Energy Score Forecast Report

Group No.: Group 18

Zizheng Chen and Kanishka Parganiha

I. Background and Introduction

- **Introduction**

In recent years, people's attention to the problem of energy shortage has increased significantly. To solve the energy over-utilization problem, government or energy company always renew their policy or cost method to reduce the waste of energy. But the problem is how to define energy waste and how to make a good policy to limit users who waste resources without harming ordinary users. This project can provide a possible solution to the above problems.

- **Problem definition**

Of the 25 variables provided by the data, we need to summarize a reliable parameter that represents the energy and water use of each building. Based on this parameter, we can score each user and determine whether this user has overused the resources. Based on the overall resource forecast and usage, we can help the government or company define new policies or pricing methods to constrain the situation of resource waste or over-utilization.

As stated in the report that we are building a predictive model that correlates the energy data to the property use details to identify the key drivers of energy use and predicts the Energy Star Score which is a measure of how well a property is performing relative to similar properties when normalized for climate and operational characteristics. The 1-100 scale is set so that 1 represents the worst-performing buildings, and 100 represents the best performing buildings. A score of 50 indicates that a building is performing at the national median, taking into account its size, location, and operating parameters.

The goal of the project is to solve the prediction of energy parameters to help make better plans for the next year.

- **Our Solution**

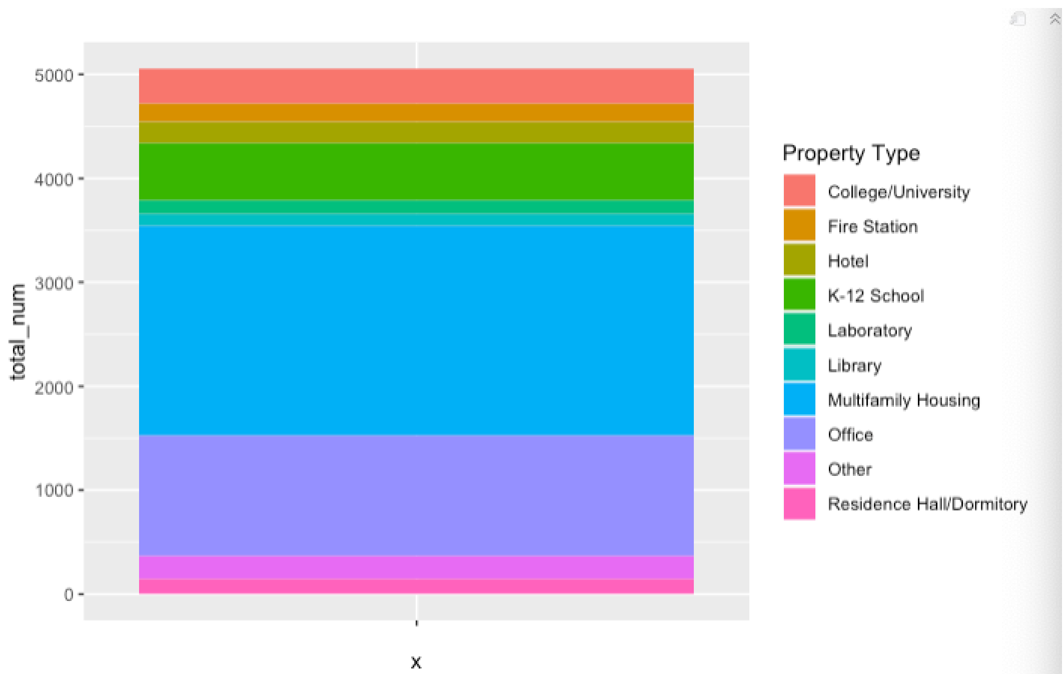
For this problem, we have two choices, first to consider these datasets as a classification model with 1 to 100 categorical variable, second to take this problem as regression problem. We performed Linear Regression, Random Forest Regression, and Gradient Boosting Regression, and the results were satisfactory compared to the classification model. Here is the table of using a different algorithm and their result.

According to the result from table above, we think Random Forest regression and Gradient Boosting regression are two possible solutions for this problem.

Machine Learning Methods	Mean Squared Error	Mean Absolute Error	Root Mean Squared Error
KNN	1727.02	32.80	41.56
Keras Classification	672.34	18.13	25.93
Linear Regression	868.75	24.38	29.47
Random Forest	620.63	16.83	24.91
Bagging Classifier	601.38	16.53	24.52
Extra Tree Classifier	459.45	14.23	21.43
Keras Regression	840.21	24.85	28.99
Support Vector with Grid Search	2273.22	36.99	47.68
Gradient Boosting	877.40	20.84	29.62
Random Forest Regression	337.09	13.61	18.36
Gradient Boosting Regression	949.05	24.48	30.81

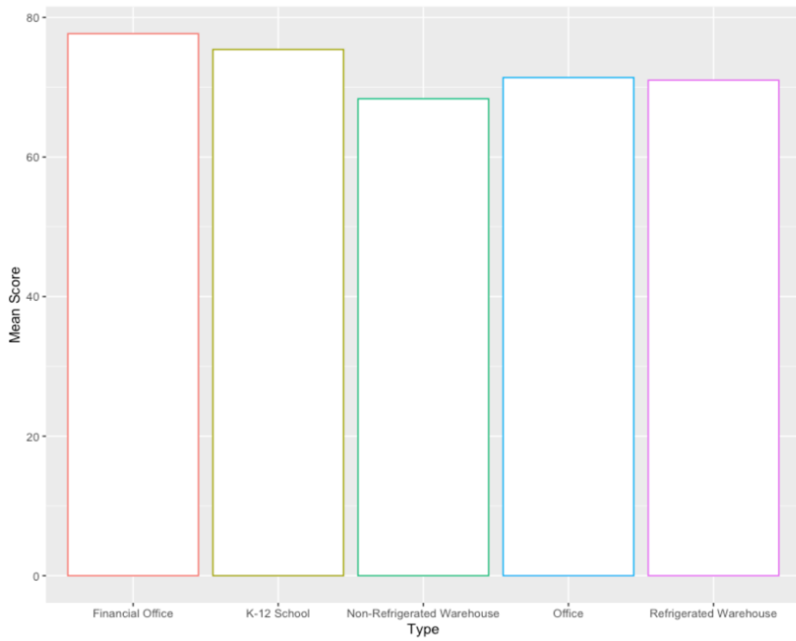
II. Data Exploration and Visualization

Our data comes from the Boston Government website. The content of the data includes the addresses of buildings, zipcode, energy usage, energy score, property type etc. In this project, we focus on energy score and energy usage variables.



