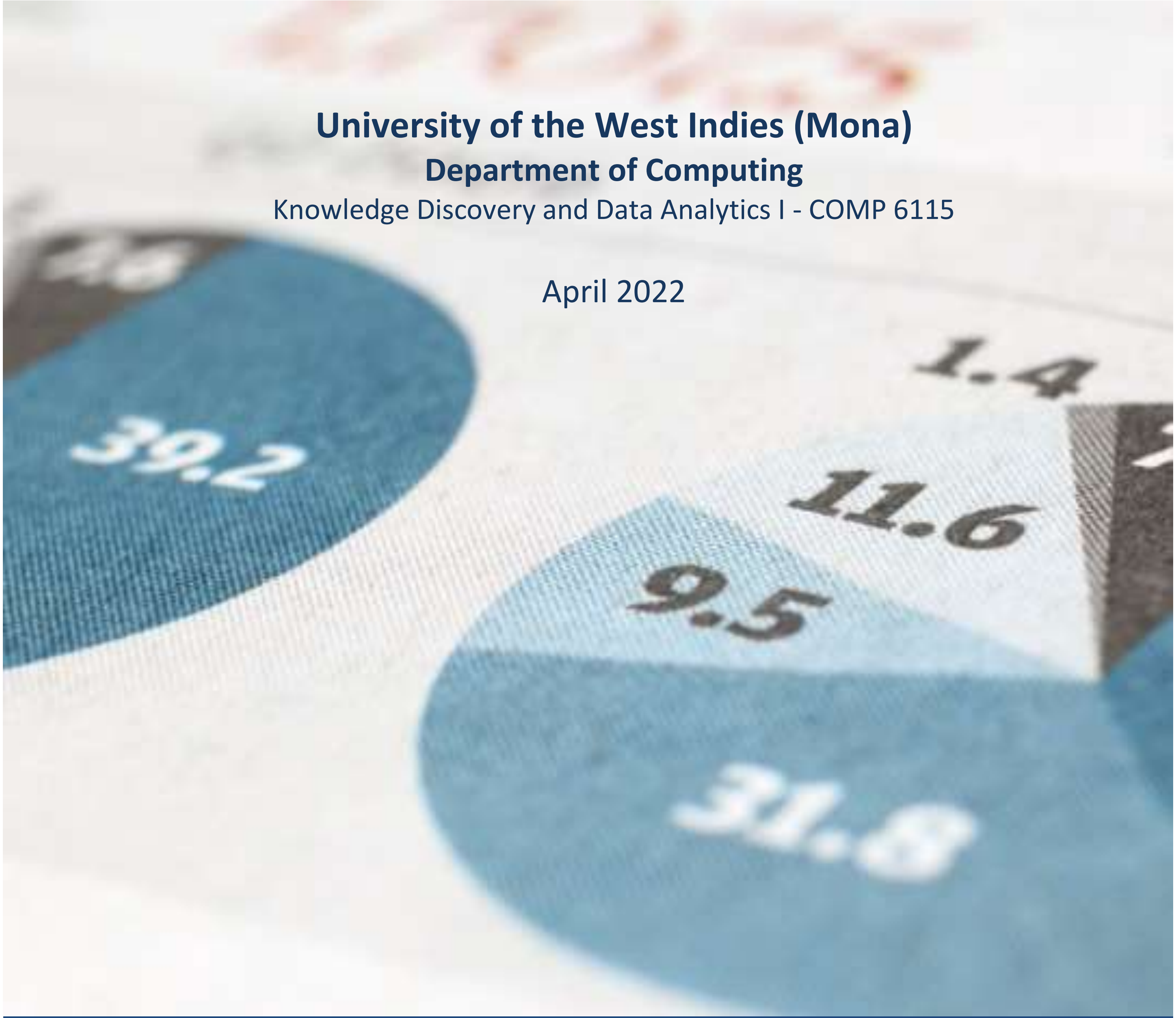


University of the West Indies (Mona)
Department of Computing
Knowledge Discovery and Data Analytics I - COMP 6115

April 2022

Modelling Heating Load

Supervised Learning - Classification

Bertland Hope 03002642 | Ottor Mills 620098373 | Owen Duckie 620040053

Knowledge Discovery and Data Analytics I - COMP6115

Due Date: April 27, 2022
Time: 11:55 pm

From online repositories (e.g. Kaggle or UCI Machine Learning repository https://archive.ics.uci.edu/ml/datasets.php) select a dataset that is suitable for the application of Supervised Machine Learning classification.

Instructions:

1. This project is to be done in groups of two or three.
2. Part I - upload the files on OurVLE as a zipped archive
    - R code
    - Data file (if not used as a link to data file).
    - Documentation for the project in pdf/doc format.

[20 marks]

Select a dataset and identify a practical use case relevant to that dataset. Execute and discuss thoroughly the activities performed in each of the phases of the KDDM cycle:

• Business Understanding	[2 marks]
• Data Understanding & Data Preparation	[4 marks]
• Modelling	[8 marks]
• Evaluation	[4 marks]
• Deployment	[2 marks]

Note:

For the selected dataset you need to identify a business problem and take that data through the various phases of the knowledge discovery process model.

You should justify your choice of modelling techniques and create multiple models (at least 4 using at least two classification algorithms) and then select the best model.

For each generated model record the relevant performance measures and show the process of selecting the best model.

Determine how the selected model satisfies the business objectives and how it could be deployed in the domain.

Show screen shots for relevant charts and plots.

See appendix for guidance on documenting & presenting your results.

Contents

Business Understanding

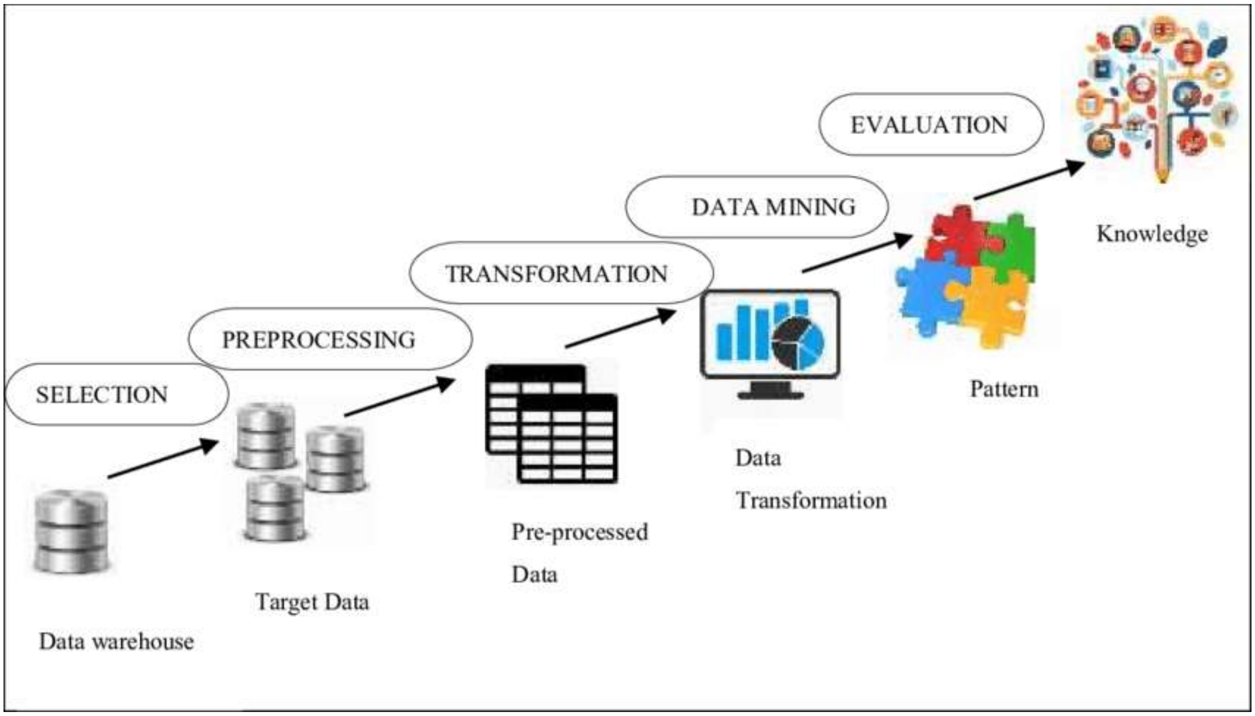
Architecture is a complex process that involves the use of various techniques in order to produce the most elegant housing solutions. The design of a building is only a part of the housing solution and usually brings practicality into question, i.e. the distinction between architecture and engineering. A building's ambient heating and cooling are salient considerations that can minimize potential wasted energy which subsequently minimizes the energy costs of its inhabitants. The ideal temperature of a room that engineers should aim for is the conventional room temperature or about 20–22 °C.

The heating load is the amount of heat energy that would need to be added to a space to maintain the temperature in an acceptable range. Depending on the design of the home, the heating load will vary and so too will the energy consumption. To minimize the energy consumption due to heating loads, which are the highest consumption class, an architecture design company wants to understand what architectural designs will help to reduce the heating loads and what combinations of factors will produce high or low heating loads. At the end of this study various models will be produced that highlight the relationship between the building construction components and its relationship to high heating loads.

This can be achieved with a large enough dataset consisting of the building construction make-up - such as that of the room's walls, roof and orientation etc. Once a suitable model has been created then it can be fed into an algorithm along with an architect's building plans in order to predict if that construction will produce a high heating load or not. The said algorithm could then be sold as a plugin for popular architecture CAD software or as a publicly available API with differently priced tiers suited for different use cases.

