Final-Project.R

Kanishk

2020-04-20

```
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)
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

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

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

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

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

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

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

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

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

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

EnergyRating<-EnergyRating %>% select(Gross.Area..sq.ft.,Site.EUI..kBTU.sf.,Energy.Star.Score,GH
G.Emissions..MTCO2e.,GHG.Intensity..kgCO2.sf.,
                                     Total.Site.Energy..kBTU.,X..Electricity,X..Gas,Water.Inten
sity..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))
```

```
## Warning: NAs introduced by coercion
```

```
EnergyRating$GHG.Emissions..MTCO2e.<-as.numeric(as.character(EnergyRating$GHG.Emissions..MTCO2
e.))
EnergyRating$GHG.Intensity..kgCO2.sf.<-as.numeric(as.character(EnergyRating$GHG.Intensity..kgCO
2.sf.))
```

```
## Warning: NAs introduced by coercion
```

```
EnergyRating$Total.Site.Energy..kBTU.<-as.numeric(as.character(EnergyRating$Total.Site.Energy..k
BTU.))
```

```
## Warning: NAs introduced by coercion
```

```
EnergyRating$X..Electricity<-as.numeric(as.character(EnergyRating$X..Electricity))
```

```
## Warning: NAs introduced by coercion
```

```
EnergyRating$X..Gas<-as.numeric(as.character(EnergyRating$X..Gas))
```

```
## Warning: NAs introduced by coercion
```

```
EnergyRating$Water.Intensity..gal.sf.<-as.numeric(as.character(EnergyRating$Water.Intensity..ga
l.sf.))
summary(EnergyRating)
```

```
##  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  
##  Min.   :   -0.60         Min.   :0.000e+00        Min.   :0.0000  
##  1st Qu.:    3.40         1st Qu.:2.604e+06        1st Qu.:0.2263  
##  Median :    4.90         Median :5.742e+06        Median :0.4068  
##  Mean   :   44.14         Mean   :3.926e+07        Mean   :0.4522  
##  3rd Qu.:    7.00         3rd Qu.:1.489e+07        3rd Qu.:0.6303  
##  Max.   :38485.10         Max.   :1.966e+10        Max.   :1.0000  
##  NA's   :30               NA's   :1834             NA's   :2108    
##      X..Gas       Water.Intensity..gal.sf.
##  Min.   :0.0000   Min.   :       0        
##  1st Qu.:0.3516   1st Qu.:      10        
##  Median :0.5789   Median :      23        
##  Mean   :0.5460   Mean   :   25284        
##  3rd Qu.:0.7707   3rd Qu.:      44        
##  Max.   :1.0000   Max.   :60595650        
##  NA's   :2613     NA's   :298             
```

```
#Visualize Missing Value in Matrix
library(dplyr)
library(wakefield)
```

```
## Warning: package 'wakefield' was built under R version 3.6.3
```

```
## 
## Attaching package: 'wakefield'
```

```
## The following object is masked from 'package:dplyr':
## 
##     id
```

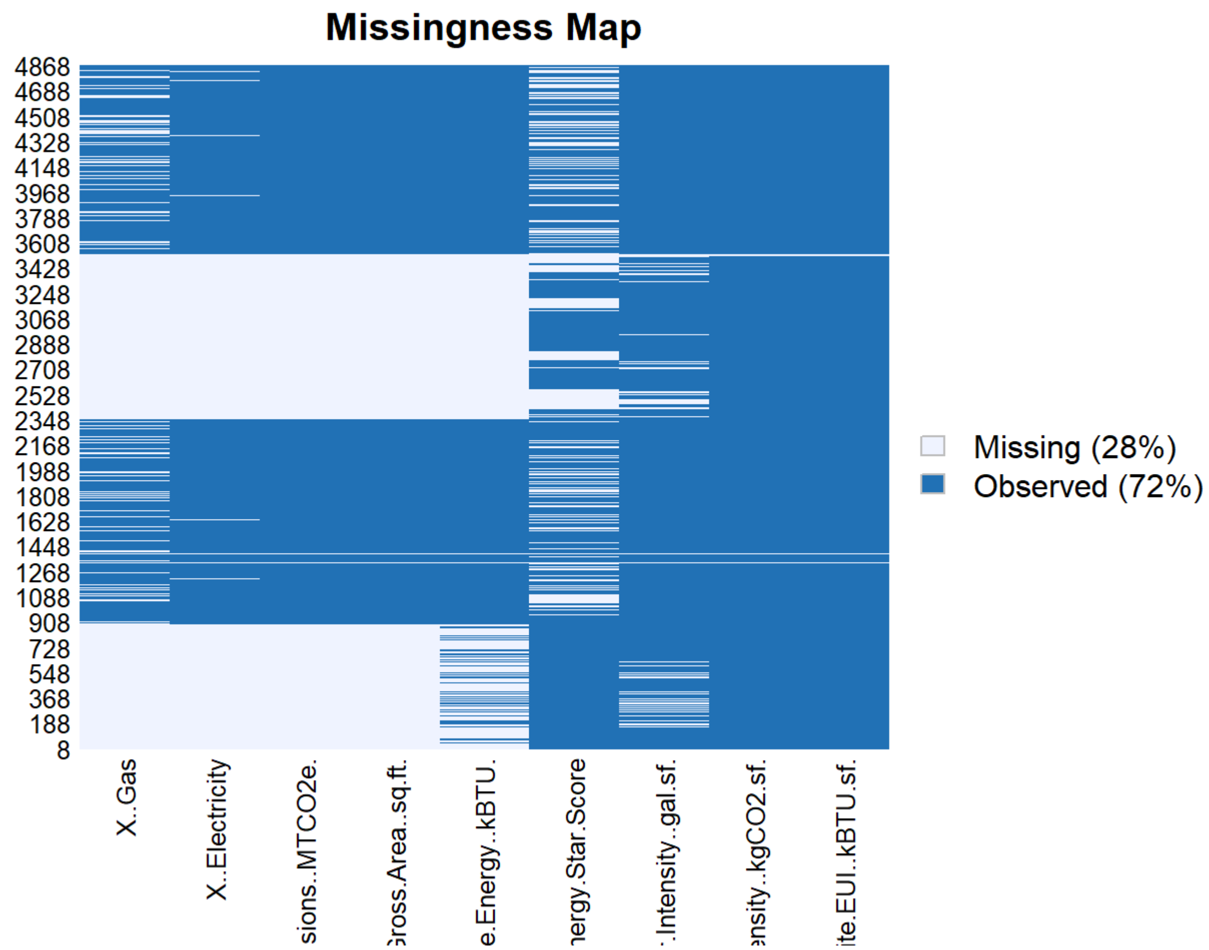
```
library(Amelia)
```

```
## Warning: package 'Amelia' was built under R version 3.6.3
```

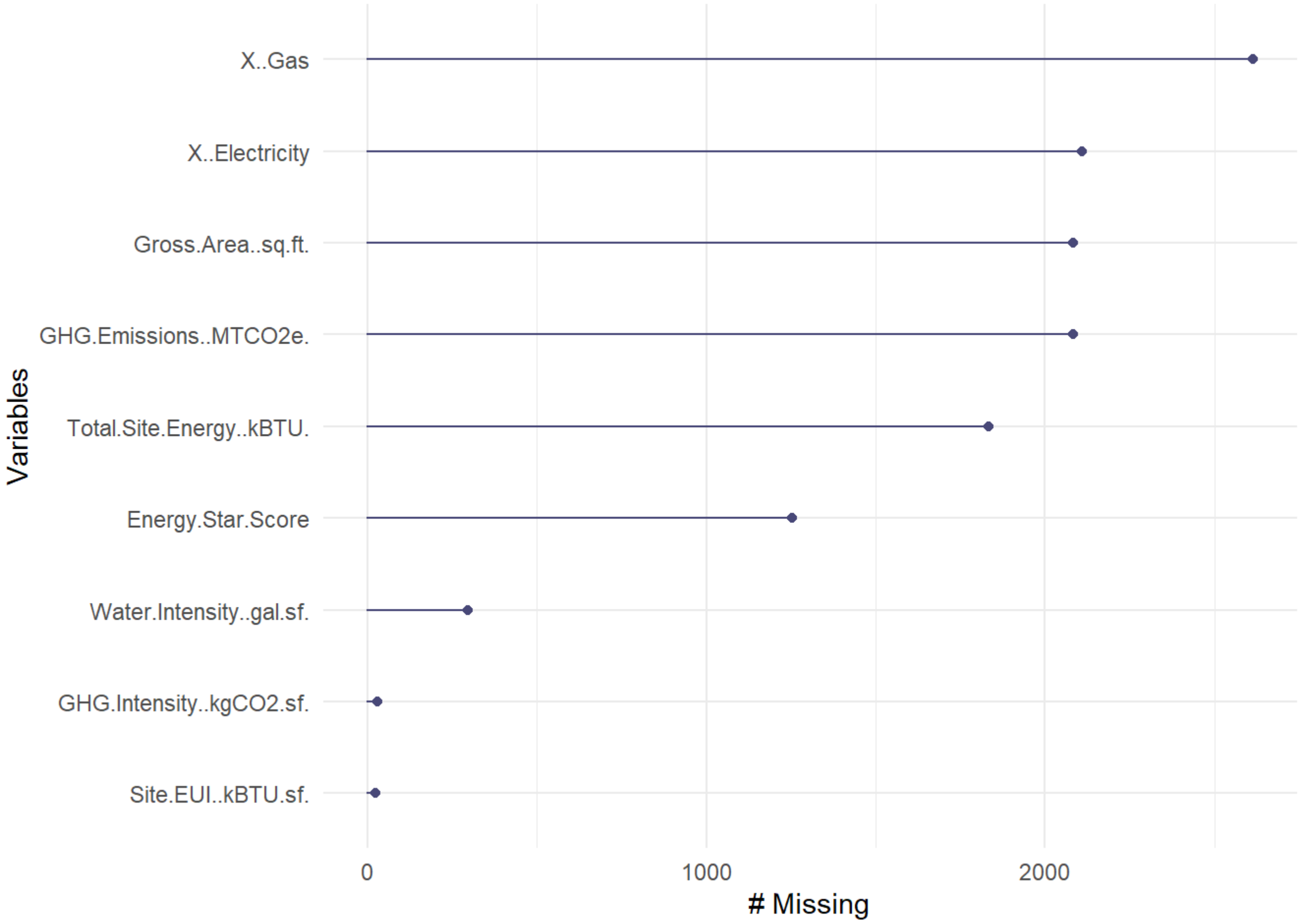
```
## Loading required package: Rcpp
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.6, built: 2019-11-24)
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
missmap(EnergyRating)
## Visualize propotion Missing datasets
library(naniar)
```

```
## Warning: package 'naniar' was built under R version 3.6.3
```



```
gg_miss_var(EnergyRating)
```



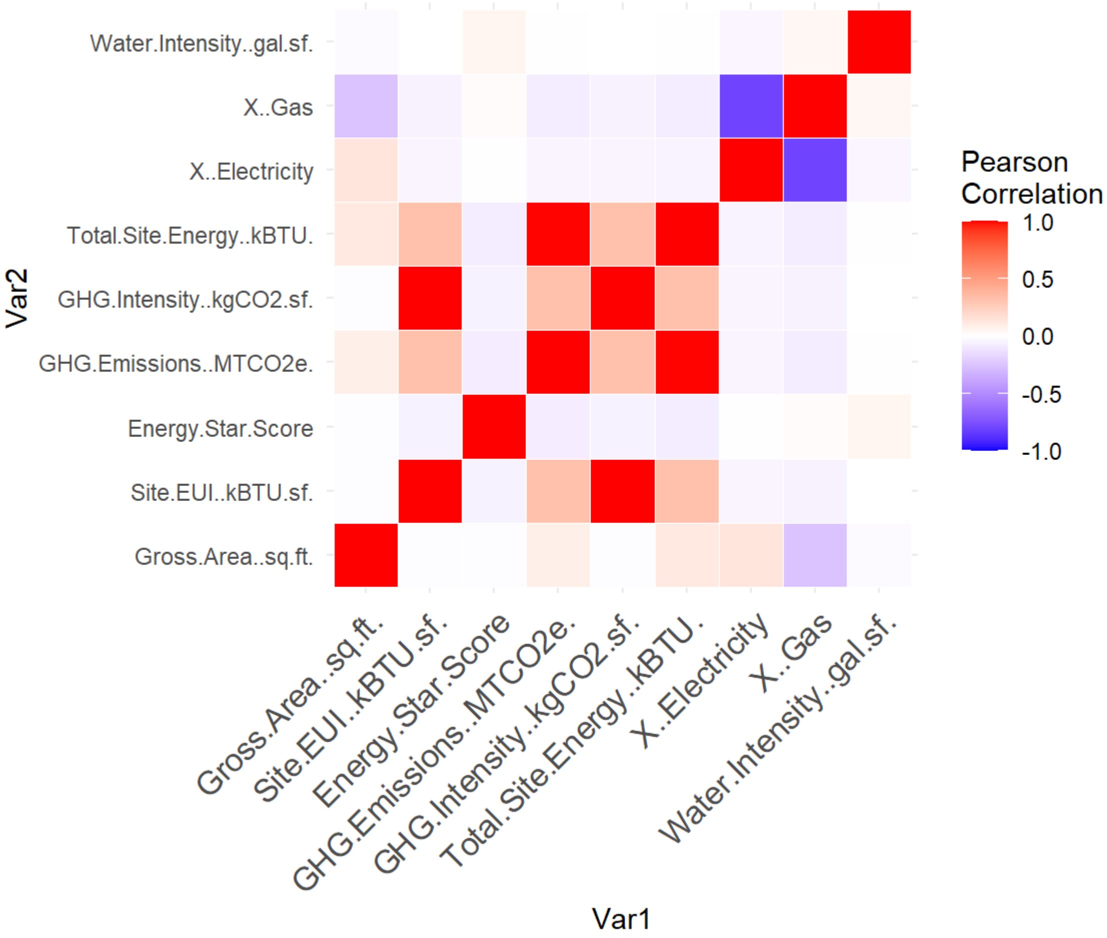
```
#Removing all the na values
EnergyRating<-EnergyRating %>% filter(!is.na(Energy.Star.Score))
EnergyRating<-EnergyRating %>% filter(!is.na(Gross.Area..sq.ft.))