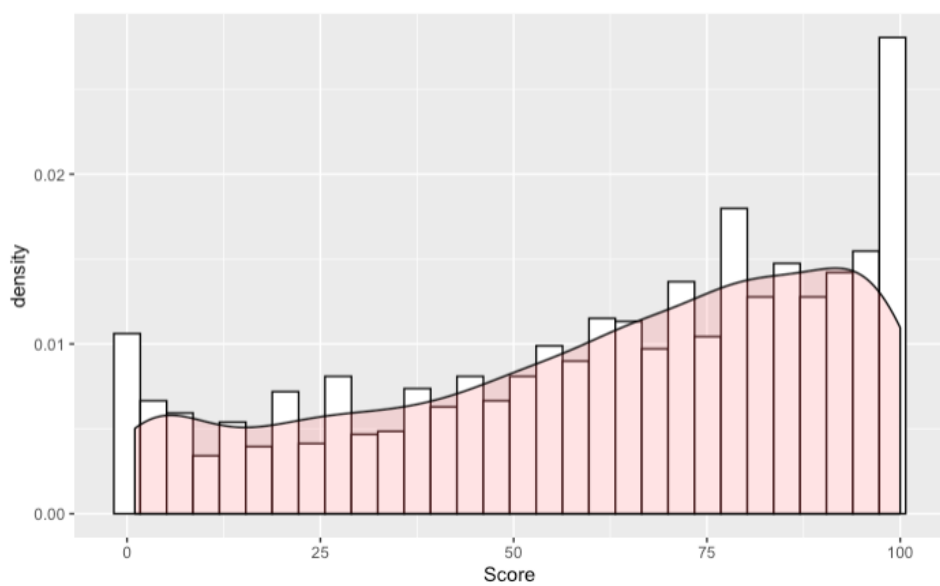
2.a

Data visualization makes us easier to understand and detect pattern, trends and outlier in the group of data. According to the bar chart from above(2.a), we obtained the top ten most building types from dataset. The Property types that accounts for a large proportion of the dataset are House, Office and school.



2.b

We created a bar chart(2.b) by using the mean of energy score group by same property type. Among all property types, we found that the building with top 5 highest energy scores are Financial Office, School, Refrigerated Warehouse, Office and Non-Refrigerated Warehouse. Their score range between 60-80, it shows that in the Boston area, these types of buildings have generally high energy scores, and their daily energy consumption is large.



2.c

Chart 2.c represent the distribution of Energy score in Boston area. We found that data with an energy score greater than 50 has a relatively high density, indicating that buildings in the Boston area generally have higher energy consumption.

III. Data Preparation and Preprocessing

Our data contains 25 variables. For this regression analysis, we only took 8 numeric variables for analysis.

From the dataset, the type of 8 variables are factors, we first convert them to numeric variable, and summary statistic information for them.

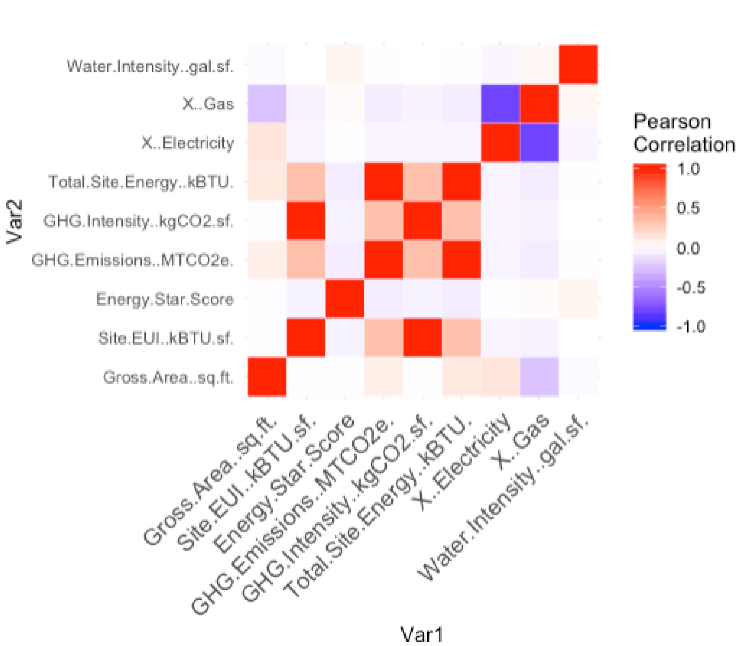
```
Gross.Area..sq.ft. Site.EUI..kBTU.sf. Energy.Star.Score GHG.Emissions..MTCO2e.
Min.   :      1    Min.   :     0.0   Min.   :  0.00    Min.   :      0.0
1st Qu.:  45000    1st Qu.:    52.0   1st Qu.: 45.00    1st Qu.:    167.4
Median :  80000    Median :    72.7   Median : 74.00    Median :    367.2
Mean   : 174551    Mean   :   666.2   Mean   : 65.23    Mean   :   2202.7
3rd Qu.: 177064    3rd Qu.:   103.9   3rd Qu.: 90.00    3rd Qu.:    943.5
Max.   :4921206    Max.   :579540.1   Max.   :100.00    Max.   :1098618.6
NA's   :2082       NA's   :25         NA's   :1255      NA's   :2082
GHG.Intensity..kgCO2.sf. Total.Site.Energy..kBTU. X..Electricity      X..Gas
Min.   :   -0.60         Min.   :0.000e+00        Min.   :0.0000   Min.   :0.0000
1st Qu.:    3.40         1st Qu.:2.604e+06        1st Qu.:0.2263   1st Qu.:0.3516
Median :    4.90         Median :5.742e+06        Median :0.4068   Median :0.5789
Mean   :   44.14         Mean   :3.926e+07        Mean   :0.4522   Mean   :0.5460
3rd Qu.:    7.00         3rd Qu.:1.489e+07        3rd Qu.:0.6303   3rd Qu.:0.7707
Max.   :38485.10         Max.   :1.966e+10        Max.   :1.0000   Max.   :1.0000
NA's   :30               NA's   :1834             NA's   :2108     NA's   :2613
Water.Intensity..gal.sf.
Min.   :       0
1st Qu.:      10
Median :      23
Mean   :   25284
3rd Qu.:      44
Max.   :60595650
NA's   :298
```

As we can see from the statistic of dataset, for each variable, we have lot of NA values in it. We visualize the NA value by using missing map.

