Capstone CYO - Energy Efficiency

Lawrence Ejike

18/07/2021

Introduction

1.1. Background

Global warming has been an issue of concern over the last couple decades and its impact has been felt largely in 2021 with several countries, especially Canada experiencing record high temperatures. Energy production from fossil fuels has been a major contributor of the greenhouse gases that lead to global warming. Buildings for residential, commercial, and industrial uses are important consumers of energy and improving the energy efficiency of such structures would contribute significantly to a reduction in energy demand and consequently, greenhouse gas production and global warming.

Efficient energy building designs involve construction of buildings with the goal of reducing energy loss. The shape of a building is an important index in building energy efficient designs and each shape is determined by various parameters including compactness, glazing area , wall, roof area, to name a few. The energy requirement, both heating and cooling load of a building can those be approximated before construction by factoring these various parameter.

1.2. Executive summary

This report details the development of a model for predicting the energy efficiency of residential buildings.

The project utilized a data set obtained from Kaggle, which was simulated in Ecotect by Angeliki Xifara (a civil/structural engineer) and Athanasois Tsanas from the University of Oxford, UK. The data set comprised 768 samples (building shapes) and 8 features (variables), with two responses.

The primary goal of this project is to create a model that predicts the energy efficiency (heating and cooling loads) of residential buildings, using variables that determines the shape of the building. A secondary goal of the project was to minimize root mean squared error, RMSE (a measure of the deviation of predicted values from observed values) for the predicted heating load and cooling load.

The development of the prediction model was achieved using machine learning as well as other data science tools. The energy efficiency data set was first split into two sets, evaluation and probe set, and the latter (probe set) was split into a train and test set. Predictive models using the train set were made that incorporated the effect of various features (relative compactness, wall area, orientation,glazing area, etc.) and these models were tested on the test set. The performance of each model was evaluated against each other. The model with the relative best rmse obtained on the test set was applied to the probe set and used to predict the heating load and cooling load in the evaluation set, yielding a final RMSE value.

2. Method/Analysis

The following packages, openxlsx, tidyverse,and caret were installed and used to download, clean, and create the data set.

i) openxlsx package was used to read, write and edit xlsx files
ii) tidyverse was used for data wrangling, web-scrapping, joining, and reshaping data tables
iii) caret package was used for machine learning and building prediction algorithms
iv) randomForest package was used for Random Forest regression

The data was downloaded from UCI Machine Learning Repository website as a xlsx file that is ready for machine learning analysis and saved in rstudio as a table, energydat.

2.1. Data Exploration and Visualization

The data in energydat data set was explored with tables and graphs where applicable, to gain useful insights that formed the basis of subsequent modeling approaches. An examination of the dimensions of the data sets, energydat, shows that it contains 768 rows and 10 columns.

```
# display number of rows and columns
dim(energydat)
```

```
[1] 768  10
```

```
#check structure of the data
str(energydat)
```

```
'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: num  2 3 4 5 2 3 4 5 2 3 ...
 $ X7: num  0 0 0 0 0 0 0 0 0 0 ...
 $ X8: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Y1: num  15.6 15.6 15.6 15.6 20.8 ...
 $ Y2: num  21.3 21.3 21.3 21.3 28.3 ...
```

```
#display first 6 rows of the data
head(energydat)
```

```
    X1    X2    X3     X4 X5 X6 X7 X8    Y1    Y2
1 0.98 514.5 294.0 110.25  7  2  0  0 15.55 21.33
2 0.98 514.5 294.0 110.25  7  3  0  0 15.55 21.33
3 0.98 514.5 294.0 110.25  7  4  0  0 15.55 21.33
4 0.98 514.5 294.0 110.25  7  5  0  0 15.55 21.33
5 0.90 563.5 318.5 122.50  7  2  0  0 20.84 28.28
6 0.90 563.5 318.5 122.50  7  3  0  0 21.46 25.38
```

The data had 768 rows and 10 columns (8 features, X1-X8 and 2 outcomes,Y1-Y2).The content of the columns were numbers and the features they represented were given by:

• X1 - Relative Compactness • X2 - Surface Area • X3 - Wall Area • X4 - Roof Area • X5 - Overall Height • X6 - Orientation • X7 - Glazing Area • X8 - Glazing Area Distribution

The outcomes and their representation are given by • Y1 - Heating Load • Y2 - Cooling Load

Although the columns were original labeled as X1 to X8 for variables and Y1 to Y2 for outcomes, the names were updated to reflect the features they represent. The first 6 rows of the updated energydat data set are shown below:

```
  relative_compactness surface_area wall_area roof_area overall_height
1                 0.98        514.5     294.0    110.25              7
2                 0.98        514.5     294.0    110.25              7
3                 0.98        514.5     294.0    110.25              7
4                 0.98        514.5     294.0    110.25              7
5                 0.90        563.5     318.5    122.50              7
6                 0.90        563.5     318.5    122.50              7
  orientation glazing_area glazing_area_distribution heating_load cooling_load
1           2            0                         0        15.55        21.33
2           3            0                         0        15.55        21.33
3           4            0                         0        15.55        21.33
4           5            0                         0        15.55        21.33
5           2            0                         0        20.84        28.28
6           3            0                         0        21.46        25.38
```

A check of the data set or each column of the data set for missing values (na) returned as false or zero.

```
#checking for missing values
any(is.na(energydat))
```

