Citibike_ML

Hongyang Yang

12/4/2017

Purpose: Is there a linear relationship bewteen the number of rents per day and the weather data in NYC?

```
library(ggplot2)
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-13
```

```
library(ISLR)
library(tree)
library(randomForest)
```

```
## randomForest 4.6-12

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

##
## Attaching package: 'randomForest'

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

```
library(e1071)
library(MASS)
library(caret)
```

```
## Loading required package: lattice
```

```
library(gbm)
```

```
## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster

## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3
```

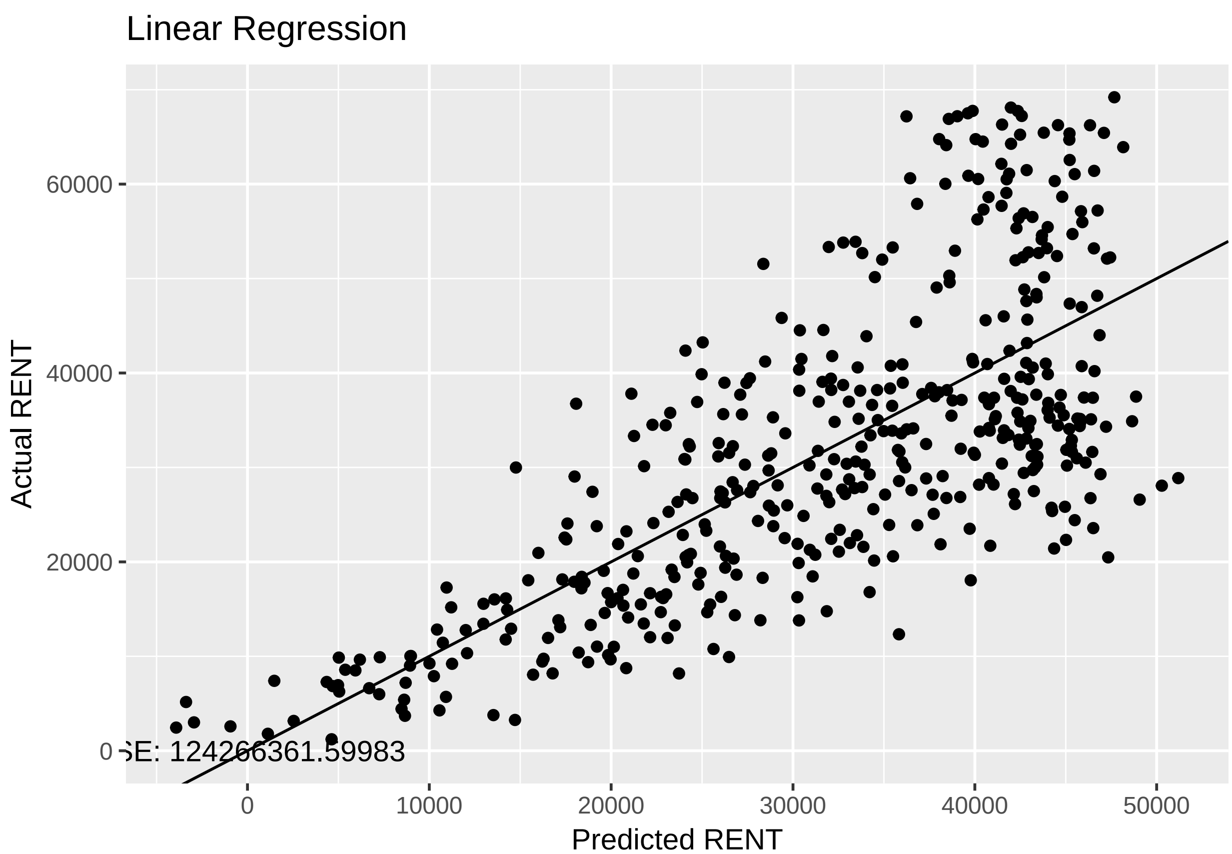
```
citibike_daily_weather=read.csv("citibike_daily_weather.csv")

sum(is.na(citibike_daily_weather))
```

```
## [1] 79
```

```
citibike_daily_weather=na.omit(citibike_daily_weather)


train_ind=sample(1:nrow(citibike_daily_weather),0.7*nrow(citibike_daily_weather))
train=citibike_daily_weather[train_ind,c(2,5:14)]
test=citibike_daily_weather[-train_ind,c(2,5:14)]

test_x=test[,-1]
test_y=test[,1]
```

Model 1: Linear Regression

```
###################################
#######Linear Regression##########
###################################

linear_model=lm(RENT~.,data=train)
linear_pre_y=predict(linear_model,test_x)

