Team14\_Project\_Perceptron

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(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)

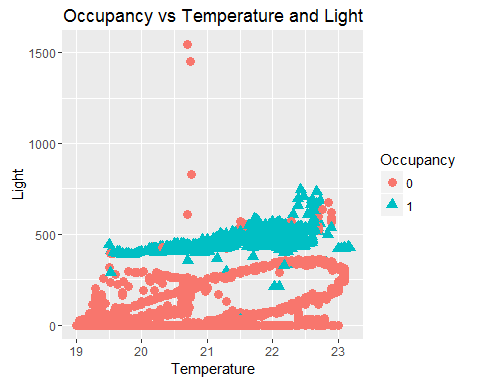
## Warning: package 'chemometrics' was built under R version 3.2.5

library(reshape2)

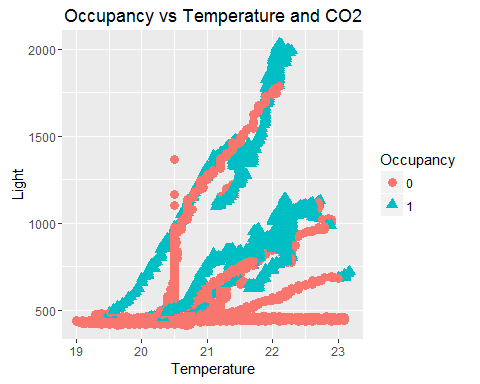
## Warning: package 'reshape2' was built under R version 3.2.5

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

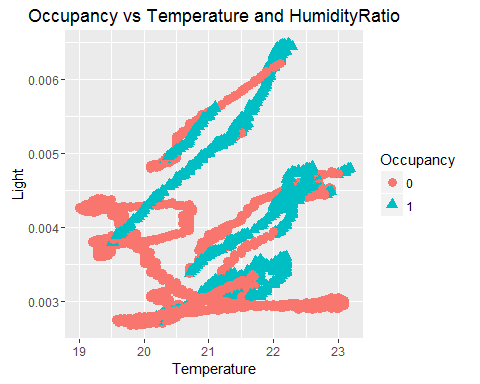
Occupancy\_Train$Occupancy <- as.factor(Occupancy\_Train$Occupancy)  
ggplot(Occupancy\_Train, aes(x = Temperature, y = Light)) +   
 geom\_point(aes(colour=Occupancy, shape=Occupancy), size = 3) +  
 xlab("Temperature") +   
 ylab("Light") +   
 ggtitle("Occupancy vs Temperature and Light")



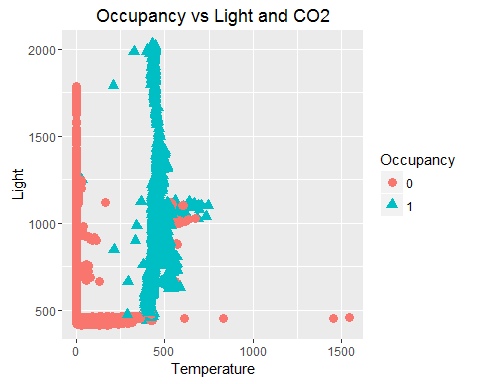
ggplot(Occupancy\_Train, aes(x = Temperature, y = CO2)) +   
 geom\_point(aes(colour=Occupancy, shape=Occupancy), size = 3) +  
 xlab("Temperature") +   
 ylab("Light") +   
 ggtitle("Occupancy vs Temperature and CO2")



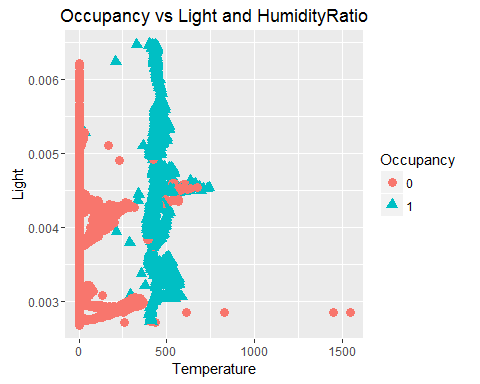
ggplot(Occupancy\_Train, aes(x = Temperature, y = HumidityRatio)) +   
 geom\_point(aes(colour=Occupancy, shape=Occupancy), size = 3) +  
 xlab("Temperature") +   
 ylab("Light") +   
 ggtitle("Occupancy vs Temperature and HumidityRatio")