Summary of the Knowledge Discovery Process

The knowledge discovery process is an organized way of identifying new, useful and reasonable patterns in data. This process starts at data gathering and continues in the illustration below until knowledge is gained.



Source: https://www.researchgate.net/figure/Knowledge-Discovery-Database-KDD-Process_fig1_334784343

How the data was taken through each step:

Data Selection

Data selection is defined as the identification and usage of a dataset relevant to the analysis. An energy analysis was performed on 12 different building shapes. The buildings differ with respect to the glazing area, the glazing area distribution, and the orientation, amongst other parameters. The dataset comprises 768 samples and 8 features, aiming to predict heating and cooling loads. We will focus on heating loads.

- X1 Relative Compactness
- X2 Surface Area
- X3 Wall Area
- X4 Roof Area
- X5 Overall Height
- X6 Orientation
- X7 Glazing Area
- X8 Glazing Area Distribution
- y1 Heating Load
- y2 Cooling Load

Link to the dataset can be found here: https://www.kaggle.com/datasets/elikplim/eergy-efficiency-dataset

Preprocessing

There was not much noise encountered in the dataset. Regardless, the data was preprocessed by noise and outlier removal. A search and removal process was also carried out for missing values. Based on the structure of the dataset it was agreed that categorization oriented analysis would be applied, which would generate classification models.

Transformation

The data was transformed through min-max normalization of the dependent variable fields (X1, X2, X3 and X4) to a scale ranging from 0 - 10. The Y2 column was removed since it was irrelevant to the objective. A field called Y1_Type was created and stored the binned results of the Y1 field. The Y1 field was binned into three ranges labeled 1, 2 and 3, with 1 and 3 being low and 2 being high. A field called Y1_ctype was created and contained boolean, categorical values of either "Low" or "High" depending on the value in the Y1_Type field

Data-Mining

Once the data had been transformed, it was used to train various classification models such as a neural network, logistic regression and decision tree.

The correlation between fields were first carried out in order to determine candidate fields that can be used to generate the most accurate model. It was discovered that X5, X6 & X8 were categorical variables that helped to understand the structure and make-up of each building type. The X1 field was a good candidate for model generation since it has a high correlation to the dependent variable Y1. X1 and X2 are highly inversely correlated, so to avoid multicollinearity in logistic regression models a choice was made to use either X1 or X2. Since X1 is a variable that spoke about relatively compactness, it made more sense to use as a predictor of Heating Load than X2; hence X1 was therefore selected as the independent variable for three of the four generated models. A combination of fields X1, X2, X3 and X4 had a very strong correlation with the target variable Y1 and they were used to generate the fourth and final model, which used logistic regression.

Since multicollinearity was not a major factor in the Decision Trees and Neural Network models, all factors were used to generate models for each algorithm.

Evaluation

An evaluation process was carried out on each model and the best model was selected for the knowledge process. Measurements such as the accuracy, simplicity, area under the curve and stability were used with weighted averages for model selection. These measurements can be found in the evaluation section of this document.

Knowledge

After the above mentioned steps a robust model was generated to provide *knowledge* to architects by making accurate predictions on the temperature of a room. By predicting the ambient heat of a room, the model could provide estimated comfort levels of future occupants of a room and by extension the building.

Data Understanding and Data Preparation

Data exploration is the initial step in data analysis, where users explore a large data set in an unstructured way to uncover initial patterns, characteristics, and points of interest. This process isn't meant to reveal every bit of information a dataset holds, but rather to help create a broad picture of important trends and major points to study in greater detail. The selected dataset contains the dimensions of rooms and the corresponding recorded ambient temperature of each room. After perusing the data for a few minutes it was discovered that the class values varied within a range. These ranged values were ideal candidates for binning and category assignment once adequate cleaning is carried out. The steps below describes the preparation process in detail:

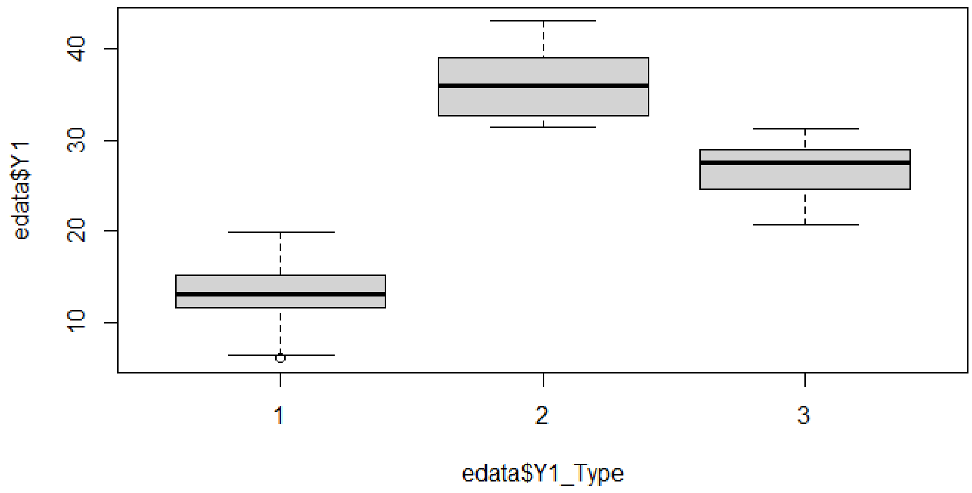
The steps taken to prepare the data are:

1. *Normalization*

   The data min-max normalized between the range of 0 - 10 in order to get the values on a similar scale

2. *Binning*

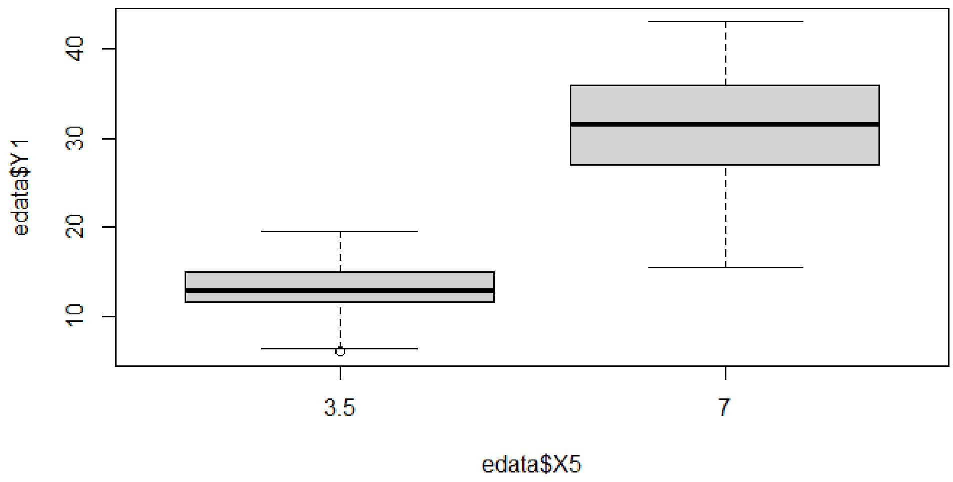
   The target variable 'Y1' was categorized into three bins of three values 1, 2 and 3 using a k-means cluster.



3. *Categorization*

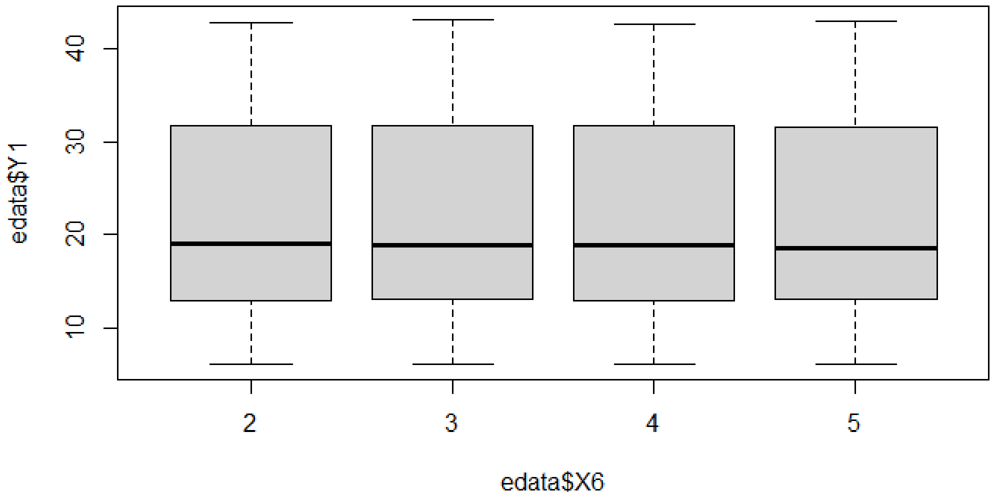
<u>X5 - Overall Height:</u>

X5 is a categorical variable as either 3.5 or 7. This variable was converted from a numeric variable to a factor variable.