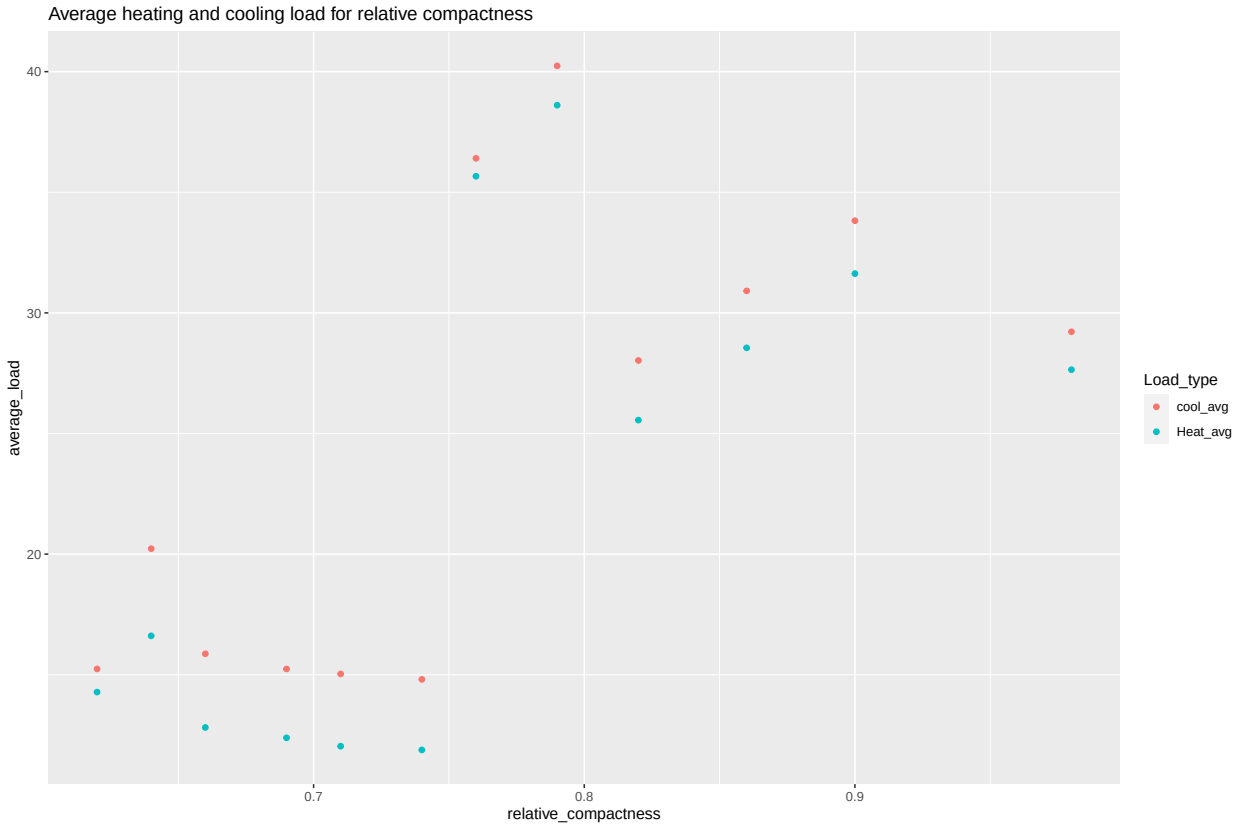
```
[1] FALSE
```

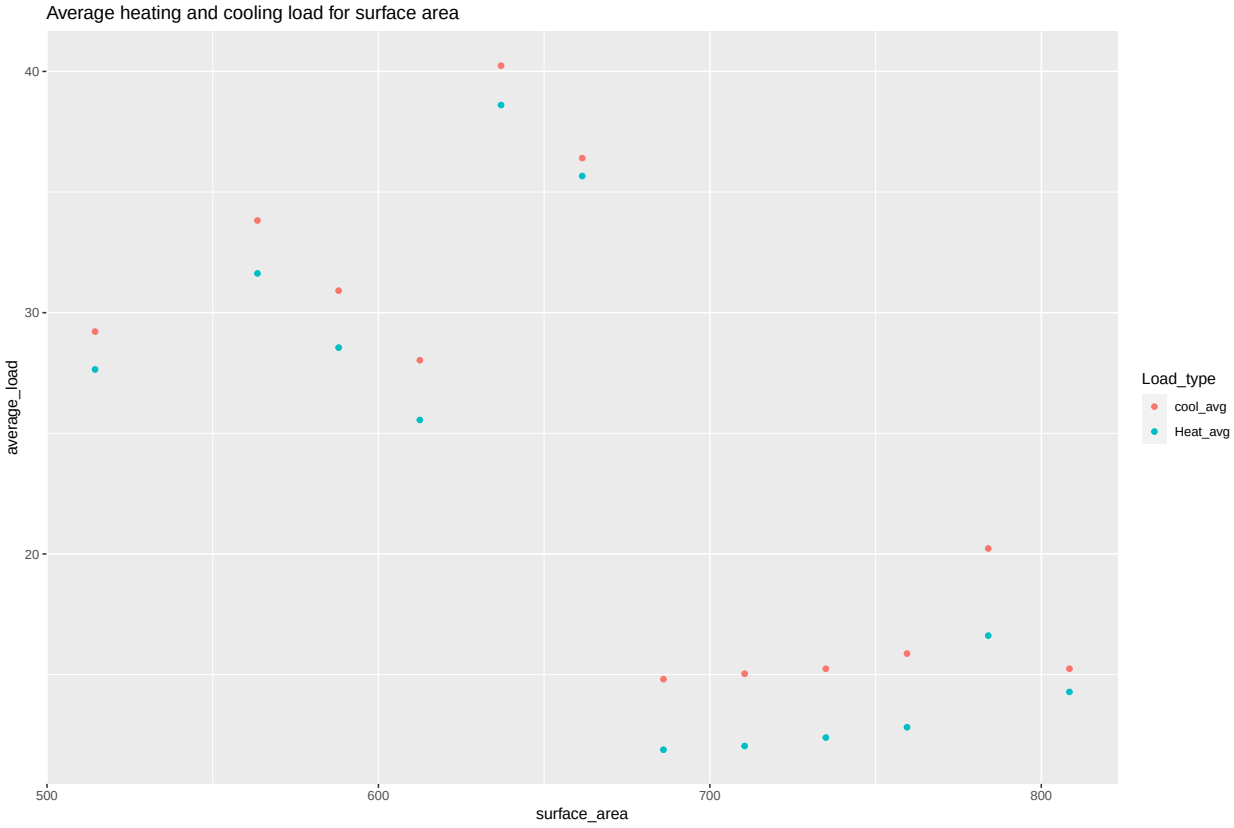
When the summary function is applied to the energydat data set, it displayed important statistics : range (min & max), mean, median, etc and showed the variability in the features and outcome.

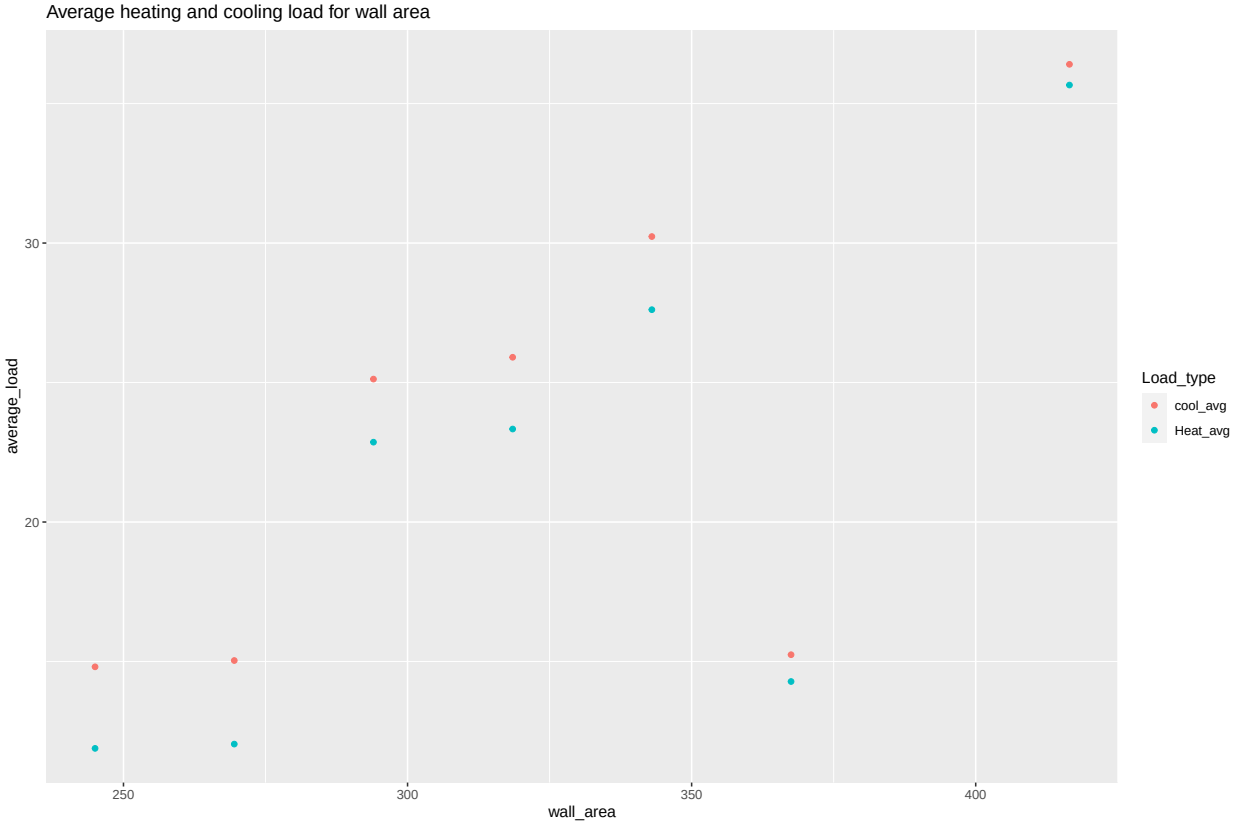
```
#summary stats of data
summary(energydat)
```