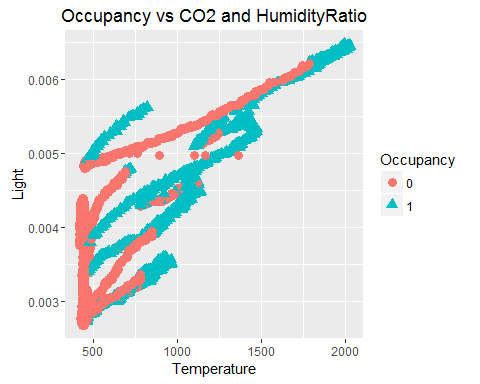
ggplot(Occupancy\_Train, aes(x = Light, y = CO2)) +   
 geom\_point(aes(colour=Occupancy, shape=Occupancy), size = 3) +  
 xlab("Temperature") +   
 ylab("Light") +   
 ggtitle("Occupancy vs Light and CO2")



ggplot(Occupancy\_Train, aes(x = Light, y = HumidityRatio)) +   
 geom\_point(aes(colour=Occupancy, shape=Occupancy), size = 3) +  
 xlab("Temperature") +   
 ylab("Light") +   
 ggtitle("Occupancy vs Light and HumidityRatio")



ggplot(Occupancy\_Train, aes(x = CO2, y = HumidityRatio)) +   
 geom\_point(aes(colour=Occupancy, shape=Occupancy), size = 3) +  
 xlab("Temperature") +   
 ylab("Light") +   
 ggtitle("Occupancy vs CO2 and HumidityRatio")

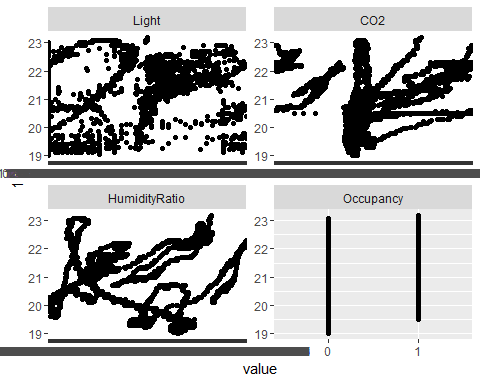


#Occupancy\_Train$Occupancy <- as.integer(Occupancy\_Train$Occupancy)

# Perceptron

Occupancy\_subset <- Occupancy\_Train[,c(2,4,5,6,7)]  
Occupancy\_response <- Occupancy\_Test2[,7]  
feature\_plot <- function (Occupancy\_subset, Occupancy\_response) {  
 mtmelt <<- melt(Occupancy\_subset, id.vars = Occupancy\_response)  
 p <- ggplot(mtmelt, aes(x = value, y = mtmelt[, 5])) +  
 facet\_wrap(~variable, scales = "free") +  
 geom\_point() +  
 labs(y = Occupancy\_response)  
 p  
}  
  
feature\_plot(Occupancy\_subset, Occupancy\_response)

## Warning: attributes are not identical across measure variables; they will  
## be dropped



Occupancy\_subset[, 6] <- 1  
Occupancy\_subset[Occupancy\_subset[, 5] == 0, 6] <- -1  
x <- Occupancy\_subset[, c(1,2,3,4)]  
y <- Occupancy\_subset[, 6]  
head(x)

## Temperature Light CO2 HumidityRatio  
## 1 23.18 426.0 721.25 0.004792988  
## 2 23.15 429.5 714.00 0.004783441  
## 3 23.15 426.0 713.50 0.004779464  
## 4 23.15 426.0 708.25 0.004771509  
## 5 23.10 426.0 704.50 0.004756993  
## 6 23.10 419.0 701.00 0.004756993

head(y)

## [1] 1 1 1 1 1 1

perceptron <- function(x, y, eta, niter) {  
  
 # initialize weight vector  
 weight <- rep(0, dim(x)[2] + 1)  
 errors <- rep(0, niter)  
  
 # loop over number of epochs niter  
 for (jj in 1:niter) {  
  
 # loop through training data set  
 for (ii in 1:length(y)) {  
  
 # Predict binary label using Heaviside activation  
 # function  
 z <- sum(weight[2:length(weight)] \*  
 as.numeric(x[ii, ])) + weight[1]  
 if(z < 0) {  
 ypred <- -1  
 } else {  
 ypred <- 1  
 }  
  
 # Change weight - the formula doesn't do anything  
 # if the predicted value is correct  
 weightdiff <- eta \* (y[ii] - ypred) \*  
 c(1, as.numeric(x[ii, ]))  
 weight <- weight + weightdiff  
  
 # Update error function  
 if ((y[ii] - ypred) != 0.0) {  
 errors[jj] <- errors[jj] + 1  
 }  
  
 }  
 }  
  
 # weight to decide between the two species  
 print(weight)  
 return(errors)  
}  
  
err <- perceptron(x, y, 0.1, 5)

## [1] 4.000000e-01 4.387833e+00 2.891000e+02 4.254667e+01 5.286345e-03

plot(0.1:5, err, type="l", lwd=2, col="red", xlab="epoch #", ylab="errors")  
title("Errors vs epoch - learning rate eta = 0.1")