MSE_linear=mean((linear_pre_y-test_y)^2,na.rm = TRUE)

p.step<-qplot((linear_pre_y), (test_y), xlab='Predicted RENT',
              ylab='Actual RENT', main='Linear Regression')
p.step + annotate("text", x = 1.5, y = 7, label = paste('MSE:',MSE_linear))+geom_abline(slope=1, interce
```



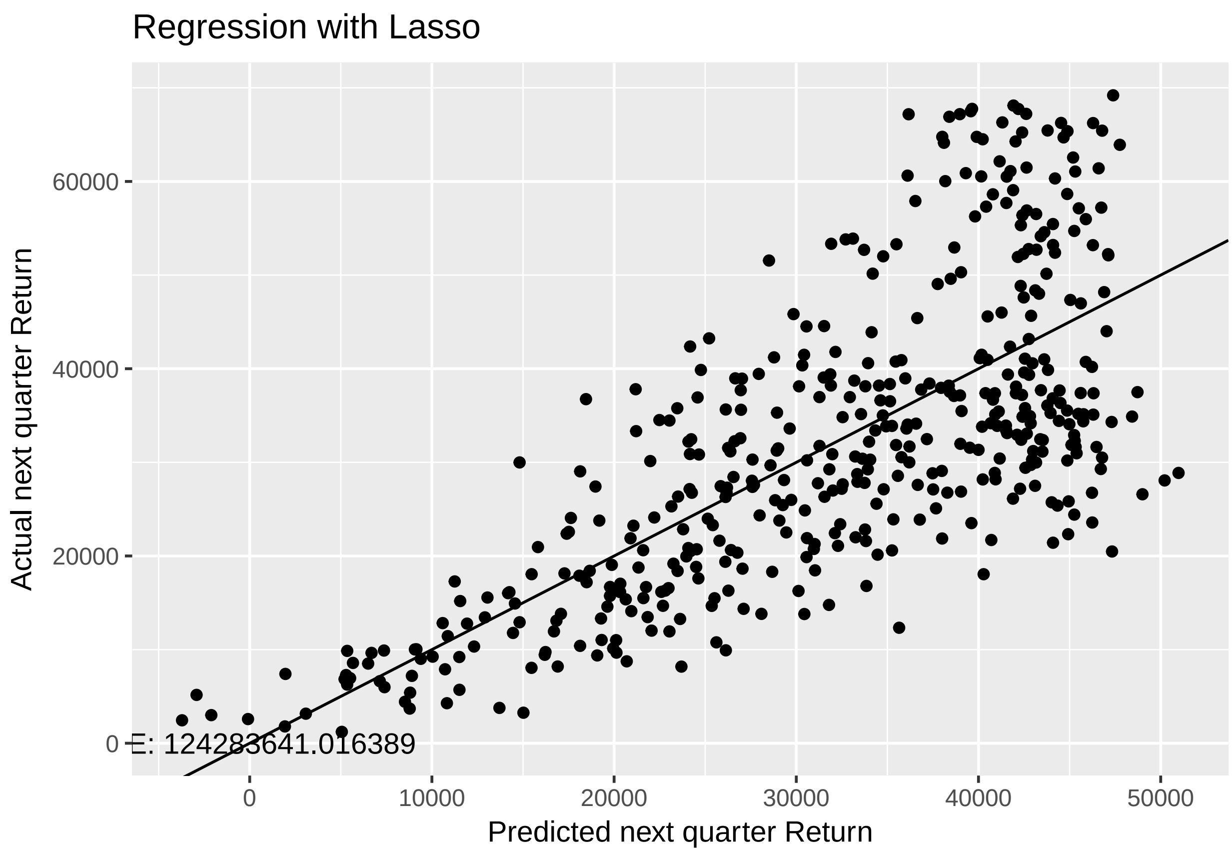
Model 2: Regression with Lasso

```
library(glmnet)
x_train_lasso=model.matrix(RENT~., train)[,1:(ncol(train)-1)]
y_train_lasso=train$RENT


x_test_lasso=model.matrix(RENT~.,test)[,1:(ncol(test)-1)]
y_test_lasso=test$RENT
grid=10^(-3:3)
#first run lasso on training set and pick the best lambda
cv.out=cv.glmnet(x_train_lasso,y_train_lasso,alpha=1,lambda = grid,nfolds = 5)

bestlam=cv.out$lambda.min

lasso_model=glmnet(x_train_lasso,y_train_lasso,alpha = 1,lambda = bestlam)
lasoo_pred_y=predict(lasso_model,x_test_lasso)

MSE_lasso=mean((lasoo_pred_y-y_test_lasso)^2,na.rm=TRUE)
#summary(lasso.mod)

coef(lasso_model)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                            s0
## (Intercept)  5283.003817
## (Intercept)     .
## AWND          -393.279479
## PRCP         -9688.023841
## SNOW             .
## SNWD          -848.754203
## TMAX           474.316849
## TMIN            38.100420
## WDF2            -6.109114
## WDF5             .
## WSF2           -95.497407
```