EnergyRating<-EnergyRating %>% filter(!is.na(Site.EUI..kBTU.sf.))
EnergyRating<-EnergyRating %>% filter(!is.na(GHG.Emissions..MTCO2e.))
EnergyRating<-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
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 3.6.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
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)
```

```
##  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
##  Min.   :-0.03228         Min.   :-0.06981         Min.   :-1.6921
##  1st Qu.:-0.02873         1st Qu.:-0.06209         1st Qu.:-0.8117
##  Median :-0.02768         Median :-0.05505         Median :-0.1396
##  Mean   : 0.00000         Mean   : 0.00000         Mean   : 0.0000
##  3rd Qu.:-0.02612         3rd Qu.:-0.03684         3rd Qu.: 0.6631
##  Max.   :40.19286         Max.   :38.74575         Max.   : 2.6667
##      X..Gas        Water.Intensity..gal.sf.
##  Min.   :-2.0551   Min.   :-0.04317
##  1st Qu.:-0.6999   1st Qu.:-0.04314
##  Median : 0.1495   Median :-0.04310
##  Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.8208   3rd Qu.:-0.04305
##  Max.   : 1.6593   Max.   :25.96820
```

```
#use k-fold cross validation and Random Forest Regression
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.6.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

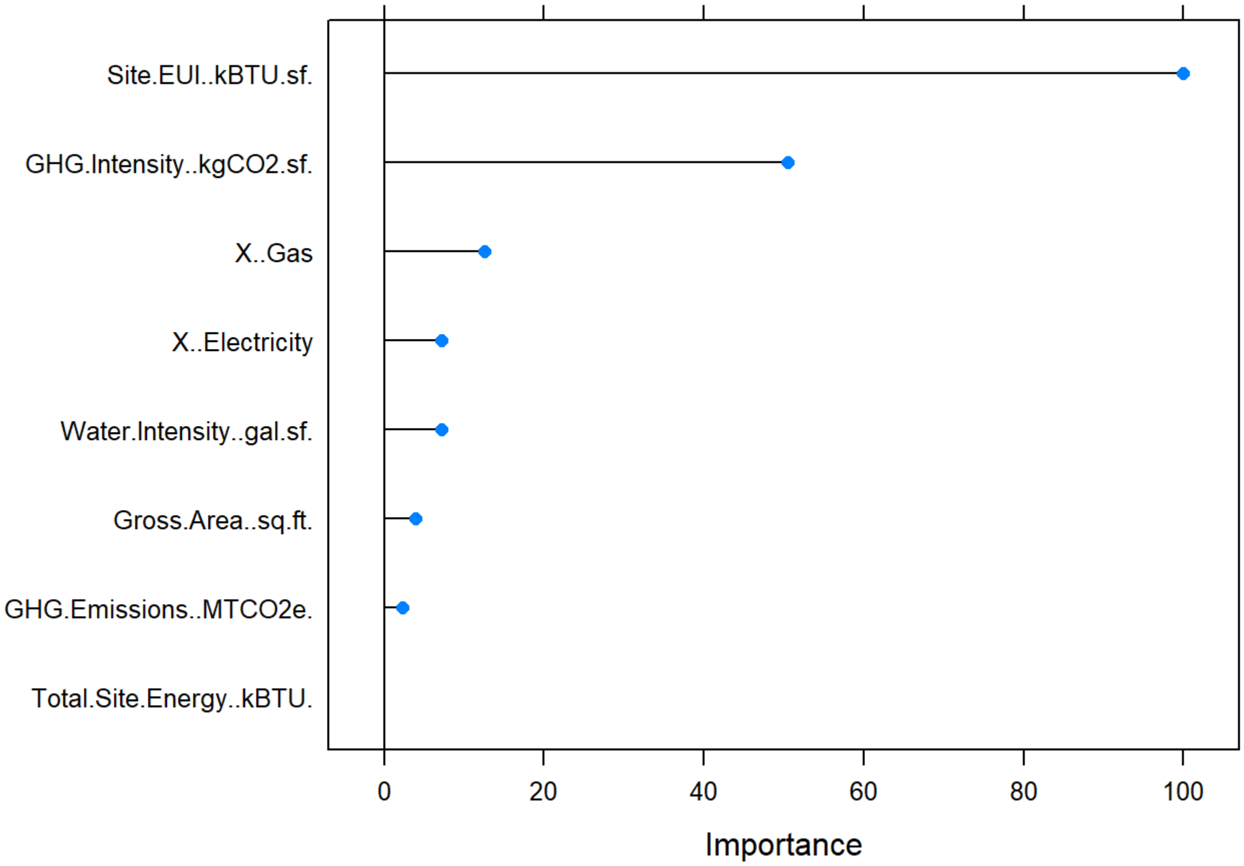
```
set.seed(131)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.3
## Loading required package: lattice
```

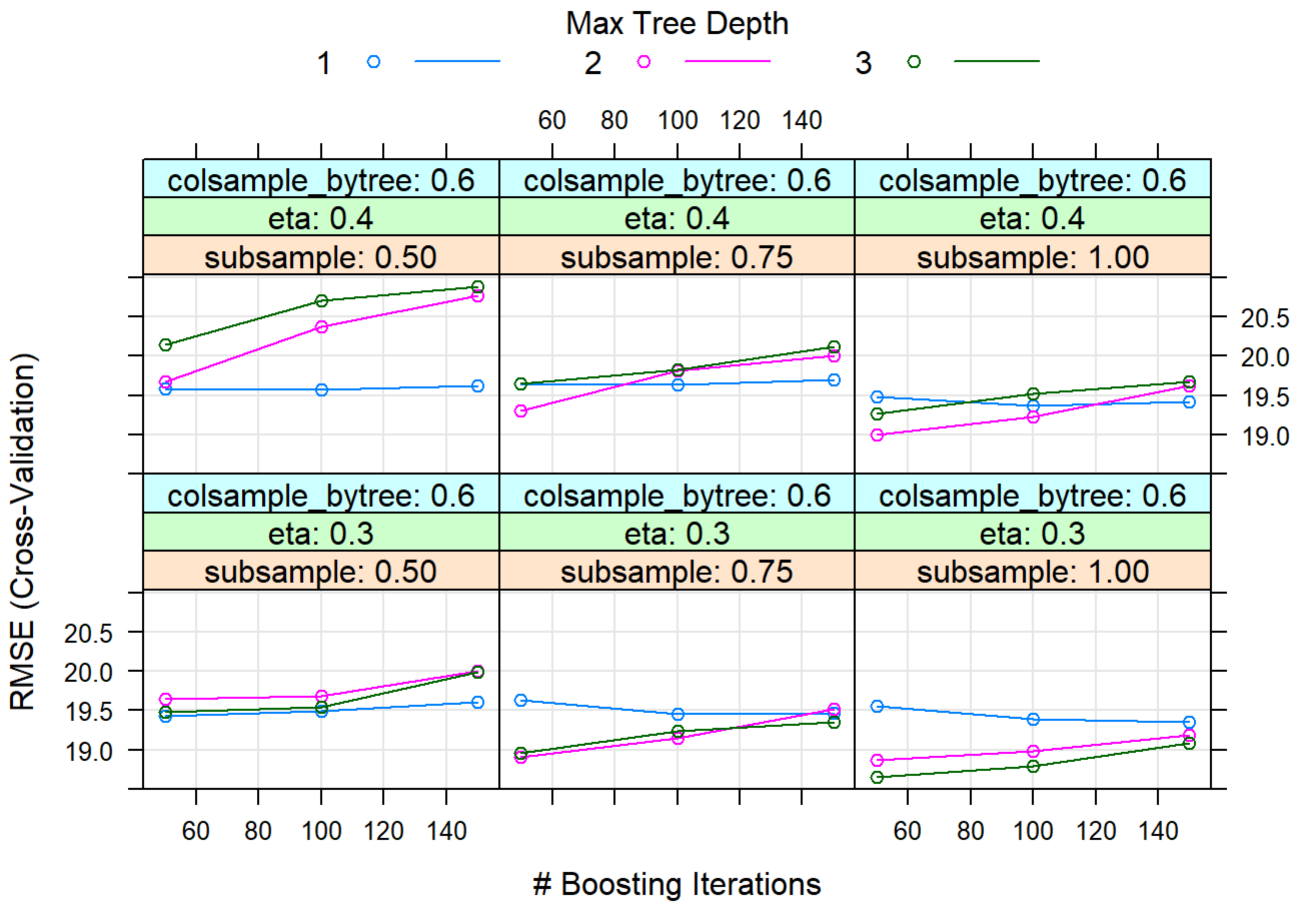
```
k_10_fold<-trainControl(method = "repeatedcv",number=10,savePredictions = TRUE)
#Tunning the parameters for Random Forest Algorithm
model_fitted <-train(Energy.Star.Score ~Gross.Area..sq.ft.+Site.EUI..kBTU.sf.+GHG.Emissions..MTC
O2e.+GHG.Intensity..kgCO2.sf.+