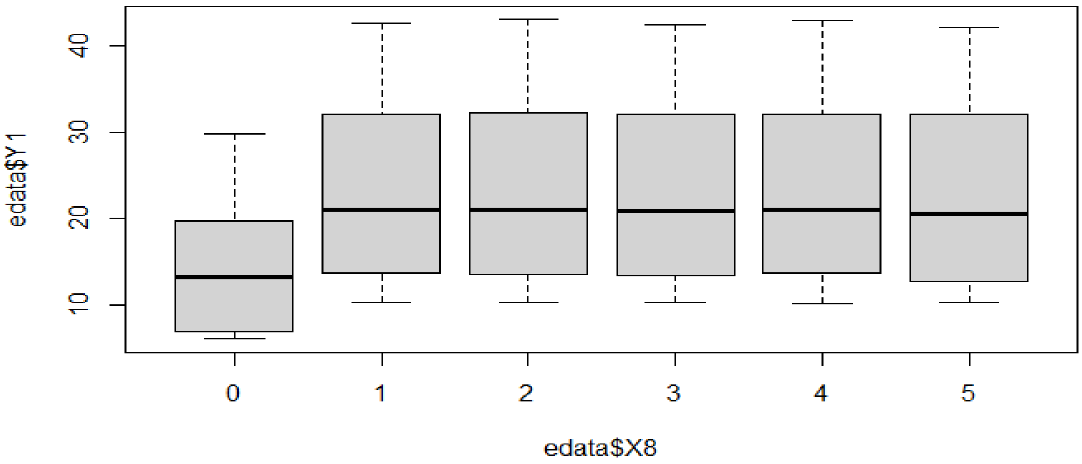
X6 - Orientation:

Data is a categorical variable that is uniformly distributed over 4 states as either 2,3,4,5.



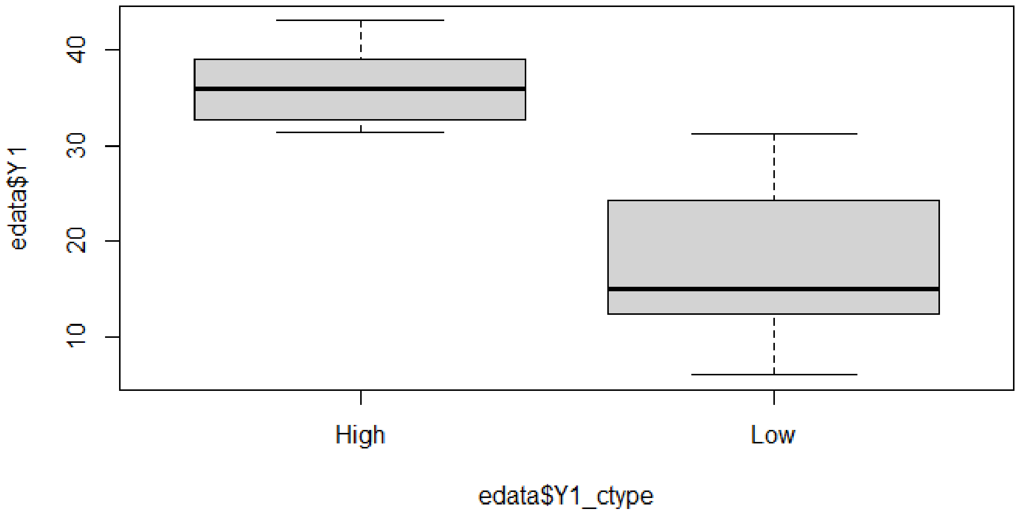
X8 - Glazing Area Distribution

Data is a categorical variable that is uniformly distributed over 6 states from 0-5. Data was converted from numeric to factor type.



Y1_ctype:

An additional field representing low and high values for the Y1_Type bins was created with 1 and 3 being "Low" and 2 being "High".

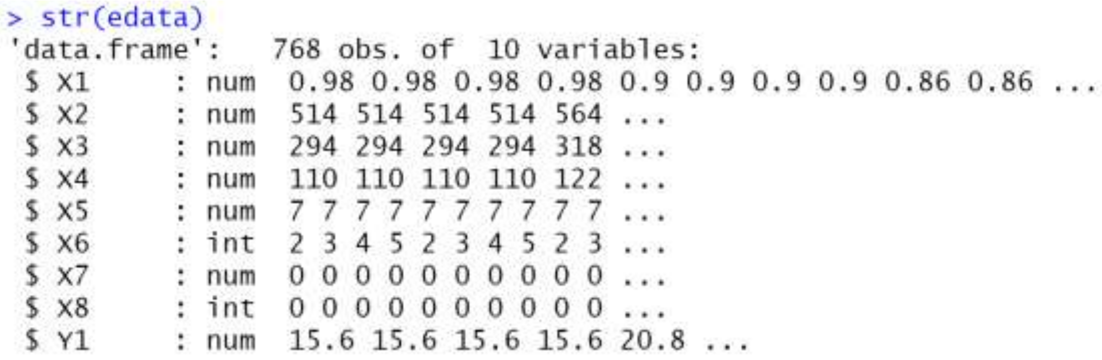


4. *Removing 'X7 - Glazing Area'*

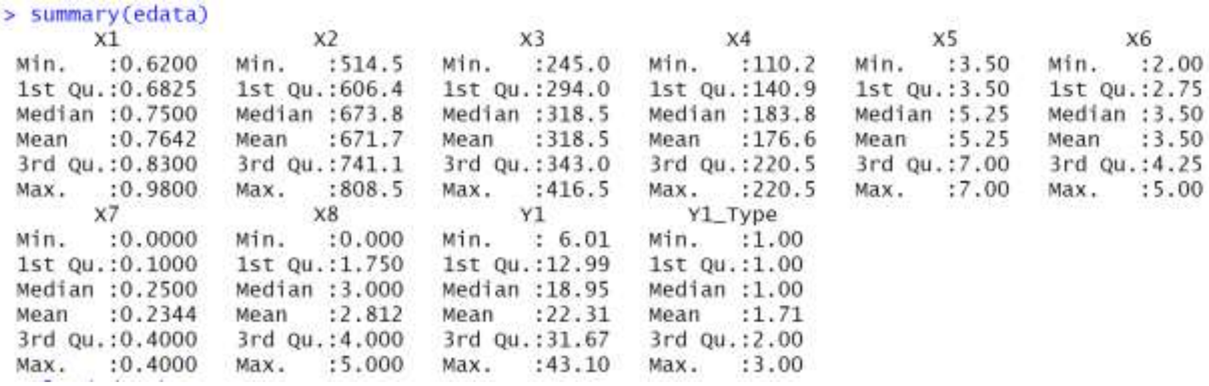
Glazing Area refers to the installation of glass in windows, doors and any other fixed opening. X7 and X8 [X8 - glazing area distribution] are representing the same data however the X8 variable is describing the X7 variable as a factor in either 0, 1, 2, 3, 4, 5 states - i.e. 6 states. Since the X8 variable would have more value in model building, X7 was removed.

5. *Data Transformation*

Before Transformation:

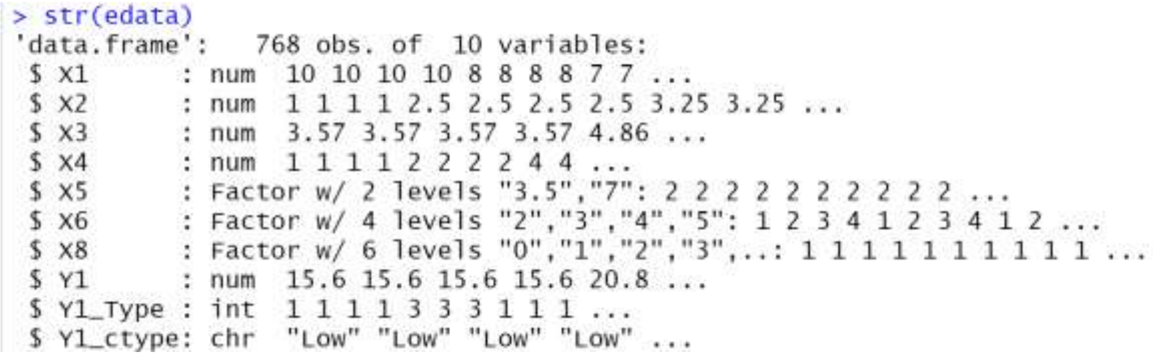


```
> str(edata)
'data.frame':	768 obs. of  10 variables:
 $ X1      : num  0.98 0.98 0.98 0.98 0.9 0.9 0.9 0.9 0.86 0.86 ...
 $ X2      : num  514 514 514 514 564 ...
 $ X3      : num  294 294 294 294 318 ...
 $ X4      : num  110 110 110 110 122 ...
 $ X5      : num  7 7 7 7 7 7 7 7 7 7 ...
 $ X6      : int  2 3 4 5 2 3 4 5 2 3 ...
 $ X7      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ X8      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Y1      : num  15.6 15.6 15.6 15.6 20.8 ...
```