```
p.lasso=qplot(as.numeric(lasoo_pred_y),y_test_lasso,xlab = 'Predicted next quarter Return',
              ylab = 'Actual next quarter Return', main = 'Regression with Lasso' )
p.lasso+annotate("text",x=1.5,y=8,label=paste('MSE:',MSE_lasso))+geom_abline(slope=1,intercept = 0)
```

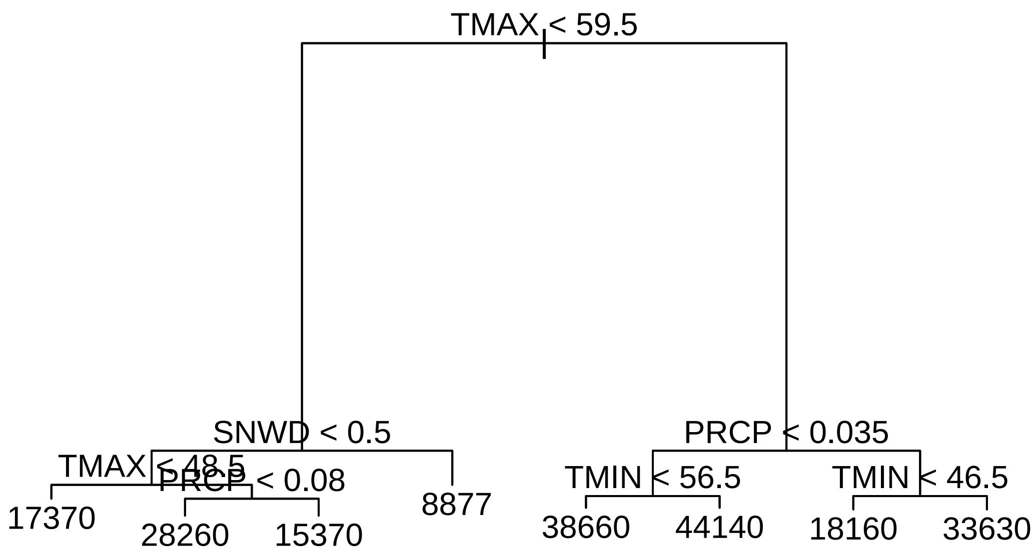



```
#MSE_lasso
```

Model 3: Regression Tree

```
library(ISLR)
library(tree)
#set.seed(1)
tree_model=tree(RENT~.,data=train)
```

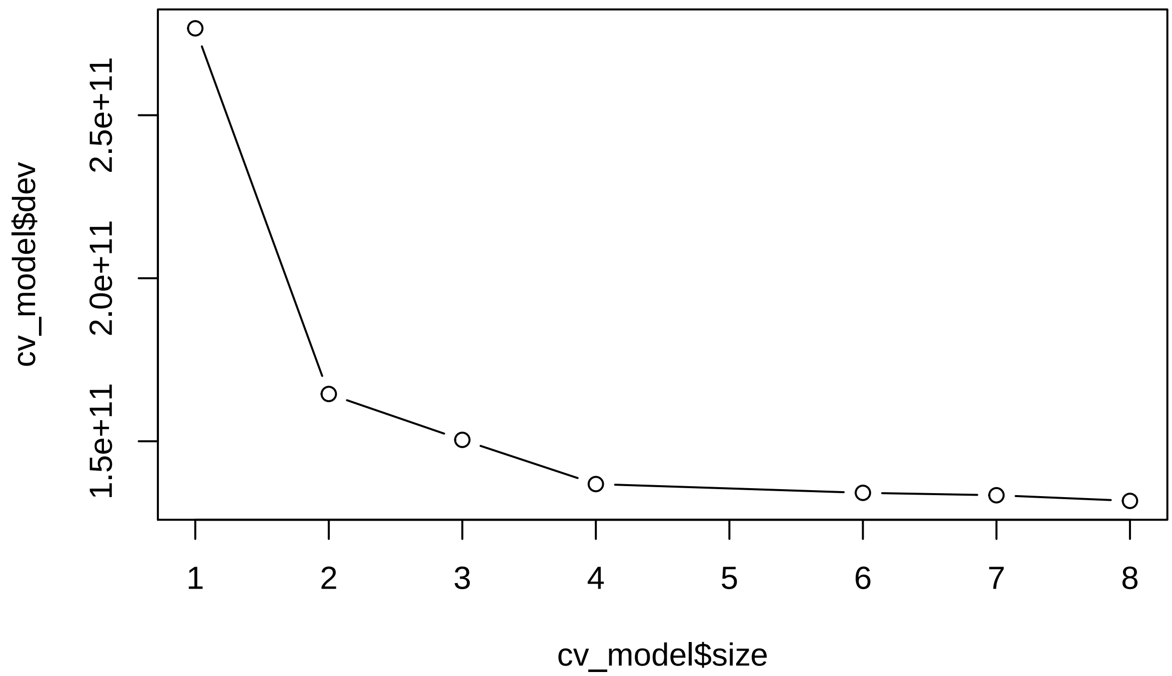
```r
plot(tree_model)
text(tree_model,pretty=1)
```




```r
tree_pred_y=predict(tree_model, test_x)

MSE_tree=mean((test_y-tree_pred_y)^2,na.rm=TRUE)


