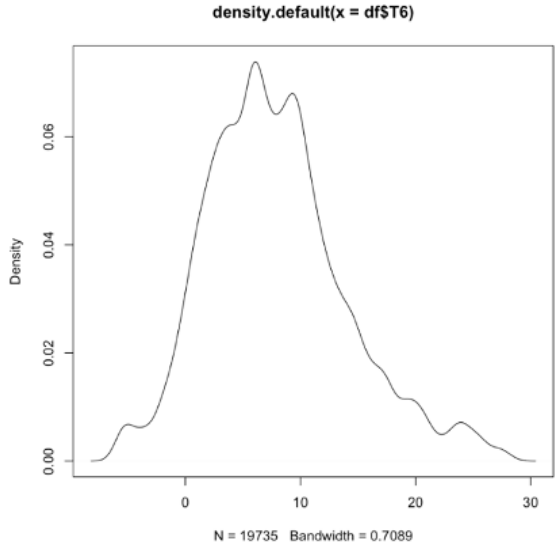

density.default(x = df$RH_3)
Density
N = 19735 Bandwidth = 0.4052

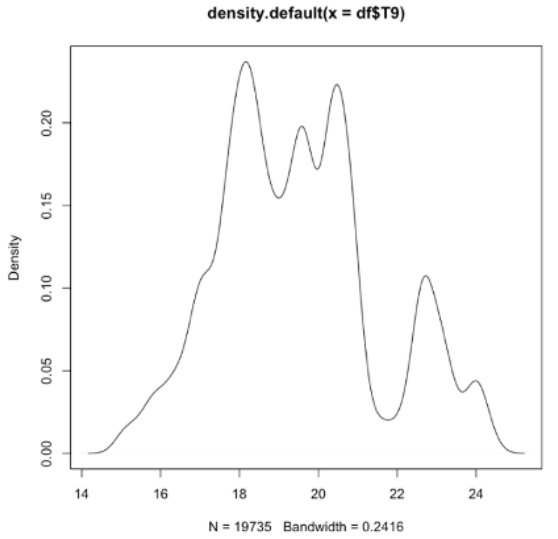

density.default(x = df$T4)
Density
N = 19735 Bandwidth = 0.2388

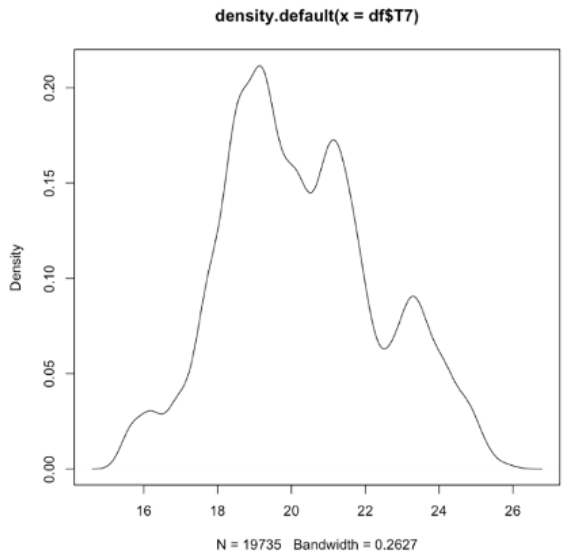

density.default(x = df$RH_4)
Density
N = 19735 Bandwidth = 0.5405

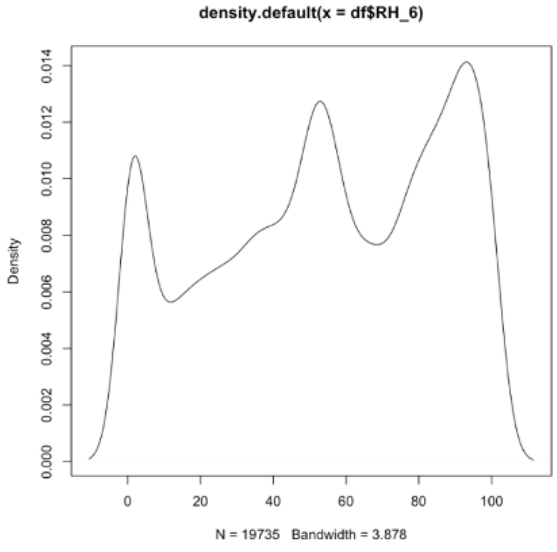

density.default(x = df$T5)
Density
0.00
0.05
0.10
0.15
0.20
16
18
20
22
24
26
N = 19735 Bandwidth = 0.2176

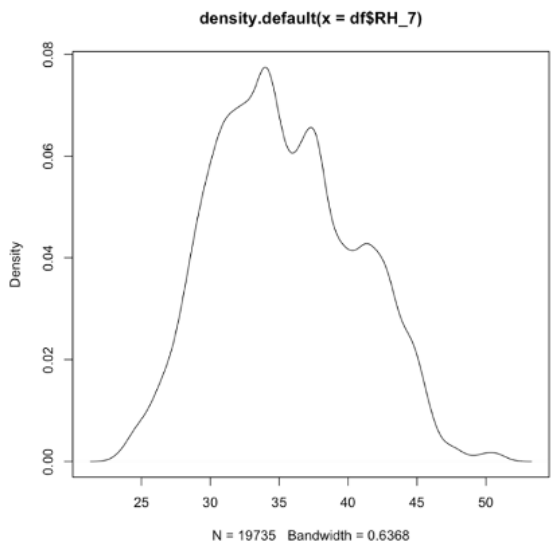

density.default(x = df$RH_5)
Density
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
30
40
50
60
70
80
90
100
N = 19735 Bandwidth = 0.7678

