Team14\_Project\_Occupancy\_Detection

Prajwal Gonnade, Supreet Nayak

December 5, 2016

library(boot)

## Warning: package 'boot' was built under R version 3.2.5

library(caret)

## Warning: package 'caret' was built under R version 3.2.5

## Warning: package 'ggplot2' was built under R version 3.2.5

library(class)  
library(ROCR)

## Warning: package 'ROCR' was built under R version 3.2.5

## Warning: package 'gplots' was built under R version 3.2.5

library(MASS)  
library(tree)

## Warning: package 'tree' was built under R version 3.2.5

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.2.5

library(chemometrics)

## Warning: package 'chemometrics' was built under R version 3.2.5

library(reshape2)

## Warning: package 'reshape2' was built under R version 3.2.5

# Load CSV

Occupancy\_Train <- read.csv(file.choose(),header=T)  
Occupancy\_Test1 <- read.csv(file.choose(),header=T)  
Occupancy\_Test2 <- read.csv(file.choose(),header=T)

# Analyzing Data

names(Occupancy\_Train)

## [1] "date" "Temperature" "Humidity" "Light"   
## [5] "CO2" "HumidityRatio" "Occupancy"

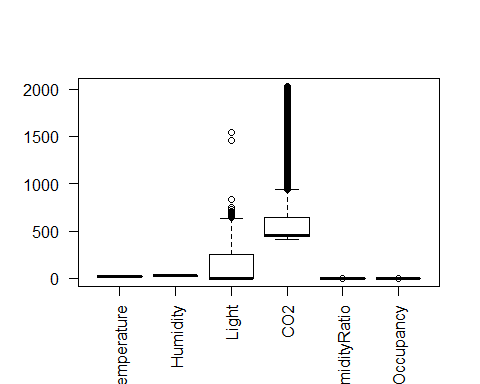
str(Occupancy\_Train)

## 'data.frame': 8143 obs. of 7 variables:  
## $ date : Factor w/ 8143 levels "2015-02-04 17:51:00",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ Temperature : num 23.2 23.1 23.1 23.1 23.1 ...  
## $ Humidity : num 27.3 27.3 27.2 27.2 27.2 ...  
## $ Light : num 426 430 426 426 426 ...  
## $ CO2 : num 721 714 714 708 704 ...  
## $ HumidityRatio: num 0.00479 0.00478 0.00478 0.00477 0.00476 ...  
## $ Occupancy : int 1 1 1 1 1 1 1 1 1 1 ...

summary(Occupancy\_Train)

## date Temperature Humidity   
## 2015-02-04 17:51:00: 1 Min. :19.00 Min. :16.75   
## 2015-02-04 17:51:59: 1 1st Qu.:19.70 1st Qu.:20.20   
## 2015-02-04 17:53:00: 1 Median :20.39 Median :26.22   
## 2015-02-04 17:54:00: 1 Mean :20.62 Mean :25.73   
## 2015-02-04 17:55:00: 1 3rd Qu.:21.39 3rd Qu.:30.53   
## 2015-02-04 17:55:59: 1 Max. :23.18 Max. :39.12   
## (Other) :8137   
## Light CO2 HumidityRatio Occupancy   
## Min. : 0.0 Min. : 412.8 Min. :0.002674 Min. :0.0000   
## 1st Qu.: 0.0 1st Qu.: 439.0 1st Qu.:0.003078 1st Qu.:0.0000   
## Median : 0.0 Median : 453.5 Median :0.003801 Median :0.0000   
## Mean : 119.5 Mean : 606.5 Mean :0.003863 Mean :0.2123   
## 3rd Qu.: 256.4 3rd Qu.: 638.8 3rd Qu.:0.004352 3rd Qu.:0.0000   
## Max. :1546.3 Max. :2028.5 Max. :0.006476 Max. :1.0000   
##

boxplot(Occupancy\_Train[,-1],las=2)



cor(Occupancy\_Train[,-1])

## Temperature Humidity Light CO2 HumidityRatio  
## Temperature 1.0000000 -0.14175931 0.64994184 0.5598938 0.1517616  
## Humidity -0.1417593 1.00000000 0.03782794 0.4390228 0.9551981  
## Light 0.6499418 0.03782794 1.00000000 0.6640221 0.2304202  
## CO2 0.5598938 0.43902276 0.66402206 1.0000000 0.6265559  
## HumidityRatio 0.1517616 0.95519808 0.23042022 0.6265559 1.0000000  
## Occupancy 0.5382197 0.13296424 0.90735211 0.7122352 0.3002816  
## Occupancy  
## Temperature 0.5382197  
## Humidity 0.1329642  
## Light 0.9073521  
## CO2 0.7122352  
## HumidityRatio 0.3002816  
## Occupancy 1.0000000

# Logistic Regression

glm(Occupancy ~ Temperature + Humidity + Light + CO2 + HumidityRatio, data = Occupancy\_Train, family = "binomial")

Occupancy\_glm <- glm(Occupancy ~ Temperature + Humidity + Light + CO2 + HumidityRatio, data = Occupancy\_Train, family = "binomial")  
summary(Occupancy\_glm)

##   
## Call:  
## glm(formula = Occupancy ~ Temperature + Humidity + Light + CO2 +   
## HumidityRatio, family = "binomial", data = Occupancy\_Train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -6.9573 -0.0654 -0.0384 -0.0186 2.7332   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.695e+00 1.367e+01 0.636 0.525   
## Temperature -9.039e-01 6.398e-01 -1.413 0.158   
## Humidity 3.099e-01 4.149e-01 0.747 0.455   
## Light 2.061e-02 7.662e-04 26.892 <2e-16 \*\*\*  
## CO2 6.430e-03 5.667e-04 11.348 <2e-16 \*\*\*  
## HumidityRatio -2.259e+03 2.682e+03 -0.842 0.400   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 8420.30 on 8142 degrees of freedom  
## Residual deviance: 956.12 on 8137 degrees of freedom  
## AIC: 968.12  
##   
## Number of Fisher Scoring iterations: 8

# Explanation -

# Removing insignificant predictors

summary(glm(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, family = "binomial"))

##   
## Call:  
## glm(formula = Occupancy ~ Temperature + Light + CO2 + HumidityRatio,   
## family = "binomial", data = Occupancy\_Train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -6.9702 -0.0650 -0.0399 -0.0191 2.7333   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.873e+01 2.838e+00 6.602 4.07e-11 \*\*\*  
## Temperature -1.373e+00 1.428e-01 -9.613 < 2e-16 \*\*\*  
## Light 2.061e-02 7.656e-04 26.915 < 2e-16 \*\*\*  
## CO2 6.259e-03 5.131e-04 12.198 < 2e-16 \*\*\*  
## HumidityRatio -2.590e+02 1.426e+02 -1.817 0.0692 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 8420.30 on 8142 degrees of freedom  
## Residual deviance: 956.68 on 8138 degrees of freedom  
## AIC: 966.68  
##   
## Number of Fisher Scoring iterations: 8

### Final Model -

Occupancy\_glm <- glm(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, family = "binomial")

# Confusion Matrix on Testing Data - 1

glm\_probs\_1 = predict(Occupancy\_glm, Occupancy\_Test1, type = "response")  
glm\_pred\_y\_1 = rep(0, length(Occupancy\_Test1$Occupancy))  
glm\_pred\_y\_1[glm\_probs\_1 > 0.5] = 1  
table(glm\_pred\_y\_1, Occupancy\_Test1$Occupancy)

##   
## glm\_pred\_y\_1 0 1  
## 0 1639 12  
## 1 54 960

mean(glm\_pred\_y\_1 != Occupancy\_Test1$Occupancy)

## [1] 0.02476548

confusionMatrix(Occupancy\_Test1$Occupancy, glm\_pred\_y\_1)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1639 54  
## 1 12 960  
##   
## Accuracy : 0.9752   
## 95% CI : (0.9686, 0.9808)  
## No Information Rate : 0.6195   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.947   
## Mcnemar's Test P-Value : 4.494e-07   
##   
## Sensitivity : 0.9927   
## Specificity : 0.9467   
## Pos Pred Value : 0.9681   
## Neg Pred Value : 0.9877   
## Prevalence : 0.6195   
## Detection Rate : 0.6150   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.9697   
##   
## 'Positive' Class : 0   
##

Test\_MSE <- sum((glm\_pred\_y\_1 - as.numeric(Occupancy\_Test1$Occupancy))^2)/nrow(Occupancy\_Test1)  
Test\_MSE