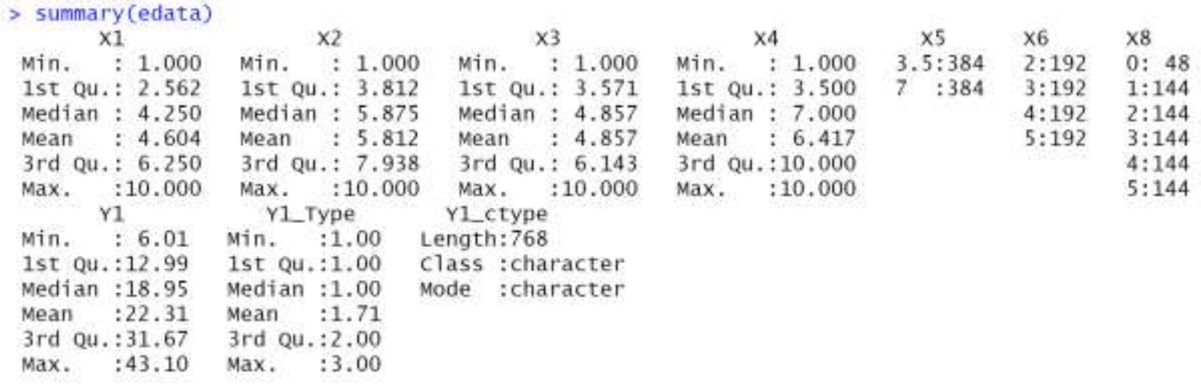


```
> summary(edata)
       X1               X2              X3              X4              X5             X6      
 Min.   :0.6200   Min.   :514.5   Min.   :245.0   Min.   :110.2   Min.   :3.50   Min.   :2.00  
 1st Qu.:0.6825   1st Qu.:606.4   1st Qu.:294.0   1st Qu.:140.9   1st Qu.:3.50   1st Qu.:2.75  
 Median :0.7500   Median :673.8   Median :318.5   Median :183.8   Median :5.25   Median :3.50  
 Mean   :0.7642   Mean   :671.7   Mean   :318.5   Mean   :176.6   Mean   :5.25   Mean   :3.50  
 3rd Qu.:0.8300   3rd Qu.:741.1   3rd Qu.:343.0   3rd Qu.:220.5   3rd Qu.:7.00   3rd Qu.:4.25  
 Max.   :0.9800   Max.   :808.5   Max.   :416.5   Max.   :220.5   Max.   :7.00   Max.   :5.00  
       X7               X8              Y1            Y1_Type    
 Min.   :0.0000   Min.   :0.000   Min.   : 6.01   Min.   :1.00  
 1st Qu.:0.1000   1st Qu.:1.750   1st Qu.:12.99   1st Qu.:1.00  
 Median :0.2500   Median :3.000   Median :18.95   Median :1.00  
 Mean   :0.2344   Mean   :2.812   Mean   :22.31   Mean   :1.71  
 3rd Qu.:0.4000   3rd Qu.:4.000   3rd Qu.:31.67   3rd Qu.:2.00  
 Max.   :0.4000   Max.   :5.000   Max.   :43.10   Max.   :3.00  
```

After Cleaning and Transformation:



```
> str(edata)
'data.frame':	768 obs. of  10 variables:
 $ X1      : num  10 10 10 10 8 8 8 8 7 7 ...
 $ X2      : num  1 1 1 1 2.5 2.5 2.5 2.5 3.25 3.25 ...
 $ X3      : num  3.57 3.57 3.57 3.57 4.86 ...
 $ X4      : num  1 1 1 1 2 2 2 2 4 4 ...
 $ X5      : Factor w/ 2 levels "3.5","7": 2 2 2 2 2 2 2 2 2 2 ...
 $ X6      : Factor w/ 4 levels "2","3","4","5": 1 2 3 4 1 2 3 4 1 2 ...
 $ X8      : Factor w/ 6 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Y1      : num  15.6 15.6 15.6 15.6 20.8 ...
 $ Y1_Type : int  1 1 1 1 3 3 3 1 1 1 ...
 $ Y1_ctype: chr  "Low" "Low" "Low" "Low" ...
```



```
> summary(edata)
       X1               X2               X3               X4            X5          X6       X8    
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   3.5:384   2:192   0: 48  
 1st Qu.: 2.562   1st Qu.: 3.812   1st Qu.: 3.571   1st Qu.: 3.500   7  :384   3:192   1:144  
 Median : 4.250   Median : 5.875   Median : 4.857   Median : 7.000             4:192   2:144  
 Mean   : 4.604   Mean   : 5.812   Mean   : 4.857   Mean   : 6.417             5:192   3:144  
 3rd Qu.: 6.250   3rd Qu.: 7.938   3rd Qu.: 6.143   3rd Qu.:10.000                     4:144  
 Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000                     5:144  
       Y1            Y1_Type        Y1_ctype        
 Min.   : 6.01   Min.   :1.00   Length:768         
 1st Qu.:12.99   1st Qu.:1.00   Class :character  
 Median :18.95   Median :1.00   Mode  :character  
 Mean   :22.31   Mean   :1.71                     
 3rd Qu.:31.67   3rd Qu.:2.00                     
 Max.   :43.10   Max.   :3.00                     
```

Modeling

Various modeling techniques are carried out in order to produce a reusable data model. A model is defined as a framework highlighting the relationships among data fields and are used for making predictions.

The three modeling techniques used for this study are:

1. **Logistic Regression**

   Logistic regression is a type of regression analysis in statistics used for prediction of outcome of a categorical dependent variable from a set of predictor or independent variables. In logistic regression the dependent variable is always binary. Logistic regression is mainly used for prediction and also calculating the probability of success. Logistic Regression was utilized in this project to predict the probability of High or Low heating or a room based on its Relative Compactness and Surface Area"

   Justification for use:

   This model was selected because the dependent variable is categorical. Since we can create linear combinations of the building construction factors (X1 – X8), to learn if the make-up of the building had a High Heating Load or not.

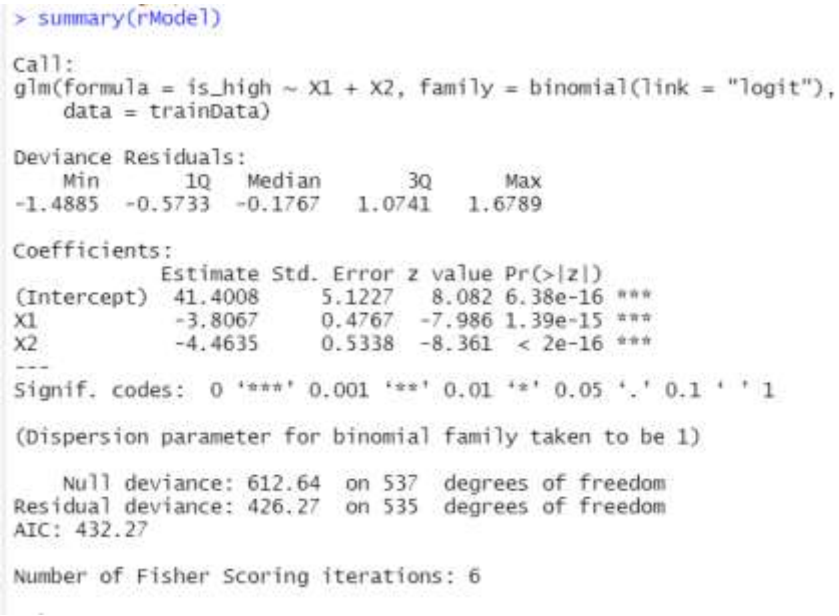
<u>Model Building</u>

2 Logistic Regression Models were built.

<u>First Model – LR 1</u> was built using X1- Relative Compactness and X2 – Surface Area.

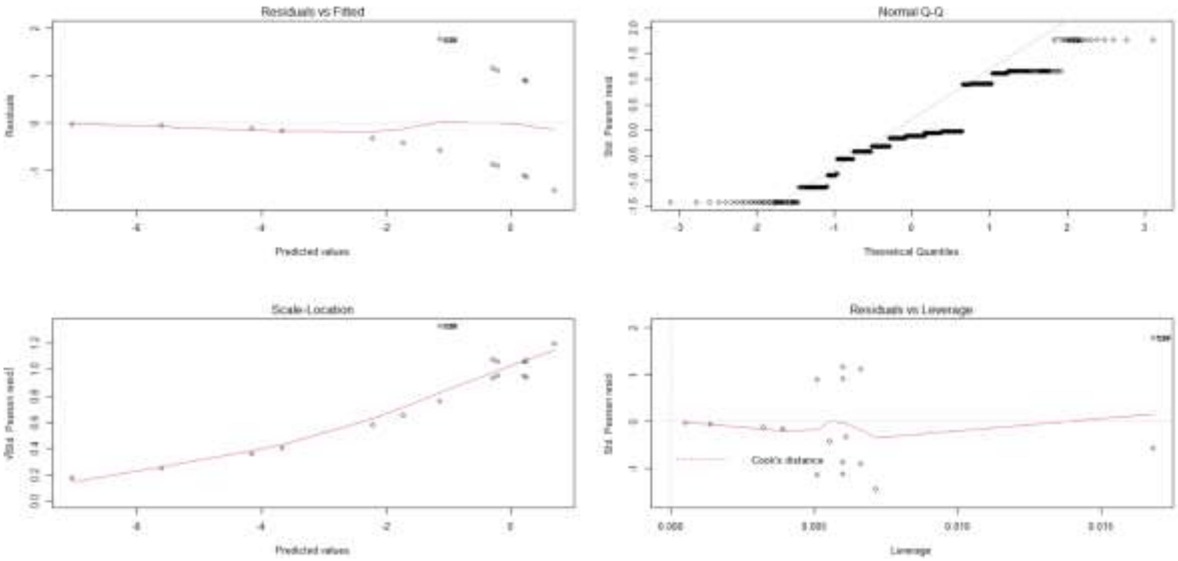
Data was split into a Training and Testing Set with a split ratio of 70%.