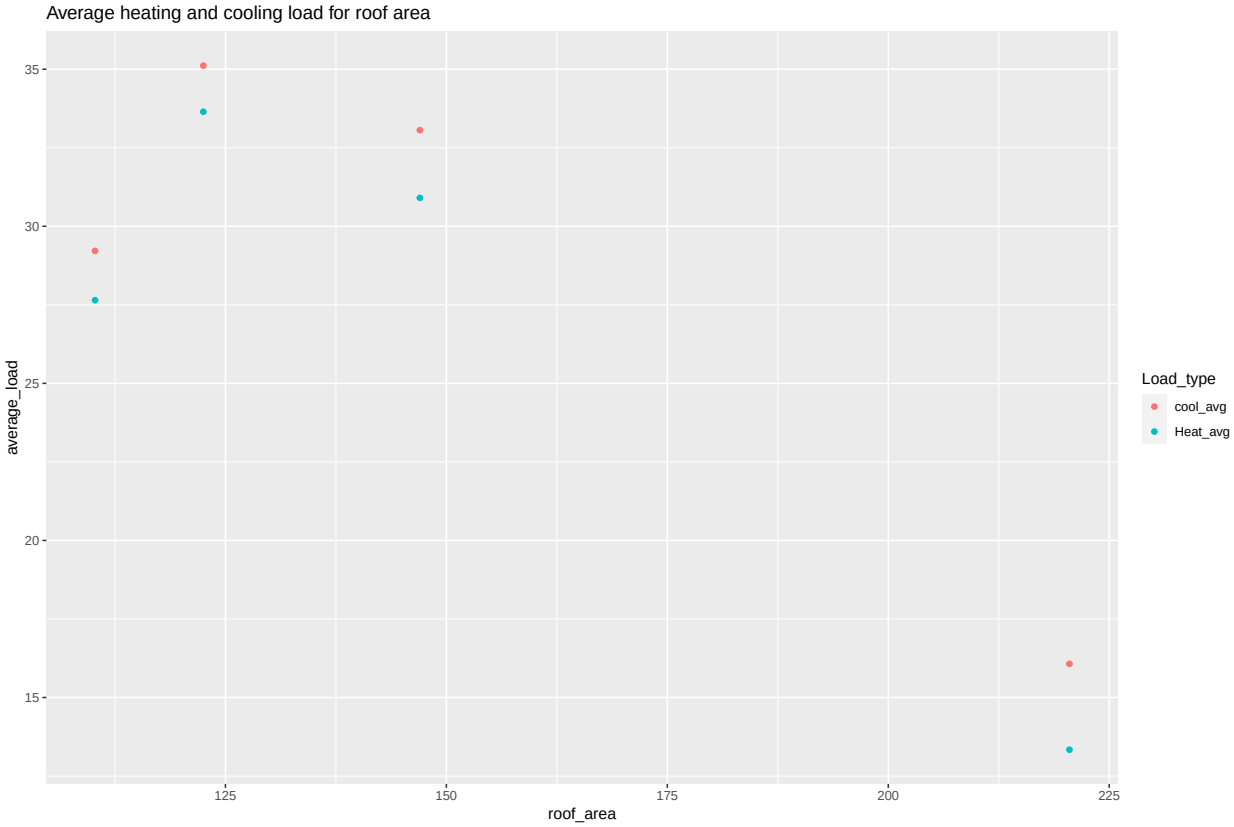
```
 relative_compactness  surface_area     wall_area       roof_area
 Min.   :0.6200       Min.   :514.5   Min.   :245.0   Min.   :110.2
 1st Qu.:0.6825       1st Qu.:606.4   1st Qu.:294.0   1st Qu.:140.9
 Median :0.7500       Median :673.8   Median :318.5   Median :183.8
 Mean   :0.7642       Mean   :671.7   Mean   :318.5   Mean   :176.6
 3rd Qu.:0.8300       3rd Qu.:741.1   3rd Qu.:343.0   3rd Qu.:220.5
 Max.   :0.9800       Max.   :808.5   Max.   :416.5   Max.   :220.5
 overall_height  orientation    glazing_area    glazing_area_distribution
 Min.   :3.50   Min.   :2.00   Min.   :0.0000   Min.   :0.000
 1st Qu.:3.50   1st Qu.:2.75   1st Qu.:0.1000   1st Qu.:1.750
 Median :5.25   Median :3.50   Median :0.2500   Median :3.000
 Mean   :5.25   Mean   :3.50   Mean   :0.2344   Mean   :2.812
 3rd Qu.:7.00   3rd Qu.:4.25   3rd Qu.:0.4000   3rd Qu.:4.000
 Max.   :7.00   Max.   :5.00   Max.   :0.4000   Max.   :5.000
  heating_load    cooling_load
 Min.   : 6.01   Min.   :10.90
 1st Qu.:12.99   1st Qu.:15.62
 Median :18.95   Median :22.08
 Mean   :22.31   Mean   :24.59
 3rd Qu.:31.67   3rd Qu.:33.13
 Max.   :43.10   Max.   :48.03
```