We have 28% NA value in our dataset, in order to not affect the training of the model, we decided to delete them. In addition, We found that in the statistical results of the data, the value range of each variable is very different, some variables have a small value range, and some have a large value range. So we take standardization on dataset, the final dataset list below.

```
Gross.Area..sq.ft.  Site.EUI..kBTU.sf.  Energy.Star.Score GHG.Emissions..MTCO2e.
Min.   :-0.61003   Min.   :-0.03264   Min.   :  1.00   Min.   :-0.06615
1st Qu.:-0.46156   1st Qu.:-0.02894   1st Qu.: 40.00   1st Qu.:-0.05910
Median :-0.32645   Median :-0.02775   Median : 68.00   Median :-0.05235
Mean   : 0.00000   Mean   : 0.00000   Mean   : 61.47   Mean   : 0.00000
3rd Qu.: 0.04719   3rd Qu.:-0.02611   3rd Qu.: 87.00   3rd Qu.:-0.03484
Max.   :13.73382   Max.   :40.19265   Max.   :100.00   Max.   :38.88169
GHG.Intensity..kgCO2.sf. Total.Site.Energy..kBTU. X..Electricity      X..Gas
Min.   :-0.03228         Min.   :-0.06981         Min.   :-1.6921   Min.   :-2.0551
1st Qu.:-0.02873         1st Qu.:-0.06209         1st Qu.:-0.8117   1st Qu.:-0.6999
Median :-0.02768         Median :-0.05505         Median :-0.1396   Median : 0.1495
Mean   : 0.00000         Mean   : 0.00000         Mean   : 0.0000   Mean   : 0.0000
3rd Qu.:-0.02612         3rd Qu.:-0.03684         3rd Qu.: 0.6631   3rd Qu.: 0.8208
Max.   :40.19286         Max.   :38.74575         Max.   : 2.6667   Max.   : 1.6593
Water.Intensity..gal.sf.
Min.   :-0.04317
1st Qu.:-0.04314
Median :-0.04310
Mean   : 0.00000
3rd Qu.:-0.04305
Max.   :25.96820
```

Our data contains 25 variables. For this regression analysis, we only took 8 numeric variables for analysis. We created correlation plot bellow. According to the plot, we can found the correlation between Gas and Electricity is close to minus 1, it means they have negative correlations, in this regard, we judge that electricity and gas are alternative energy sources, the main energy types of different companies may be different. Besides, the correlation between kgCO2 and KBTU are very high, We think that these two variables may be two ways of measuring the same energy source. For other variables, the correlation between them is close to 0. For the target variable Energy Score, it don't have high correlation with other variables.



IV. Data Mining Techniques and Implementation

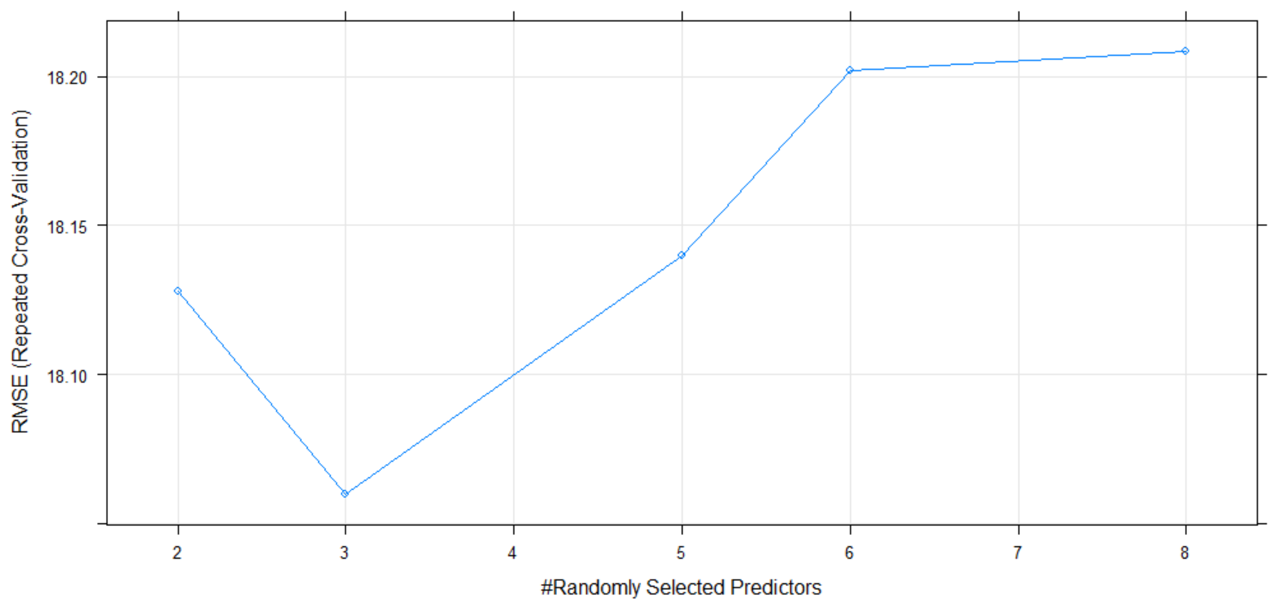
We chose two algorithms for the problem, one is Random Forest and Gradient Boosting Algorithm for predicting the **Energy Star Score.** As the output variable is a continuous variable so we consider this problem as a **Regression Problem.**

Random Forest and Boosting Tree overcome the challenges of **Decision Tree's** overfitting problem, that is, having highly correlated independent variables results in the incorrect variable being selected for splitting the root node.

Random Forest: The Random forest technique uses the decision tree model for parametrization. But it integrates a sampling technique, a subspace method, and an ensemble approach, to optimize the model. The sampling approach is also called bootstrap, which adopts a random sampling approach with replacement.
For the problem, we used **Tuning Algorithm** which helps to control training process and gain better result. For tuning the parameter of Random Forest we used **train function** and for cross validation we used k-fold method with splits the datasets into k sets. Here k=10. In this, we mainly focus on important parameters namely **mtry, trConrol and tuneLength**.