                       Total.Site.Energy..kBTU.+X..Electricity+X..Gas+Water.Intensity..gal.sf.,
 data=train.df, family
                     = identity,trControl = k_10_fold, tuneLength =5)

print(model_fitted)
```

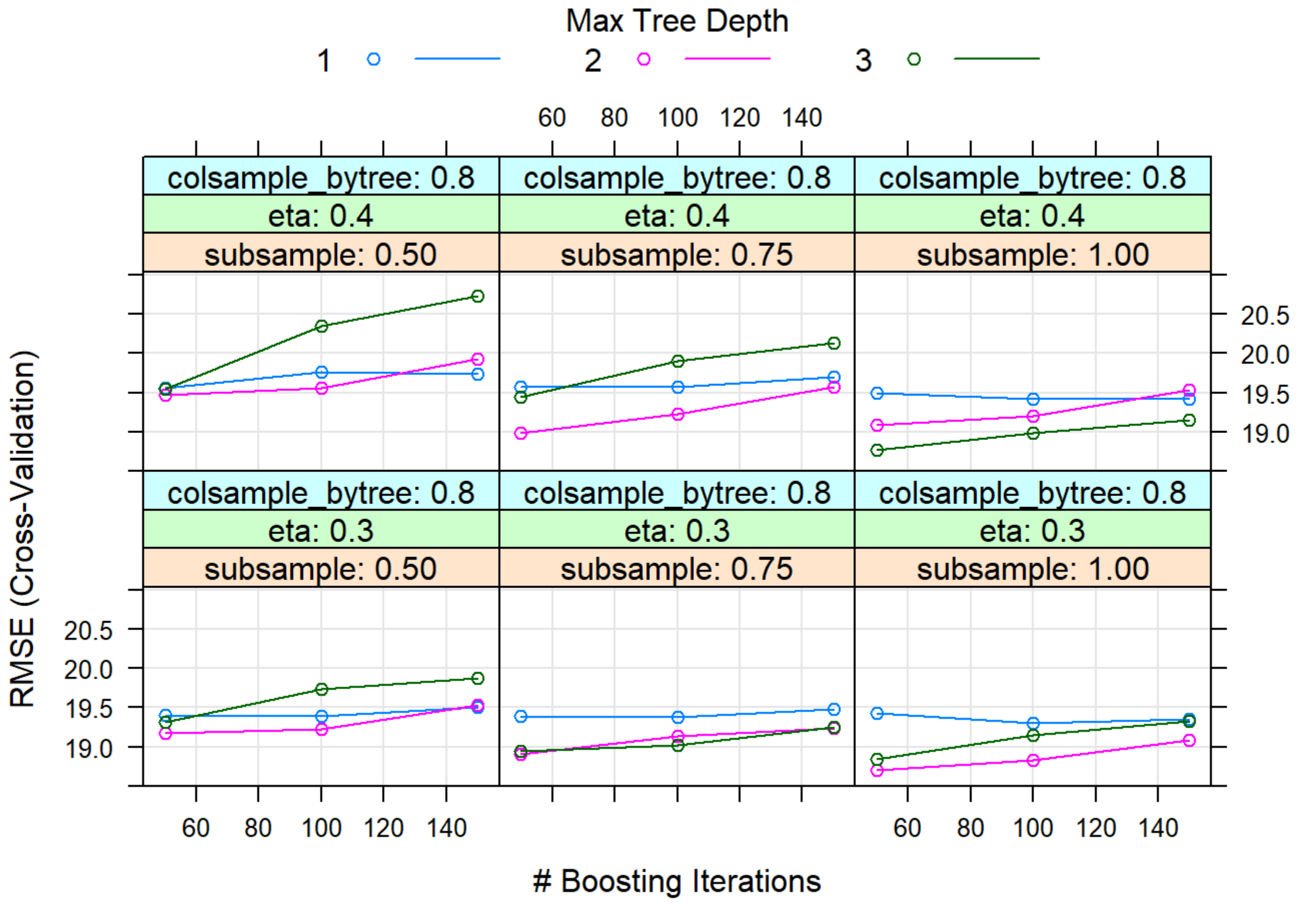
```
## Random Forest
##
## 977 samples
##   8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 879, 879, 881, 878, 880, 879, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##   2     17.75913  0.6344741  13.57877
##   3     17.57440  0.6416219  13.27619
##   5     17.44720  0.6467705  13.05776
##   6     17.46384  0.6467152  13.04445
##   8     17.49625  0.6453840  13.02921
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 5.
```

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


Site.EUI..kBTU.sf.
GHG.Intensity..kgCO2.sf.
X..Gas
X..Electricity
Water.Intensity..gal.sf.
Gross.Area..sq.ft.
GHG.Emissions..MTCO2e.
Total.Site.Energy..kBTU.