## [1] 0.02476548

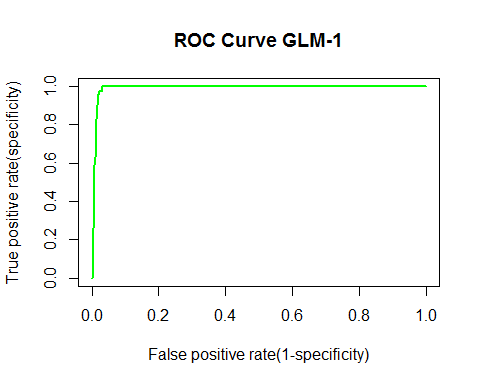
# ROC Analysis GLM - 1

roc.curve=function(s,print=FALSE){  
Ps=(glm\_probs\_1>s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 0))/sum(Occupancy\_Test1$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 1))/sum(Occupancy\_Test1$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test1$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 1639 54  
## 1 12 960

## FPR TPR   
## 0.03189604 0.98765432

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,], xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)',main = 'ROC Curve GLM-1', col="green",lwd=2,type="l")

  
 #Confusion Matrix on Testing Data - 2

glm\_probs\_1 = predict(Occupancy\_glm, Occupancy\_Test2, type = "response")  
glm\_pred\_y\_1 = rep(0, length(Occupancy\_Test2$Occupancy))  
glm\_pred\_y\_1[glm\_probs\_1 > 0.5] = 1  
table(glm\_pred\_y\_1, Occupancy\_Test2$Occupancy)

##   
## glm\_pred\_y\_1 0 1  
## 0 7648 135  
## 1 55 1914

mean(glm\_pred\_y\_1 != Occupancy\_Test2$Occupancy)

## [1] 0.01948318

confusionMatrix(Occupancy\_Test2$Occupancy, glm\_pred\_y\_1)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7648 55  
## 1 135 1914  
##   
## Accuracy : 0.9805   
## 95% CI : (0.9776, 0.9832)  
## No Information Rate : 0.7981   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9404   
## Mcnemar's Test P-Value : 9.969e-09   
##   
## Sensitivity : 0.9827   
## Specificity : 0.9721   
## Pos Pred Value : 0.9929   
## Neg Pred Value : 0.9341   
## Prevalence : 0.7981   
## Detection Rate : 0.7842   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.9774   
##   
## 'Positive' Class : 0   
##

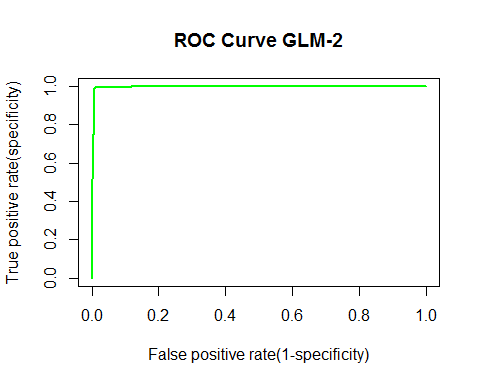
# ROC Analysis GLM - 2

roc.curve=function(s,print=FALSE){  
Ps=(glm\_probs\_1>s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 0))/sum(Occupancy\_Test2$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 1))/sum(Occupancy\_Test2$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test2$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 7648 55  
## 1 135 1914

## FPR TPR   
## 0.007140075 0.934114202

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,], xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)',main = 'ROC Curve GLM-2', col="green",lwd=2,type="l")

  
 #LDA - Testing Data 1

Occupancy\_lda <- lda(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
Occupancy\_lda

## Call:  
## lda(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.7876704 0.2123296   
##   
## Group means:  
## Temperature Light CO2 HumidityRatio  
## 0 20.33493 27.77644 490.3203 0.003729632  
## 1 21.67319 459.85435 1037.7048 0.004355428  
##   
## Coefficients of linear discriminants:  
## LD1  
## Temperature -4.092751e-01  
## Light 1.227500e-02  
## CO2 2.326517e-03  
## HumidityRatio -1.060925e+02

lda\_pred = predict(Occupancy\_lda, Occupancy\_Test1)  
names(lda\_pred)

## [1] "class" "posterior" "x"

table(lda\_pred$class, Occupancy\_Test1$Occupancy)

##   
## 0 1  
## 0 1638 1  
## 1 55 971

mean(lda\_pred$class != Occupancy\_Test1$Occupancy)

## [1] 0.02101313

confusionMatrix(Occupancy\_Test1$Occupancy,lda\_pred$class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1638 55  
## 1 1 971  
##   
## Accuracy : 0.979   
## 95% CI : (0.9728, 0.9841)  
## No Information Rate : 0.615   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9552   
## Mcnemar's Test P-Value : 1.417e-12   
##   
## Sensitivity : 0.9994   
## Specificity : 0.9464   
## Pos Pred Value : 0.9675   
## Neg Pred Value : 0.9990   
## Prevalence : 0.6150   
## Detection Rate : 0.6146   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.9729   
##   
## 'Positive' Class : 0   
##

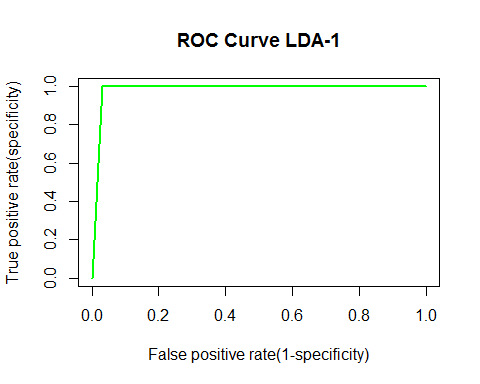
# ROC Analysis LDA - Testing Data 1

S = lda\_pred$posterior[,2]  
roc.curve=function(s,print=FALSE){  
Ps=(S>s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 0))/sum(Occupancy\_Test1$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 1))/sum(Occupancy\_Test1$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test1$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 1638 55  
## 1 1 971

## FPR TPR   
## 0.03248671 0.99897119

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,],xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)',main = 'ROC Curve LDA-1', col="green",lwd=2,type="l")

  
 #LDA - Testing Data 2

Occupancy\_lda <- lda(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
Occupancy\_lda

## Call:  
## lda(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.7876704 0.2123296   
##   
## Group means:  
## Temperature Light CO2 HumidityRatio  
## 0 20.33493 27.77644 490.3203 0.003729632  
## 1 21.67319 459.85435 1037.7048 0.004355428  
##   
## Coefficients of linear discriminants:  
## LD1  
## Temperature -4.092751e-01  
## Light 1.227500e-02  
## CO2 2.326517e-03  
## HumidityRatio -1.060925e+02

lda\_pred = predict(Occupancy\_lda, Occupancy\_Test2)  
names(lda\_pred)

## [1] "class" "posterior" "x"

table(lda\_pred$class, Occupancy\_Test2$Occupancy)

##   
## 0 1  
## 0 7626 8  
## 1 77 2041

mean(lda\_pred$class != Occupancy\_Test2$Occupancy)

## [1] 0.008716161

confusionMatrix(Occupancy\_Test2$Occupancy,lda\_pred$class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7626 77  
## 1 8 2041  
##   
## Accuracy : 0.9913   
## 95% CI : (0.9892, 0.993)  
## No Information Rate : 0.7828   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9741   
## Mcnemar's Test P-Value : 1.636e-13   
##   
## Sensitivity : 0.9990   
## Specificity : 0.9636   
## Pos Pred Value : 0.9900   
## Neg Pred Value : 0.9961   
## Prevalence : 0.7828   
## Detection Rate : 0.7820   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.9813   
##   
## 'Positive' Class : 0   
##

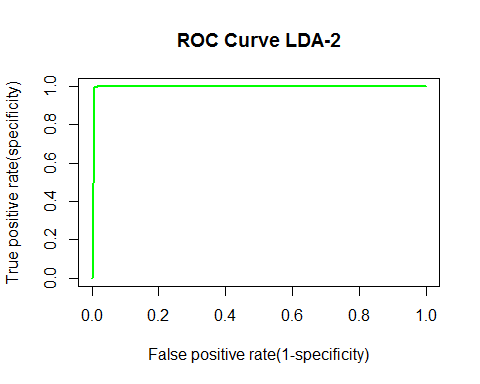
# ROC Analysis LDA - Testing Data 2

S = lda\_pred$posterior[,2]  
roc.curve=function(s,print=FALSE){  
Ps=(S>s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 0))/sum(Occupancy\_Test2$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 1))/sum(Occupancy\_Test2$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test2$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 7626 77  
## 1 8 2041

## FPR TPR   
## 0.009996105 0.996095656

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,],xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)',main = 'ROC Curve LDA-2', col="green",lwd=2,type="l")



# Cross Validation LDA

inds=sample(1:nrow(Occupancy\_Train),0.9\*nrow(Occupancy\_Train))  
df.train=Occupancy\_Train[inds,]  
df.test=Occupancy\_Train[-inds,]  
train.model = lda(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
preds=predict(train.model, df.test)