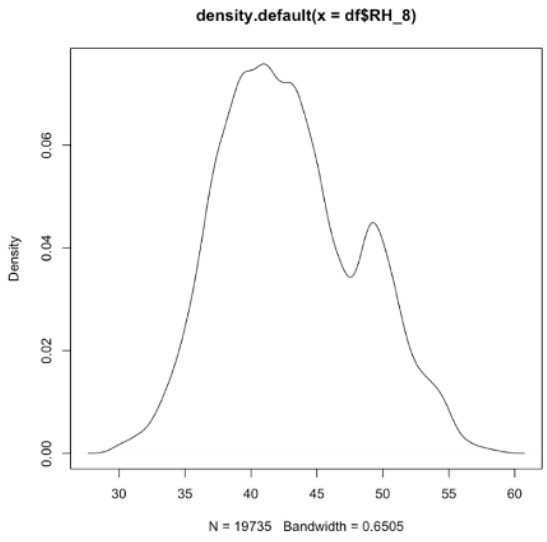

density.default(x = df$T6)
Density
0.00
0.02
0.04
0.06
0
10
20
30
N = 19735 Bandwidth = 0.7089


density.default(x = df$T9)
Density
0.00
0.05
0.10
0.15
0.20
14
16
18
20
22
24
N = 19735 Bandwidth = 0.2416


density.default(x = df$T7)
Density
0.00
0.05
0.10
0.15
0.20
16
18
20
22
24
26
N = 19735 Bandwidth = 0.2627

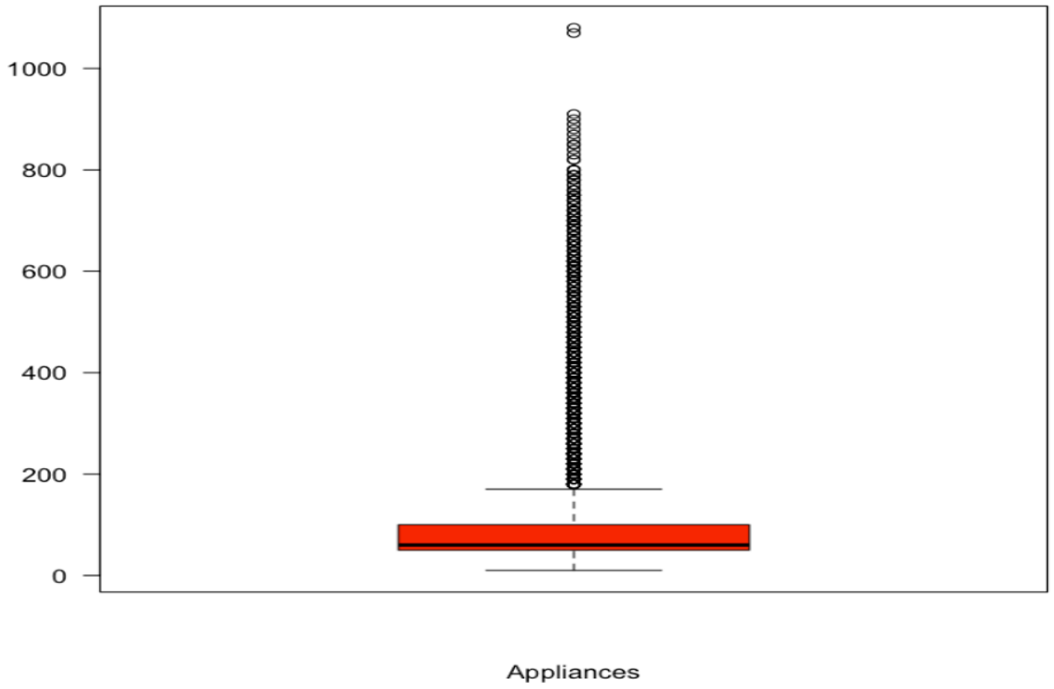

density.default(x = df$RH_6)
Density
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
0
20
40
60
80
100
N = 19735 Bandwidth = 3.878