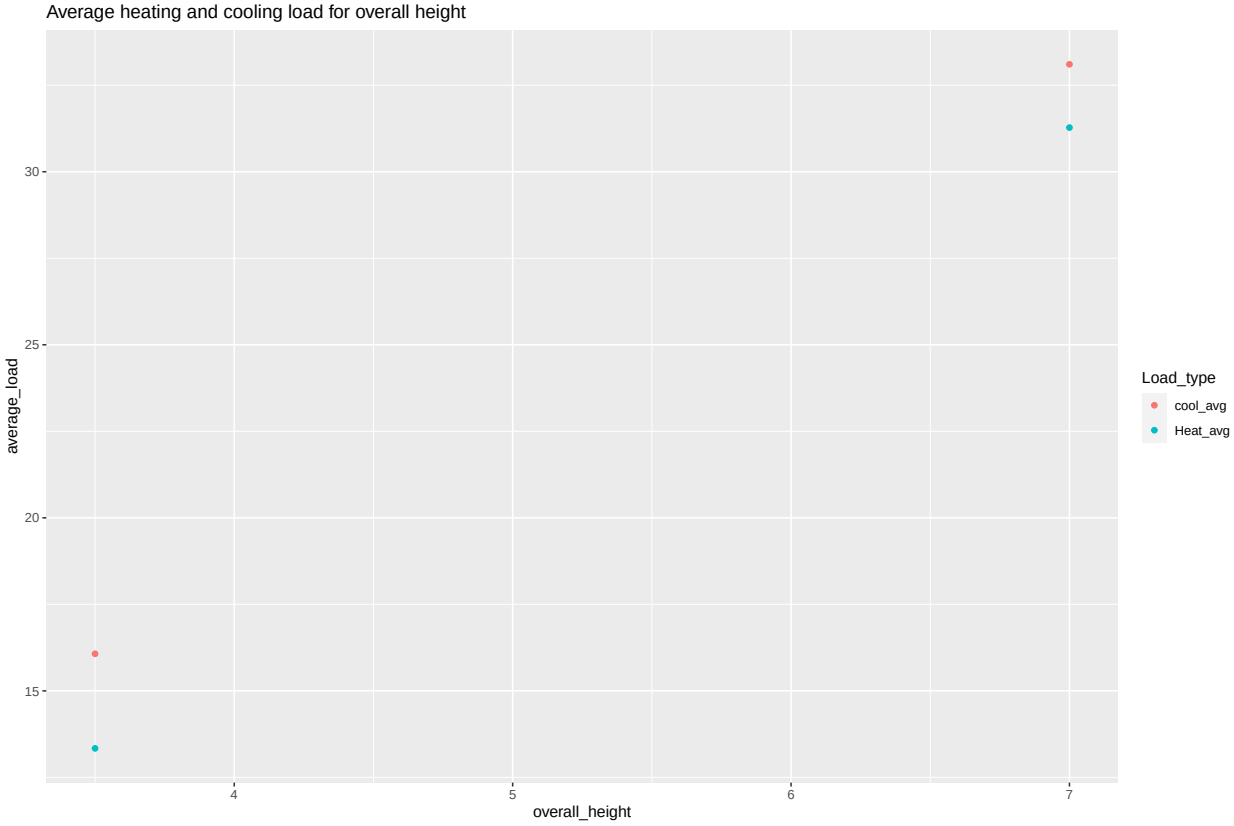
The average heating load and average cooling load for each unique value of each variable are shown in the subsequent graphs