Model Summary:

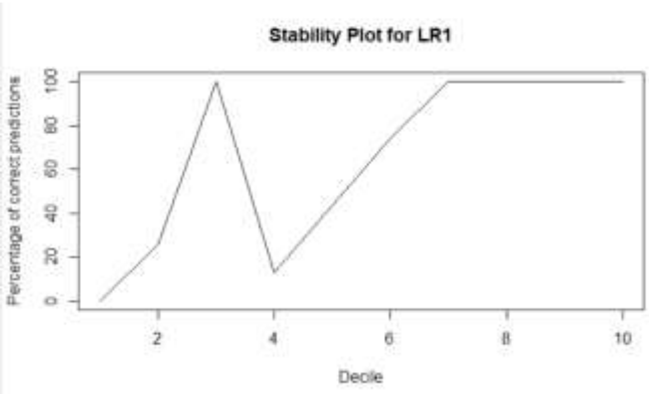


```
> summary(rModel)

Call:
glm(formula = is_high ~ X1 + X2, family = binomial(link = "logit"),
    data = trainData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4885  -0.5733  -0.1767   1.0741   1.6789

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  41.4008     5.1227   8.082 6.38e-16 ***
X1           -3.8067     0.4767  -7.986 1.39e-15 ***
X2           -4.4635     0.5338  -8.361  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 612.64  on 537  degrees of freedom
Residual deviance: 426.27  on 535  degrees of freedom
AIC: 432.27

Number of Fisher Scoring iterations: 6
```

Residual Plots:



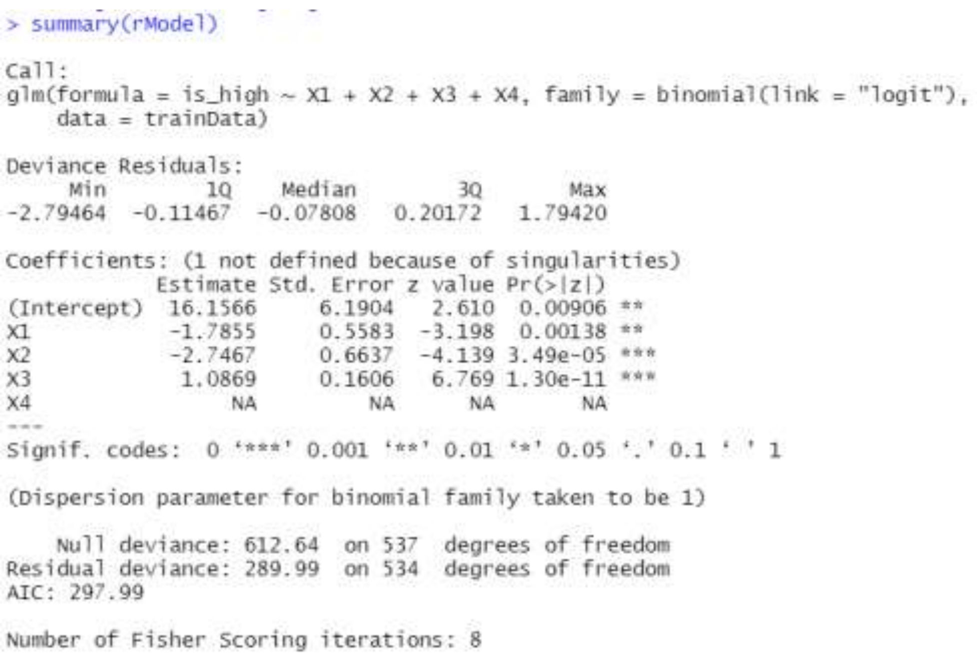
Stability Plot:



Based on the first 3 deciles we can say that the model is not stable.

__Second Model – LR 2__ was built using X1- Relative Compactness, X2 – Surface Area, X3- Wall Area and X4 – Roof Area. Data was split into a Training and Testing Set with a split ratio of 70%.

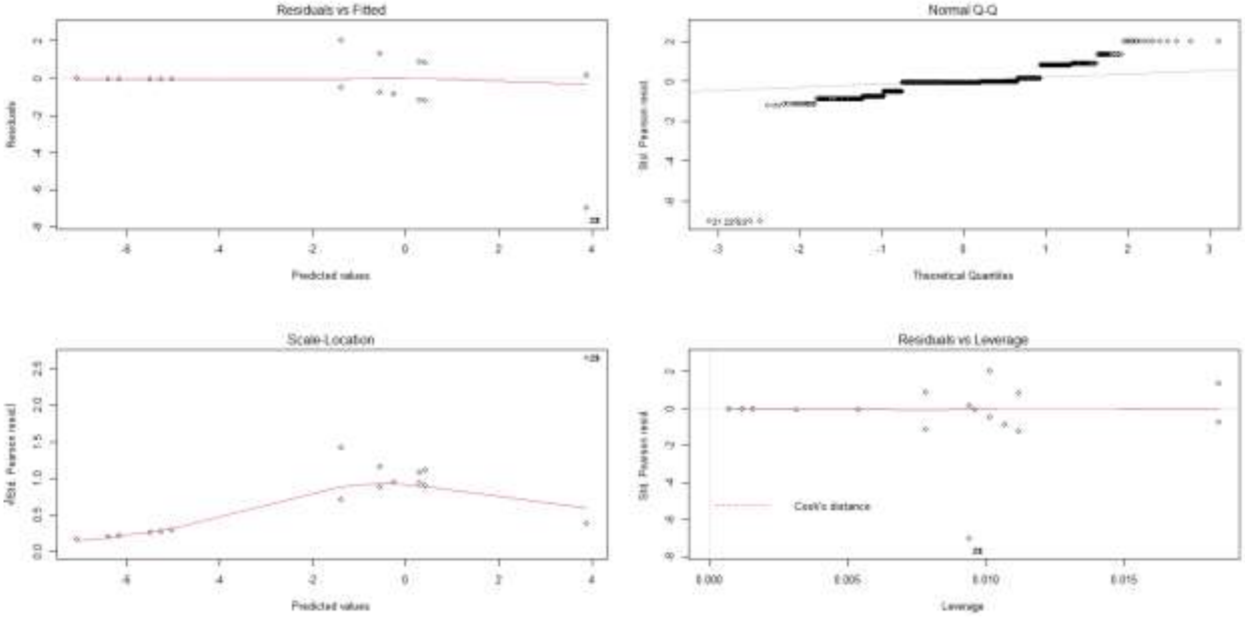
Model Summary



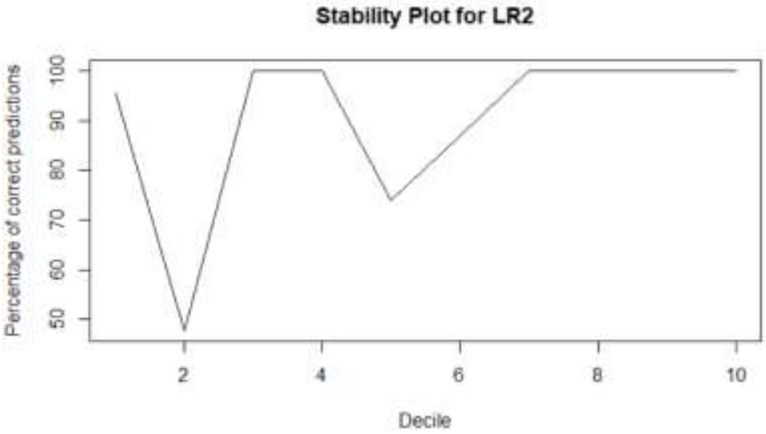
```
> summary(rModel)

Call:
glm(formula = is_high ~ X1 + X2 + X3 + X4, family = binomial(link = "logit"),
    data = trainData)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.79464  -0.11467  -0.07808   0.20172   1.79420

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  16.1566     6.1904   2.610  0.00906 **
X1           -1.7855     0.5583  -3.198  0.00138 **
X2           -2.7467     0.6637  -4.139 3.49e-05 ***
X3            1.0869     0.1606   6.769 1.30e-11 ***
X4                NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 612.64  on 537  degrees of freedom
Residual deviance: 289.99  on 534  degrees of freedom
AIC: 297.99

Number of Fisher Scoring iterations: 8
```

X4 – Roof Area did not produce an estimated beta value and so did not contribute to the model building.

Residual Plot



Stability Plot



Based on the first 2 deciles we can say that the model is stable.

2. **Decision Tree**

Predictive (Explanatory) Modeling - Classification Value prediction Problem (High or Low) - Hold out method Stratified Sampling - Rules. The most useful way of visualizing what's happening in the decision process. Load functions caret and e1071, for classification as well as rattle which makes plots for decision trees.

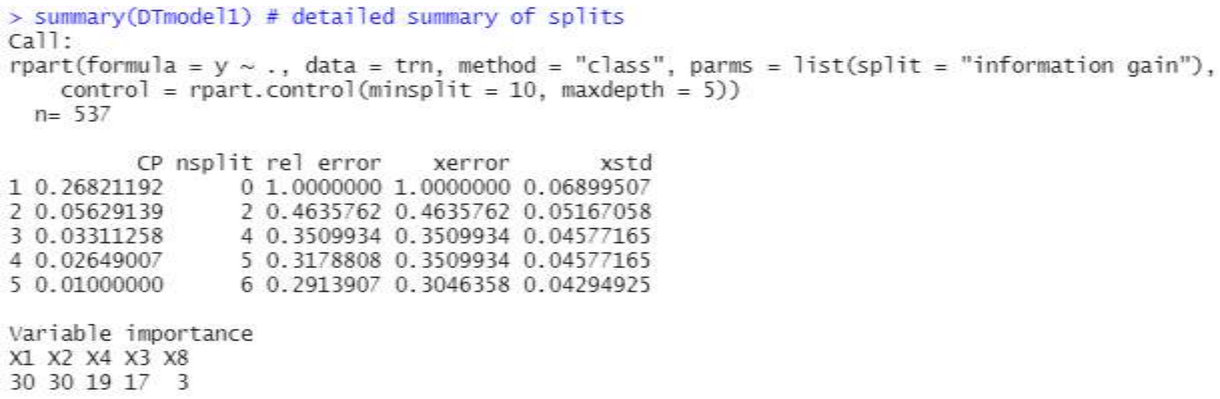
Justification for use:

The dependent variable is categorical and the resulting model provides a semantic representation of the data effectively breaking down the decision making into pathways that will help us understand what combination of factors create High Heating Loads.