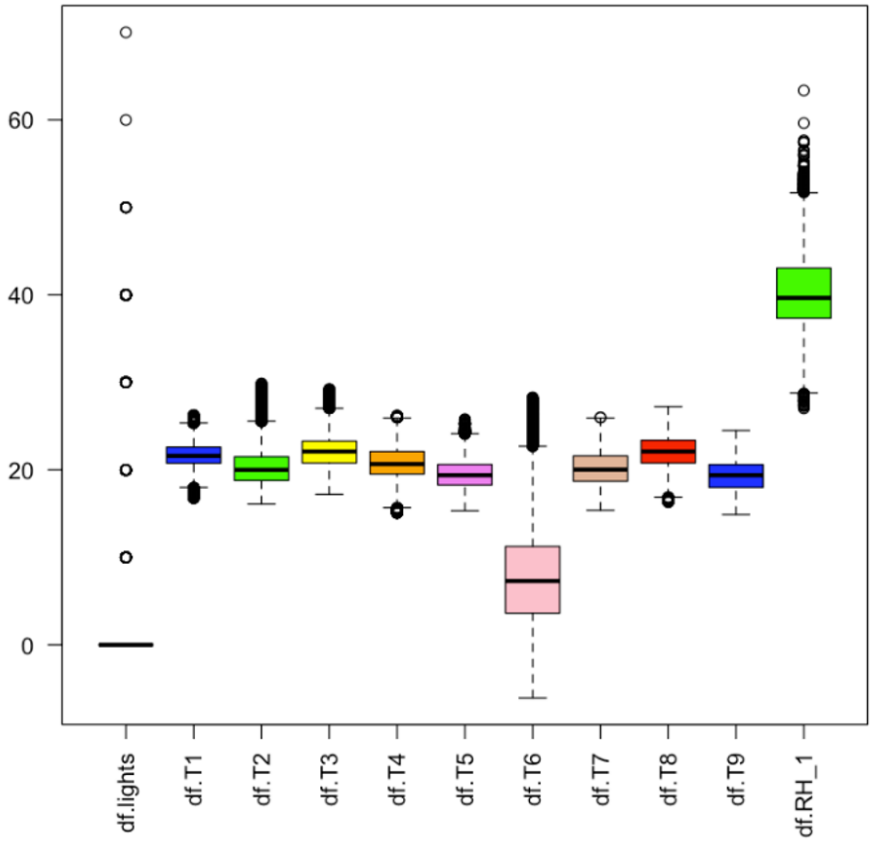




Plot for outliers

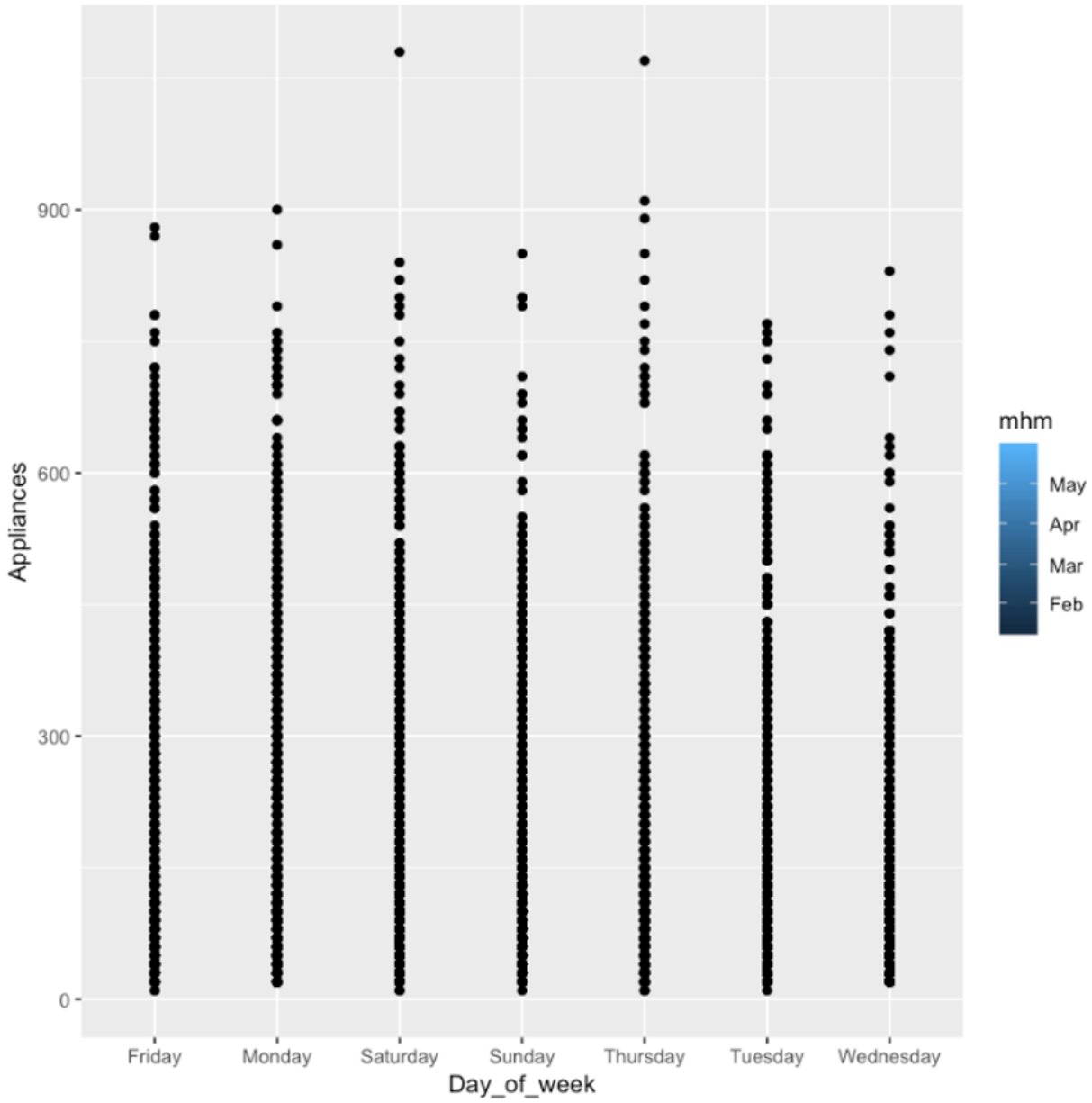
The use of boxplot is done to find outliers in the dataset. All features were subjected to this. We were able to notice that "Appliances" had the majority number of outliers. This is followed by RH_1 and T6.