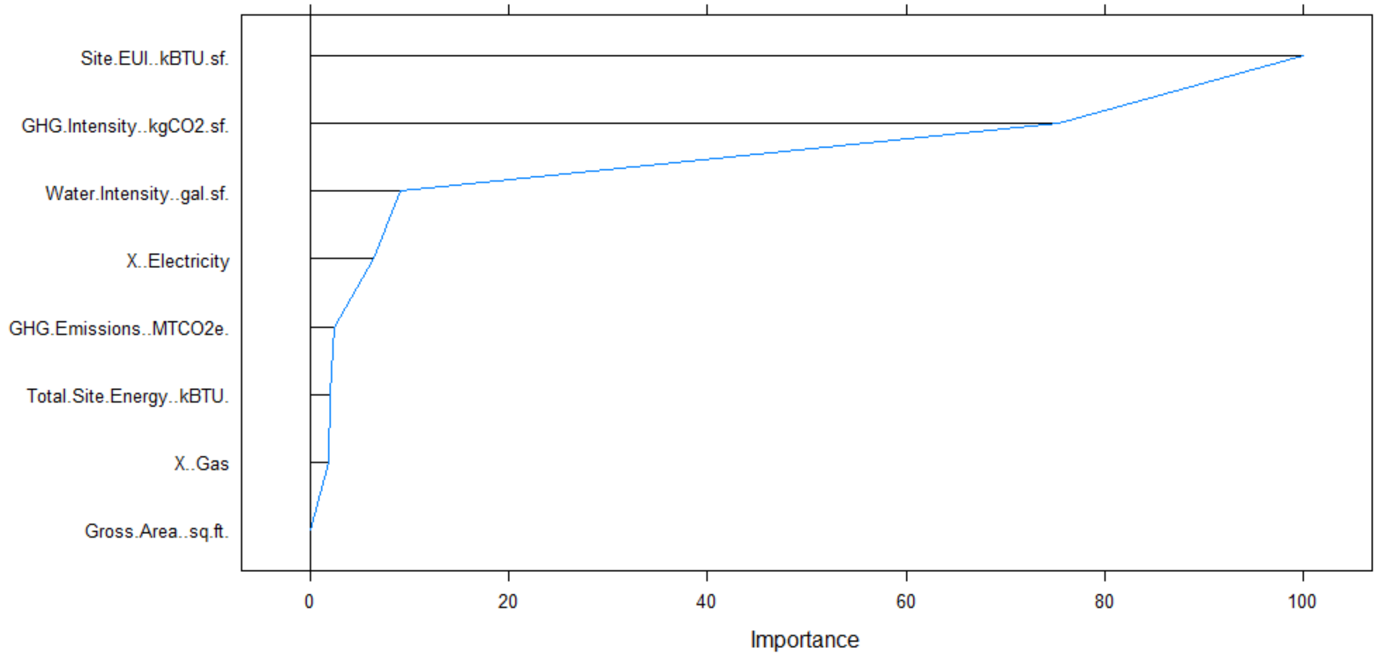
- **mtry**: Number of variables is randomly collected to be sampled at each split time.
- **trControl**: a list of values that defines how this function acts. Here we use k-fold repeated cross validation where k=10.
- **tuneLength**: an integer denoting the number of levels for each tuning parameters.

While tuning the parameters we used Root mean squared error as a metrics to selected the best model. After running the tuning algorithm we got found out that RMSE is least when **mtry** is **3** i.e the number of variables randomly collected to be sampled at each split is 3.



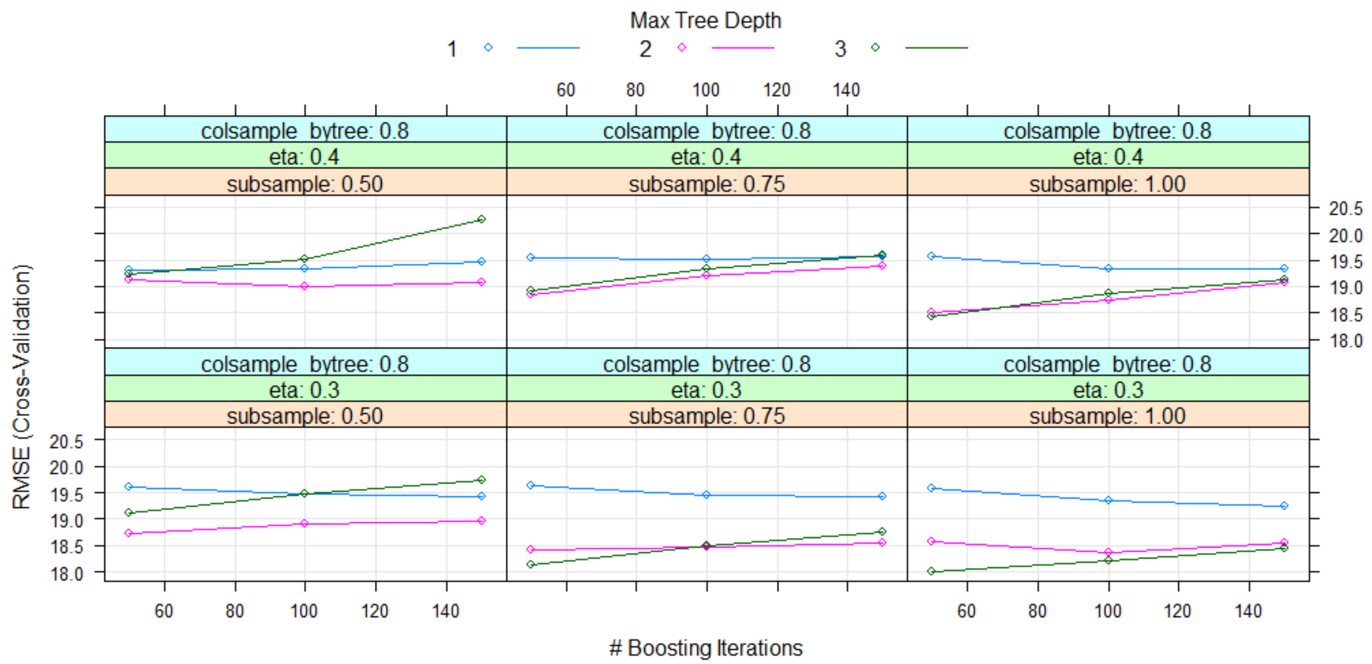

```
Random Forest

977 samples
  8 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 1 times)
Summary of sample sizes: 880, 880, 880, 879, 879, 880, ...
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
  2     17.86921  0.6469711  13.52959
  3     17.78285  0.6488445  13.38693
  5     17.74126  0.6496627  13.28327
  6     17.74572  0.6491137  13.22955
  8     17.72595  0.6494220  13.14669

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 8.
```