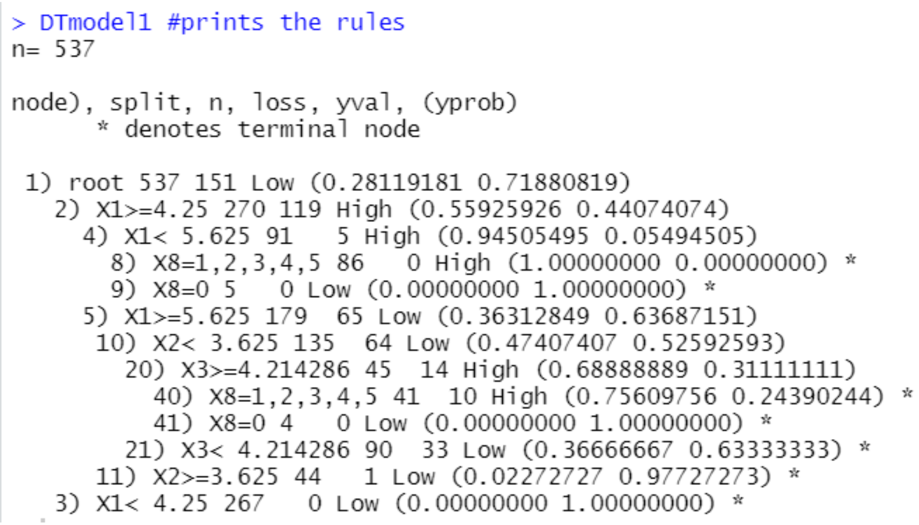
Model Building

A decision tree model was created by splitting data in to a test and training set.

Model Summary:



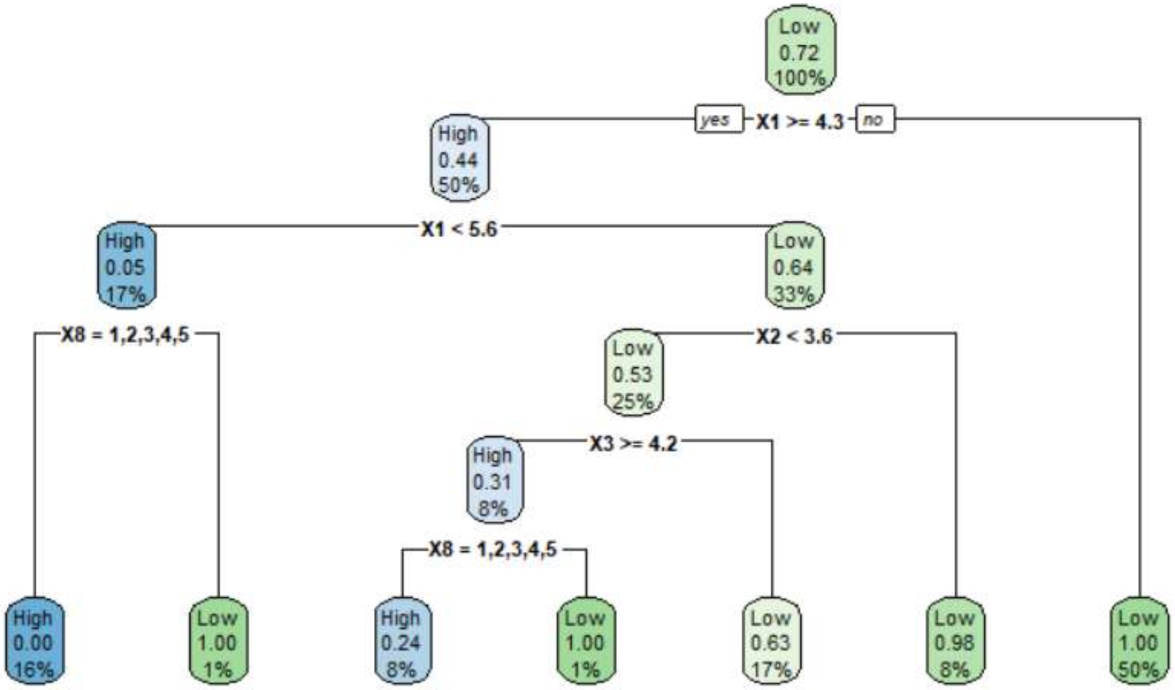
```
> summary(DTmodel1) # detailed summary of splits
Call:
rpart(formula = y ~ ., data = trn, method = "class", parms = list(split = "information gain"),
    control = rpart.control(minsplit = 10, maxdepth = 5))
  n= 537

          CP nsplit rel error    xerror       xstd
1 0.26821192      0 1.0000000 1.0000000 0.06899507
2 0.05629139      2 0.4635762 0.4635762 0.05167058
3 0.03311258      4 0.3509934 0.3509934 0.04577165
4 0.02649007      5 0.3178808 0.3509934 0.04577165
5 0.01000000      6 0.2913907 0.3046358 0.04294925

Variable importance
X1 X2 X4 X3 X8
30 30 19 17  3
```



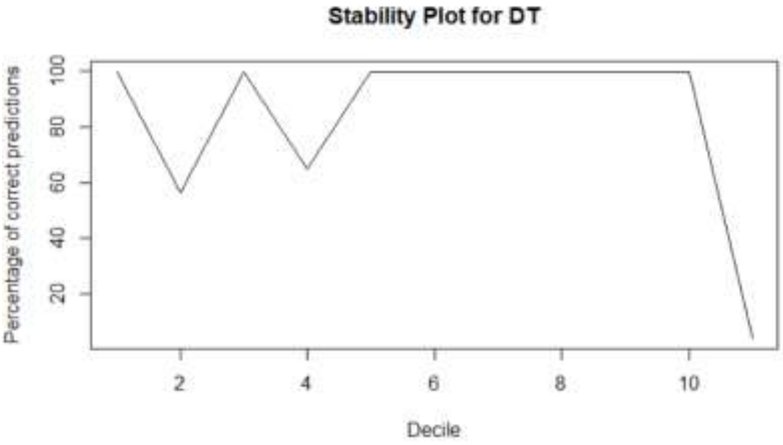
```
> DTmodel1 #prints the rules
n= 537

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 537 151 Low (0.28119181 0.71880819)
   2) X1>=4.25 270 119 High (0.55925926 0.44074074)
     4) X1< 5.625 91   5 High (0.94505495 0.05494505)
       8) X8=1,2,3,4,5 86   0 High (1.00000000 0.00000000) *
       9) X8=0 5   0 Low (0.00000000 1.00000000) *
     5) X1>=5.625 179  65 Low (0.36312849 0.63687151)
      10) X2< 3.625 135  64 Low (0.47407407 0.52592593)
        20) X3>=4.214286 45  14 High (0.68888889 0.31111111)
          40) X8=1,2,3,4,5 41  10 High (0.75609756 0.24390244) *
          41) X8=0 4   0 Low (0.00000000 1.00000000) *
        21) X3< 4.214286 90  33 Low (0.36666667 0.63333333) *
      11) X2>=3.625 44   1 Low (0.02272727 0.97727273) *
   3) X1< 4.25 267   0 Low (0.00000000 1.00000000) *
```

Model Plot:

See Appendix for Model Plot. A small version can be seen below:



Stability Plot



Based on the first 2 deciles we can say that the model is stable.

3. **Neural Network**

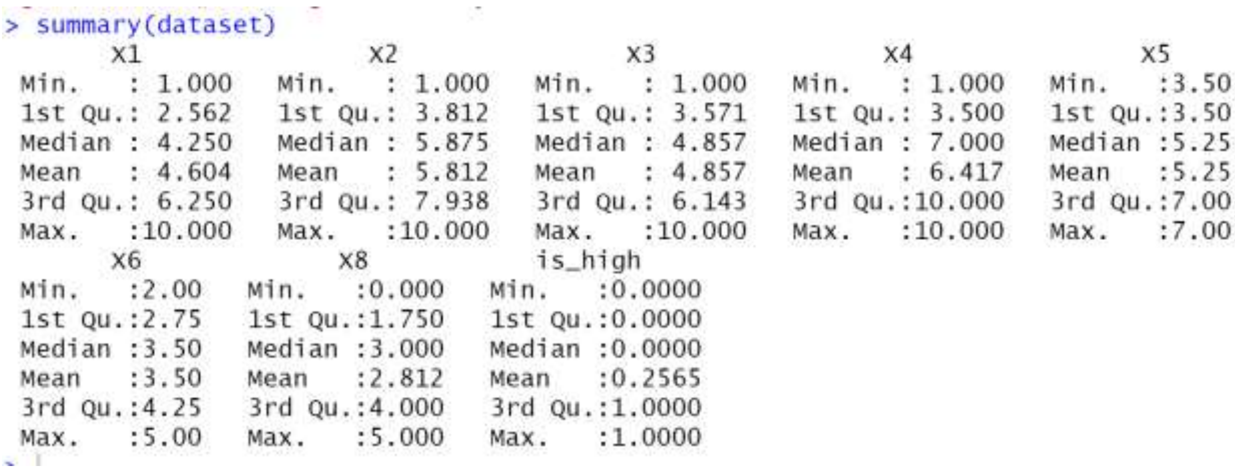
A neural network is a series of algorithms that seek to identify relationships in a dataset based on a process that mimics how the human brain works.

Justification for use:

Based on the construction of a Neural Network, it can help us to identify the hidden pattern that exists between the building construction factors and if they contribute to high heating loads.