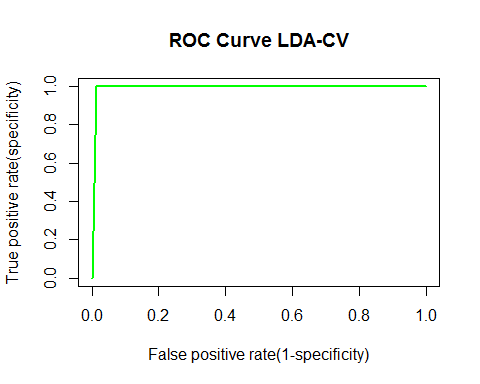
# ROC Analysis LDA - CV

S = preds$posterior[,2]  
roc.curve=function(s,print=FALSE){  
Ps=(S>s)\*1  
FP=sum((Ps==1)\*(df.test$Occupancy == 0))/sum(df.test$Occupancy == 0)  
TP=sum((Ps==1)\*(df.test$Occupancy == 1))/sum(df.test$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=df.test$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 629 9  
## 1 0 177

## FPR TPR   
## 0.01410658 1.00000000

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,],xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)',main = 'ROC Curve LDA-CV', col="green",lwd=2,type="l")



# QDA - Testing Data 1

Occupancy\_qda <- qda(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
Occupancy\_qda

## Call:  
## qda(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.7876704 0.2123296   
##   
## Group means:  
## Temperature Light CO2 HumidityRatio  
## 0 20.33493 27.77644 490.3203 0.003729632  
## 1 21.67319 459.85435 1037.7048 0.004355428

qda.pred <- predict(Occupancy\_qda, Occupancy\_Test1)  
table(qda.pred$class, Occupancy\_Test1$Occupancy)

##   
## 0 1  
## 0 1639 6  
## 1 54 966

mean(qda.pred$class != Occupancy\_Test1$Occupancy)

## [1] 0.02251407

confusionMatrix(Occupancy\_Test1$Occupancy,qda.pred$class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1639 54  
## 1 6 966  
##   
## Accuracy : 0.9775   
## 95% CI : (0.9711, 0.9828)  
## No Information Rate : 0.6173   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9519   
## Mcnemar's Test P-Value : 1.298e-09   
##   
## Sensitivity : 0.9964   
## Specificity : 0.9471   
## Pos Pred Value : 0.9681   
## Neg Pred Value : 0.9938   
## Prevalence : 0.6173   
## Detection Rate : 0.6150   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.9717   
##   
## 'Positive' Class : 0   
##

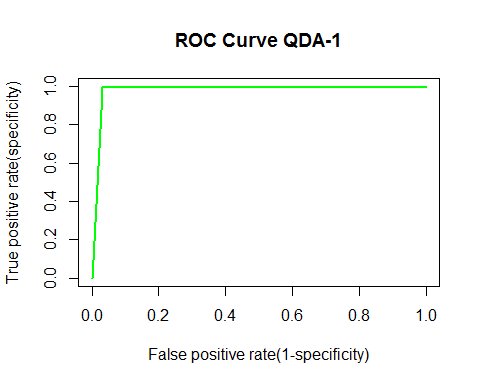
# ROC Analysis QDA - Testing Data 1

S = qda.pred$posterior[,2]  
roc.curve=function(s,print=FALSE){  
Ps=(S > s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 0))/sum(Occupancy\_Test1$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 1))/sum(Occupancy\_Test1$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test1$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 1639 54  
## 1 6 966

## FPR TPR   
## 0.03189604 0.99382716

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,],xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)',main = 'ROC Curve QDA-1', col="green",lwd=2,type="l")

  
 #QDA - Testing Data 2

Occupancy\_qda <- qda(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
Occupancy\_qda

## Call:  
## qda(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.7876704 0.2123296   
##   
## Group means:  
## Temperature Light CO2 HumidityRatio  
## 0 20.33493 27.77644 490.3203 0.003729632  
## 1 21.67319 459.85435 1037.7048 0.004355428

qda.pred <- predict(Occupancy\_qda, Occupancy\_Test2)  
table(qda.pred$class, Occupancy\_Test2$Occupancy)

##   
## 0 1  
## 0 7645 166  
## 1 58 1883

mean(qda.pred$class != Occupancy\_Test2$Occupancy)

## [1] 0.02296965

confusionMatrix(Occupancy\_Test2$Occupancy,qda.pred$class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7645 58  
## 1 166 1883  
##   
## Accuracy : 0.977   
## 95% CI : (0.9739, 0.9799)  
## No Information Rate : 0.801   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9294   
## Mcnemar's Test P-Value : 8.726e-13   
##   
## Sensitivity : 0.9787   
## Specificity : 0.9701   
## Pos Pred Value : 0.9925   
## Neg Pred Value : 0.9190   
## Prevalence : 0.8010   
## Detection Rate : 0.7839   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.9744   
##   
## 'Positive' Class : 0   
##

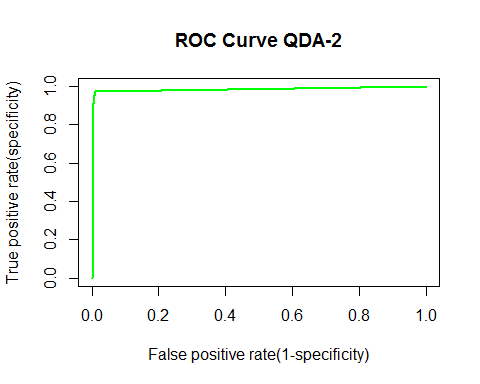
# ROC Analysis QDA - Testing Data 2

S = qda.pred$posterior[,2]  
roc.curve=function(s,print=FALSE){  
Ps=(S > s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 0))/sum(Occupancy\_Test2$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 1))/sum(Occupancy\_Test2$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test2$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 7645 58  
## 1 166 1883

## FPR TPR   
## 0.007529534 0.918984871

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,],xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)',main = 'ROC Curve QDA-2', col="green",lwd=2,type="l")

  
 #Cross Validation QDA - 10 Fold

inds=sample(1:nrow(Occupancy\_Train),0.9\*nrow(Occupancy\_Train))  
df.train=Occupancy\_Train[inds,]  
df.test=Occupancy\_Train[-inds,]  
train.model = qda(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
preds=predict(train.model, df.test)

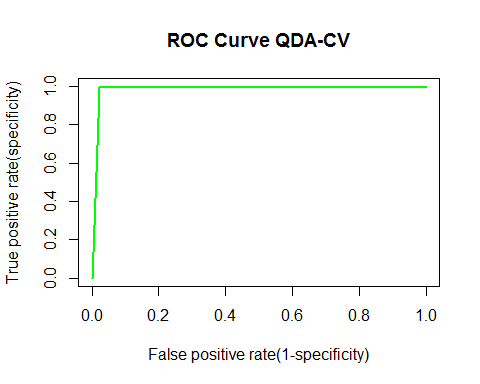
# ROC Analysis QDA - CV

S = preds$posterior[,2]  
roc.curve=function(s,print=FALSE){  
Ps=(S>s)\*1  
FP=sum((Ps==1)\*(df.test$Occupancy == 0))/sum(df.test$Occupancy == 0)  
TP=sum((Ps==1)\*(df.test$Occupancy == 1))/sum(df.test$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=df.test$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 641 14  
## 1 2 158

## FPR TPR   
## 0.02137405 0.98750000

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,],xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)',main = 'ROC Curve QDA-CV', col="green",lwd=2,type="l")



# CART - Testing Data 1

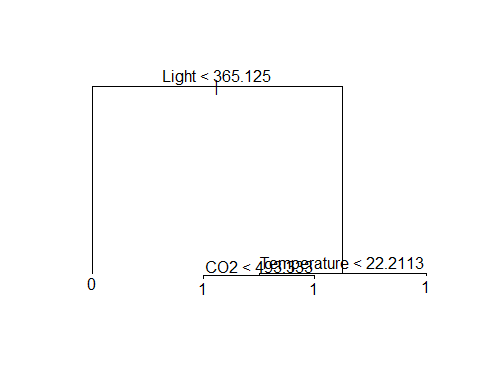
set.seed(2)  
Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
str(Occupancy\_Train)

## 'data.frame': 8143 obs. of 7 variables:  
## $ date : Factor w/ 8143 levels "2015-02-04 17:51:00",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ Temperature : num 23.2 23.1 23.1 23.1 23.1 ...  
## $ Humidity : num 27.3 27.3 27.2 27.2 27.2 ...  
## $ Light : num 426 430 426 426 426 ...  
## $ CO2 : num 721 714 714 708 704 ...  
## $ HumidityRatio: num 0.00479 0.00478 0.00478 0.00477 0.00476 ...  
## $ Occupancy : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

tree.Occupancy\_Train <- tree(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
plot(tree.Occupancy\_Train)  
summary(tree.Occupancy\_Train)