The important Variables in the Random Forest are shown:



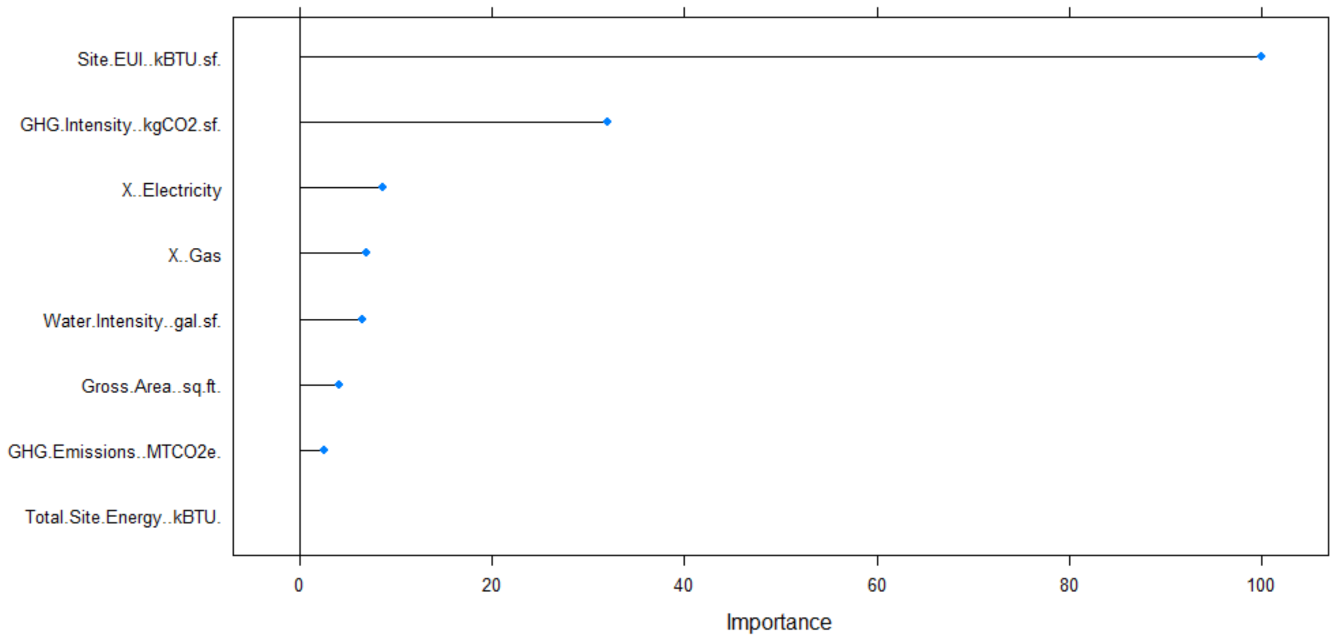
So by using Random Forest Regression, go the Root mean Square error of **18.05990**

Gradient Boosting Tree Algorithm: The main idea of boosting is to add new models to the ensemble sequentially. Ensemble learning offers a systematic solution to combine the predictive power of multiple learners. **Bagging** and **boosting** are two widely used ensemble learners. So we are tuning the Boosting algorithm parameter **eta and sub-sample.** The cross validation method used here is k-fold method with k=10 and the boosting algorithm used is **xgboost.** After tuning and running the algorithm we get this.



The best model is when the parameter **eta=0.3, subsample=1 and colsample_bytree=0.8.**

The important parameters of Gradient Boosting algorithm are:



V. Performance Evaluation

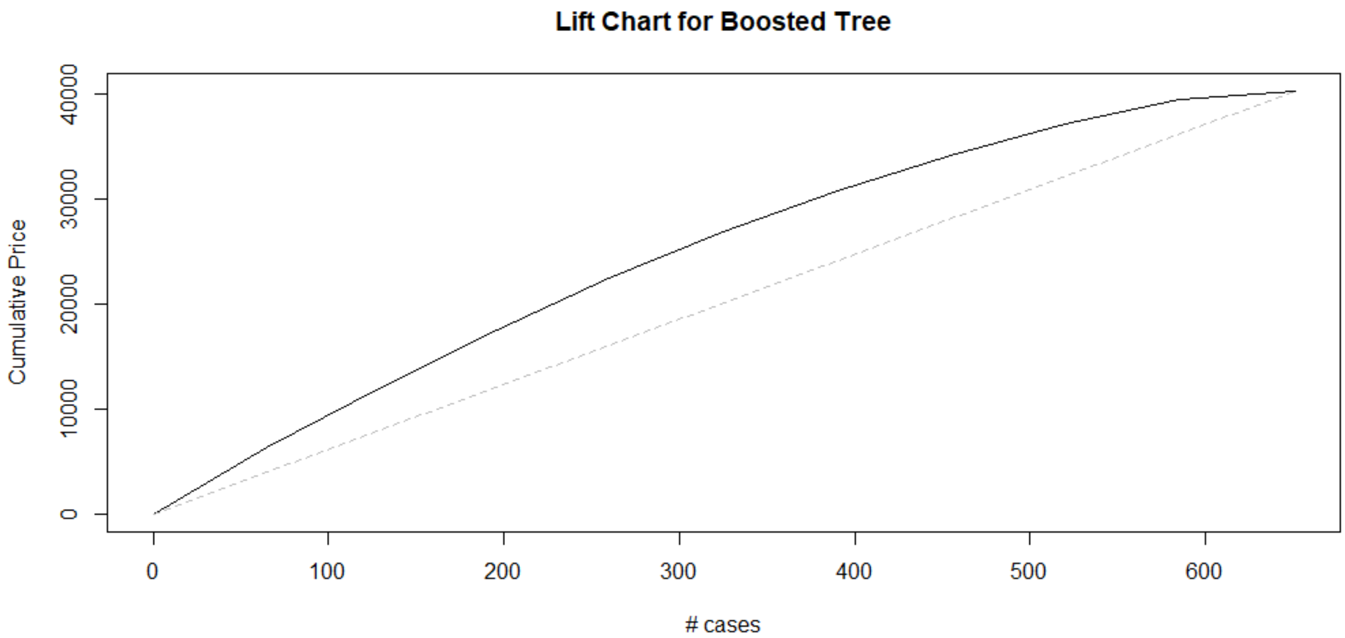
We used Root Mean Squared error, MAPE, Mean Error, Mean Absolute Error along with Lift Charts and Decile Chart for model Performance of the two methods used.