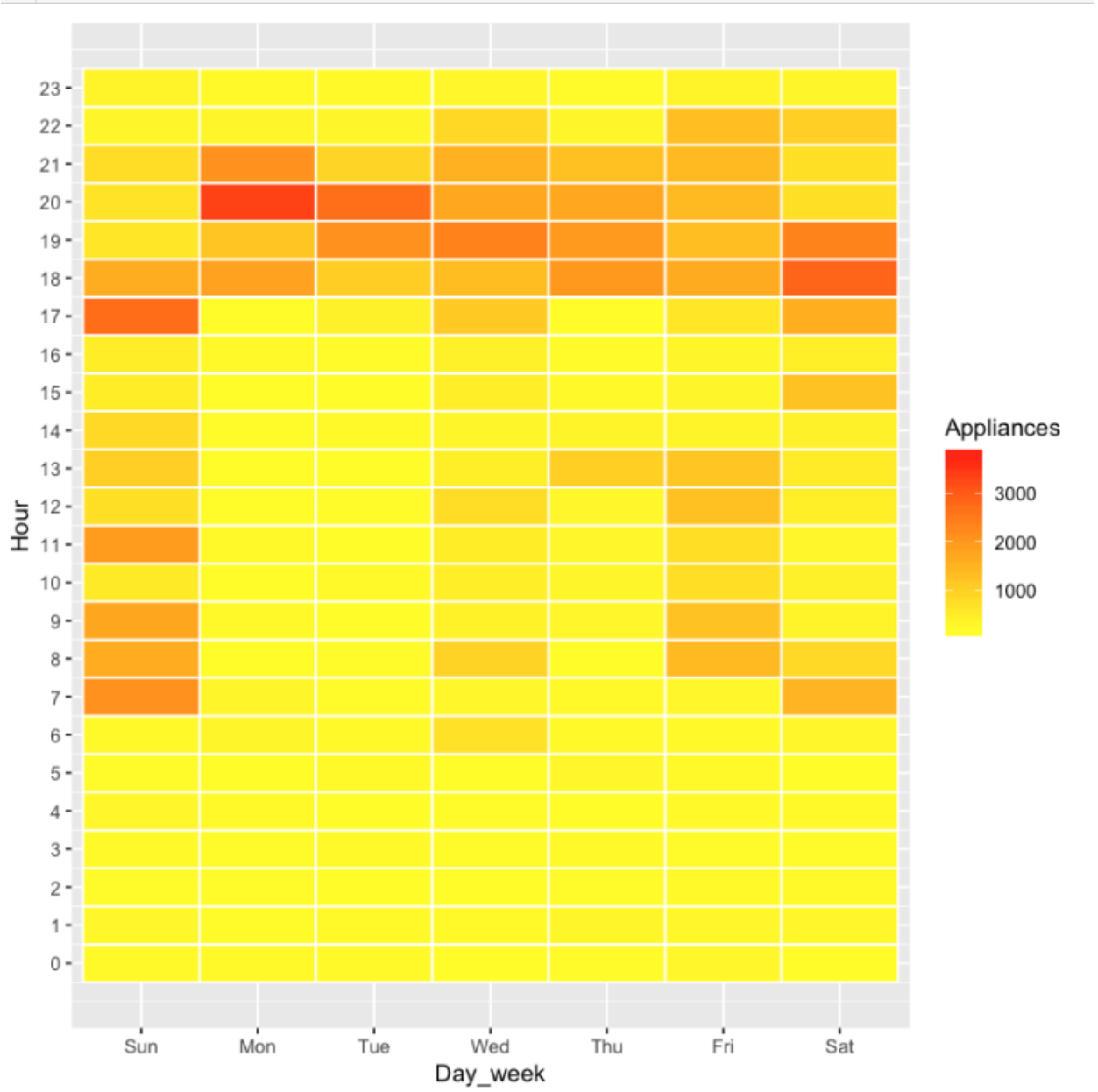

60
40
20
0
df.lights
df.T1
df.T2
df.T3
df.T4
df.T5
df.T6
df.T7
df.T8
df.T9
df.RH_1

Bivariate Analysis

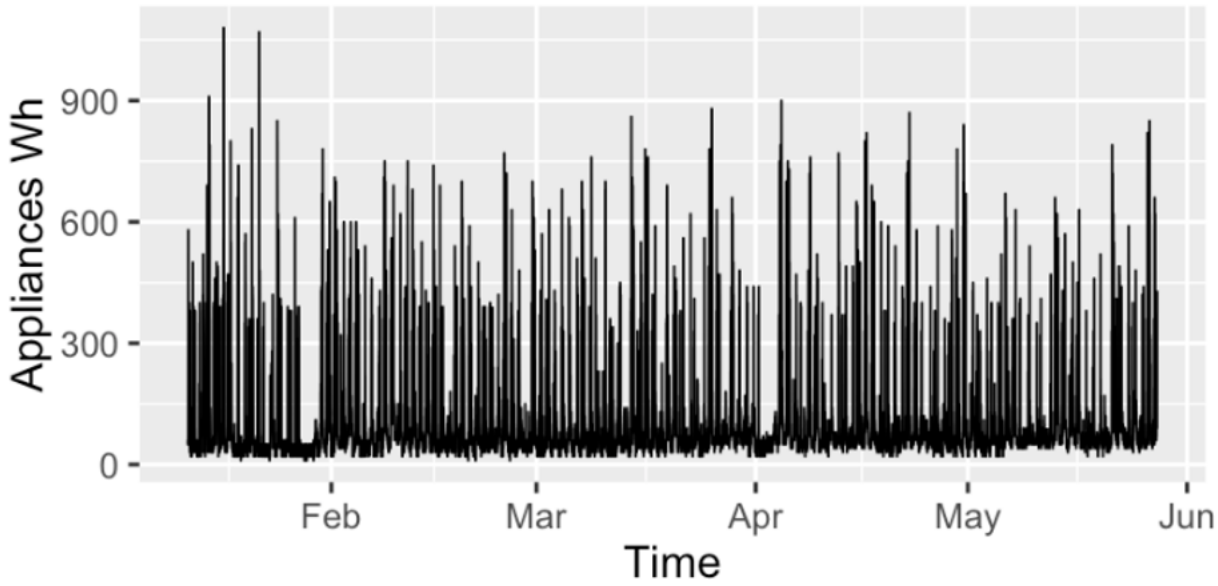
We moved forward with Bivariate Analysis. Bivariate analysis allows us to find correlation between two features (in this case) and allows us to identify patterns and usage. “Day_of_week'' and “Appliances” were used for comparison.



Correlation Analysis was performed between “Day_of_week” and “Hour.” The darker the color, the higher it is. We see higher correlation values at Monday, 8PM; Saturday, 6PM; Sunday, 5PM.



We also analyzed the relationship between time and Appliances. However, in this case, we analyzed the usage by month. January seems to be having the highest energy usage followed by April.



