

# FAT LAB EXAM

Register Number:19BLC1186

Name:Tarun Sidhu

Date:7/6/2021

Dataset:SeoulBikeData(d5)

Question(Q\_id-6):Develop a Random Forest Regression Tree for the givendataset. Identify the target variable and describe the goal of the analysis. Summarize the performance of the model. Highlight the steps taken to improve model performance.

```
#Importing the Given dataset and analysing it
sbd <- read.csv("/Users/tarunsidhu/Downloads/SeoulBikeData.csv")
dim(sbd)
```

```
## [1] 8760 14
```

```
head(sbd)
```

```
##          Date Rented.Bike.Count Hour Temperature..C. Humidity...
## 1 01/12/2017          254      0          -5.2          37
## 2 01/12/2017          204      1          -5.5          38
## 3 01/12/2017          173      2          -6.0          39
## 4 01/12/2017          107      3          -6.2          40
## 5 01/12/2017           78      4          -6.0          36
## 6 01/12/2017          100      5          -6.4          37
## Wind.speed..m.s. Visibility..10m. Dew.point.temperature..C.
## 1          2.2          2000          -17.6
## 2          0.8          2000          -17.6
## 3          1.0          2000          -17.7
## 4          0.9          2000          -17.6
## 5          2.3          2000          -18.6
## 6          1.5          2000          -18.7
## Solar.Radiation..MJ.m2. Rainfall.mm. Snowfall..cm. Seasons      Holiday
## 1          0          0          0 Winter No Holiday
## 2          0          0          0 Winter No Holiday
## 3          0          0          0 Winter No Holiday
## 4          0          0          0 Winter No Holiday
## 5          0          0          0 Winter No Holiday
## 6          0          0          0 Winter No Holiday
## Functioning.Day
## 1          Yes
## 2          Yes
## 3          Yes
## 4          Yes
## 5          Yes
## 6          Yes
```

```
sapply(sbd, class)
```

```
##          Date          Rented.Bike.Count          Hour
##          "character"          "integer"          "integer"
##          Temperature..C.          Humidity...          Wind.speed..m.s.
##          "numeric"          "integer"          "numeric"
##          Visibility..10m. Dew.point.temperature..C.          Solar.Radiation..MJ.m2.
##          "integer"          "numeric"          "numeric"
##          Rainfall.mm.          Snowfall..cm.          Seasons
##          "numeric"          "numeric"          "character"
##          Holiday          Functioning.Day
##          "character"          "character"
```

```
summary(sbd)
```

```
##      Date      Rented.Bike.Count      Hour      Temperature..C.
## Length:8760    Min.      :    0.0    Min.      : 0.00    Min.      : -17.80
## Class :character 1st Qu.: 191.0    1st Qu.: 5.75    1st Qu.: 3.50
## Mode  :character Median : 504.5    Median :11.50    Median : 13.70
##                Mean   : 704.6    Mean   :11.50    Mean   : 12.88
##                3rd Qu.:1065.2    3rd Qu.:17.25    3rd Qu.: 22.50
##                Max.   :3556.0    Max.   :23.00    Max.   : 39.40
## Humidity... Wind.speed..m.s. Visibility..10m. Dew.point.temperature..C.
## Min.      : 0.00    Min.      :0.000    Min.      : 27    Min.      : -30.600
## 1st Qu.:42.00    1st Qu.:0.900    1st Qu.: 940    1st Qu.: -4.700
## Median :57.00    Median :1.500    Median :1698    Median : 5.100
## Mean   :58.23    Mean   :1.725    Mean   :1437    Mean   : 4.074
## 3rd Qu.:74.00    3rd Qu.:2.300    3rd Qu.:2000    3rd Qu.: 14.800
## Max.   :98.00    Max.   :7.400    Max.   :2000    Max.   : 27.200
## Solar.Radiation..MJ.m2. Rainfall.mm. Snowfall..cm. Seasons
## Min.      :0.0000    Min.      : 0.0000    Min.      :0.00000    Length:8760
## 1st Qu.:0.0000    1st Qu.: 0.0000    1st Qu.:0.00000    Class :character
## Median :0.0100    Median : 0.0000    Median :0.00000    Mode  :character
## Mean   :0.5691    Mean   : 0.1487    Mean   :0.07507
## 3rd Qu.:0.9300    3rd Qu.: 0.0000    3rd Qu.:0.00000
## Max.   :3.5200    Max.   :35.0000    Max.   :8.80000
## Holiday      Functioning.Day
## Length:8760    Length:8760
## Class :character Class :character
## Mode  :character Mode  :character
##
##
##
```