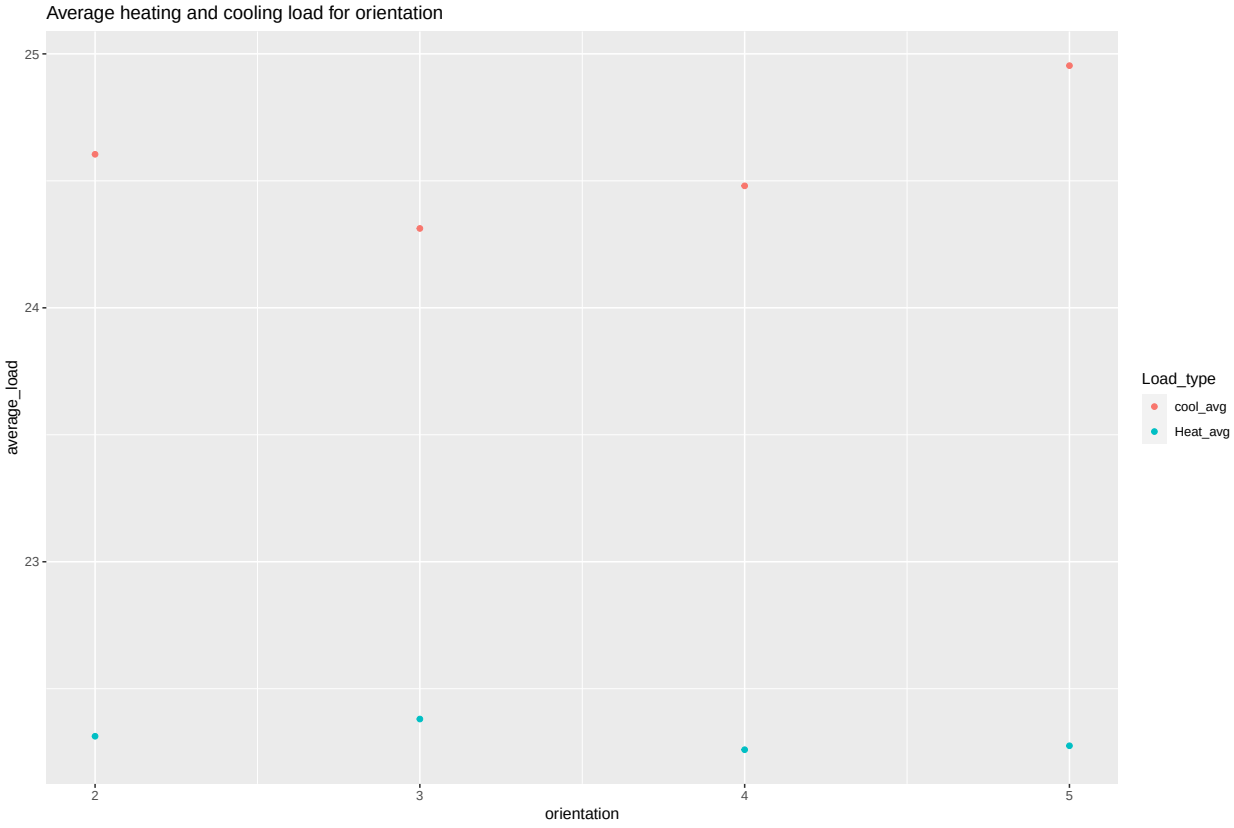


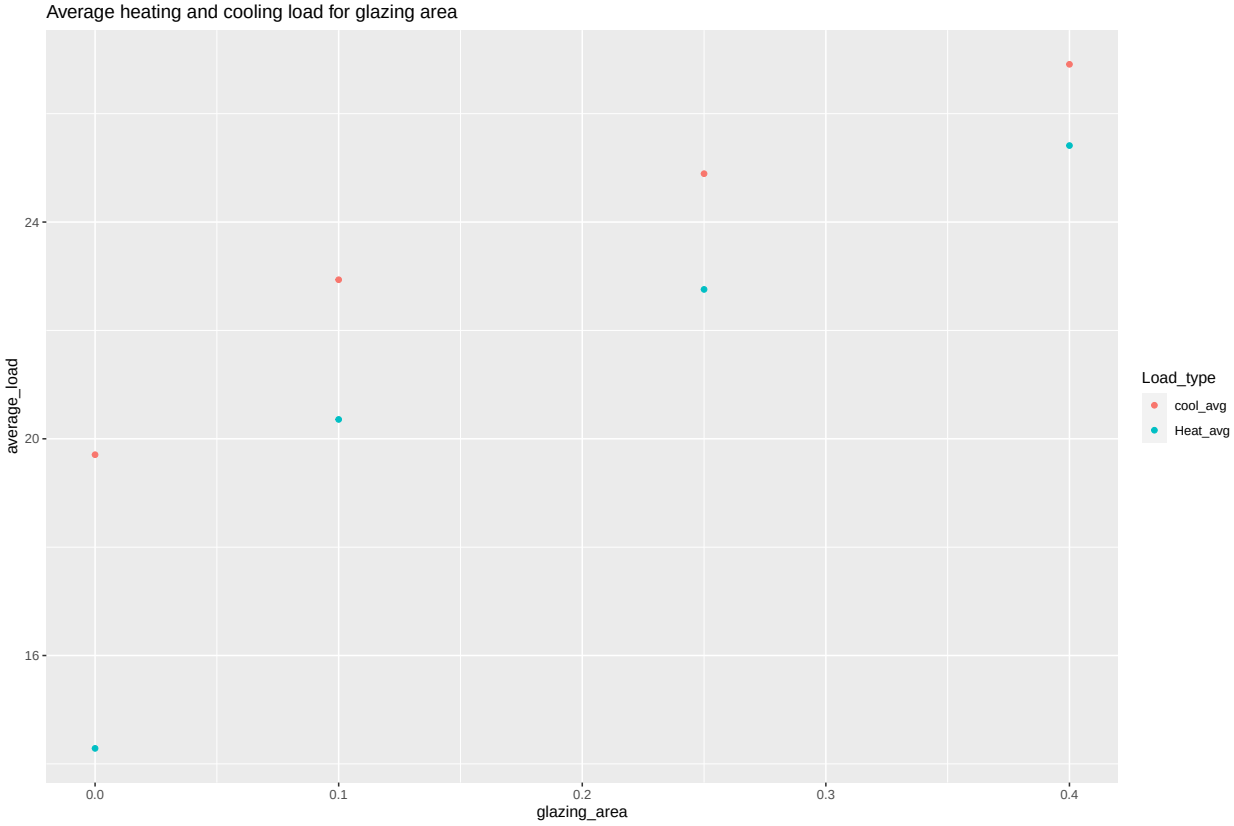

Average heating and cooling load for surface area
average_load
surface_area
40
30
20
500
600
700
800
Load_type
cool_avg
Heat_avg

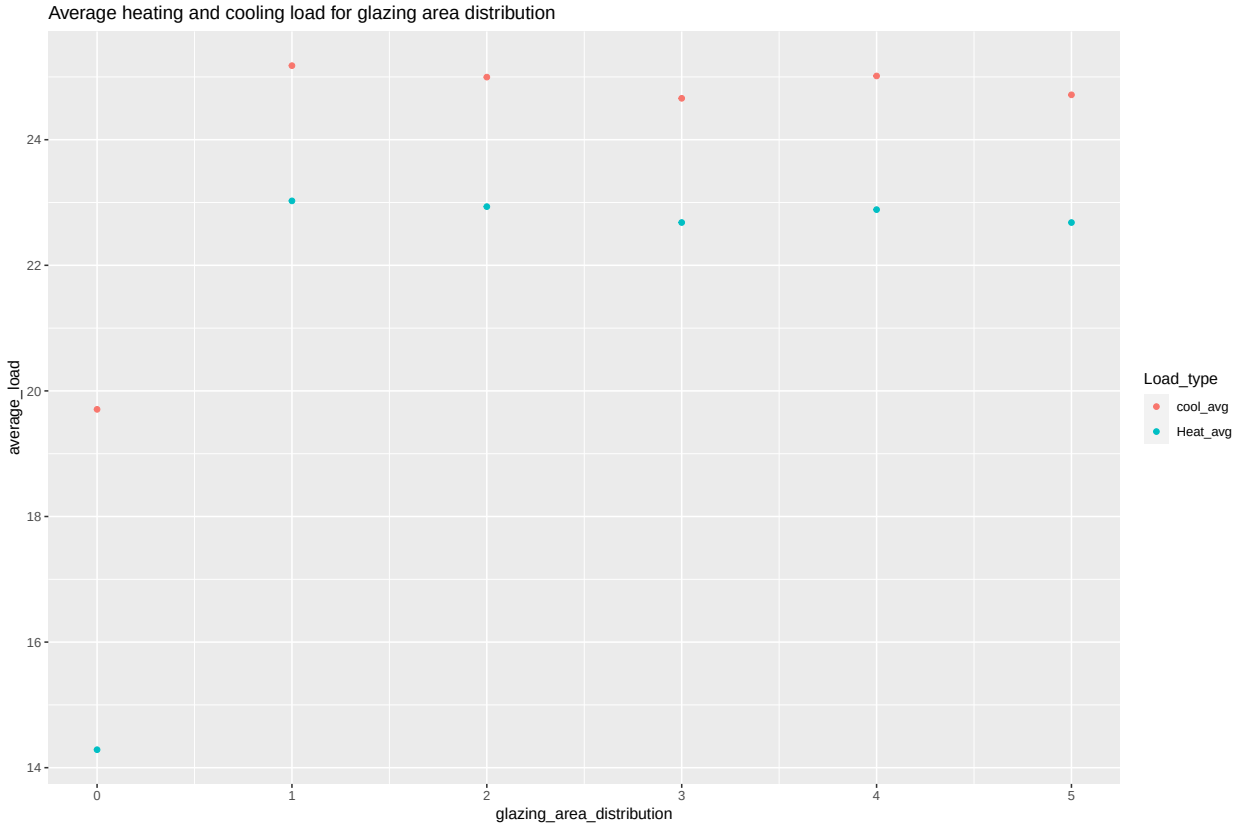

Average heating and cooling load for wall area
average_load
wall_area
30
20
250
300
350
400
Load_type
cool_avg
Heat_avg


Average heating and cooling load for roof area
average_load
35
30
25
20
15
125
150
175
200
225
roof_area
Load_type
cool_avg
Heat_avg


Average heating and cooling load for overall height
average_load
30
25
20
15
4
5
6
7
overall_height
Load_type
cool_avg
Heat_avg


Average heating and cooling load for orientation
25
24
23
average_load
2
3
4
5
orientation
Load_type
cool_avg
Heat_avg


Average heating and cooling load for glazing area
24
20
16
average_load
0.0
0.1
0.2
0.3
0.4
glazing_area
Load_type
cool_avg
Heat_avg



Each graph show the changes in the average heating load and average cooling load as the corresponding variable changes. However, the relationships displayed in the graph could be confounded by the impact of changes in other features for each unique entry of a feature under consideration. The graphs also shows the number of unique values for each variable: - relative compactness and surface area have 12 each, - wall area has 7, - glazing area distribution has 6, - roof area, orientation,and glazing area have 4 each, and - overall height has 2

2.2 Data Cleaning

2.2.1. Creating training (probe) set and hold out test (evaluation) set The energydat data set was split using createDataPartition into probe set (90% of the data) and evaluation set (remaining 10%) using createDataPartition function in the caret package.

2.2.2. Creating train set and test set from edx To be able to test the performance of the models being developed, the probe data set was split into a train set and a test set. The train_set was used for model building while the test set was used to check the performance of the developed models. The method for creating the train and test set from the probe data set was similar to that for creating the probe and evaluation set from the energydat data set.

The train and test sets were created using the createDataPartition function in the caret package. The train set consisted of 80% of the probe data set while the test set constituted the remaining 20%.