MSE_tree
```

```
## [1] 136301321
```

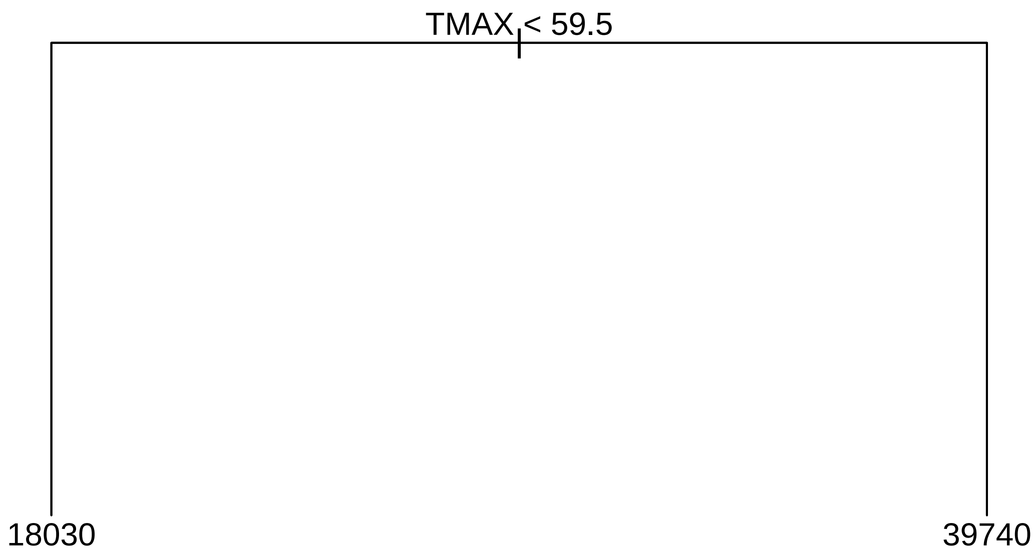
```r
##### CROSS VALIDATION #####
cv_model=cv.tree(tree_model)
plot(cv_model$size,cv_model$dev,type='b')
```




```r
bestSize=which.min(cv_model$dev)
print(bestSize)
```

```
## [1] 1
```

```r
# Prune Tree
prune.tree=prune.tree(tree_model,best=2)
plot(prune.tree)
text(prune.tree,pretty=0)
```




```
pred.prune.tree = predict(prune.tree, newdata=test)
MSE_prune_tree=mean((test_y-pred.prune.tree)^2)
MSE_prune_tree
```

```
## [1] 161233240
```

Model 4: Random Forest

```
library(randomForest)
library(e1071)
library(MASS)
library(caret)


RF_Model=randomForest(RENT~.,data = na.omit(train) ,importance=TRUE, na.rm = TRUE)

RF_Model
```

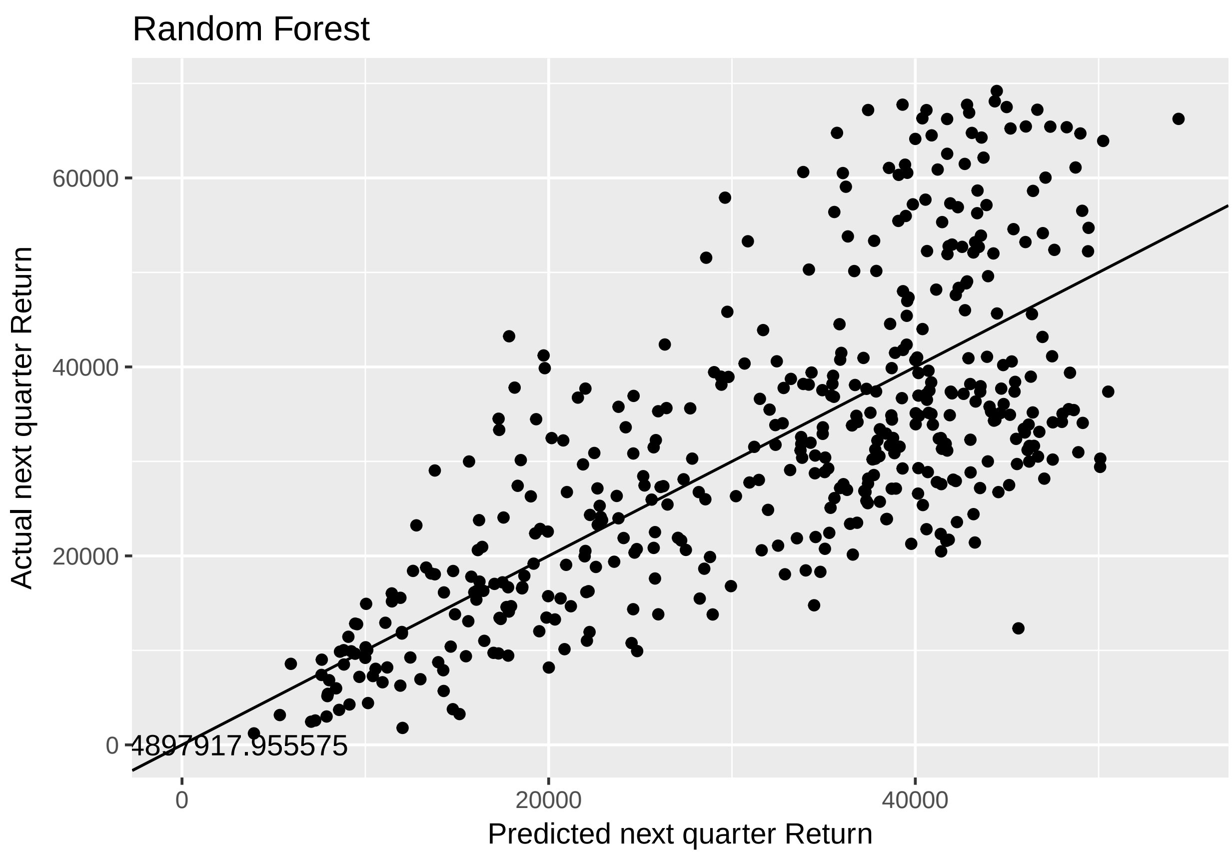
```
##
## Call:
##  randomForest(formula = RENT ~ ., data = na.omit(train), importance = TRUE,      na.rm = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
```