##   
## Classification tree:  
## tree(formula = Occupancy ~ Temperature + Light + CO2 + HumidityRatio,   
## data = Occupancy\_Train)  
## Variables actually used in tree construction:  
## [1] "Light" "Temperature" "CO2"   
## Number of terminal nodes: 4   
## Residual mean deviance: 0.07946 = 646.8 / 8139   
## Misclassification error rate: 0.01216 = 99 / 8143

text(tree.Occupancy\_Train, pretty = 0)



tree.pred <- predict(tree.Occupancy\_Train, Occupancy\_Test1, type = "class")  
table(tree.pred, Occupancy\_Test1$Occupancy)

##   
## tree.pred 0 1  
## 0 1639 3  
## 1 54 969

mean(tree.pred != Occupancy\_Test1$Occupancy)

## [1] 0.02138837

confusionMatrix(Occupancy\_Test1$Occupancy, tree.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1639 54  
## 1 3 969  
##   
## Accuracy : 0.9786   
## 95% CI : (0.9724, 0.9838)  
## No Information Rate : 0.6161   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9544   
## Mcnemar's Test P-Value : 3.528e-11   
##   
## Sensitivity : 0.9982   
## Specificity : 0.9472   
## Pos Pred Value : 0.9681   
## Neg Pred Value : 0.9969   
## Prevalence : 0.6161   
## Detection Rate : 0.6150   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.9727   
##   
## 'Positive' Class : 0   
##

# Cross Validation and Pruning - Testing Data 1

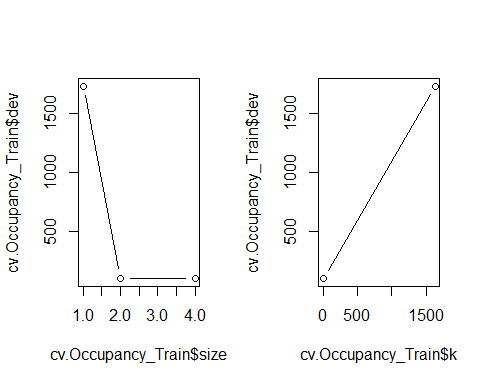
set.seed(3)  
cv.Occupancy\_Train <- cv.tree(tree.Occupancy\_Train, FUN=prune.misclass)  
names(cv.Occupancy\_Train)

## [1] "size" "dev" "k" "method"

cv.Occupancy\_Train

## $size  
## [1] 4 2 1  
##   
## $dev  
## [1] 101 101 1729  
##   
## $k  
## [1] -Inf 0 1630  
##   
## $method  
## [1] "misclass"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

par(mfrow = c(1,2))  
plot(cv.Occupancy\_Train$size, cv.Occupancy\_Train$dev, type = "b")  
plot(cv.Occupancy\_Train$k, cv.Occupancy\_Train$dev, type = "b")



prune.Occupancy\_Train = prune.misclass(tree.Occupancy\_Train, best = 9)

## Warning in prune.tree(tree = tree.Occupancy\_Train, best = 9, method =  
## "misclass"): best is bigger than tree size

#prune.mt2Data\_Train = prune.misclass(tree.mt2Data\_Train)  
plot(prune.Occupancy\_Train)  
text(prune.Occupancy\_Train, pretty = 0)  
tree.pred = predict(prune.Occupancy\_Train, Occupancy\_Test1, type = "class")  
#tree.pred = predict(prune.mt2Data\_Train, mt2Data\_Test)  
table(tree.pred, Occupancy\_Test1$Occupancy)

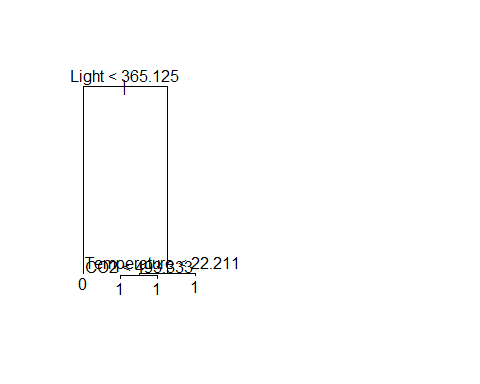
##   
## tree.pred 0 1  
## 0 1639 3  
## 1 54 969

mean(tree.pred != Occupancy\_Test1$Occupancy)

## [1] 0.02138837

confusionMatrix(Occupancy\_Test1$Occupancy, tree.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1639 54  
## 1 3 969  
##   
## Accuracy : 0.9786   
## 95% CI : (0.9724, 0.9838)  
## No Information Rate : 0.6161   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9544   
## Mcnemar's Test P-Value : 3.528e-11   
##   
## Sensitivity : 0.9982   
## Specificity : 0.9472   
## Pos Pred Value : 0.9681   
## Neg Pred Value : 0.9969   
## Prevalence : 0.6161   
## Detection Rate : 0.6150   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.9727   
##   
## 'Positive' Class : 0   
##

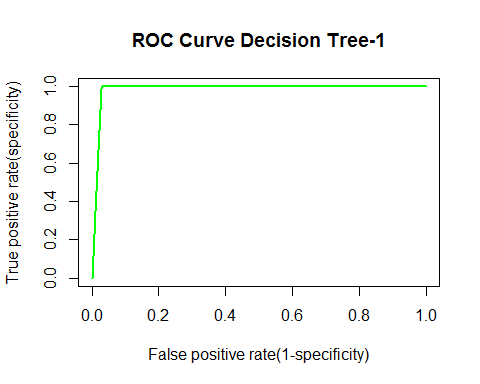
  
 #ROC Analysis Decision Trees - Testing Data 1

tree.pred = predict(prune.Occupancy\_Train, Occupancy\_Test1, type = "vector")  
#tree.prob <- attr(tree.pred, "vector")   
roc.curve=function(s,print=FALSE){  
Ps=(tree.pred[,2]>s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 0))/sum(Occupancy\_Test1$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 1))/sum(Occupancy\_Test1$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test1$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 1639 54  
## 1 3 969

## FPR TPR   
## 0.03189604 0.99691358

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,],xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)', main = 'ROC Curve Decision Tree-1', col="green",lwd=2,type="l")



# CART - Testing Data 2

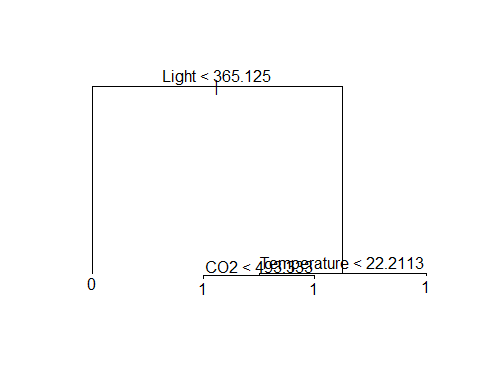
set.seed(2)  
Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
str(Occupancy\_Train)

## 'data.frame': 8143 obs. of 7 variables:  
## $ date : Factor w/ 8143 levels "2015-02-04 17:51:00",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ Temperature : num 23.2 23.1 23.1 23.1 23.1 ...  
## $ Humidity : num 27.3 27.3 27.2 27.2 27.2 ...  
## $ Light : num 426 430 426 426 426 ...  
## $ CO2 : num 721 714 714 708 704 ...  
## $ HumidityRatio: num 0.00479 0.00478 0.00478 0.00477 0.00476 ...  
## $ Occupancy : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

tree.Occupancy\_Train <- tree(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train)  
plot(tree.Occupancy\_Train)  
summary(tree.Occupancy\_Train)

##   
## Classification tree:  
## tree(formula = Occupancy ~ Temperature + Light + CO2 + HumidityRatio,   
## data = Occupancy\_Train)  
## Variables actually used in tree construction:  
## [1] "Light" "Temperature" "CO2"   
## Number of terminal nodes: 4   
## Residual mean deviance: 0.07946 = 646.8 / 8139   
## Misclassification error rate: 0.01216 = 99 / 8143

text(tree.Occupancy\_Train, pretty = 0)



tree.pred <- predict(tree.Occupancy\_Train, Occupancy\_Test2, type = "class")  
table(tree.pred, Occupancy\_Test2$Occupancy)

##   
## tree.pred 0 1  
## 0 7648 12  
## 1 55 2037

mean(tree.pred != Occupancy\_Test2$Occupancy)