The metric used for evaluating the performance of each model built was residual mean squared error, RMSE. RMSE represents the difference between values predicted by each model and the actual values observed. In relative terms, it is the error made when predicting the heating load and cooling load, i.e. the deviation of prediction values from real values. Hence, the lower the rmse value, the better the predictive power of a

given model. The target for this project was to obtain as low an rmse value as possible. The RMSE equation is shown below

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

In R, the RMSE function was defined as follows:

```
# define function for computing RMSE for vectors of energy load and corresponding predictors
RMSE <- function(true_load, predicted_load){sqrt(mean((true_load - predicted_load)^2))}
```

2.3 Modelling

For each of the following models, the RMSE function was applied with the predicted load to calculate the rmse of the model. A table, RMSE_results was also set up to which all models developed and their corresponding rmse values are stored for easy comparison. The number of digits to be printed was also set with the option function as shown below:

2.3.1. Model 1: Using the average The simplest model for predicting heating load and cooling load based on the data is one that gives the average heating load of all the entries as an estimate of the heating load, and similarly, gives the average cooling load of as the estimate for cooling load for a combination of variable. This model gives the same average value of heating load for each combination of variables and likewise for cooling load.

2.3.2 Model 2: Linear Regression Beyond the simple prediction of average values for cooling and heating load, regression models were applied and evaluated for rmse. For each outcome being predicted, the other outcome was excluded from the model fitting. For example, when fitting a model for heating load, the 8 features were used while the cooling load was excluded from the model. For linear regression, after fitting, the summary function was used to display the co-efficient and p-values for each variable. Variables with p-value > 0.05 were then excluded, since they are not statically significant to the fitting. NA value was obtained for co-efficient and p-value for roof_area. This indicates that this variable could also be excluded from the model. The alias function showed that roof_area was dependent on the wall area and surface area. For the final linear regression model, roof area (NA as p-value), and both orientation and glazing area distribution (p-values > 0.05) were excluded. . . However, these exclusions only marginally improved the rmse of the model for heating load (from 3.0340 to 2.9953) for heating load, but did not have a significant change on that of the cooling load (3.3389 vs 3.3434).

2.3.3 Model 3 : K-nearest neighbours (knn) All features were used in the fitting of KNN model using the train_set data. Cross-validation was performed to optimize the k value(number of nearest neighbors). A best tune value of 3 was obtained for both heating load and cooling load. This optimized value of k was automatically used in the fitted model to predicted both outcomes on the test set. The rmse obtained for predictions on the test set were 1.6959 and 1.6053 for heating load and cooling load respectively.

```
## KNN model for heating load
set.seed(1, sample.kind = "Rounding") #set seed to 1

#cross-validation for knn model for heating load
knn_cv_heat <- train(heating_load ~ . - cooling_load, method = "knn",
                     data = train_set,
                     tuneGrid = data.frame(k = seq(3, 51, 2)),
                     trControl = trainControl(method = "cv", number = 10, p = .9))

knn_cv_heat$bestTune # display optimized complexity parameter
```

```
  k
1 3
```

```
knn_heat_pred <- predict(knn_cv_heat, test_set) # predict heating load

knn_heat_rmse  <- RMSE(test_set$heating_load,knn_heat_pred) # compute rmse for heating load

knn_heat_rmse # print rmse
```

```
[1] 1.6959
```

```
## KNN model for cooling load
set.seed(1, sample.kind = "Rounding") # set seed to 1

# use cross-validation to choose k
knn_cv_cool <- train(cooling_load ~ . - heating_load, method = "knn",
                     data = train_set,
                     tuneGrid = data.frame(k = seq(3, 51, 2)),
                     trControl = trainControl(method = "cv", number = 10, p = .9))

knn_cv_cool$bestTune # display optimized k value
```

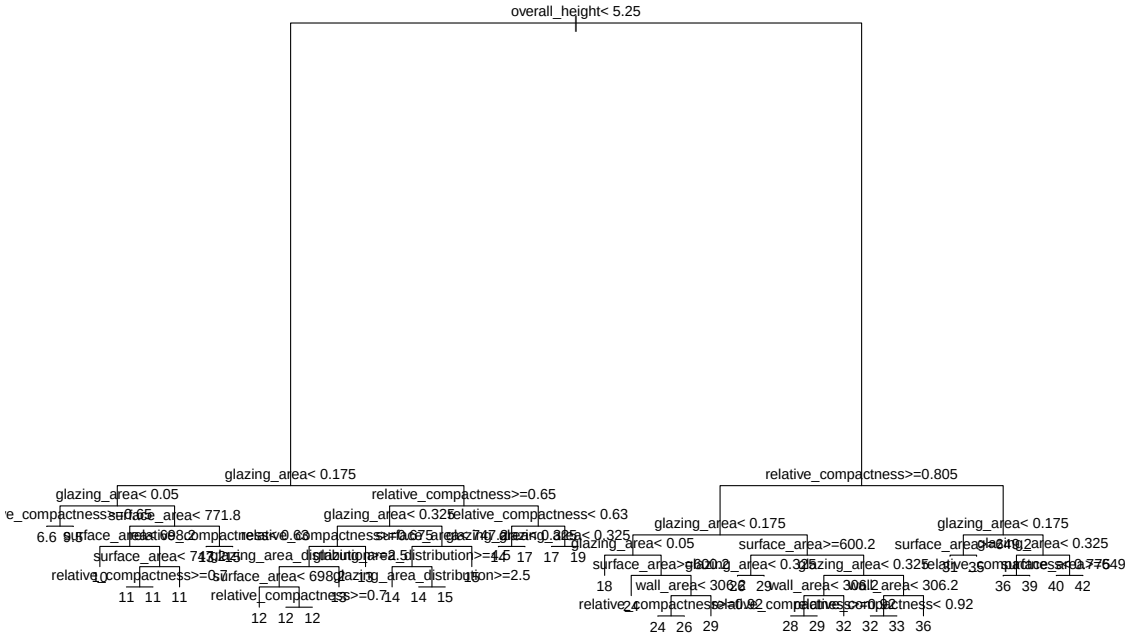
```
  k
1 3
```

```
knn_cool_pred <- predict(knn_cv_cool, test_set) # predict cooling load

knn_cool_rmse  <- RMSE(test_set$cooling_load,knn_cool_pred) # compute rmse for cooling load

knn_cool_rmse # print rmse
```