```
##           Mean of squared residuals: 121627027
##                     % Var explained: 53.68
```

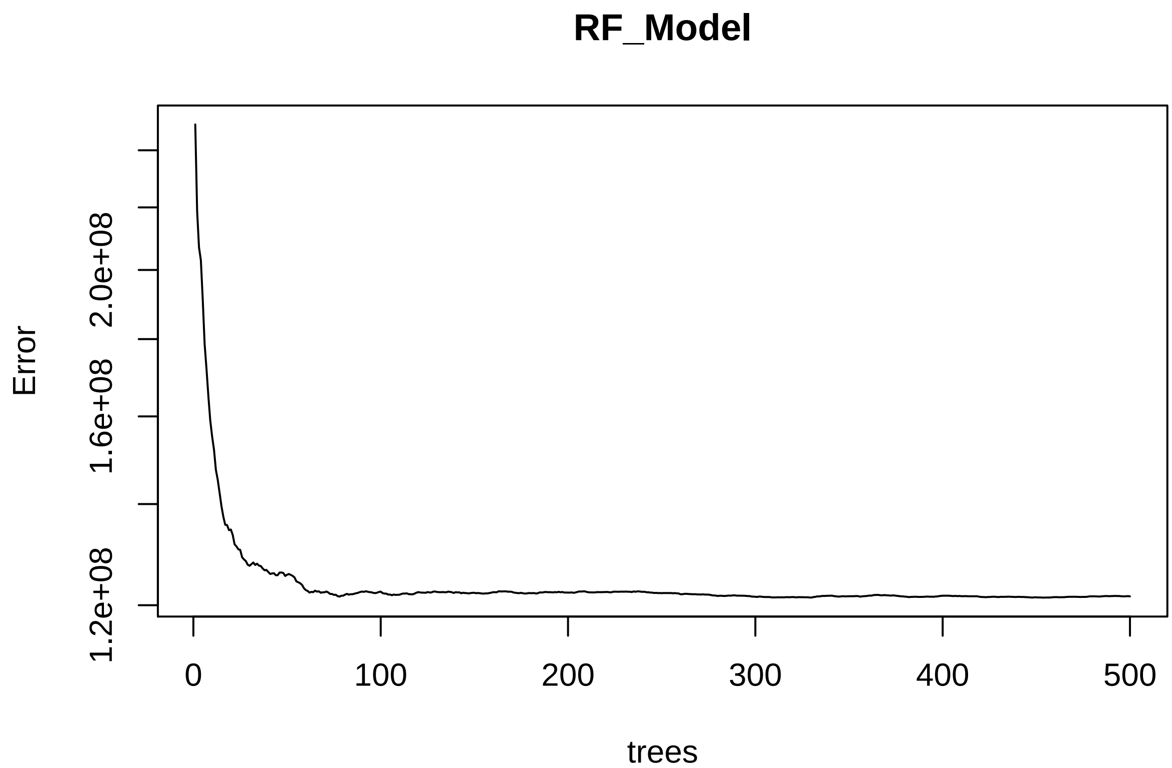
```r
yhat_bag=predict(RF_Model,test_x)
MSE_RF=mean((yhat_bag-test_y)^2,na.rm=TRUE)