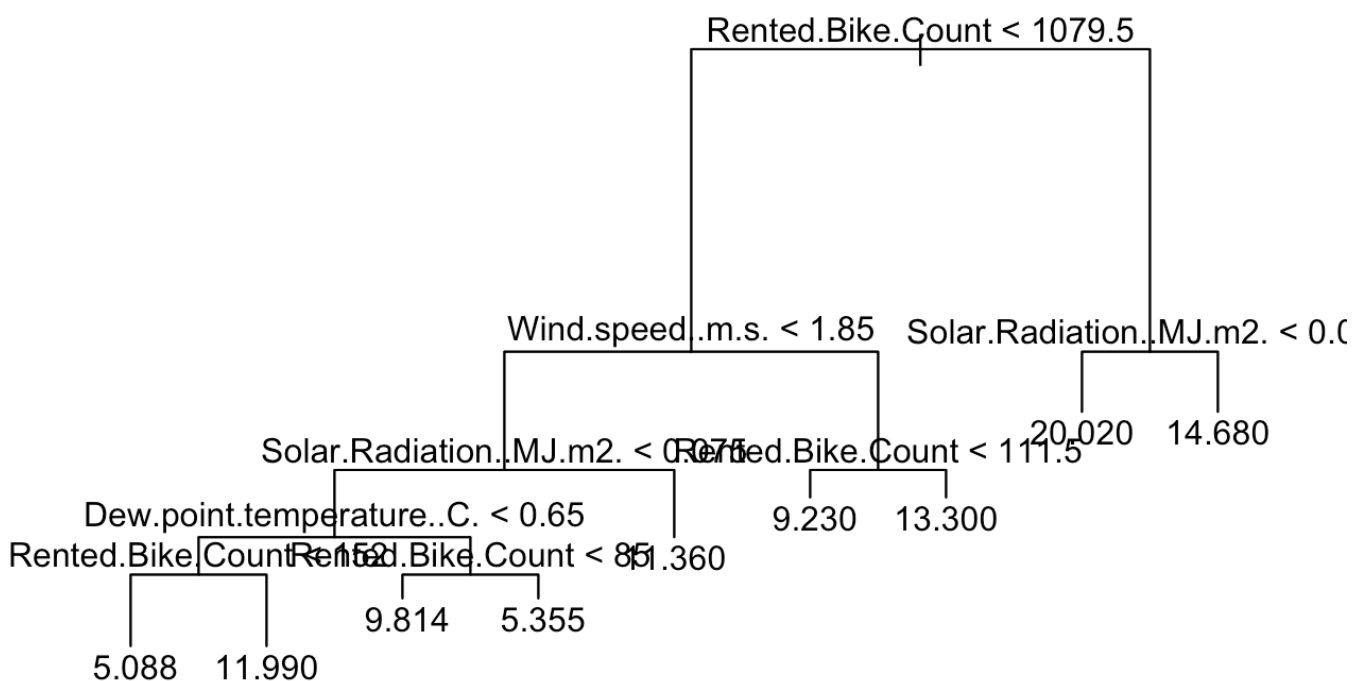
```
# Fitting Regression Trees
```

```
library(tree)
library(ISLR)
library(MASS)
set.seed(1)
train = sample(1:nrow(sbd), nrow(sbd)/2)
tree.sbd=tree(Hour~.,sbd,subset=train)
```

```
## Warning in tree(Hour ~ ., sbd, subset = train): NAs introduced by coercion
```

```
summary(tree.sbd)
```

```
plot(tree.sbd)
text(tree.sbd, pretty=0)
```



```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by  