## [1] 0.006870386

confusionMatrix(Occupancy\_Test2$Occupancy, tree.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7648 55  
## 1 12 2037  
##   
## Accuracy : 0.9931   
## 95% CI : (0.9913, 0.9947)  
## No Information Rate : 0.7855   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9795   
## Mcnemar's Test P-Value : 2.88e-07   
##   
## Sensitivity : 0.9984   
## Specificity : 0.9737   
## Pos Pred Value : 0.9929   
## Neg Pred Value : 0.9941   
## Prevalence : 0.7855   
## Detection Rate : 0.7842   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.9861   
##   
## 'Positive' Class : 0   
##

# Cross Validation and Pruning - Testing Data 2

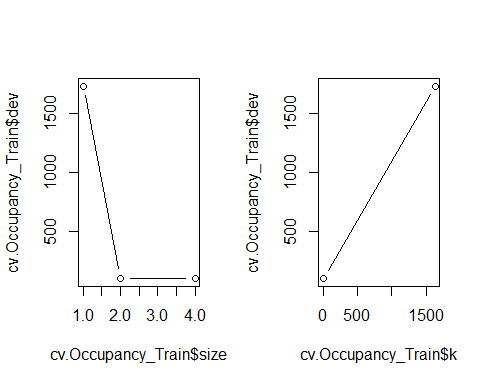
set.seed(3)  
cv.Occupancy\_Train <- cv.tree(tree.Occupancy\_Train, FUN=prune.misclass)  
names(cv.Occupancy\_Train)

## [1] "size" "dev" "k" "method"

cv.Occupancy\_Train

## $size  
## [1] 4 2 1  
##   
## $dev  
## [1] 101 101 1729  
##   
## $k  
## [1] -Inf 0 1630  
##   
## $method  
## [1] "misclass"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

par(mfrow = c(1,2))  
plot(cv.Occupancy\_Train$size, cv.Occupancy\_Train$dev, type = "b")  
plot(cv.Occupancy\_Train$k, cv.Occupancy\_Train$dev, type = "b")



prune.Occupancy\_Train = prune.misclass(tree.Occupancy\_Train, best = 9)

## Warning in prune.tree(tree = tree.Occupancy\_Train, best = 9, method =  
## "misclass"): best is bigger than tree size

#prune.mt2Data\_Train = prune.misclass(tree.mt2Data\_Train)  
plot(prune.Occupancy\_Train)  
text(prune.Occupancy\_Train, pretty = 0)  
tree.pred = predict(prune.Occupancy\_Train, Occupancy\_Test2, type = "class")  
#tree.pred = predict(prune.mt2Data\_Train, mt2Data\_Test)  
table(tree.pred, Occupancy\_Test2$Occupancy)

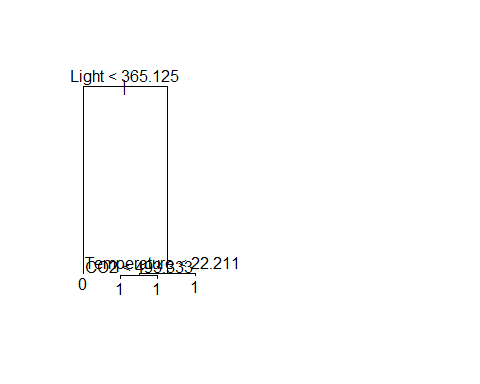
##   
## tree.pred 0 1  
## 0 7648 12  
## 1 55 2037

mean(tree.pred != Occupancy\_Test2$Occupancy)

## [1] 0.006870386

confusionMatrix(Occupancy\_Test2$Occupancy, tree.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7648 55  
## 1 12 2037  
##   
## Accuracy : 0.9931   
## 95% CI : (0.9913, 0.9947)  
## No Information Rate : 0.7855   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9795   
## Mcnemar's Test P-Value : 2.88e-07   
##   
## Sensitivity : 0.9984   
## Specificity : 0.9737   
## Pos Pred Value : 0.9929   
## Neg Pred Value : 0.9941   
## Prevalence : 0.7855   
## Detection Rate : 0.7842   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.9861   
##   
## 'Positive' Class : 0   
##

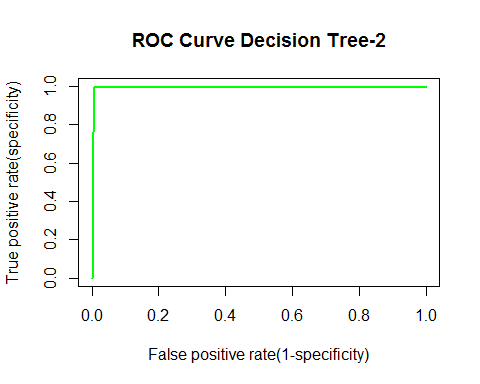
  
 #ROC Analysis Decision Trees - Testing Data 2

tree.pred = predict(prune.Occupancy\_Train, Occupancy\_Test2, type = "vector")  
#tree.prob <- attr(tree.pred, "vector")   
roc.curve=function(s,print=FALSE){  
Ps=(tree.pred[,2]>s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 0))/sum(Occupancy\_Test2$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 1))/sum(Occupancy\_Test2$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test2$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 7648 55  
## 1 12 2037

## FPR TPR   
## 0.007140075 0.994143485

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,],xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)', main = 'ROC Curve Decision Tree-2', col="green",lwd=2,type="l")

  
 ###KNN Testing Data 1 #KNN k = 1

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test1$Occupancy <- factor(Occupancy\_Test1$Occupancy)  
attach(Occupancy\_Train)

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test1))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 1, prob = TRUE)  
table(knn.pred, Occupancy\_Test1$Occupancy)

##   
## knn.pred 0 1  
## 0 1231 525  
## 1 462 447

mean(knn.pred != Occupancy\_Test1$Occupancy)

## [1] 0.3703565

confusionMatrix(Occupancy\_Test1$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1231 462  
## 1 525 447  
##   
## Accuracy : 0.6296   
## 95% CI : (0.611, 0.648)  
## No Information Rate : 0.6589   
## P-Value [Acc > NIR] : 0.99929   
##   
## Kappa : 0.1896   
## Mcnemar's Test P-Value : 0.04844   
##   
## Sensitivity : 0.7010   
## Specificity : 0.4917   
## Pos Pred Value : 0.7271   
## Neg Pred Value : 0.4599   
## Prevalence : 0.6589   
## Detection Rate : 0.4619   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.5964   
##   
## 'Positive' Class : 0   
##

# KNN k = 3

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test1$Occupancy <- factor(Occupancy\_Test1$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test1))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 3, prob = TRUE)  
table(knn.pred, Occupancy\_Test1$Occupancy)

##   
## knn.pred 0 1  
## 0 1231 523  
## 1 462 449

mean(knn.pred != Occupancy\_Test1$Occupancy)

## [1] 0.369606

confusionMatrix(Occupancy\_Test1$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1231 462  
## 1 523 449  
##   
## Accuracy : 0.6304   
## 95% CI : (0.6117, 0.6488)  
## No Information Rate : 0.6582   
## P-Value [Acc > NIR] : 0.99876   
##   
## Kappa : 0.1916   
## Mcnemar's Test P-Value : 0.05591   
##   
## Sensitivity : 0.7018   
## Specificity : 0.4929   
## Pos Pred Value : 0.7271   
## Neg Pred Value : 0.4619   
## Prevalence : 0.6582   
## Detection Rate : 0.4619   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.5973   
##   
## 'Positive' Class : 0   
##

# KNN k = 5

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test1$Occupancy <- factor(Occupancy\_Test1$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test1))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 5, prob = TRUE)  
table(knn.pred, Occupancy\_Test1$Occupancy)

##   
## knn.pred 0 1  
## 0 1240 524  
## 1 453 448

mean(knn.pred != Occupancy\_Test1$Occupancy)

## [1] 0.3666041

confusionMatrix(Occupancy\_Test1$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1240 453  
## 1 524 448  
##   
## Accuracy : 0.6334   
## 95% CI : (0.6148, 0.6517)  
## No Information Rate : 0.6619   
## P-Value [Acc > NIR] : 0.99908   
##   
## Kappa : 0.1964   
## Mcnemar's Test P-Value : 0.02512   
##   
## Sensitivity : 0.7029   
## Specificity : 0.4972   
## Pos Pred Value : 0.7324   
## Neg Pred Value : 0.4609   
## Prevalence : 0.6619   
## Detection Rate : 0.4653   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.6001   
##   
## 'Positive' Class : 0   
##

# KNN k = 10

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test1$Occupancy <- factor(Occupancy\_Test1$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test1))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 10, prob = TRUE)  
table(knn.pred, Occupancy\_Test1$Occupancy)

##   
## knn.pred 0 1  
## 0 1249 528  
## 1 444 444

mean(knn.pred != Occupancy\_Test1$Occupancy)