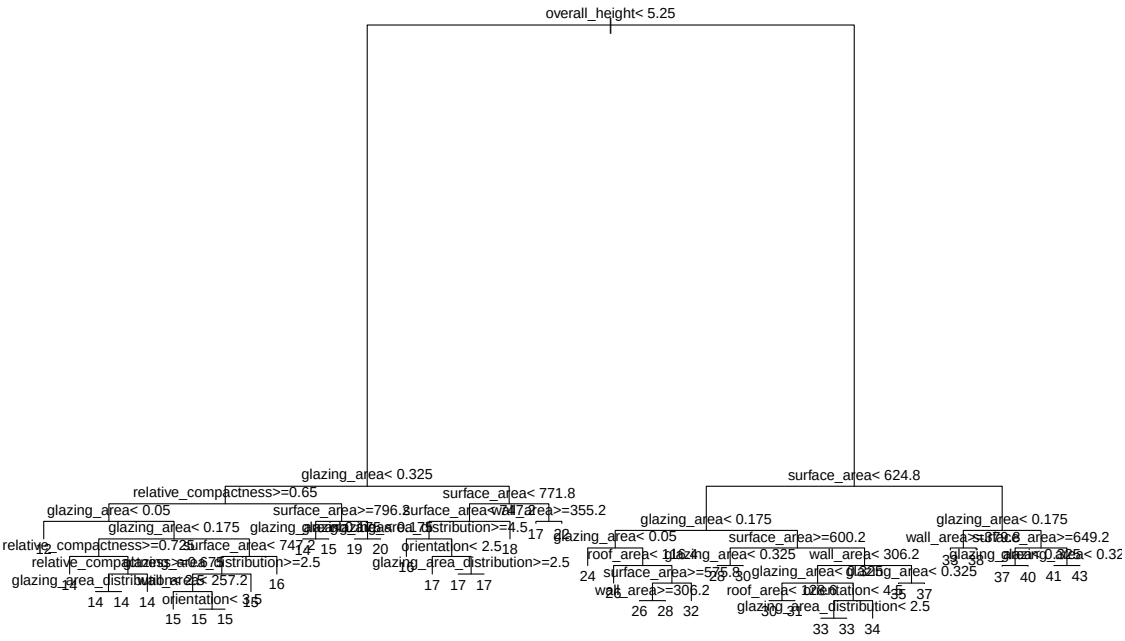
```
[1] 1.6043
```

2.3.4 Model 4 : Decision Tree A decision tree is a tree based algorithm used to solve both regression and classification problem. Here, a regression tree was applied since the outcomes are continuous. The train function in the caret package was used with the method set as 'rpart' for decision trees. Cross validation was performed at the onset,to optimize the complexity parameter (the minimum improvement in the model at each node),cp over a range of 0 to 0.1. For both heating and cooling load, the optimized cp value was 0. The model was fitted with all variables and the rmse obtained for heating load and cooling load based on values predicted on the test set were 0.75732 and 1.8205 respectively. The rmse for cooling was surprisingly larger than expected considering the value for heating load. Several attempts to prune the fitted tree for the cooling load only produced large rmse

```
# plot decision tree for heating load
plot(rpart_heat$finalModel)
text(rpart_heat$finalModel, cex = 0.8)
```

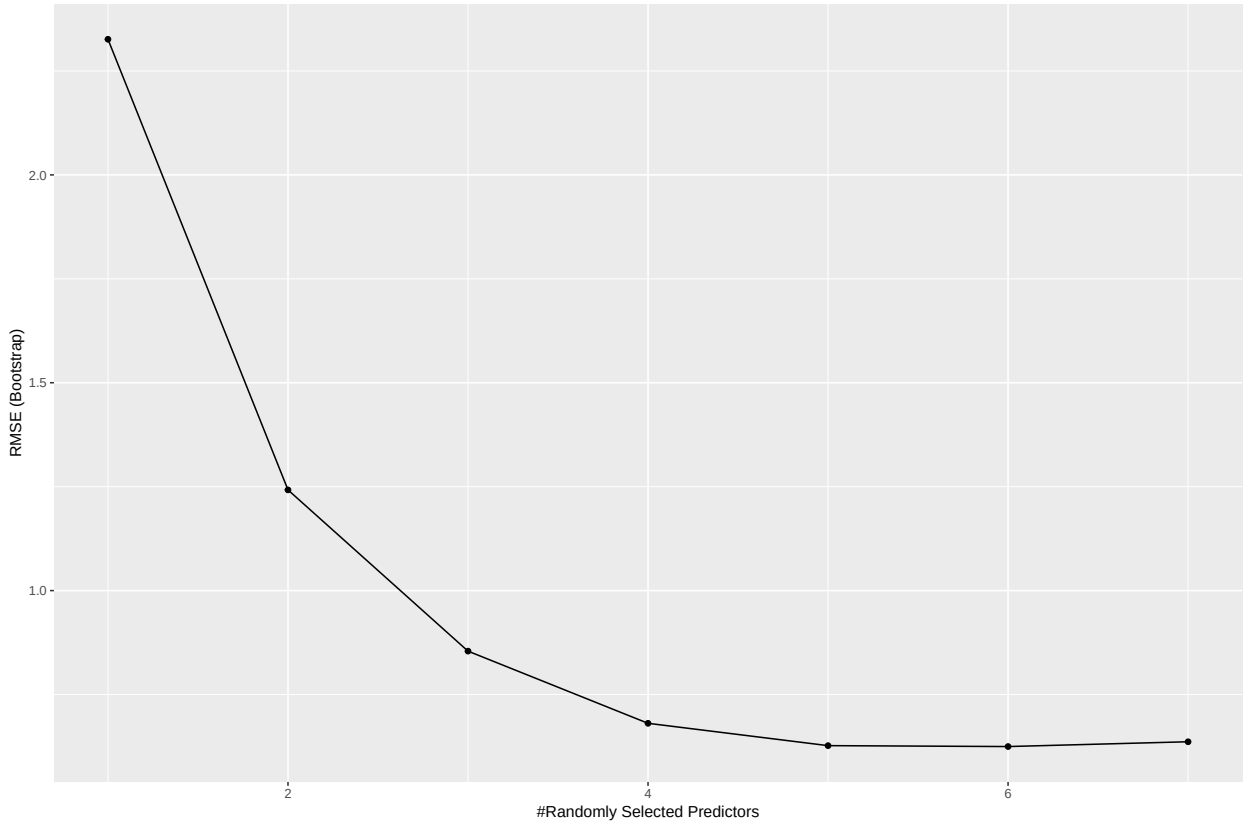



```r
# plot decision tree for cooling load
plot(rpart_cool$finalModel)
text(rpart_cool$finalModel, cex = 0.8)
```



2.3.4 Model 5 : Random Forest Random forest is a collection of decision trees. It improves predictions performance and reduces instability by averaging multiple decision tress[1]. The final prediction is the average prediction of these trees.For this model, the train function with method set as rf was used to perform cross validation and train the model. For cross validation, the value optimized was mtry, the number of variables available for splitting at each tree node. The best value obtained for mtry was 6 for heating load and 4 for cooling load. Model fitting was performed with best tune mtry values and ntree set to 100. A plot of the error vs number of tree showed that ntree at 100 was a sufficient choice. Application of the VarImp function in caret package also display the importance of each variable to the model

```
## random forest model for heating load
set.seed(5, sample.kind = "Rounding") # set seed to 5

# use cross-validation to select mtry
rf_heat <- train(heating_load ~ . - cooling_load, method = "rf", data= train_set,
                 ntree = 100, tuneGrid = data.frame(mtry = seq(1:7)))

#plot of mtry vs RMSE
ggplot(rf_heat)
```




```
rf_heat$bestTune #display optimized mtry
```

```
  mtry
6    6
```

```
# important variables for predicting heating load
varImp(rf_heat)
```

```
rf variable importance

                          Overall
surface_area               100.00
overall_height              99.76
roof_area                   82.71
relative_compactness        79.86
glazing_area                30.49
wall_area                    8.62
glazing_area_distribution    8.33
orientation                  0.00
```

```
## random forest model for cooling load
set.seed(5, sample.kind = "Rounding") # set seed to 5

# use cross-validation to select mtry
rf_cool <- train(cooling_load ~ . - heating_load, method = "rf", data= train_set,
```

```
                ntree = 100, tuneGrid = data.frame(mtry = seq(1:7)))

#mtry value that maximizes accuracy
rf_cool$bestTune
```

```
  mtry
4    4
```

```
# important variables for predicting cooling load
varImp(rf_cool)
```

```
rf variable importance

                         Overall
surface_area              100.00
roof_area                  60.52
overall_height             37.83
relative_compactness       36.01
glazing_area               10.12
wall_area                   8.97
glazing_area_distribution   2.60
orientation                 0.00
```