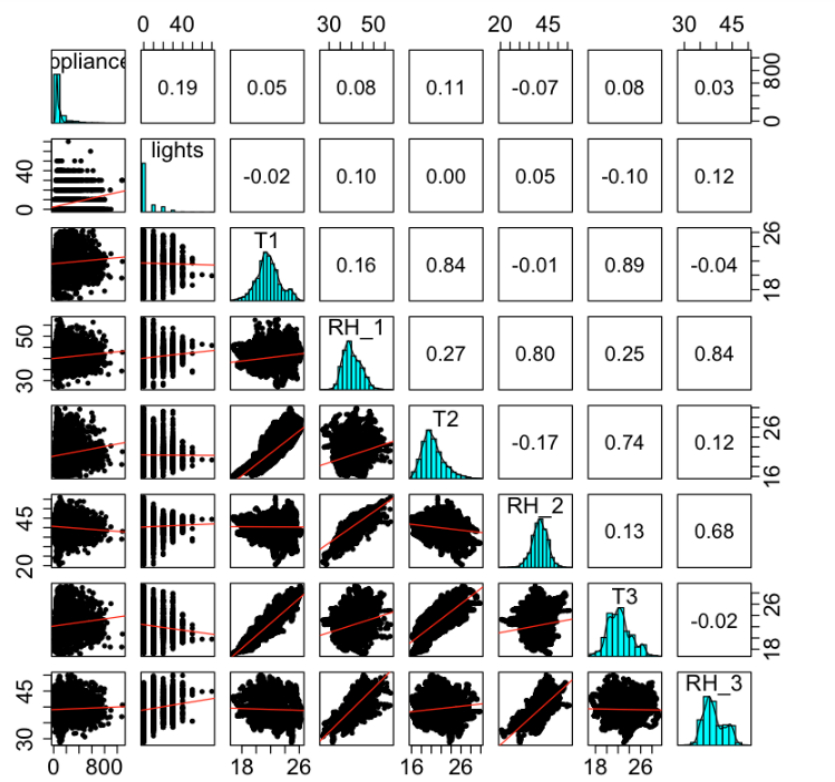
Correlation

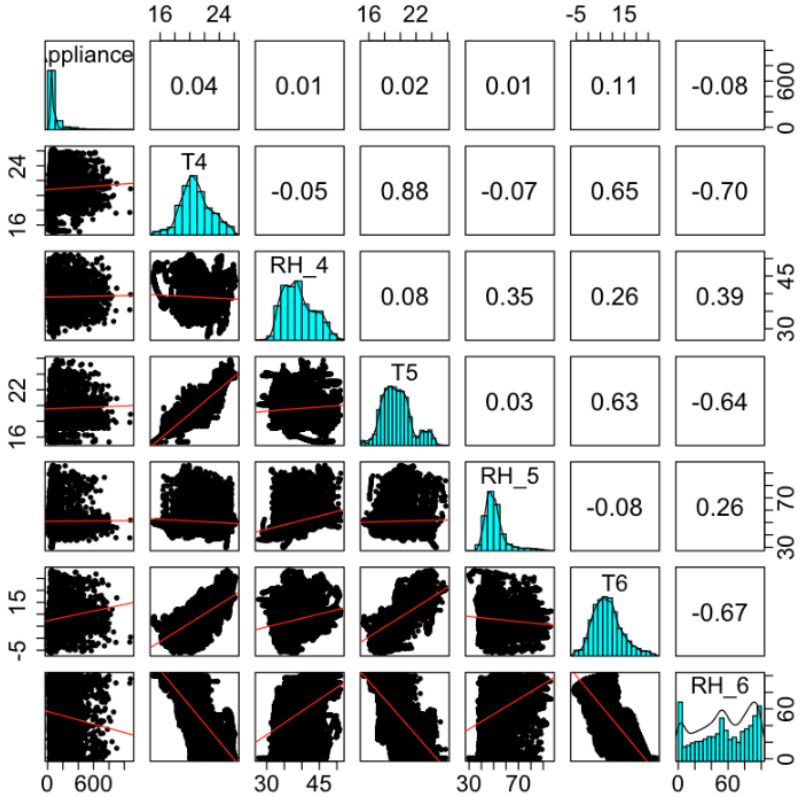
A matrix (Matrices) with a combination of histograms and scatter plots is plotted. To understand the correlation between all the columns, we used the "pairs.panels()" function to plot scatterplots with linear regression and histograms to understand the various patterns, significance, and trends between the features.

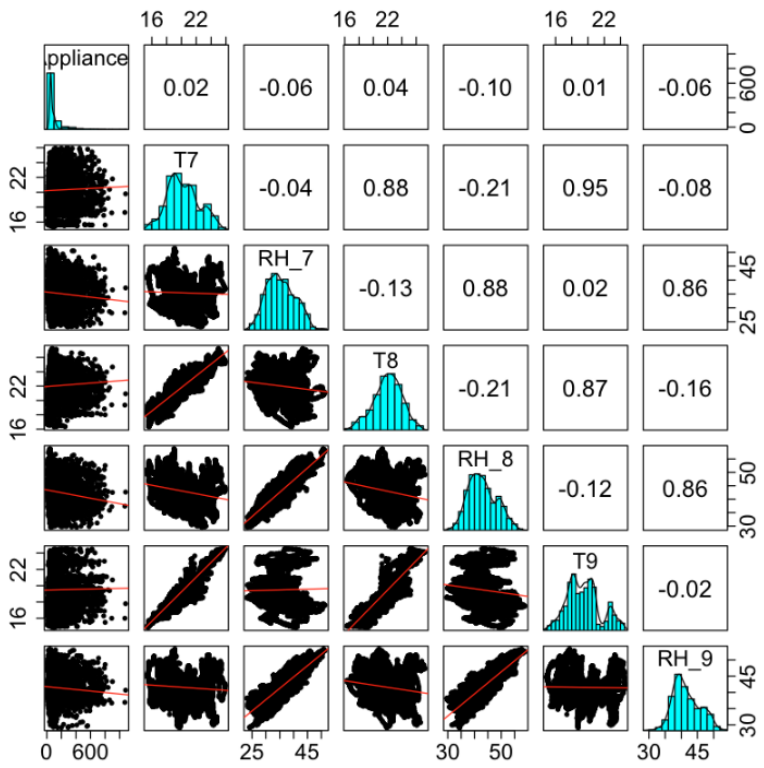
T1 seems to have high correlation with T2 and T3 and vice versa. RH_1 seems to be highly correlated with RH_2 and RH_3 and vice versa. T4 has high correlation with T5 and T6 and converse seems to be true for both T5 and T6. RH_6 seems to be negatively correlated with T6, T5 and T4.