#running the result
MSE_RF
```

```
## [1] 124897918
```

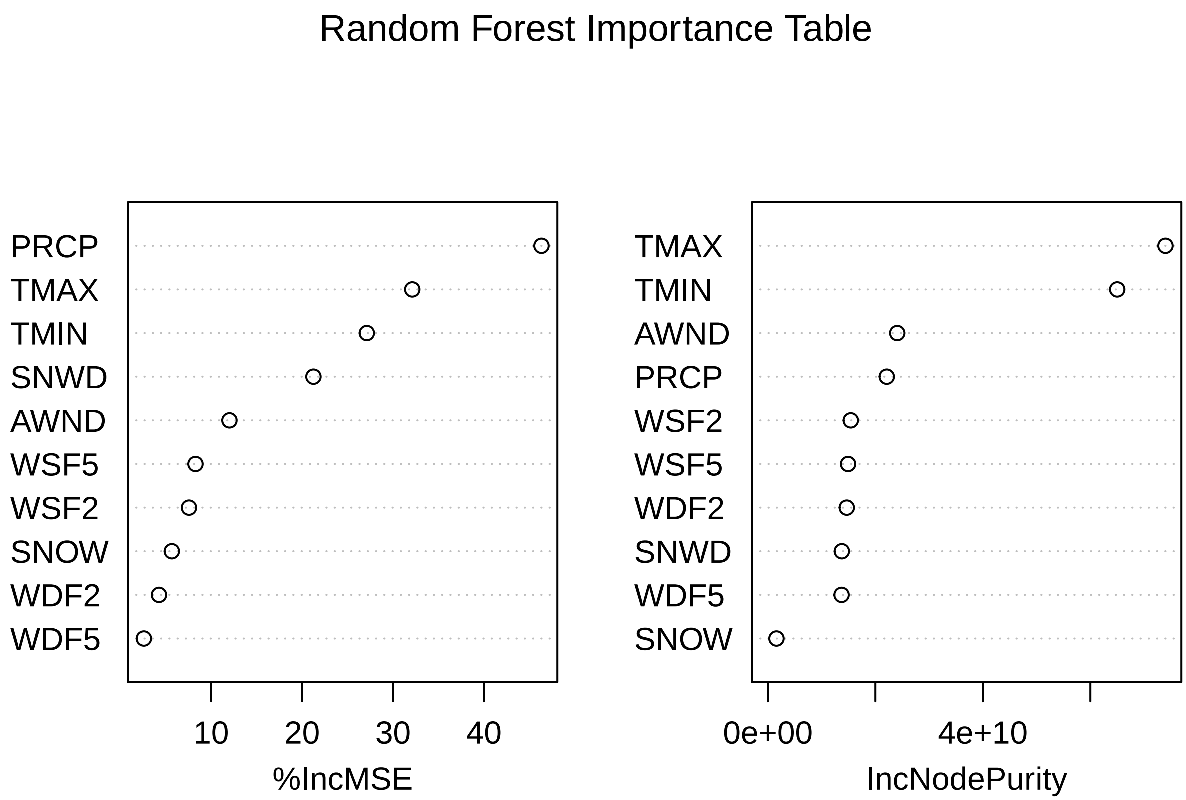
```r
p.rf<-qplot((yhat_bag), (test_y), xlab='Predicted next quarter Return',
            ylab='Actual next quarter Return', main='Random Forest')
p.rf + annotate("text", x = 1.5, y = 9, label = paste('MSE:',MSE_RF))+
  annotate("text", x = 1.5, y = 7, label = "")+
  geom_abline(slope=1, intercept=0)
```




```r
plot(RF_Model, log="y")
```




```r
varImpPlot(RF_Model,main='Random Forest Importance Table')
```

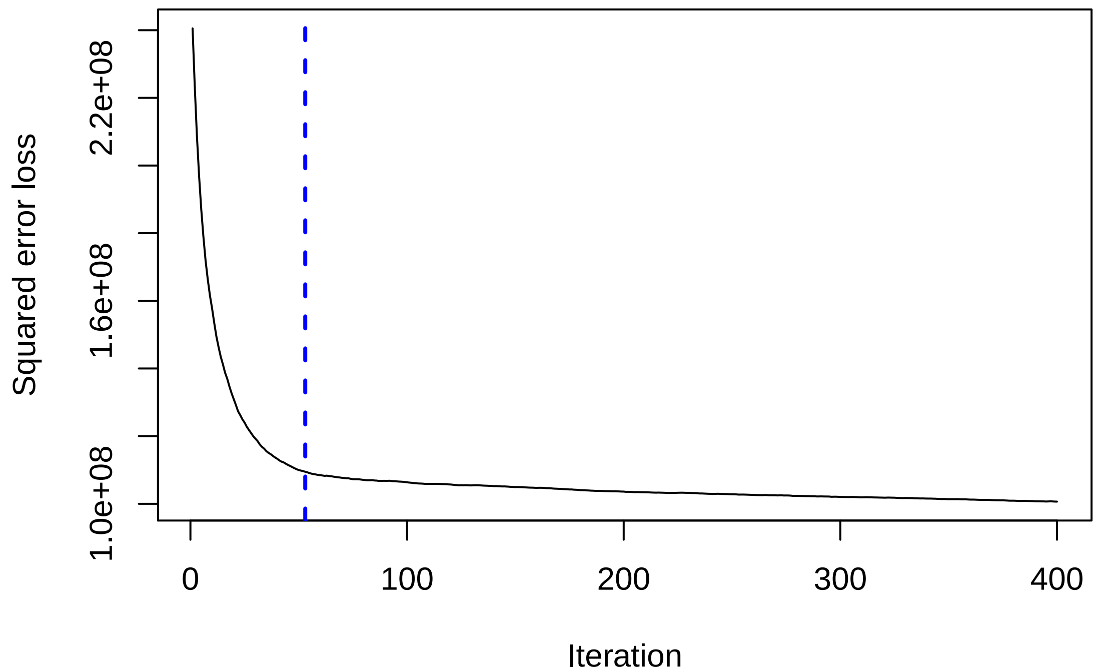



```r
varImp(RF_Model)
```

```
##        Overall
## AWND 12.011491
## PRCP 46.329839
## SNOW  5.669385
## SNWD 21.246177
## TMAX 32.115100
## TMIN 27.115211
## WDF2  4.258008
## WDF5  2.593359
## WSF2  7.558715
## WSF5  8.274484
```

Model 5: GBM