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by  
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by  
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by  
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by  
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by  
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by  
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

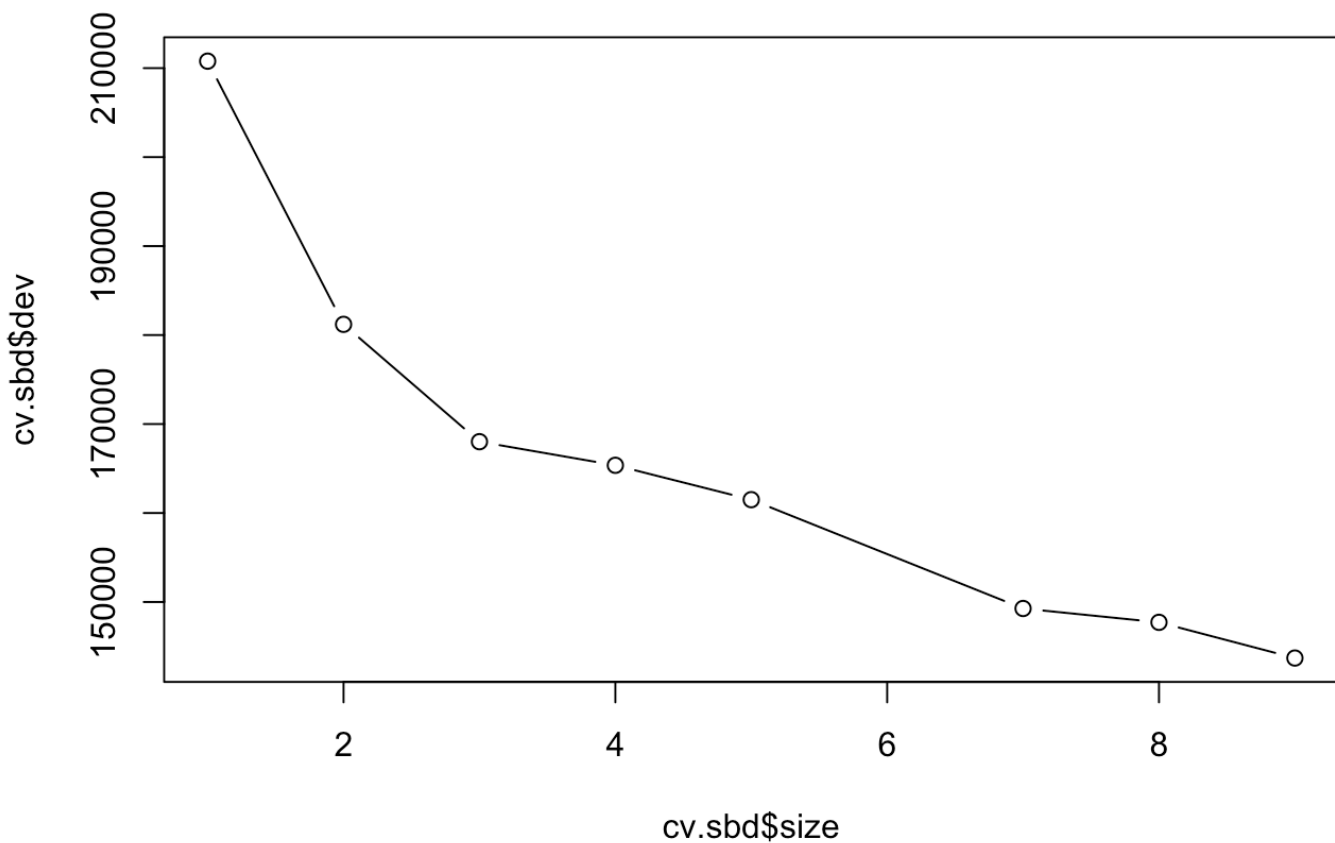
```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