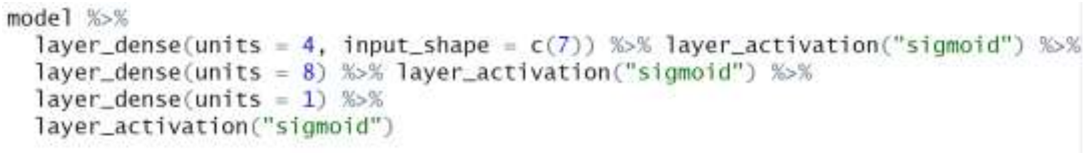
Model Building

A neural network was fitted using the independent variables X1, X2, X3, X4, X5, X6, X8 to classify the heating load as either High or Low. To create the dependent variable to supervise the model fitting, we use the Y1_ctype variable [which contains Low or High categorical variable], and create a new variable called is_high that is a boolean variable with '1' is high and 'o' is low.:



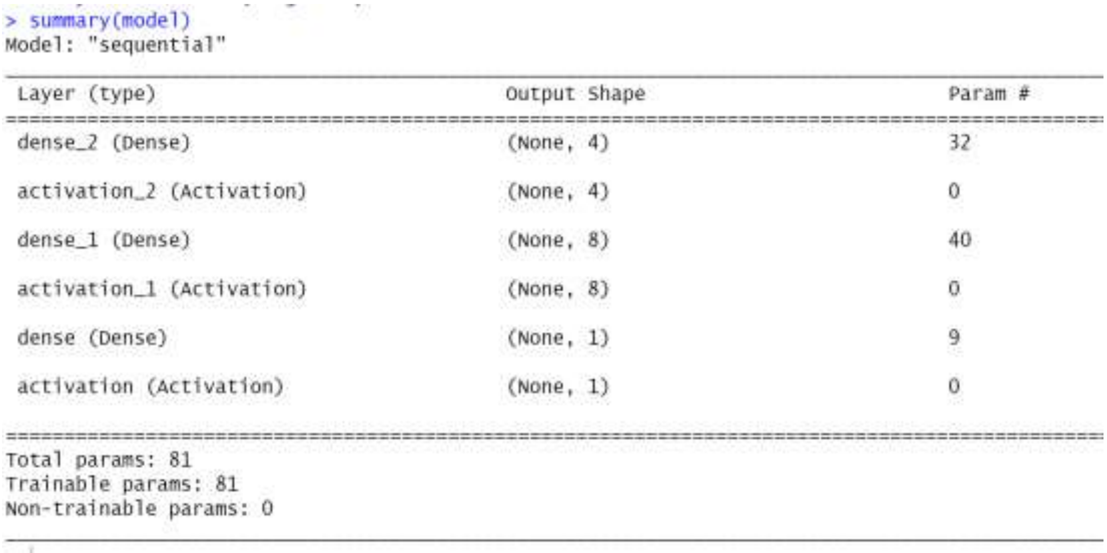
```
> summary(dataset)
       X1               X2               X3               X4               X5
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   :3.50
 1st Qu.: 2.562   1st Qu.: 3.812   1st Qu.: 3.571   1st Qu.: 3.500   1st Qu.:3.50
 Median : 4.250   Median : 5.875   Median : 4.857   Median : 7.000   Median :5.25
 Mean   : 4.604   Mean   : 5.812   Mean   : 4.857   Mean   : 6.417   Mean   :5.25
 3rd Qu.: 6.250   3rd Qu.: 7.938   3rd Qu.: 6.143   3rd Qu.:10.000   3rd Qu.:7.00
 Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :7.00
       X6              X8            is_high
 Min.   :2.00   Min.   :0.000   Min.   :0.0000
 1st Qu.:2.75   1st Qu.:1.750   1st Qu.:0.0000
 Median :3.50   Median :3.000   Median :0.0000
 Mean   :3.50   Mean   :2.812   Mean   :0.2565
 3rd Qu.:4.25   3rd Qu.:4.000   3rd Qu.:1.0000
 Max.   :5.00   Max.   :5.000   Max.   :1.0000
```

The data was split into a Test and Train dataset using 70% of the data as the training set and 30% as the testing set. Both training and testing set were separated into dependent variables as ‘Ys’ and the independent variables as 'X1’ to ‘X8’. Since Keras algorithm cannot consume dataframe, the datasets were converted into vectors and fitted a NN Model with 120 epochs with batch size of 10 using the following layers:



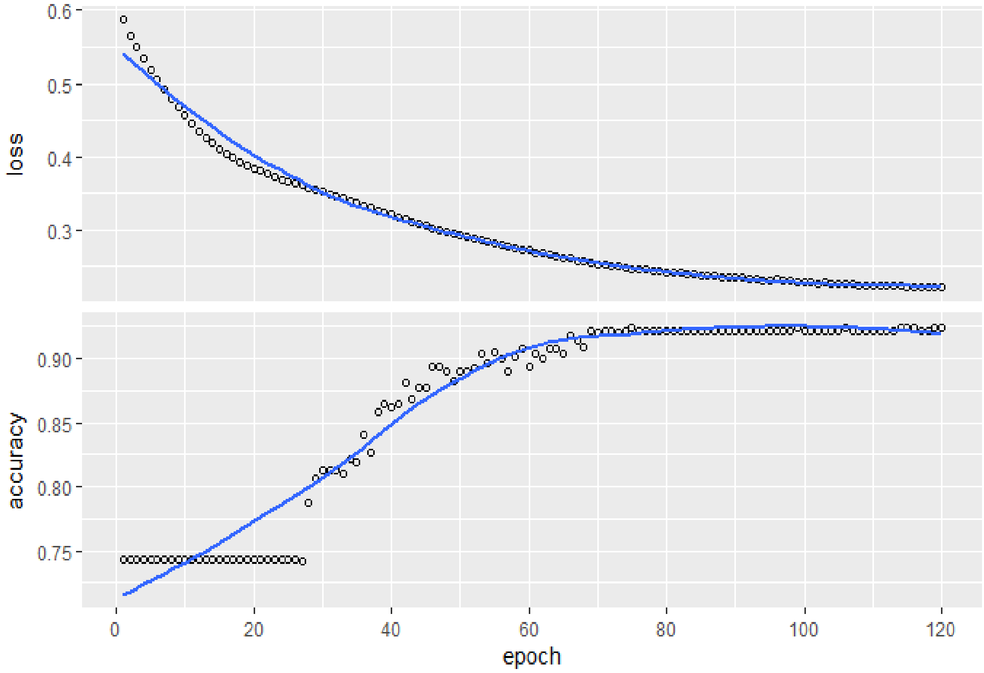
```
model %>%
  layer_dense(units = 4, input_shape = c(7)) %>% layer_activation("sigmoid") %>%
  layer_dense(units = 8) %>% layer_activation("sigmoid") %>%
  layer_dense(units = 1) %>%
  layer_activation("sigmoid")
```

Model Summary:



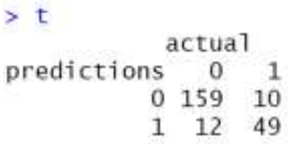
```
> summary(model)
Model: "sequential"
________________________________________________________________________________________
 Layer (type)                        Output Shape                      Param #
========================================================================================
 dense_2 (Dense)                     (None, 4)                         32

 activation_2 (Activation)           (None, 4)                         0

 dense_1 (Dense)                     (None, 8)                         40

 activation_1 (Activation)           (None, 8)                         0

 dense (Dense)                       (None, 1)                         9

 activation (Activation)             (None, 1)                         0

========================================================================================
Total params: 81
Trainable params: 81
Non-trainable params: 0
________________________________________________________________________________________
```

The NN Model produced the following Loss/Accuracy plot given each epoch:



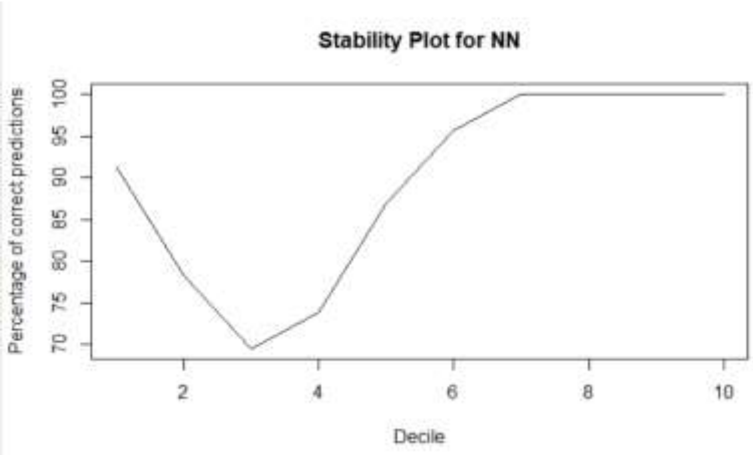
The NN Model was evaluated using the Test Data and produced the following

Confusion matrix:



```
> t
           actual
predictions   0   1
          0 159  10
          1  12  49
```

Stability Plot



Based on the first 3 deciles we can say that the model is stable.

Evaluation

Performance Measures

Measure	**Description**	**Definition of Value Function**	**Weight**	**Threshold**
Accuracy	Proportion of correctly classified	-	50%	> 0.6
Simplicity	Logistic Regression: # of significant variables DT : # of Leaves NN: # of Layers	$f(x) = 1- (x/7), 0<x\leq7$ [7 is total number of independent variables in the dataset.] $f(x) = 1 - (x/100), 0<x\leq100$ [We choose 100 leaves as the maximum number of leaves that is able to run during deployment] $f(x) = 1 - (x/20), 0<x\leq20$ [We choose 20 layers as the maximum number of layers that is able to run during deployment]	10%	
AUC	Area Under Curve	-	25%	>0.6
Stability	Visual inspection of % of correct predictions chart	0 - Unstable 1 - Stable	15%	-

Summary Results

No	**Model**	**Accuracy**	**Simplicity Measure**	**Simplicity Score**	**AUC Score**	**Stability**	**Overall Score**
1	LR 1	0.65	2 Variables	0.71	0.72	0	0.576
2	LR 2	0.9	3 Variables	0.57	0.94	1	0.892
3	DT	0.92	10 Leaves	0.90	0.97	1	0.943
4	NN	0.9	3 Layers	0.85	0.96	1	0.925

Model evaluation consists of examining Accuracy, Simplicity, Area under the Curve (AUC) and Stability. Each component was given a weight and threshold that will help to assess the overall score of each model. For Simplicity, a value function was given to help produce a score that is between 0 and 1. Each value function was given as a function of the primary model variables – leaves, layers, independent variables.