```r
#Generalized Boosted Regression Modeling
library(gbm)
gbm_model=gbm(RENT~.,data = train,dist="gaussian",n.tree = 400,shrinkage=0.1, cv.folds = 5)

best.iter <- gbm.perf(gbm_model,method="OOB")


gbm.perf(gbm_model,method="OOB")
```




```
## [1] 53
```

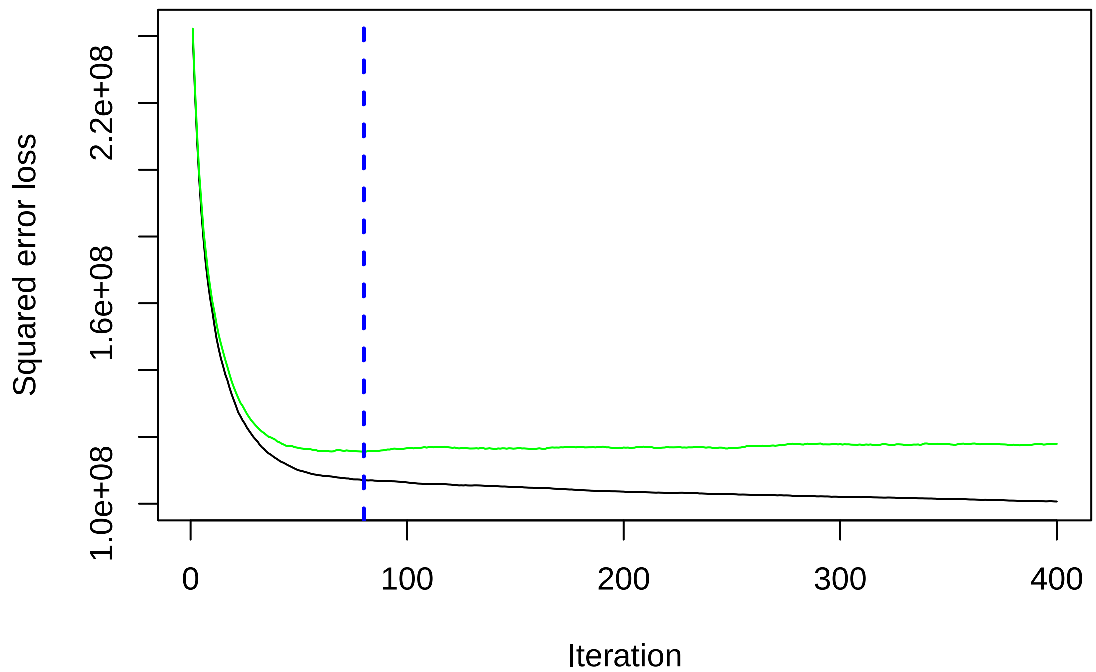
```r
print(best.iter)
```

```
## [1] 53
```

```r
best.iter <- gbm.perf(gbm_model,method="cv")
print(best.iter)
```

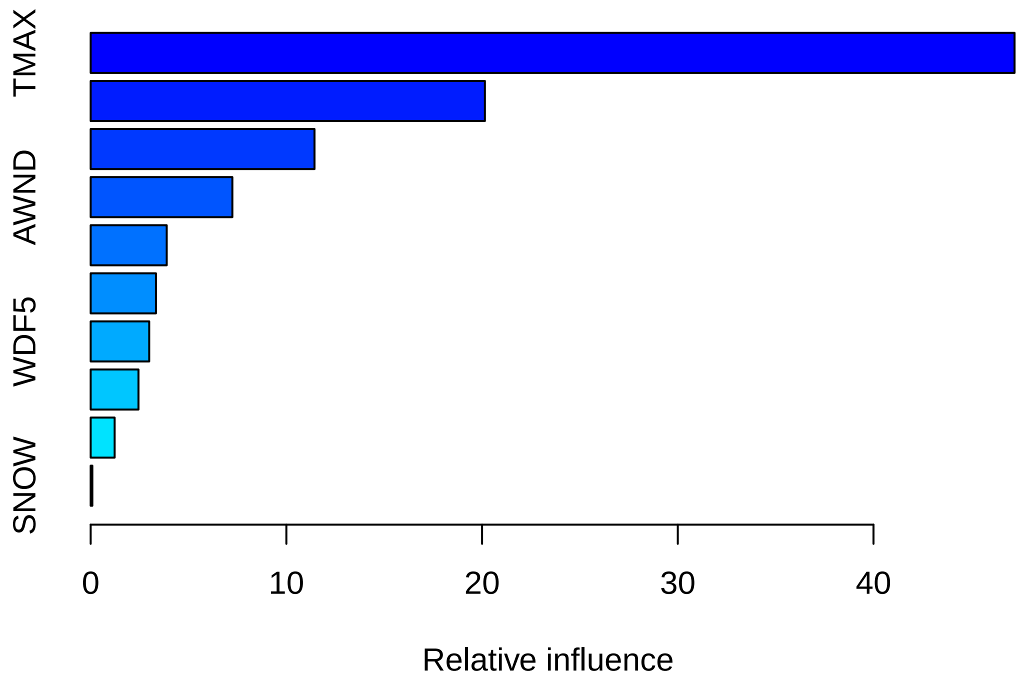
```
## [1] 80
```