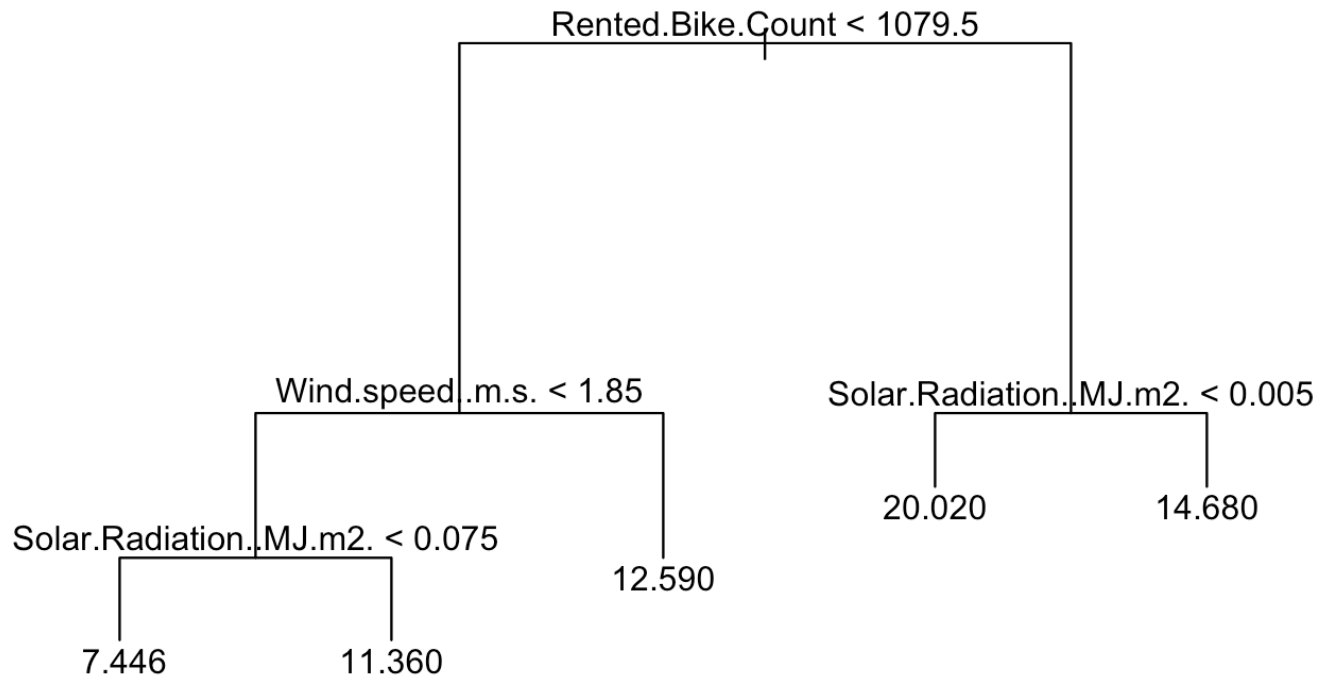
```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
plot(cv.sbd$size,cv.sbd$dev,type='b')
```