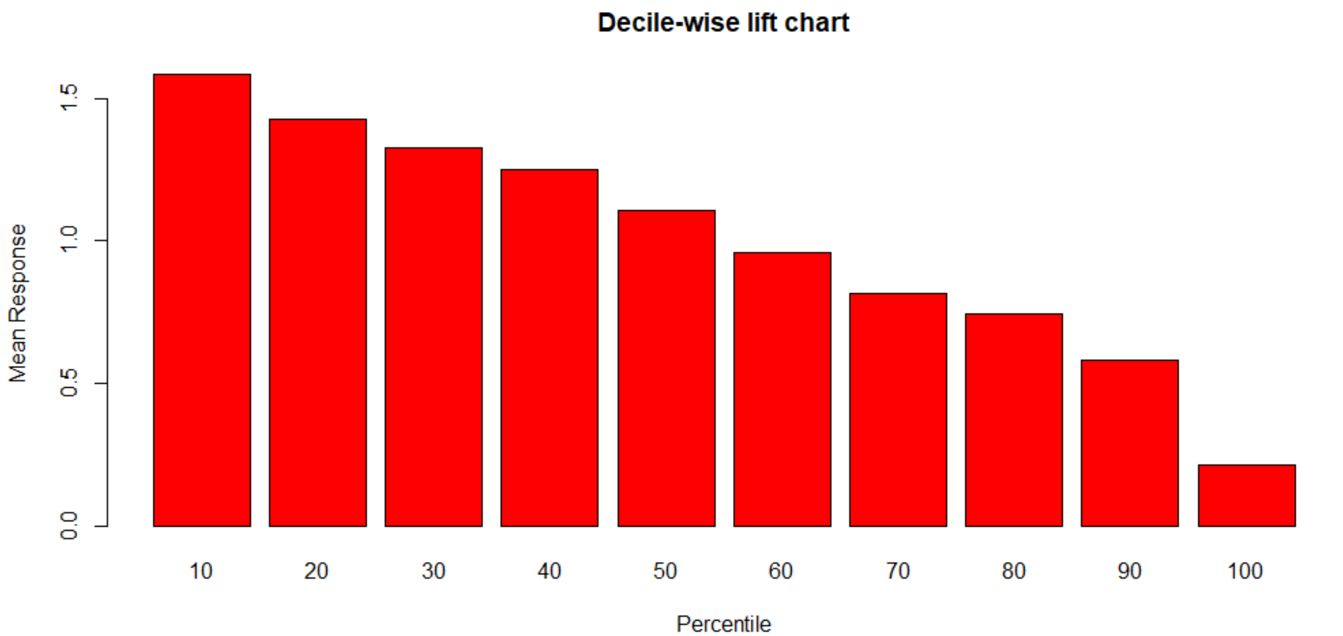
Gradient Boosting Forest:

Accuracy

```
                 ME     RMSE      MAE       MPE     MAPE
Test set -0.1021298 16.19094 11.90085 -89.14044 106.7116
```