## [1] 0.364728

confusionMatrix(Occupancy\_Test1$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1249 444  
## 1 528 444  
##   
## Accuracy : 0.6353   
## 95% CI : (0.6167, 0.6536)  
## No Information Rate : 0.6668   
## P-Value [Acc > NIR] : 0.999719   
##   
## Kappa : 0.1982   
## Mcnemar's Test P-Value : 0.007763   
##   
## Sensitivity : 0.7029   
## Specificity : 0.5000   
## Pos Pred Value : 0.7377   
## Neg Pred Value : 0.4568   
## Prevalence : 0.6668   
## Detection Rate : 0.4687   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.6014   
##   
## 'Positive' Class : 0   
##

# KNN k = 25

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test1$Occupancy <- factor(Occupancy\_Test1$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 6):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test1))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 25, prob = TRUE)  
table(knn.pred, Occupancy\_Test1$Occupancy)

##   
## knn.pred 0 1  
## 0 1255 529  
## 1 438 443

mean(knn.pred != Occupancy\_Test1$Occupancy)

## [1] 0.3628518

confusionMatrix(Occupancy\_Test1$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1255 438  
## 1 529 443  
##   
## Accuracy : 0.6371   
## 95% CI : (0.6186, 0.6554)  
## No Information Rate : 0.6694   
## P-Value [Acc > NIR] : 0.999797   
##   
## Kappa : 0.2011   
## Mcnemar's Test P-Value : 0.003801   
##   
## Sensitivity : 0.7035   
## Specificity : 0.5028   
## Pos Pred Value : 0.7413   
## Neg Pred Value : 0.4558   
## Prevalence : 0.6694   
## Detection Rate : 0.4709   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.6032   
##   
## 'Positive' Class : 0   
##

# KNN k = 50

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test1$Occupancy <- factor(Occupancy\_Test1$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 6):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 7):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test1))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 50, prob = TRUE)  
table(knn.pred, Occupancy\_Test1$Occupancy)

##   
## knn.pred 0 1  
## 0 1290 551  
## 1 403 421

mean(knn.pred != Occupancy\_Test1$Occupancy)

## [1] 0.3579737

confusionMatrix(Occupancy\_Test1$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1290 403  
## 1 551 421  
##   
## Accuracy : 0.642   
## 95% CI : (0.6235, 0.6603)  
## No Information Rate : 0.6908   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.2016   
## Mcnemar's Test P-Value : 1.943e-06   
##   
## Sensitivity : 0.7007   
## Specificity : 0.5109   
## Pos Pred Value : 0.7620   
## Neg Pred Value : 0.4331   
## Prevalence : 0.6908   
## Detection Rate : 0.4841   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.6058   
##   
## 'Positive' Class : 0   
##

# KNN k = 100

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test1$Occupancy <- factor(Occupancy\_Test1$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 6):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 7):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 8):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test1))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 100, prob = TRUE)  
table(knn.pred, Occupancy\_Test1$Occupancy)

##   
## knn.pred 0 1  
## 0 1320 548  
## 1 373 424

mean(knn.pred != Occupancy\_Test1$Occupancy)

## [1] 0.345591

confusionMatrix(Occupancy\_Test1$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1320 373  
## 1 548 424  
##   
## Accuracy : 0.6544   
## 95% CI : (0.636, 0.6725)  
## No Information Rate : 0.7009   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.2245   
## Mcnemar's Test P-Value : 9.838e-09   
##   
## Sensitivity : 0.7066   
## Specificity : 0.5320   
## Pos Pred Value : 0.7797   
## Neg Pred Value : 0.4362   
## Prevalence : 0.7009   
## Detection Rate : 0.4953   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.6193   
##   
## 'Positive' Class : 0   
##

### KNN Testing Data 2

# KNN k = 1

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test2$Occupancy <- factor(Occupancy\_Test2$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 6):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 7):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 8):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 9):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test2))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 1, prob = TRUE)  
table(knn.pred, Occupancy\_Test2$Occupancy)

##   
## knn.pred 0 1  
## 0 6310 1209  
## 1 1393 840

mean(knn.pred != Occupancy\_Test2$Occupancy)

## [1] 0.2668171

confusionMatrix(Occupancy\_Test2$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 6310 1393  
## 1 1209 840  
##   
## Accuracy : 0.7332   
## 95% CI : (0.7243, 0.7419)  
## No Information Rate : 0.771   
## P-Value [Acc > NIR] : 1.0000000   
##   
## Kappa : 0.2218   
## Mcnemar's Test P-Value : 0.0003338   
##   
## Sensitivity : 0.8392   
## Specificity : 0.3762   
## Pos Pred Value : 0.8192   
## Neg Pred Value : 0.4100   
## Prevalence : 0.7710   
## Detection Rate : 0.6470   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.6077   
##   
## 'Positive' Class : 0   
##

# KNN k = 3

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test2$Occupancy <- factor(Occupancy\_Test2$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 6):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 7):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 8):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 9):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 10):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test2))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 3, prob = TRUE)  
table(knn.pred, Occupancy\_Test2$Occupancy)

##   
## knn.pred 0 1  
## 0 6314 1210  
## 1 1389 839

mean(knn.pred != Occupancy\_Test2$Occupancy)

## [1] 0.2665094

confusionMatrix(Occupancy\_Test2$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 6314 1389  
## 1 1210 839  
##   
## Accuracy : 0.7335   
## 95% CI : (0.7246, 0.7422)  
## No Information Rate : 0.7715   
## P-Value [Acc > NIR] : 1.0000000   
##   
## Kappa : 0.222   
## Mcnemar's Test P-Value : 0.0004802   
##   
## Sensitivity : 0.8392   
## Specificity : 0.3766   
## Pos Pred Value : 0.8197   
## Neg Pred Value : 0.4095   
## Prevalence : 0.7715   
## Detection Rate : 0.6475   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.6079   
##   
## 'Positive' Class : 0   
##

# KNN k = 5

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test2$Occupancy <- factor(Occupancy\_Test2$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 6):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 7):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 8):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 9):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 10):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 11):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test2))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 5, prob = TRUE)  
table(knn.pred, Occupancy\_Test2$Occupancy)

##   
## knn.pred 0 1  
## 0 6966 1217  
## 1 737 832

mean(knn.pred != Occupancy\_Test2$Occupancy)

## [1] 0.2003692

confusionMatrix(Occupancy\_Test2$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 6966 737  
## 1 1217 832  
##   
## Accuracy : 0.7996   
## 95% CI : (0.7915, 0.8075)  
## No Information Rate : 0.8391   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.3396   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8513   
## Specificity : 0.5303   
## Pos Pred Value : 0.9043   
## Neg Pred Value : 0.4061   
## Prevalence : 0.8391   
## Detection Rate : 0.7143   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.6908   
##   
## 'Positive' Class : 0   
##

# KNN k = 10

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test2$Occupancy <- factor(Occupancy\_Test2$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 6):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 7):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 8):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 9):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 10):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 11):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 12):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test2))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 10, prob = TRUE)  
table(knn.pred, Occupancy\_Test2$Occupancy)

##   
## knn.pred 0 1  
## 0 6988 1219  
## 1 715 830

mean(knn.pred != Occupancy\_Test2$Occupancy)

## [1] 0.1983183

confusionMatrix(Occupancy\_Test2$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 6988 715  
## 1 1219 830  
##   
## Accuracy : 0.8017   
## 95% CI : (0.7936, 0.8096)  
## No Information Rate : 0.8416   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.3432   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8515   
## Specificity : 0.5372   
## Pos Pred Value : 0.9072   
## Neg Pred Value : 0.4051   
## Prevalence : 0.8416   
## Detection Rate : 0.7166   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.6943   
##   
## 'Positive' Class : 0   
##

# KNN k = 25

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test2$Occupancy <- factor(Occupancy\_Test2$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 6):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 7):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 8):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 9):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 10):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 11):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 12):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 13):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test2))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 25, prob = TRUE)  
table(knn.pred, Occupancy\_Test2$Occupancy)

##   
## knn.pred 0 1  
## 0 6984 1247  
## 1 719 802

mean(knn.pred != Occupancy\_Test2$Occupancy)

## [1] 0.2015997

confusionMatrix(Occupancy\_Test2$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 6984 719  
## 1 1247 802  
##   
## Accuracy : 0.7984   
## 95% CI : (0.7903, 0.8063)  
## No Information Rate : 0.844   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.3292   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8485   
## Specificity : 0.5273   
## Pos Pred Value : 0.9067   
## Neg Pred Value : 0.3914   
## Prevalence : 0.8440   
## Detection Rate : 0.7162   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.6879   
##   
## 'Positive' Class : 0   
##

# KNN k = 50

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test2$Occupancy <- factor(Occupancy\_Test2$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 6):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 7):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 8):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 9):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 10):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 11):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 12):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 13):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 14):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test2))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 50, prob = TRUE)  
table(knn.pred, Occupancy\_Test2$Occupancy)