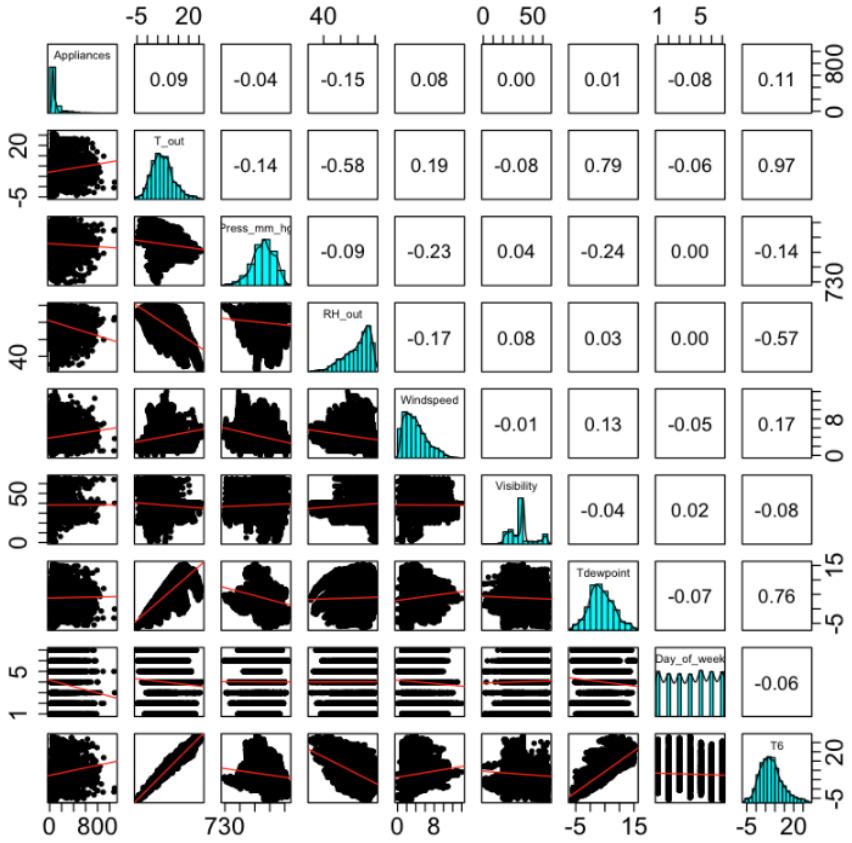
T7 has high correlation with T8 and T9 and converse seems to be true for both T8 and T9.RH_7 has positive high correlation with RH_8 and RH_9.

T_out also has a high correlation with Tdewpoint and T6.


ppliance
lights
T1
RH_1
T2
RH_2
T3
RH_3
0.19
0.05
0.08
0.11
-0.07
0.08
0.03
-0.02
0.10
0.00
0.05
-0.10
0.12
0.16
0.84
-0.01
0.89
-0.04
0.27
0.80
0.25
0.84
-0.17
0.74
0.12
0.13
0.68
-0.02

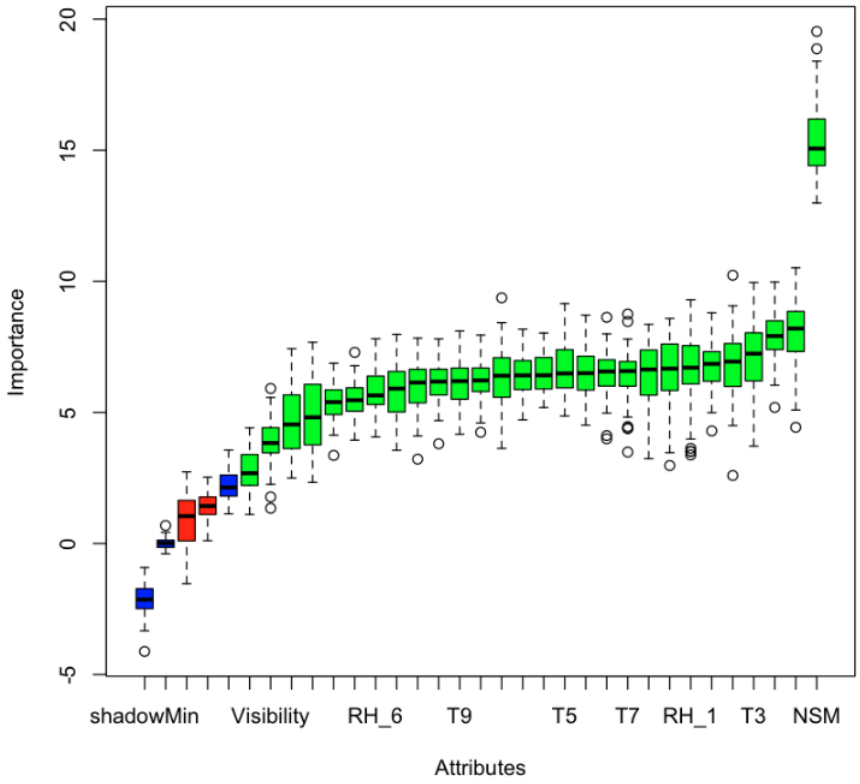

ppliance
T4
RH_4
T5
RH_5
T6
RH_6
0.04
0.01
0.02
0.01
0.11
-0.08
-0.05
0.88
-0.07
0.65
-0.70
0.08
0.35
0.26
0.39
0.03
0.63
-0.64
-0.08
0.26
-0.67


ppliance
T7
RH_7
T8
RH_8
T9
RH_9
0.02
-0.06
0.04
-0.10
0.01
-0.04
0.88
-0.21
0.95
-0.13
0.02
-0.16
-0.12
-0.02