0
20
40
60
80
100
Importance

```
plot(model)
```


Max Tree Depth
1
2
3
colsample_bytree: 0.6
eta: 0.4
subsample: 0.50
subsample: 0.75
subsample: 1.00
eta: 0.3
RMSE (Cross-Validation)
20.5
20.0
19.5
19.0
60
80
100
120
140
# Boosting Iterations


Max Tree Depth
1
2
3
colsample_bytree: 0.8
eta: 0.4
subsample: 0.50
subsample: 0.75
subsample: 1.00
eta: 0.3
RMSE (Cross-Validation)
20.5
20.0
19.5
19.0
60
80
100
120
140
# Boosting Iterations

```
#Predicting the model
Predict_valid_rf<-predict(model_fitted,valid.df)
Predict_valid_xgb<-predict(model,valid.df)
#Result Interpretation
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

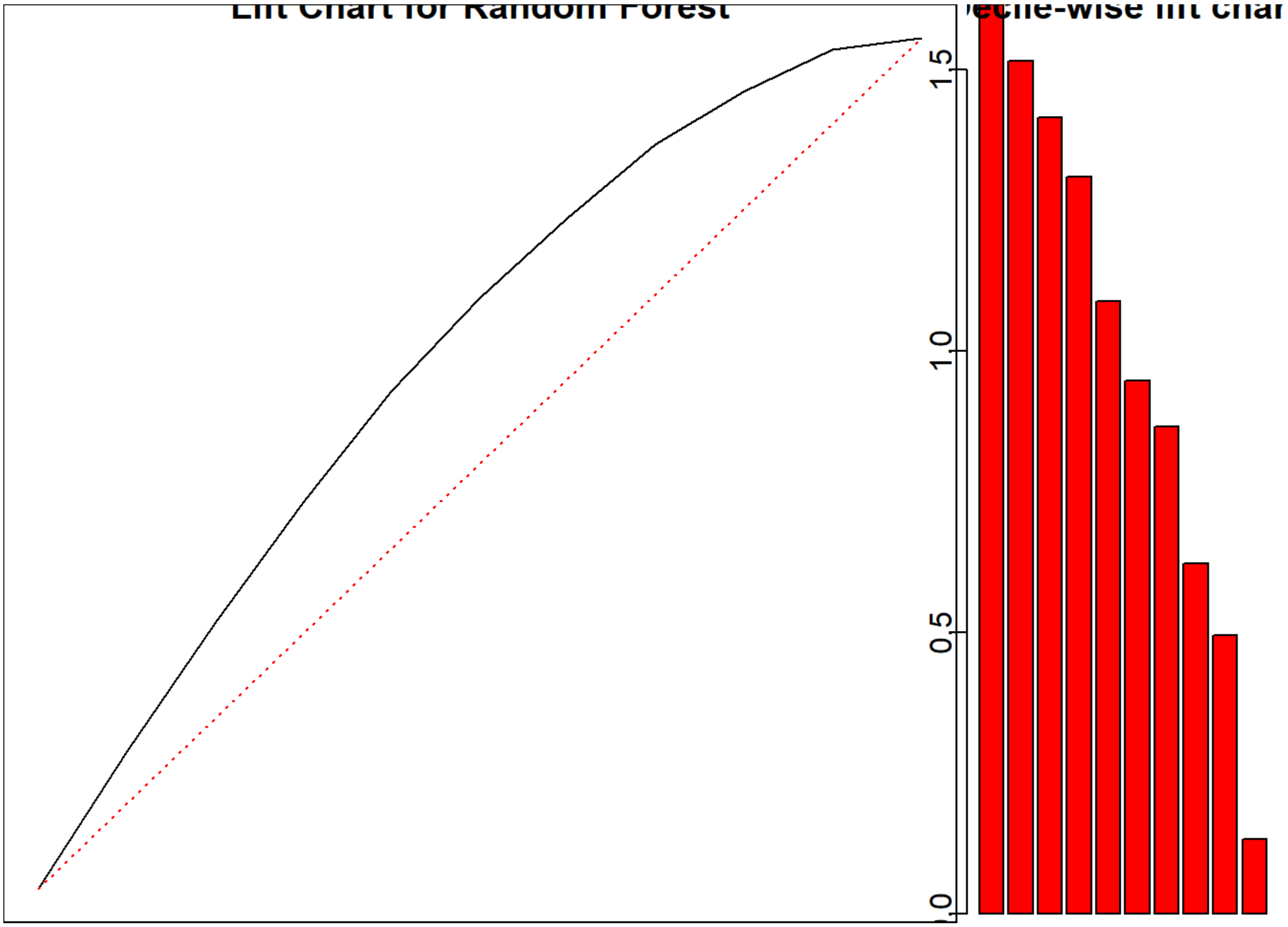
```
accuracy(Predict_valid_rf,valid.df$Energy.Star.Score) #Random_Forest_Regression