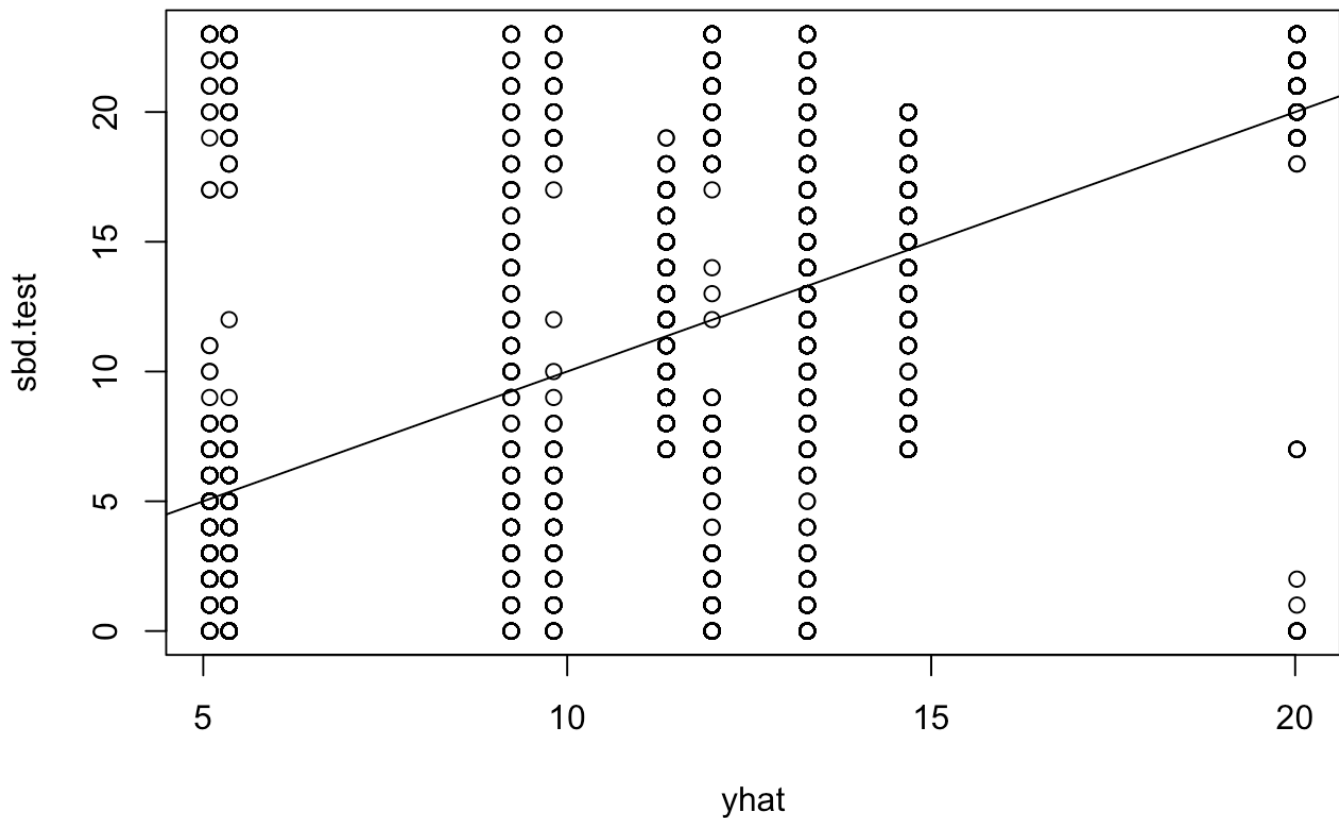
```
prune.sbd=prune.tree(tree.sbd,best=5)  
plot(prune.sbd)  
text(prune.sbd,pretty=0)
```



```
yhat=predict(tree.sbd,newdata=sbd[-train,])
```

```
## Warning in pred1.tree(object, tree.matrix(newdata)): NAs introduced by coercion
```

```
sbd.test=sbd[-train,"Hour"]
plot(yhat,sbd.test)
abline(0,1)
```



```
mean((yhat-sbd.test)^2)
```

```
## [1] 31.53642
```

```
# Bagging and Random Forests  
library(randomForest)
```