Appliances
T_out
Press_mm_hg
RH_out
Windspeed
Visibility
Tdewpoint
Day_of_week
T6
0.09
-0.04
-0.15
0.08
0.00
0.01
-0.08
0.11
-0.14
-0.58
0.19
-0.08
0.79
-0.06
0.97
-0.09
-0.23
0.04
-0.24
0.00
-0.14
-0.17
0.08
0.03
0.00
-0.57
-0.01
0.13
-0.05
0.17
-0.04
0.02
-0.08
-0.07
0.76
-0.06

Feature Importance and Selection

We moved forward with feature importance and used the same boxplot method to identify. The use of "boruta()" library was essential in this case. The most important variable was identified as "NSM".

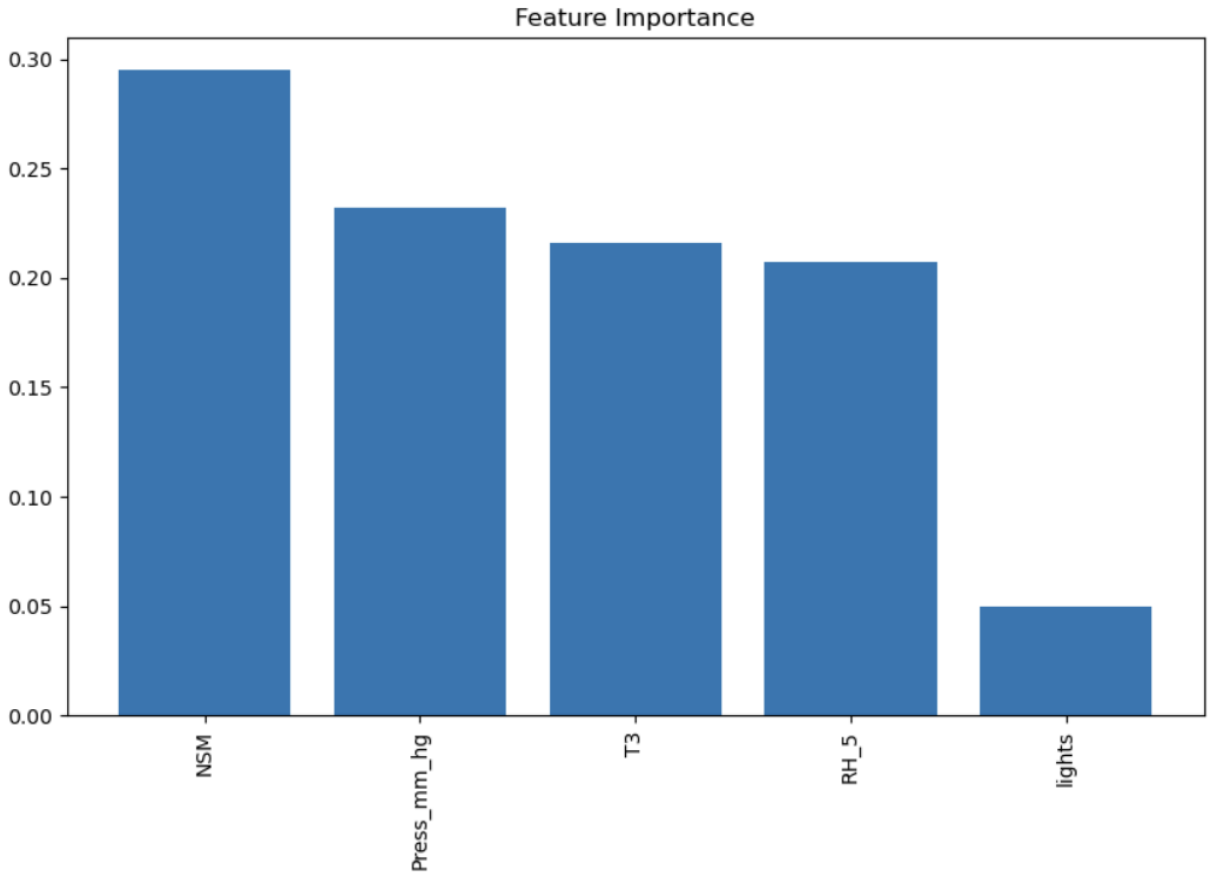


RFE was used for feature selection among the 37 features and we found that NSM, lights, Press_mm_hg, RH_5, T3 are the most important features as shown in the screenshot below.

```
Recursive feature selection

Outer resampling method: Cross-Validated (5 fold)

Resampling performance over subset size:

 Variables  RMSE Rsquared   MAE RMSESD RsquaredSD  MAESD Selected
         1 94.58   0.1463 51.78  2.529    0.01316 1.1667
         3 91.93   0.2019 49.71  3.322    0.02836 1.8177
         5 85.02   0.3475 44.76  3.221    0.02930 1.5871
         7 72.82   0.5004 35.01  2.451    0.01833 1.0350
         9 72.26   0.5057 34.45  2.046    0.01211 0.9410
        11 71.94   0.5115 34.27  2.014    0.01233 0.8864
        13 71.85   0.5109 34.06  2.158    0.01687 0.9402
        15 71.52   0.5149 33.95  2.122    0.01714 0.9251
        35 70.42   0.5298 33.48  2.076    0.01857 0.8240        *

The top 5 variables (out of 35):
   NSM, lights, Press_mm_hg, RH_5, T3
```

Statistical model design

Based on our Data Analysis and EDA and the feature importance and selection process, we see that NSM is the most important feature and the top five variables to be used in our models are NSM, lights, Press_mm_hg, RH_5, and T3 shown below.