```r
gbm.perf(gbm_model,method="cv")
```




```
## [1] 80
```

```r
sumary_GBM=summary(gbm_model)
```



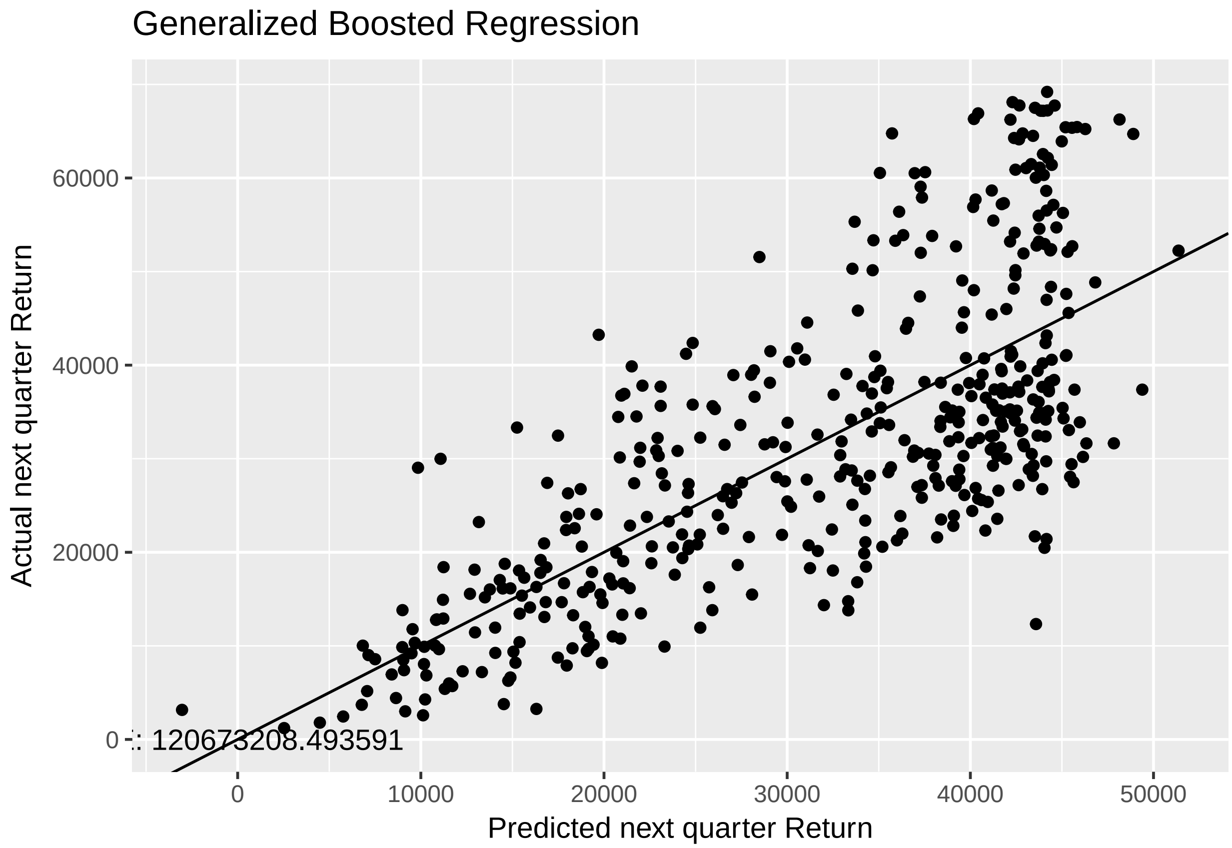

```
sumary_GBM
```

```
##       var     rel.inf
## TMAX TMAX 47.20815126
## TMIN TMIN 20.14344720
## PRCP PRCP 11.43817976
## AWND AWND  7.24097453
## WDF2 WDF2  3.88647390
## SNWD SNWD  3.33626299
## WDF5 WDF5  2.99283091
## WSF5 WSF5  2.44273745
## WSF2 WSF2  1.22564796
## SNOW SNOW  0.08529404
```

```
gbm_pred_y = predict(gbm_model, test, n.tree = 400, type = 'response')
MSE_gbm=mean((gbm_pred_y-test_y)^2,na.rm=TRUE)
MSE_gbm
```

```
## [1] 120673208
```

```
p.rf<-qplot((gbm_pred_y), (test_y), xlab='Predicted next quarter Return',
            ylab='Actual next quarter Return', main='Generalized Boosted Regression')
p.rf + annotate("text", x = 1.5, y = 9, label = paste('MSE:',MSE_gbm))+
  annotate("text", x = 1.5, y = 7, label = "")+
  geom_abline(slope=1, intercept=0)
```



Conclusion: Based on the 5 models we have, it can be concluded that there is a linear relationship between the number of rents per day and the weather data in NYC.