Team14\_Project\_SVM

Prajwal Gonnade; Supreet Nayak

December 9, 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(e1071)

## Warning: package 'e1071' 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)

Occupancy\_subset=data.frame(x=Occupancy\_Train[,c(2,4,5,6)], y=as.factor(Occupancy\_Train[,7]))  
Occupancy.svm=svm(y~., data=Occupancy\_subset, kernel="linear",cost=10)  
summary(Occupancy.svm)

##   
## Call:  
## svm(formula = y ~ ., data = Occupancy\_subset, kernel = "linear",   
## cost = 10)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 10   
## gamma: 0.25   
##   
## Number of Support Vectors: 325  
##   
## ( 162 163 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## 0 1

table(Occupancy.svm$fitted, Occupancy\_subset$y)

##   
## 0 1  
## 0 6321 21  
## 1 93 1708

Occupancy\_subset.te=data.frame(x=Occupancy\_Test1[,c(2,4,5,6)], y=as.factor(Occupancy\_Test1[,7]))  
pred.te=predict(Occupancy.svm, newdata=Occupancy\_subset.te,decision.values=TRUE)  
Occupancy.svm.probs<-attr(pred.te,"decision.values")  
table(pred.te, Occupancy\_subset.te$y)

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

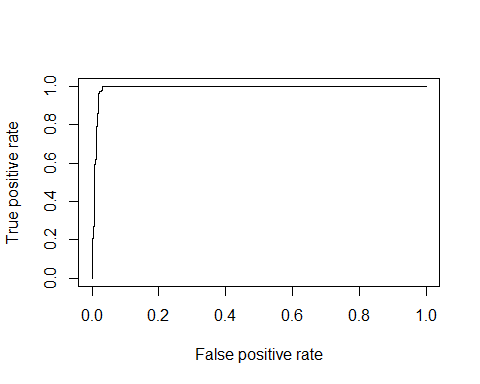
Occupancy.svm.confusion <- confusionMatrix(Occupancy\_subset.te$y, pred.te)  
Occupancy.svm.confusion

## 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   
##

Occupancy.svm.accuracy <- mean(pred.te == Occupancy\_subset.te$y)  
Occupancy.svm.accuracy

## [1] 0.9786116

Occupancy.svm.prediction<-prediction(Occupancy.svm.probs,Occupancy\_subset.te$y)  
Occupancy.svm.performance<-performance(Occupancy.svm.prediction,"tpr","fpr")  
Occupancy.svm.auc<-performance(Occupancy.svm.prediction,"auc")@y.values[[1]]  
plot(Occupancy.svm.performance)



Occupancy\_subset.te=data.frame(x=Occupancy\_Test2[,c(2,4,5,6)], y=as.factor(Occupancy\_Test2[,7]))  
pred.te=predict(Occupancy.svm, newdata=Occupancy\_subset.te,decision.values=TRUE)  
Occupancy.svm.probs<-attr(pred.te,"decision.values")  
table(pred.te, Occupancy\_subset.te$y)

##   
## pred.te 0 1  
## 0 7644 14  
## 1 59 2035

Occupancy.svm.confusion <- confusionMatrix(Occupancy\_subset.te$y, pred.te)  
Occupancy.svm.confusion

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 7644 59  
## 1 14 2035  
##   
## Accuracy : 0.9925   
## 95% CI : (0.9906, 0.9941)  
## No Information Rate : 0.7853   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9776   
## Mcnemar's Test P-Value : 2.607e-07   
##   
## Sensitivity : 0.9982   
## Specificity : 0.9718   
## Pos Pred Value : 0.9923   
## Neg Pred Value : 0.9932   
## Prevalence : 0.7853   
## Detection Rate : 0.7838   
## Detection Prevalence : 0.7899   
## Balanced Accuracy : 0.9850   
##   
## 'Positive' Class : 0   
##

Occupancy.svm.accuracy <- mean(pred.te == Occupancy\_subset.te$y)  
Occupancy.svm.accuracy

## [1] 0.9925144

Occupancy.svm.prediction<-prediction(Occupancy.svm.probs,Occupancy\_subset.te$y)  
Occupancy.svm.performance<-performance(Occupancy.svm.prediction,"tpr","fpr")  
Occupancy.svm.auc<-performance(Occupancy.svm.prediction,"auc")@y.values[[1]]  
plot(Occupancy.svm.performance)