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

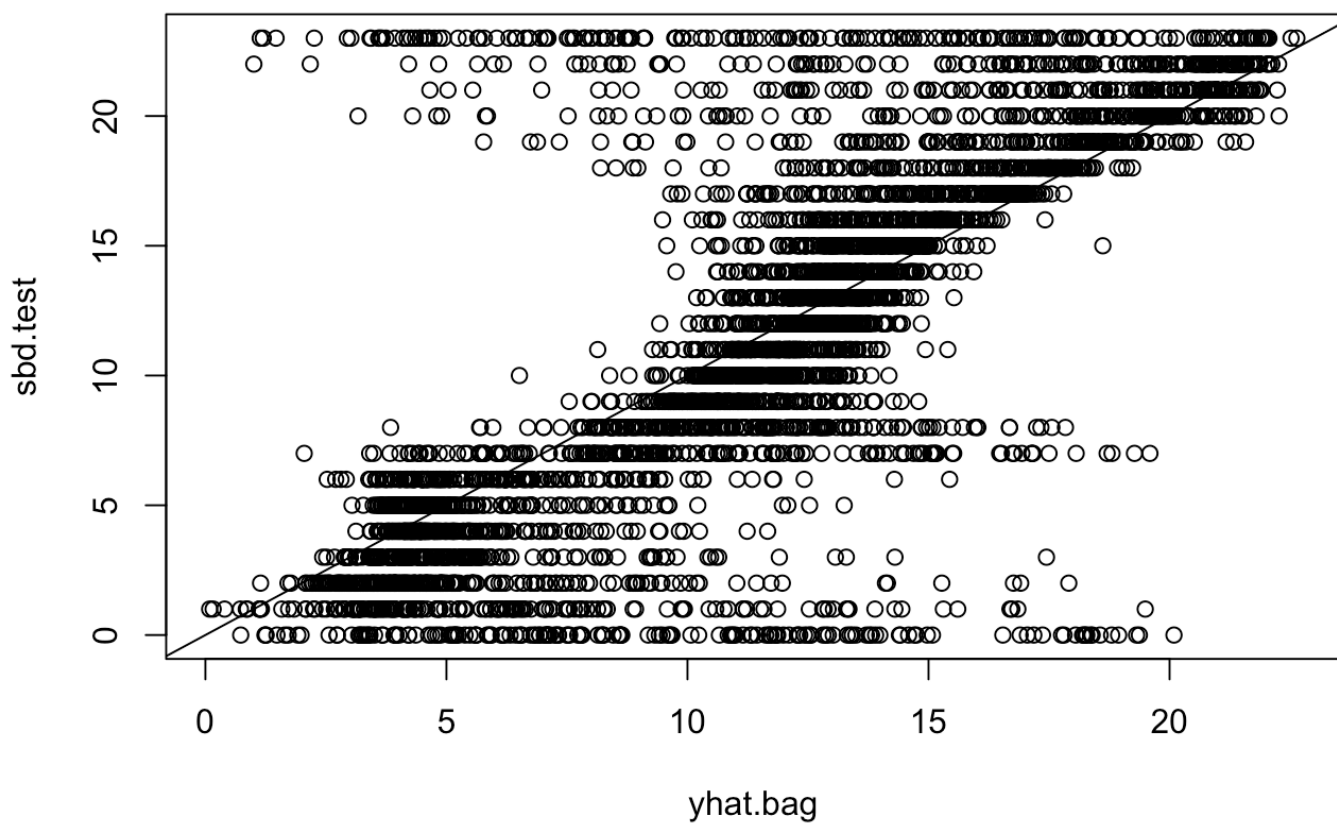
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(1)  
bag.sbd=randomForest(Hour~.,data=sbd,subset=train,mtry=13,importance=TRUE)  
bag.sbd
```



```
##
## Call:
## randomForest(formula = Hour ~ ., data = sbd, mtry = 13, importance = TRUE,
subset = train)
##
##           Type of random forest: regression
##           Number of trees: 500
## No. of variables tried at each split: 13
##
##           Mean of squared residuals: 20.50458
##           % Var explained: 57.37
```

```
yhat.bag = predict(bag.sbd,newdata=sbd[-train,])
plot(yhat.bag, sbd.test)
abline(0,1)
```



```
mean((yhat.bag-sbd.test)^2)
```

```
## [1] 20.95013
```

```
bag.sbd=randomForest(Hour~.,data=sbd,subset=train,mtry=13,ntree=25)
yhat.bag = predict(bag.sbd,newdata=sbd[-train,])
mean((yhat.bag-sbd.test)^2)
```