##   
## knn.pred 0 1  
## 0 7042 1277  
## 1 661 772

mean(knn.pred != Occupancy\_Test2$Occupancy)

## [1] 0.1987285

confusionMatrix(Occupancy\_Test2$Occupancy, knn.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7042 661  
## 1 1277 772  
##   
## Accuracy : 0.8013   
## 95% CI : (0.7932, 0.8092)  
## No Information Rate : 0.8531   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.327   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8465   
## Specificity : 0.5387   
## Pos Pred Value : 0.9142   
## Neg Pred Value : 0.3768   
## Prevalence : 0.8531   
## Detection Rate : 0.7221   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.6926   
##   
## 'Positive' Class : 0   
##

# KNN k = 100

Occupancy\_Train$Occupancy <- factor(Occupancy\_Train$Occupancy)  
Occupancy\_Test2$Occupancy <- factor(Occupancy\_Test2$Occupancy)  
attach(Occupancy\_Train)

## The following objects are masked from Occupancy\_Train (pos = 3):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 4):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 5):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 6):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 7):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 8):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 9):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 10):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 11):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 12):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 13):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 14):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following objects are masked from Occupancy\_Train (pos = 15):  
##   
## CO2, date, Humidity, HumidityRatio, Light, Occupancy,  
## Temperature

## The following object is masked from package:datasets:  
##   
## CO2

test.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Test2))]  
train.X = cbind(Temperature, Light, CO2, HumidityRatio)[as.numeric(rownames(Occupancy\_Train))]  
Occupancy\_Train.Occupancy = Occupancy[1:nrow(Occupancy\_Train)]  
set.seed(1)  
knn.pred = knn(data.frame(train.X), data.frame(test.X), Occupancy\_Train.Occupancy, k = 100, prob = TRUE)  
table(knn.pred, Occupancy\_Test2$Occupancy)

##   
## knn.pred 0 1  
## 0 7076 1272  
## 1 627 777

mean(knn.pred != Occupancy\_Test2$Occupancy)

## [1] 0.1947293

confusionMatrix(Occupancy\_Test2$Occupancy, knn.pred)

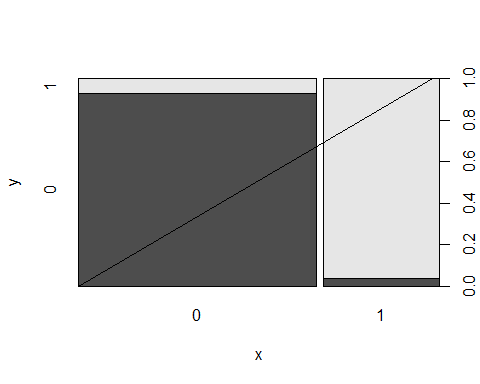
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7076 627  
## 1 1272 777  
##   
## Accuracy : 0.8053   
## 95% CI : (0.7973, 0.8131)  
## No Information Rate : 0.856   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.3367   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8476   
## Specificity : 0.5534   
## Pos Pred Value : 0.9186   
## Neg Pred Value : 0.3792   
## Prevalence : 0.8560   
## Detection Rate : 0.7256   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.7005   
##   
## 'Positive' Class : 0   
##

# Random Forest Bag - Testing Data 1

set.seed(123)  
bag.Occupancy\_Train = randomForest(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, importance = TRUE)  
bag.Occupancy\_Train

##   
## Call:  
## randomForest(formula = Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 0.6%  
## Confusion matrix:  
## 0 1 class.error  
## 0 6390 24 0.003741815  
## 1 25 1704 0.014459225

yhat.bag = predict(bag.Occupancy\_Train, Occupancy\_Test1)  
plot(yhat.bag, Occupancy\_Test1$Occupancy)  
abline(0,1)



mean((as.numeric(yhat.bag) - as.numeric(Occupancy\_Test1$Occupancy)) ^ 2)

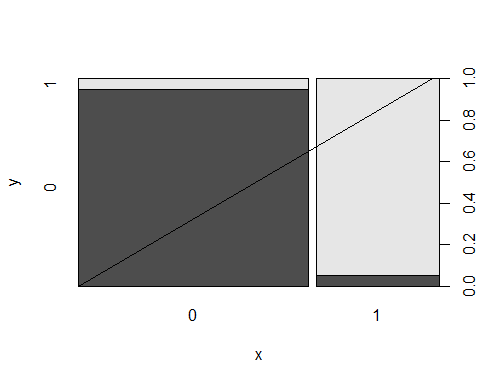
## [1] 0.06003752

# Random Forest Ntree

set.seed(123)  
bag.Occupancy\_Train = randomForest(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, ntree = 15)  
bag.Occupancy\_Train

##   
## Call:  
## randomForest(formula = Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, ntree = 15)   
## Type of random forest: classification  
## Number of trees: 15  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 0.71%  
## Confusion matrix:  
## 0 1 class.error  
## 0 6385 25 0.003900156  
## 1 33 1693 0.019119351

yhat.bag = predict(bag.Occupancy\_Train, Occupancy\_Test1)  
plot(yhat.bag, Occupancy\_Test1$Occupancy)  
abline(0,1)



mean((as.numeric(yhat.bag) - as.numeric(Occupancy\_Test1$Occupancy)) ^ 2)

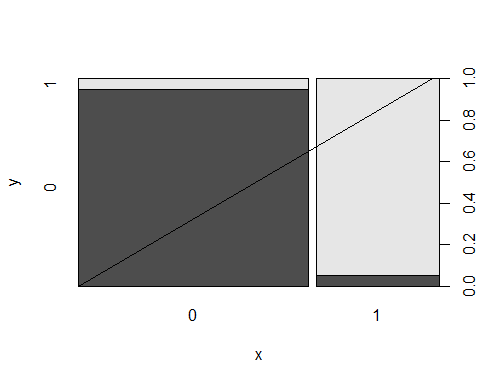
## [1] 0.05140713

# Random Forest

set.seed(123)  
rf.Occupancy\_Train = randomForest(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, importance = TRUE)  
rf.Occupancy\_Train

##   
## Call:  
## randomForest(formula = Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 0.6%  
## Confusion matrix:  
## 0 1 class.error  
## 0 6390 24 0.003741815  
## 1 25 1704 0.014459225

yhat.rf = predict(bag.Occupancy\_Train, Occupancy\_Test1)  
plot(yhat.rf, Occupancy\_Test1$Occupancy)  
abline(0,1)



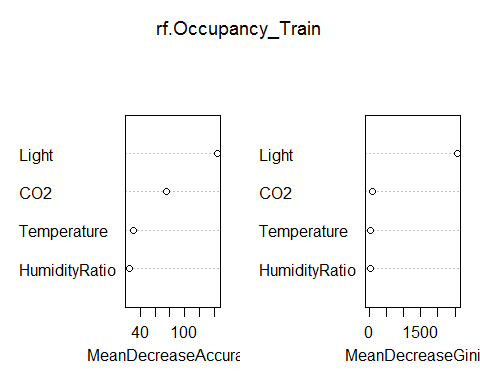
mean((as.numeric(yhat.rf) - as.numeric(Occupancy\_Test1$Occupancy)) ^ 2)

## [1] 0.05140713

importance(rf.Occupancy\_Train)

## 0 1 MeanDecreaseAccuracy MeanDecreaseGini  
## Temperature 37.41279 25.49205 31.54602 47.34943  
## Light 46.78300 454.59543 143.40751 2566.50569  
## CO2 26.80236 66.48899 74.95026 84.60173  
## HumidityRatio 26.44377 7.57482 25.56969 25.09048

varImpPlot(rf.Occupancy\_Train)

  
 #Random Forst Confusion Matrix

rf.pred <- predict(rf.Occupancy\_Train, Occupancy\_Test1, type = "class")  
table(rf.pred, Occupancy\_Test1$Occupancy)

##   
## rf.pred 0 1  
## 0 1663 130  
## 1 30 842

mean(rf.pred != Occupancy\_Test1$Occupancy)

## [1] 0.06003752

confusionMatrix(Occupancy\_Test1$Occupancy, rf.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1663 30  
## 1 130 842  
##   
## Accuracy : 0.94   
## 95% CI : (0.9303, 0.9487)  
## No Information Rate : 0.6728   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8675   
## Mcnemar's Test P-Value : 5.011e-15   
##   
## Sensitivity : 0.9275   
## Specificity : 0.9656   
## Pos Pred Value : 0.9823   
## Neg Pred Value : 0.8663   
## Prevalence : 0.6728   
## Detection Rate : 0.6240   
## Detection Prevalence : 0.6353   
## Balanced Accuracy : 0.9465   
##   
## 'Positive' Class : 0   
##