```

```
##                   ME     RMSE      MAE       MPE     MAPE
## Test set -0.07351628 13.67206 9.076902 -85.60907 97.51587
```

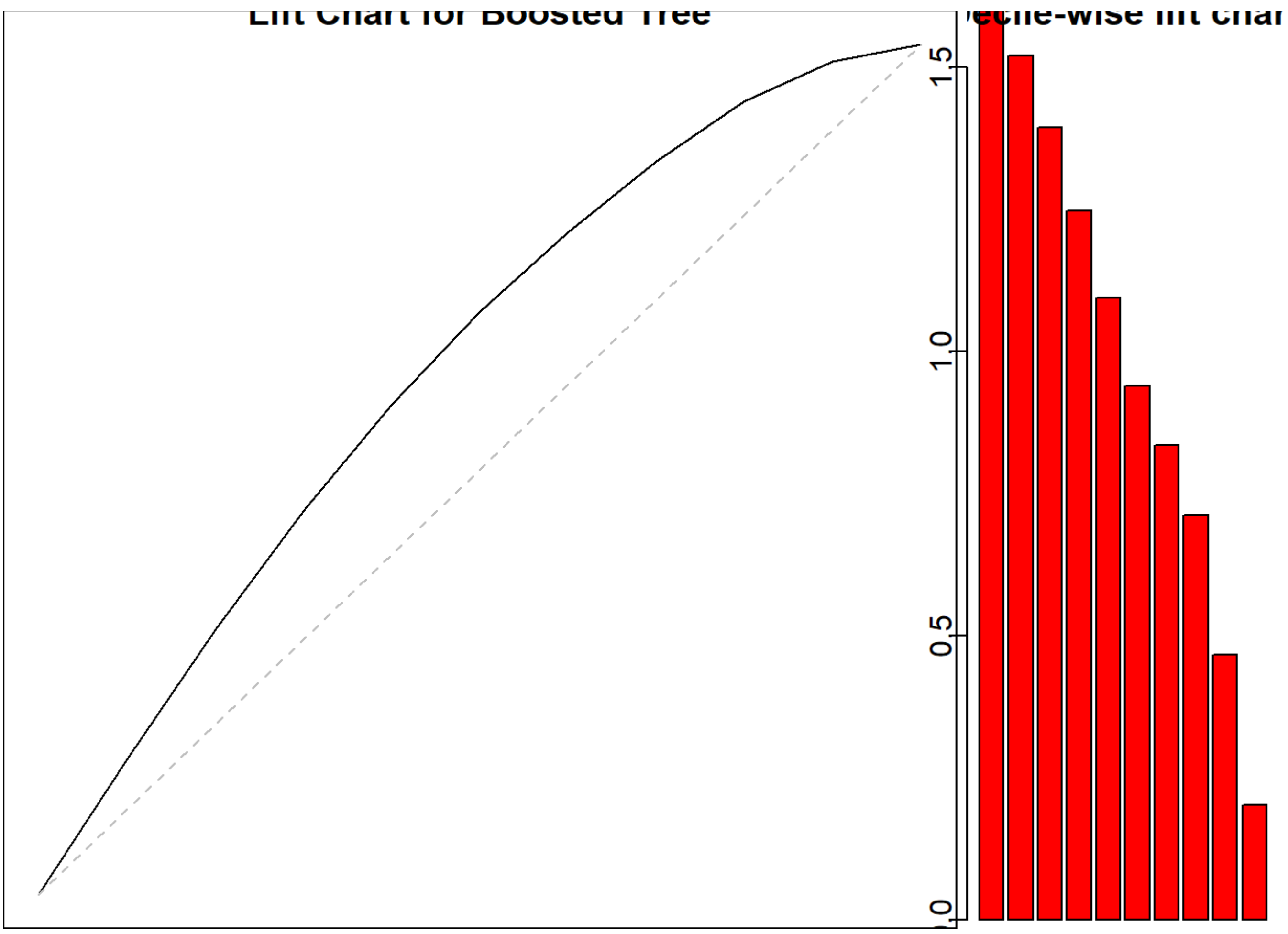
```
accuracy(Predict_valid_xgb,valid.df$Energy.Star.Score) #XG Gradient Boosting Algorithm
```

```
##                  ME     RMSE      MAE       MPE     MAPE
## Test set -0.2181431 16.17104 11.65232 -108.9711 124.7968
```

```
#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$Energy.Star.Score[!is.na(Predict_valid_xgb)], Predict_valid_xgb[!is.na(Pr
edict_valid_xgb)])
rating <- valid.df$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)
```

```
##                   ME     RMSE      MAE       MPE     MAPE
## Test set -0.2181431 16.17104 11.65232 -108.9711 124.7968
```