```
## [1] 21.67848
```

```
set.seed(1)
rf.sbd=randomForest(Hour~.,data=sbd,subset=train,mtry=6,importance=TRUE)
yhat.rf = predict(rf.sbd,newdata=sbd[-train,])
mean((yhat.rf-sbd.test)^2)
```

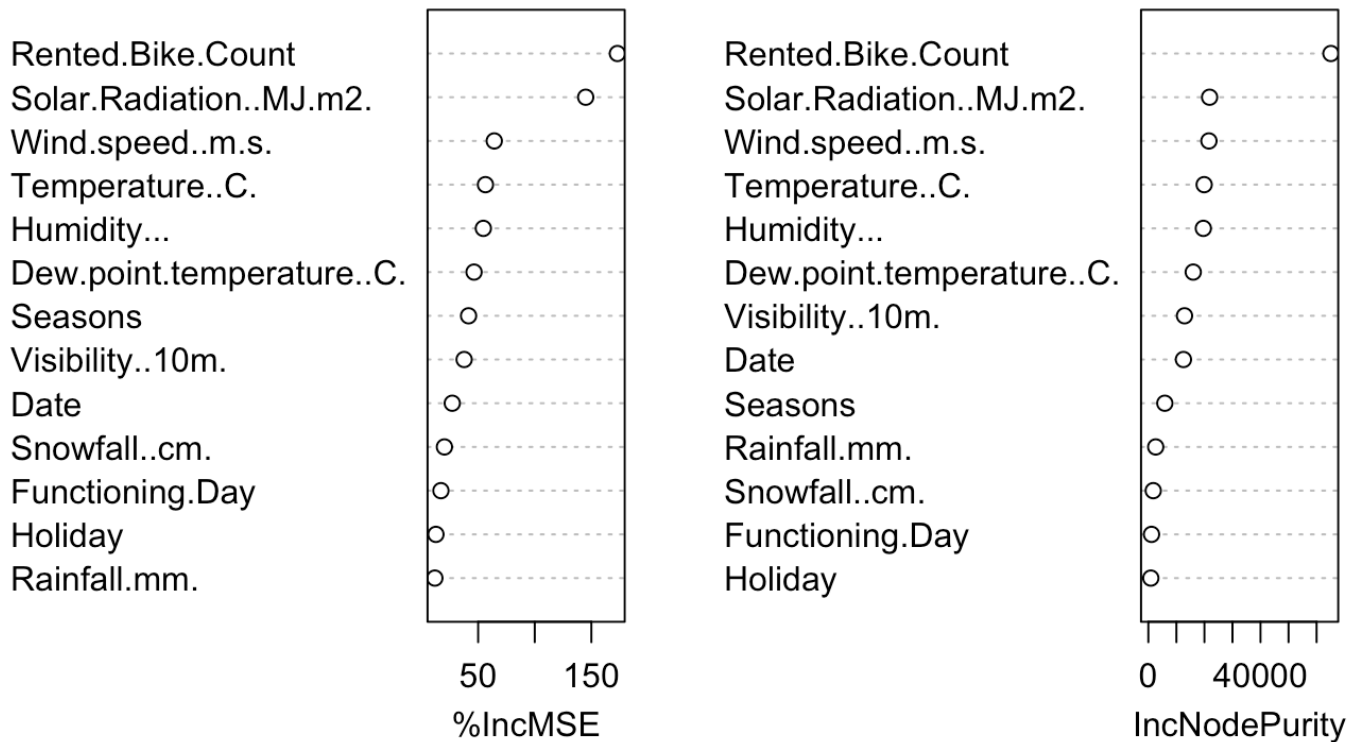
```
## [1] 20.86522
```

```
importance(rf.sbd)
```

##	%IncMSE	IncNodePurity
## Date	27.13718	12513.9417
## Rented.Bike.Count	173.01939	65115.1476
## Temperature..C.	56.45510	19905.0929
## Humidity...	54.58848	19611.8849
## Wind.speed..m.s.	64.24826	21602.2777
## Visibility..10m.	37.62303	12899.3860
## Dew.point.temperature..C.	46.37766	16006.9368
## Solar.Radiation..MJ.m2.	145.00822	21838.4070
## Rainfall.mm.	11.76214	2621.5133
## Snowfall..cm.	20.22296	1742.4241
## Seasons	41.53848	5868.6368
## Holiday	12.71431	879.7728
## Functioning.Day	17.11581	1132.8967

```
varImpPlot(rf.sbd)
```

## rf.sbd



```

oob.err=double(13)
test.err=double(13)
for (mtry in 1:13){
  fit=randomForest(Hour~.,data = sbd,subset=train,mtry=mtry,ntree=400)
  oob.err[mtry]=fit$mse[400]
  pred=predict(fit,sbd[-train,])
  test.err[mtry]=with(sbd[-train,],mean((Hour-Rented.Bike.Count)^2))
  cat(mtry," ")
}

```

```
## 1 2 3 4 5 6 7 8 9 10 11 12 13
```

```

matplot(1:mtry,cbind(test.err,oob.err),pch=19,col=c("red","blue"),type = "b",ylab=
"Mean Squared Error")
legend("topright",legend=c("Oob","Test"),pch=19,col=c("red","blue"))

```