2.3.6. Predicting energy load on the evaluation set Based on rmse values, Model 5 (random Forest) and Model 3 (Knn) was then updated with the probe data set as a training set and then used to make prediction on the evaluation set for heating load and cooling load respectively.

```
################################
# Application to evaluation set
################################

## random forest model for heating load in evaluation set
set.seed(5, sample.kind = "Rounding") # set seed to 5

# use cross-validation to select mtry for
rf_heat <- train(heating_load ~ . - cooling_load, method = "rf", data= probe,
                ntree = 100, tuneGrid = data.frame(mtry = seq(1:7)))

#display optimized mtry
rf_heat$bestTune
```

```
  mtry
5    5
```

```
pred_heat <- predict(rf_heat, evaluation) # predict heating load

# compute and assign rmse for heating load
heat_rmse <- RMSE(evaluation$heating_load,pred_heat)

heat_rmse # print rmse
```

```
[1] 0.51654
```

```
## Model 3 (knn) for predicting cooling load in evaluation set
set.seed(5, sample.kind = "Rounding") # set seed to 5

# use cross-validation to choose k
knn_cool <- train(cooling_load ~ . - heating_load, method = "knn",
                      data = probe,
                      tuneGrid = data.frame(k = seq(3, 51, 2)),
                      trControl = trainControl(method = "cv", number = 10, p = .9))

knn_cool$bestTune # display optimized k value
```

```
  k
1 3
```

```
pred_cool <- predict(knn_cool, evaluation) # predict cooling load

cool_rmse  <- RMSE(evaluation$cooling_load,pred_cool) # compute rmse for cooling load

cool_rmse # print rmse
```

```
[1] 1.402
```

3. Result & Discussion

Table 1: RMSE results for different models

Model	RMSE_heat	RMSE_cool
Using the average	9.8110	9.4237
Linear Regression	2.9953	3.3434
knn	1.6959	1.6043
Decision Tree	0.7573	1.8205
Random Forest	0.5329	1.7989

Section 2.3 outlined the different modeling approaches to prediction the heating load and cooling load for buildings.The rmse results for each model are presented in the table above. Model 1 (using the average) predicted the average heating load of the train set as the heating load for each entry in the test set, and likewise, the average cooling load as the cooling load for the test set entries. This gave an RMSE of 9.8110 for heating load and 9.4237 for cooling load.

Model 2 (linear regression) applied linear regression in predicting energy load using variables. This model significantly improved the rmse obtained from prediction on test_set, yielding 3.0340 for heating load and 3.3389 for cooling load. When variables considered statistically insignificant by the linear regression model were removed, the rmse for heating and cooling load changed to 2.9953 and 3.3434 respectively.

In Model 3, knn model was applied for regression and subsequent prediction.This model produced much better rmses than the linear regression model, giving values of 1.6959 and 1.6043 for heating load and cooling loads respectively.

Model 4 (Decision Tree) entailed a decision tree regression using the train set data. The model was fitted with all variables and then used to make prediction on the test set. The rmse obtained for heating load

and cooling load based on values predicted on the test set were 0.7573 and 1.8205 respectively. Model 4 significantly improved prediction on the heating load compared to knn(1.6959) or linear regression(2.9953) as shown by its lower rmse. However, Model 4 underperformed slightly on cooling load when compared to knn (1.6043), although outperforming linear regression rmse. Attempts to improve the rmse on cooling load by pruning proved futile.

Model 5 (random Forest) used random forest model to perform regression on the train set data, prior to prediction on the test set. RMSE values of 0.5329 and 1.7989 were obtained for heating load and cooling load for this model, which out perform all previous models except for the knn model for cooling load test model.For the heating model,surface area was the most important variable, followed by overall height (99.76) , roof area (82.71), and then relative compactness (79.86). For cooling load, surface area had the highest importance, followed by roof area (60.52); overall height was third (37.83), and relative compactness was the fourth (36.01). These differences in the position and extent of importance of variables to the model for heating and cooling load, coupled with the differences in the number of unique values for each variable likely accounts for the difference in the effectiveness of the random forest model as seen in the rmse for each outcome

Considering the outcomes from all five models, the best performing model based on rmse value for prediction on the test set was chosen. Model 5 (Random forest) provided the best overall prediction based on rmse, with an overwhelming low rmse on heating load compared to others. However, Model 3 (knn) gave the best rmse for cooling load (1.6053), being lower than that Model 5 (1.7929). Therefore, Model 5 was applied for heating load modeling & prediction while Model 3 will be applied for cooling load modeling and prediction. These were then applied using 'probe' data set as a training set and used to make prediction on the evaluation set. RMSE values of **0.51654** and **1.4020** were obtained for heating load and cooling load respectively.

4. Conclusion

This report outlines the development of a model for predicting energy (heating and cooling load) for buildings. The project was built on data set generated by a civil/structural engineer and staff at the University of Oxford, which was made available on UCI Machine Learning Repository website. The data set comprised of 768 samples with 8 features that define the shape of the residential buildings and two predictable outcomes, heating load and cooling load. The ability to predict the energy demand for a structure before construction is important for creating energy efficient building designs. The primary goal of this project is to create a model that predicts the energy efficiency of buildings using variables that determines the shape of the building. A secondary goal of the project was to minimize root mean squared error, RMSE (a measure of the deviation of predicted values from observed values) for the predicted heating load and cooling load.

Five models were developed ranging from a basic use of averages(Model 1) to linear regression (Model 2), knn (Model 3), decision tree (Model 4) and random forest (Model 5). Model parameters were optimized were possible and a final model chosen based on performance of trained models in prediction on test_set. Model 5 produced the best rmse for heating load and while Model 3 gave the best for cooling load.These model were then applied to the evaluation set, yielding 0.51654 and 1.4020 for heating and cooling loads respectively

Some limitations of the current method employed in this project is that (1) it assumes an additive relationship between all the predictors, (2) there might be other data structures the model did not account for. For example, it does not account for multi-colinearity in the data set (although this may not impact prediction if not severe)

For the future, one major improvement on this model should also apply an ensemble model. An Ensemble would combine multiple machine learning algorithms into one model to improve prediction.This method could combine both the random forest and knn models into one, and/ or include other appropriate models. Other considerations could be to apply Principle Component Analysis (PCA) which would reduce the number of variables considered, based on variance. However, there may be a trade off in accuracy with the dimensional reduction of PCA.