Lift Chart and Decile Chart



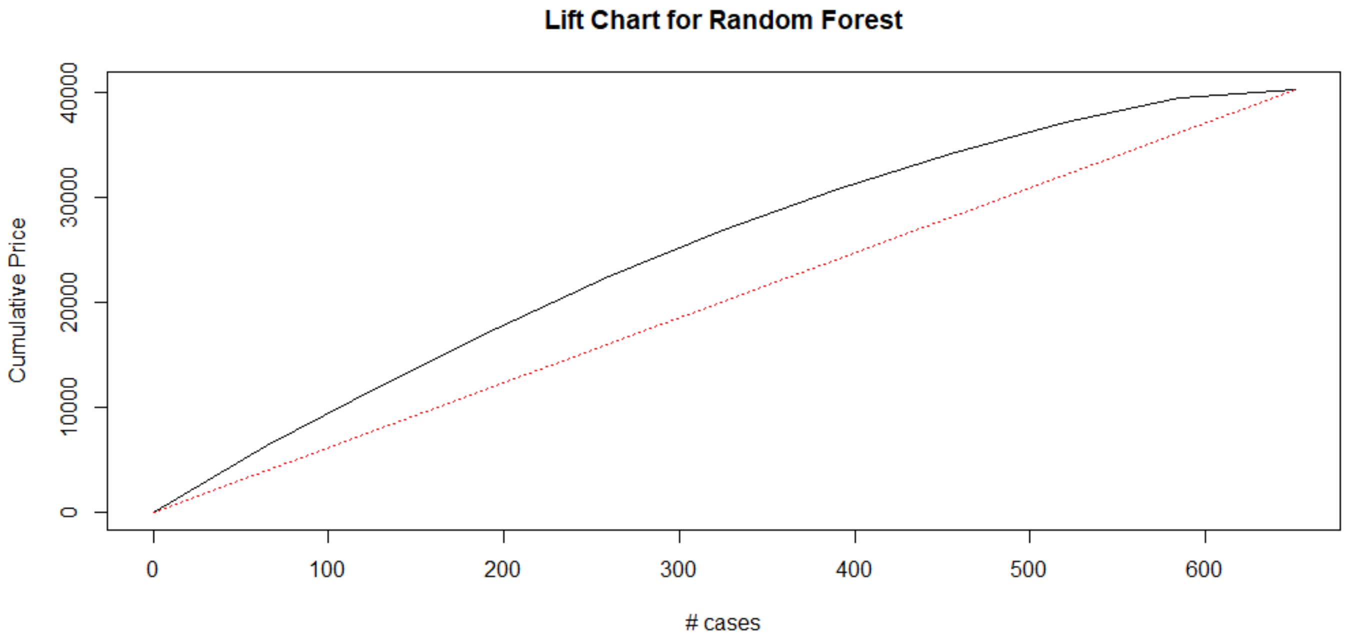


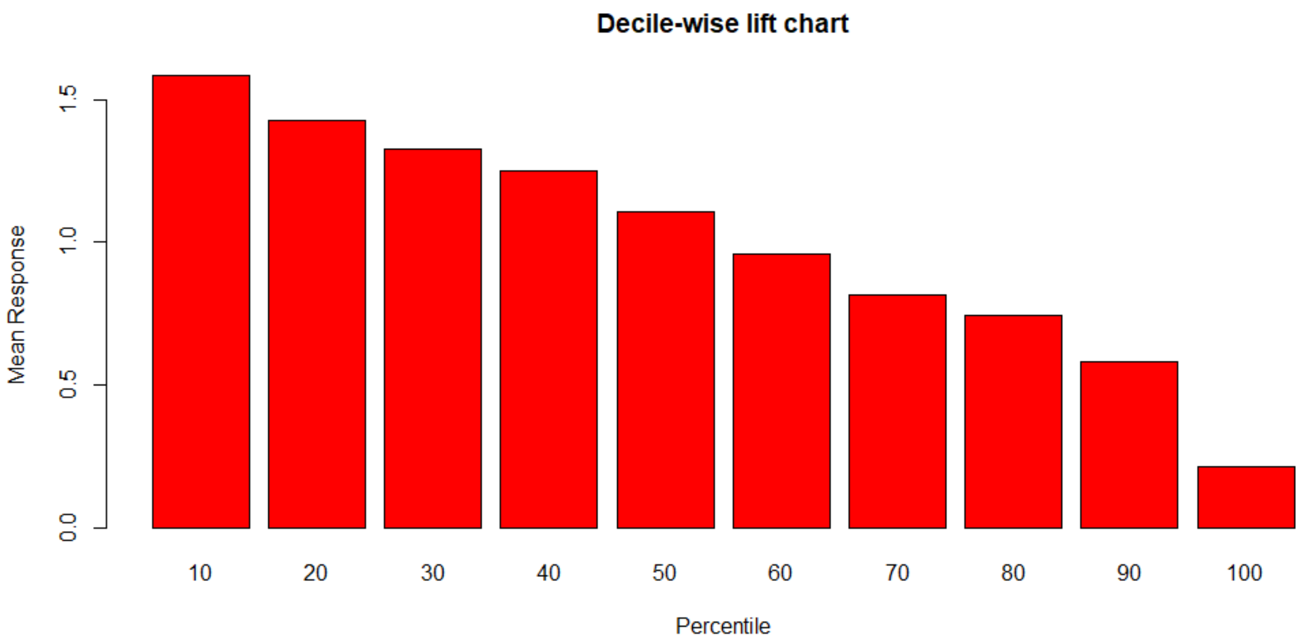
Random Forest

Accuracy

```
                ME     RMSE      MAE       MPE     MAPE
Test set 0.2136858 12.71033 8.872855 -61.68488 73.90248
```

Lift Chart and Decile Chart:





So as we see that the Accuracy and Lift Chart of **Random Forest Algorithm** using Tuning Algorithm is performing better than Gradient Boosting Algorithms.

VI. Discussion and Recommendation

As shown above the Random Forest Regression preformed better than Gradient Boosting Algorithm. Random Forests are fairly easy to tune since there are only a handful of tuning parameter. They provide a very powerful out-of-the-box algorithm that has a great predictive accuracy.

VII. Summary

The Use Case Study has demonstrated the Data Mining techniques, Algorithm, Data visualization, Data Preprocessing and Performance metrics to measure the accuracy of the two data mining techniques used in the case study.