The overall score was calculated but summing the weight scores based on the outline given above. Based on the Overall Evaluation Score, we can clearly see that a Decision Tree model is the best model to use in identifying high heat loads.

Deployment

The deployment phase involves making use of the selected data model in such a way where its results become easily accessible to customers.

Once the model has been selected, it could reside on a server which exposes a public API that would be called using HTTP requests. The body of each request to the server would contain values X1 to X8; with at least X1, X2, X3, X4 and X8, for the model to produce a result. When the server receives a new request it will extract the values from the request body and run them through the appropriate model.

Example:

<u>Decision Tree Model1 – High</u>

➔ [X1=8,X2=2.5,X3=4.857143,X4=2,X5=7,X6=4,X8=4] is transferred via API

➔ DTHLnewData <- data.frame(X1=8,X2=2.5,X3=4.857143,X4=2,X5=7,X6=4,X8=4)

➔ DTHLpred<- predict(DTmodel1, DTHLnewData , type="class")

➔ Return DTHLpred [High]via API

<u>Decision Tree Model2 - Low</u>

➔ [X1=2.75,X2=7.75,X3=3.571429,X4=10,X5=3.5,X6=2,X8=2] is transferred via API

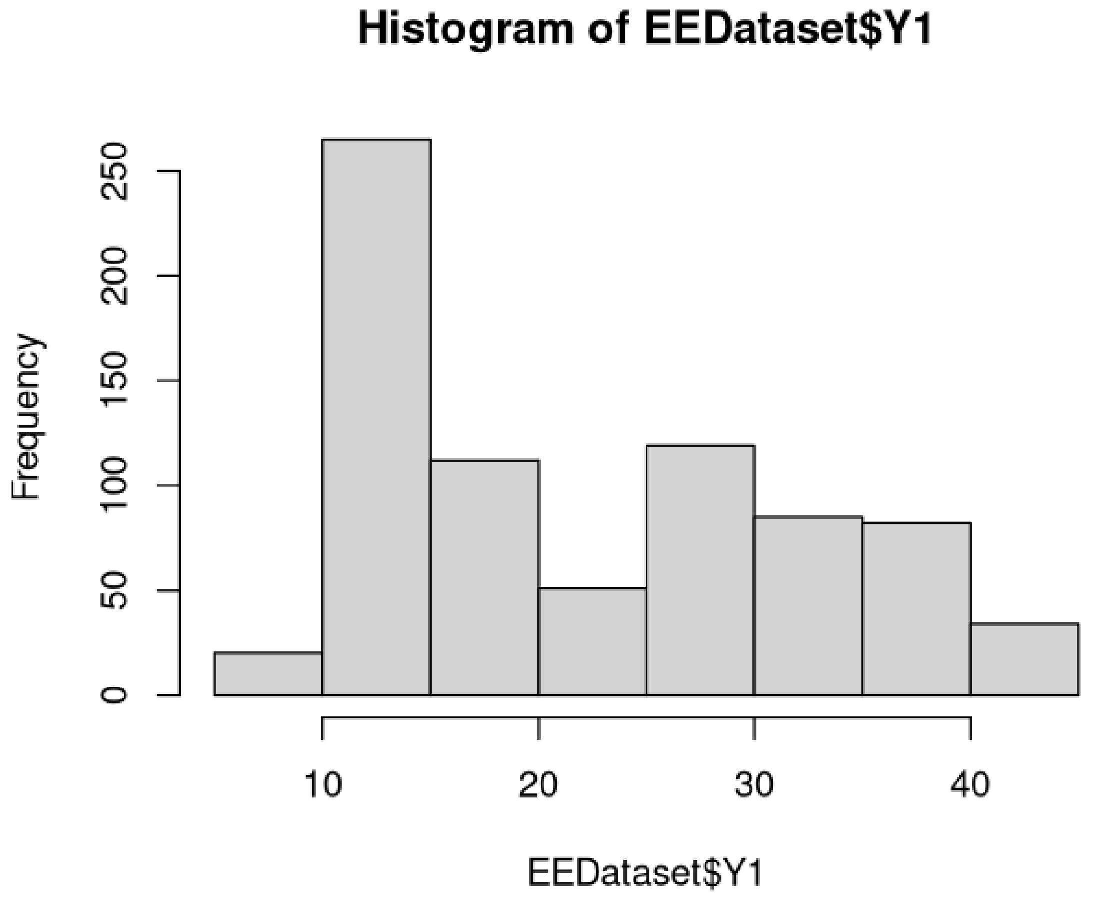
➔ DTHLnewData <- data.frame(X1=2.75,X2=7.75,X3=3.571429,X4=10,X5=3.5,X6=2,X8=2)

➔ DTHLpred<- predict(DTmodel1, DTHLnewData , type="class")

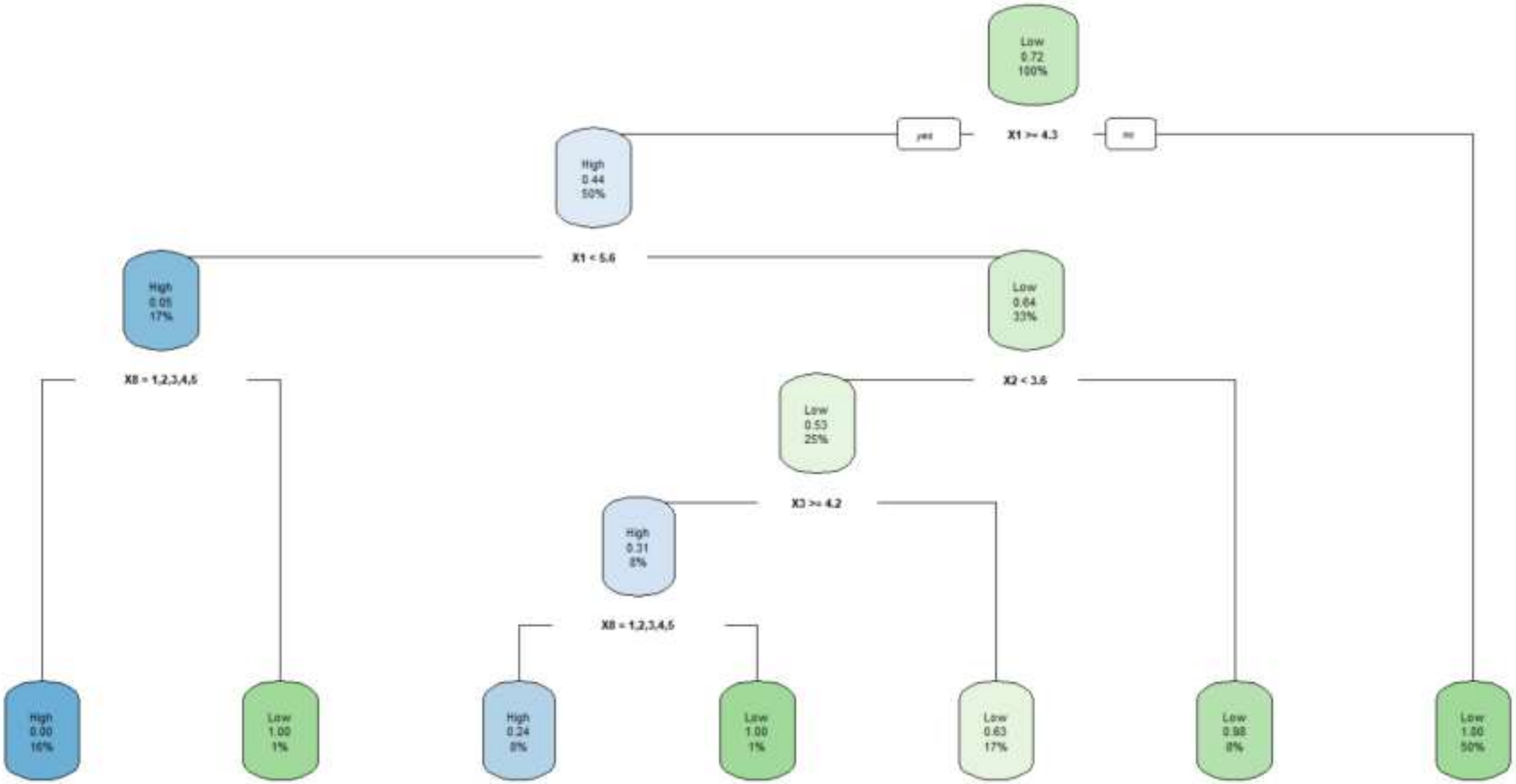
➔ Return DTHLpred [Low] via API

Appendix

Histogram showing the frequency of the target variable



Fitting the model - Decision Tree Heating Load


Low
0.72
100%
yes
X1 >= 4.3
no
High
0.44
50%
X1 < 5.6
High
0.05
17%
Low
0.64
33%
X8 = 1,2,3,4,5
X2 < 3.6
Low
0.53
25%
X3 >= 4.2
High
0.31
8%
X8 = 1,2,3,4,5
High
0.00
16%
Low
1.00
1%
High
0.24
8%
Low
1.00
1%
Low
0.63
17%
Low
0.98
8%
Low
1.00
50%