Following are the list of models we used to train and test on our dataset:

- RandomForest
- GradientBoostingClassifier
- ExtraTreeRegressor
- XGBRegressor

The metrics used are RMSE and R2_Score, but RMSE is the primary metric to compare the models. Following table shows the results of various models.

	Name	Train_R2_Score	Test_R2_Score	Test_RMSE_Score
0	RandomForest	0.934229	0.435089	0.751606
1	ExtraTreeRegressor :	1.000000	0.477074	0.723136
2	GradientBoostingClassifier:	0.283220	0.226379	0.879557
3	XGBRegressor:	0.728595	0.362877	0.798200

Amongst the four models used, ExtraTreeRegressor showed comparatively better performance. This model was selected to be fine-tuned. After fine tuning the model, the RMSE score decreased by 1% and the R2_score for the test increased by 1%. Screenshot attached for reference.

```
Training set R2 Score -  1.0
Testing set R2 Score -  0.4831749374033385
Testing set RMSE Score -  0.718905461515394
```

Key insights/findings and conclusion

Several inferences can be drawn from this project. Firstly, amongst the original 32 features, five additional features were engineered. With this, we performed RFE which narrowed down to only five important features - NSM, lights, Press_mm_hg, RH_5, and T3. With this, we were able to maximize the performance of the model (reduce the risk of overfitting).

Secondly, we are not aware of the environmental conditions the data was retrieved from. This means that this is an ideal case where we just compare the use case and not consider other factors that might come into play. Some may include the material of the house, material of doors and windows and whether they were kept open, etc. Third, several columns present in the dataframe were not considered in the final model implementation. This suggests improvements are required to get more data to experiment with.

The best model evaluated with the dataset we used in the limited time is ExtraTreeRegressor with its hyperparameter tuned. There are many ways the findings of this project can be utilized. One way is to use the findings to optimize energy consumption in households and other buildings. By identifying the energy usage patterns of various IoT devices, households and building managers can adjust their energy consumption habits and optimize energy usage, resulting in reduced energy costs and a smaller carbon footprint.

In addition, the results can be used to inform the development and implementation of energy policies and regulations. Policymakers can use the findings to identify areas where energy efficiency can be improved and to incentivize the adoption of energy-efficient IoT devices. This can lead to a more sustainable and efficient energy system.

Moreover, the results of the project can be used to inform the development of new IoT devices and technologies that are more energy-efficient. The findings can be used to identify areas where energy efficiency can be improved and to develop new technologies that are more efficient and cost-effective. This can help to drive innovation in the IoT industry and lead to the development of more sustainable and efficient devices.

Overall, the best results of the project can have significant implications for the environment, energy costs, and technological innovation, and can be utilized by various stakeholders such as households, building managers, policymakers, and technology companies to optimize energy consumption and promote sustainability.

Potential real-world applications of project

Several potential real-world applications of this endeavor are highlighted below:

A strategy for controlling the energy consumption of IoT devices by optimizing the scheduling of their operation and the use of sensors is called energy-efficient scheduling of IoT devices and their respective sensors. With this method, IoT device carbon footprints will be reduced, efficiency will be increased, costs will be brought down, maintenance will be predicted, and smart home automation will be made possible.

IoT device scheduling can be improved to make sure that they are only active when necessary. An IoT device that tracks a room's temperature, for instance, can be set up to turn off when the space is empty or when the temperature falls within a predetermined range. Similar to this, an IoT device that manages lighting can be programmed to shut off when nobody is present or when ambient light is adequate.

Sensor use is a crucial component of energy-efficient scheduling. Sensors can be used to track environmental changes and only activate IoT devices when necessary. As an illustration, a temperature sensor that senses a change in temperature can activate an IoT device and modify the temperature. This strategy guarantees that energy is only used when it is required, lowering the total amount of energy used by IoT devices.

By implementing energy-efficient scheduling and sensor usage, the carbon footprint of IoT devices can be significantly reduced. This reduction is achieved by minimizing the amount of energy consumed by the devices, thus reducing the amount of energy that needs to be generated. Additionally, the reduction in energy consumption results in cost savings, as less energy is required to operate the devices.

Sensor use is a crucial component of energy-efficient scheduling. Sensors can be used to track environmental changes and only activate IoT devices when necessary. As an illustration, a temperature sensor that senses a change in temperature can activate an IoT device and modify the temperature. This strategy guarantees that energy is only used when it is required, lowering the total amount of energy used by IoT devices.

Finally, smart-home automation systems can incorporate energy-efficient scheduling and sensor usage. The automatic operation of IoT devices based on environmental conditions, occupancy levels in a room, and other variables is made possible by this integration. For instance, an IoT device that manages lighting can be programmed to automatically turn on and off depending on the amount of natural light present and whether a room is occupied. With this strategy, the

house becomes more energy-efficient overall, the user experience is improved, and energy consumption is decreased.

Limitations of project work

Possibility of further research is expected and therefore the current work can go extinct sooner than expected. Furthermore, the cost of implementing such a system may not be feasible for all homes due to several reasons such as cost, age of home, etc. In addition to this, the capital investment required can be a major factor for people not adapting to smart-homes. As stated above in the key-insights, there were several factors; there were several factors that were not. This project ws computational expensive for a relatively small dataset. This is because of the complexity of the model that was required. Finally, the scale of implementation ought to be large so as to have an impact suggested above.