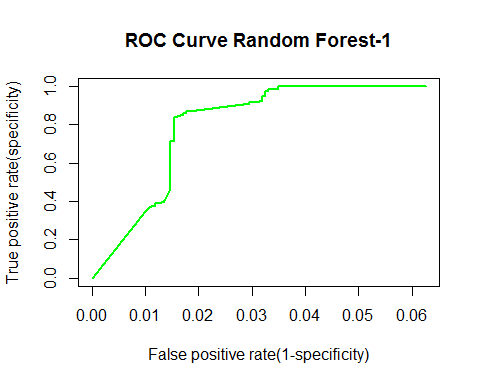
# ROC Analysis Random Forest

rf.pred = predict(rf.Occupancy\_Train, Occupancy\_Test1, type = "prob")  
#tree.prob <- attr(tree.pred, "vector")   
roc.curve=function(s,print=FALSE){  
Ps=(rf.pred[,2]>s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 0))/sum(Occupancy\_Test1$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test1$Occupancy == 1))/sum(Occupancy\_Test1$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test1$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 1663 30  
## 1 130 842

## FPR TPR   
## 0.01772002 0.86625514

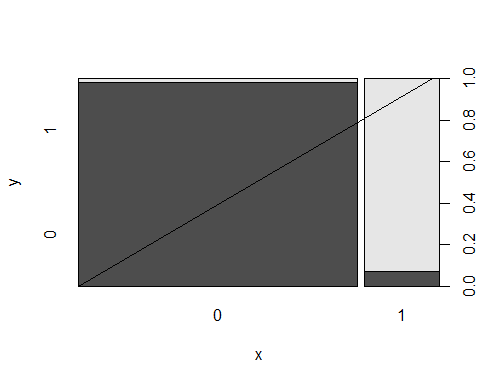
ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,],xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)', main = 'ROC Curve Random Forest-1', col="green",lwd=2,type="l")

  
 #Random Forest Bag - Testing Data 2

set.seed(123)  
bag.Occupancy\_Train = randomForest(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, importance = TRUE)  
bag.Occupancy\_Train

##   
## Call:  
## randomForest(formula = Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 0.6%  
## Confusion matrix:  
## 0 1 class.error  
## 0 6390 24 0.003741815  
## 1 25 1704 0.014459225

yhat.bag = predict(bag.Occupancy\_Train, Occupancy\_Test2)  
plot(yhat.bag, Occupancy\_Test2$Occupancy)  
abline(0,1)



mean((as.numeric(yhat.bag) - as.numeric(Occupancy\_Test2$Occupancy)) ^ 2)

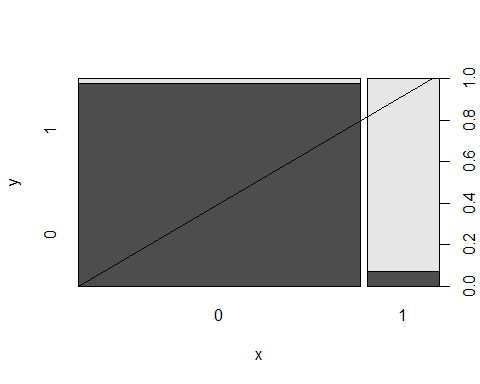
## [1] 0.0288146

# Random Forest Ntree

set.seed(123)  
bag.Occupancy\_Train = randomForest(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, ntree = 15)  
bag.Occupancy\_Train

##   
## Call:  
## randomForest(formula = Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, ntree = 15)   
## Type of random forest: classification  
## Number of trees: 15  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 0.71%  
## Confusion matrix:  
## 0 1 class.error  
## 0 6385 25 0.003900156  
## 1 33 1693 0.019119351

yhat.bag = predict(bag.Occupancy\_Train, Occupancy\_Test2)  
plot(yhat.bag, Occupancy\_Test2$Occupancy)  
abline(0,1)



mean((as.numeric(yhat.bag) - as.numeric(Occupancy\_Test2$Occupancy)) ^ 2)

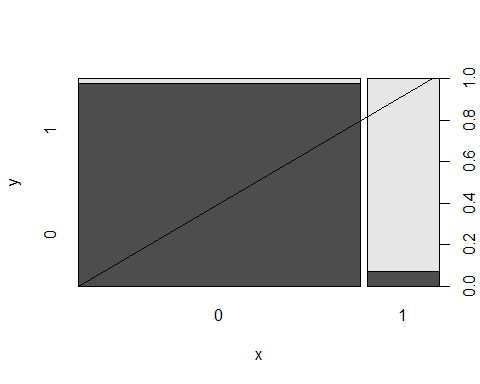
## [1] 0.03373667

# Random Forest

set.seed(123)  
rf.Occupancy\_Train = randomForest(Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, importance = TRUE)  
rf.Occupancy\_Train

##   
## Call:  
## randomForest(formula = Occupancy ~ Temperature + Light + CO2 + HumidityRatio, data = Occupancy\_Train, mtry = 4, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 0.6%  
## Confusion matrix:  
## 0 1 class.error  
## 0 6390 24 0.003741815  
## 1 25 1704 0.014459225

yhat.rf = predict(bag.Occupancy\_Train, Occupancy\_Test2)  
plot(yhat.rf, Occupancy\_Test2$Occupancy)  
abline(0,1)



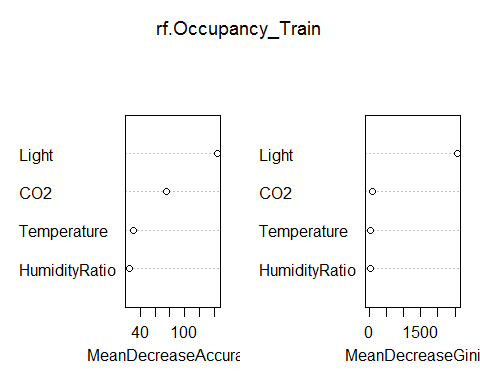
mean((as.numeric(yhat.rf) - as.numeric(Occupancy\_Test2$Occupancy)) ^ 2)

## [1] 0.03373667

importance(rf.Occupancy\_Train)

## 0 1 MeanDecreaseAccuracy MeanDecreaseGini  
## Temperature 37.41279 25.49205 31.54602 47.34943  
## Light 46.78300 454.59543 143.40751 2566.50569  
## CO2 26.80236 66.48899 74.95026 84.60173  
## HumidityRatio 26.44377 7.57482 25.56969 25.09048

varImpPlot(rf.Occupancy\_Train)

  
 #Random Forst Confusion Matrix

rf.pred <- predict(rf.Occupancy\_Train, Occupancy\_Test2, type = "class")  
table(rf.pred, Occupancy\_Test2$Occupancy)

##   
## rf.pred 0 1  
## 0 7554 132  
## 1 149 1917

mean(rf.pred != Occupancy\_Test2$Occupancy)

## [1] 0.0288146

confusionMatrix(Occupancy\_Test2$Occupancy, rf.pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7554 149  
## 1 132 1917  
##   
## Accuracy : 0.9712   
## 95% CI : (0.9677, 0.9744)  
## No Information Rate : 0.7881   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9135   
## Mcnemar's Test P-Value : 0.3398   
##   
## Sensitivity : 0.9828   
## Specificity : 0.9279   
## Pos Pred Value : 0.9807   
## Neg Pred Value : 0.9356   
## Prevalence : 0.7881   
## Detection Rate : 0.7746   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.9554   
##   
## 'Positive' Class : 0   
##

# ROC Analysis Random Forest

rf.pred = predict(rf.Occupancy\_Train, Occupancy\_Test2, type = "prob")  
#tree.prob <- attr(tree.pred, "vector")   
roc.curve=function(s,print=FALSE){  
Ps=(rf.pred[,2]>s)\*1  
FP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 0))/sum(Occupancy\_Test2$Occupancy == 0)  
TP=sum((Ps==1)\*(Occupancy\_Test2$Occupancy == 1))/sum(Occupancy\_Test2$Occupancy == 1)  
if(print==TRUE){  
print(table(Observed=Occupancy\_Test2$Occupancy,Predicted=Ps))  
}  
vect=c(FP,TP)  
names(vect)=c("FPR","TPR")  
return(vect)  
}  
threshold = 0.5  
roc.curve(threshold,print=TRUE)

## Predicted  
## Observed 0 1  
## 0 7554 149  
## 1 132 1917

## FPR TPR   
## 0.01934311 0.93557833

ROC.curve=Vectorize(roc.curve)  
M.ROC=ROC.curve(seq(0,1,by=.01))  
plot(M.ROC[1,],M.ROC[2,],xlab='False positive rate(1-specificity)', ylab='True positive rate(specificity)', main = 'ROC Curve Random Forest-2', col="green",lwd=2,type="l")