Appendix: R Code for use case study

```
EnergyRating<-read.csv('C:/Users/Kanishk/Downloads/IE Courses/Data Mining/Project/Combine.csv')

EnergyRating<- EnergyRating[ , -c(1 , 2 , 3 , 4 , 5 , 6, 10, 11, 12 ,18, 20, 21 ,22 ,23 ,24 ,25)]#Removing unwanted columns

library(dplyr)
#########################3#Filtering of Datasets####################################
EnergyRating<-EnergyRating %>%
select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GHG.Emissions..MTCO2e.,GHG.Intensity..kgCO2.sf.,
                    Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Intensity..gal.sf.,) %>%
filter(!Energy.Star.Score=='Not Available')

EnergyRating<-EnergyRating %>%
select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GHG.Emissions..MTCO2e.,GHG.Intensity..kgCO2.sf.,
                    Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Intensity..gal.sf.,) %>%
filter(!EnergyRating$Gross.Area..sq.ft.=='Not Available')

EnergyRating<-EnergyRating %>%
select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GHG.Emissions..MTCO2e.,GHG.Intensity..kgCO2.sf.,
                    Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Intensity..gal.sf.,) %>%
filter(!EnergyRating$Site.EUI..kBTU.sf.=='Not Available')

EnergyRating<-EnergyRating %>%
select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GHG.Emissions..MTCO2e.,GHG.Intensity..kgCO2.sf.,
                    Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Intensity..gal.sf.,) %>%
filter(!EnergyRating$GHG.Emissions..MTCO2e.=='Not Available')
```

```
EnergyRating<-EnergyRating %>%
select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GHG.Emissions..MTCO2e.,GHG.Intensity..kgC
O2.sf.,
                          Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Intensity..gal.sf.,) %>%
filter(!EnergyRating$GHG.Intensity..kgCO2.sf.=='Not Available')

EnergyRating<-EnergyRating %>%
select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GHG.Emissions..MTCO2e.,GHG.Intensity..kgC
O2.sf.,
                          Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Intensity..gal.sf.,) %>%
filter(!EnergyRating$Total.Site.Energy..kBTU.=='Not Available')

EnergyRating<-EnergyRating %>%
select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GHG.Emissions..MTCO2e.,GHG.Intensity..kgC
O2.sf.,
                          Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Intensity..gal.sf.,) %>%
filter(!EnergyRating$X..Electricity=='Not Available')

EnergyRating<-EnergyRating %>%
select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GHG.Emissions..MTCO2e.,GHG.Intensity..kgC
O2.sf.,
                          Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Intensity..gal.sf.,) %>%
filter(!EnergyRating$X..Gas=='Not Available')

EnergyRating<-EnergyRating %>%
select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GHG.Emissions..MTCO2e.,GHG.Intensity..kgC
O2.sf.,
                          Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Intensity..gal.sf.,) %>%
filter(!EnergyRating$Water.Intensity..gal.sf.=='Not Available')

EnergyRating<-EnergyRating %>%
select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GHG.Emissions..MTCO2e.,GHG.Intensity..kgC
O2.sf.,
                          Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Intensity..gal.sf.,) %>%
filter(!EnergyRating$Gross.Area..sq.ft.=='Not Available')

##########Converting Datasets to numeric data type######################

EnergyRating$Gross.Area..sq.ft.<-as.numeric(as.character(EnergyRating$Gross.Area..sq.ft.))
EnergyRating$Site.EUI..kBTU.sf.<-as.numeric(as.character(EnergyRating$Site.EUI..kBTU.sf.))
EnergyRating$Energy.Star.Score<-as.numeric(as.character(EnergyRating$Energy.Star.Score))
EnergyRating$GHG.Emissions..MTCO2e.<-
as.numeric(as.character(EnergyRating$GHG.Emissions..MTCO2e.))
EnergyRating$GHG.Intensity..kgCO2.sf.<-as.numeric(as.character(EnergyRating$GHG.Intensity..kgCO2.sf.))
EnergyRating$Total.Site.Energy..kBTU.<-as.numeric(as.character(EnergyRating$Total.Site.Energy..kBTU.))
EnergyRating$X..Electricity<-as.numeric(as.character(EnergyRating$X..Electricity))
EnergyRating$X..Gas<-as.numeric(as.character(EnergyRating$X..Gas))
EnergyRating$Water.Intensity..gal.sf.<-as.numeric(as.character(EnergyRating$Water.Intensity..gal.sf.))
summary(EnergyRating)
```

```
###############3Visualize Missing Value in Matrix#################
library(dplyr)
library(wakefield)
missmap(EnergyRating)
library(naniar)
gg_miss_var(EnergyRating)

######################Removing all the na values################################
EnergyRating<-EnergyRating %>% filter(!is.na(Energy.Star.Score))
EnergyRating<-EnergyRating %>% filter(!is.na(Gross.Area..sq.ft.))
EnergyRating<-EnergyRating %>% filter(!is.na(Site.EUI..kBTU.sf.))
EnergyRating<-EnergyRating %>% filter(!is.na(GHG.Emissions..MTCO2e.))
EnergyRating<-EnergyRating %>% filter(!is.na(GHG.Intensity..kgCO2.sf.))
EnergyRating<-EnergyRating %>% filter(!is.na(Total.Site.Energy..kBTU.))
EnergyRating<-EnergyRating %>% filter(!is.na(X..Electricity))
EnergyRating<-EnergyRating %>% filter(!is.na(X..Gas))
EnergyRating<-EnergyRating %>% filter(!is.na(Water.Intensity..gal.sf.))

##########Visualizing HeatMap of correlation Matrix####################3
library(ggcorrplot)
library(reshape2)

qplot(x=Var1, y=Var2, data=melt(cor(EnergyRating)), fill=value, geom="tile")+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal()+ # minimal theme
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed()

#################################Pre-Processing##################
StandardScale <- function(x){
  return((x-mean(x))/sd(x))
}

EnergyRating.norm<-EnergyRating
EnergyRating.norm[,c(1:2,4:9)]<-data.frame(lapply(EnergyRating[,c(1:2,4:9)],FUN =StandardScale))
train.index <- sample(c(1:dim(EnergyRating.norm)[1]), dim(EnergyRating.norm)[1]*0.6)
train.df <- EnergyRating.norm[train.index, ]
valid.index <- sample(c(1:dim(EnergyRating.norm)[1]), dim(EnergyRating.norm)[1]*0.4)
valid.df<-EnergyRating.norm[valid.index,]
summary(EnergyRating.norm)

######use k-fold cross validation and Random Forest Regression#############################

library(randomForest)
set.seed(131)
library(caret)
k_10_fold<-trainControl(method = "repeatedcv",number=10,savePredictions = TRUE)
```

```
#Tunning the parameters for Random Forest Algorithm
model_fitted <-train(Energy.Star.Score
~Gross.Area..sq.ft.+Site.EUI..kBTU.sf.+GHG.Emissions..MTCO2e.+GHG.Intensity..kgCO2.sf.+
                      Total.Site.Energy..kBTU.+X..Electricity+X..Gas+Water.Intensity..gal.sf., data=train.df, family
                    = identity,trControl = k_10_fold, tuneLength =5)

print(model_fitted)

##################XG Boosting Algorithm###########################
set.seed(123)
model <- train(
  Energy.Star.Score
~Gross.Area..sq.ft.+Site.EUI..kBTU.sf.+GHG.Emissions..MTCO2e.+GHG.Intensity..kgCO2.sf.+Total.Site.Ene
rgy..kBTU.
  +X..Electricity+X..Gas+Water.Intensity..gal.sf., data = train.df, method = "xgbTree",
  trControl = trainControl("cv", number = 10)
)
plot(varImp(model))
plot(model)

#######################Predicting the model######################
Predict_valid_rf<-predict(model_fitted,valid.df)
Predict_valid_xgb<-predict(model,valid.df)

###########################Result Interpretation#####################

library(forecast)
accuracy(Predict_valid_rf,valid.df$Energy.Star.Score) #Random_Forest_Regression
accuracy(Predict_valid_xgb,valid.df$Energy.Star.Score) #XG Gradient Boosting Algorithm

###############Lift Charts################

library(gains)
gain <- gains(valid.df$Energy.Star.Score[!is.na(Predict_valid_rf)], Predict_valid_rf[!is.na(Predict_valid_rf)])

rating <- valid.df$Energy.Star.Score[!is.na(valid.df$Energy.Star.Score)]
plot(c(0,gain$cume.pct.of.total*sum(rating))~c(0,gain$cume.obs),
     xlab="# cases", ylab="Cumulative Price", main="Lift Chart for Random Forest", type="l")
lines(c(0,sum(rating))~c(0,dim(valid.df)[1]), col="red", lty=3)

#####################Decile Chart#################

barplot(gain$mean.resp/mean(rating), names.arg = gain$depth,
        xlab = "Percentile", ylab = "Mean Response", main = "Decile-wise lift chart",col=c("red"))

###############Lift Charts################
library(gains)
gain <- gains(valid.df$Energy.Star.Score[!is.na(Predict_valid_xgb)],
Predict_valid_xgb[!is.na(Predict_valid_xgb)])
rating <- valid.df$Energy.Star.Score[!is.na(valid.df$Energy.Star.Score)]
plot(c(0,gain$cume.pct.of.total*sum(rating))~c(0,gain$cume.obs),
```

```
    xlab="# cases", ylab="Cumulative Price", main="Lift Chart for Boosted Tree", type="l")
lines(c(0,sum(rating))~c(0,dim(valid.df)[1]), col="gray", lty=2)

barplot(gain$mean.resp/mean(rating), names.arg = gain$depth,
        xlab = "Percentile", ylab = "Mean Response", main = "Decile-wise lift chart",col=c("red"))
accuracy(Predict_valid_xgb,valid.df$Energy.Star.Score)
```