library(caret)

library(neuralnet)

library(vcd)

data(shuttle)

str(shuttle)

table(shuttle$use)

table1=structable(wind+magn~use, shuttle)

table1

mosaic(table1, gp=shading\_hcl)

mosaic(use~error+vis, shuttle)

table(shuttle$use, shuttle$stability)

prop.table(table(shuttle$use, shuttle$stability))

chisq.test(shuttle$use, shuttle$stability)

dummies = dummyVars(use~.,shuttle, fullRank=TRUE)

dummies

shuttle.2 = as.data.frame(predict(dummies, newdata=shuttle))

names(shuttle.2)

head(shuttle.2)

shuttle.2$use = ifelse(shuttle$use=="auto",1,0)

table(shuttle.2$use)

set.seed(123)

trainIndex = createDataPartition(shuttle.2$use, p = .7,

list = FALSE,

times = 1)

head(trainIndex)

shuttleTrain = shuttle.2[ trainIndex,]

shuttleTest = shuttle.2[-trainIndex,]

table(shuttleTrain$use)

table(shuttleTest$use)

n = names(shuttleTrain)

n

form = as.formula(paste("use ~", paste(n[!n %in% "use"], collapse = " + ")))

form

fit = neuralnet(form, data=shuttleTrain, err.fct="ce", linear.output=FALSE)

fit$result.matrix

head(fit$generalized.weights[[1]])

plot(fit)

par(mfrow=c(1,2))

gwplot(fit, selected.covariate = "vis.yes")

gwplot(fit, selected.covariate = "wind.tail")

res = compute(fit, shuttleTrain[,1:10])

predTrain = res$net.result

predTrain = ifelse(predTrain>=0.5,1,0)

table(predTrain, shuttleTrain$use)

res2 = compute(fit, shuttleTest[,1:10])

predTest = res2$net.result

predTest = ifelse(predTest>=0.5,1,0)

table(predTest, shuttleTest$use)

which(predTest==1 & shuttleTest$use==0)

shuttleTest[62,]

fit2 = neuralnet(form, data=shuttleTrain, hidden=c(3,2), err.fct="ce", linear.output=FALSE)

plot(fit2)

res = compute(fit2, shuttleTrain[,1:10])

predTrain = res$net.result

predTrain = ifelse(predTrain>=0.5,1,0)

table(predTrain, shuttleTrain$use)

res2 = compute(fit2, shuttleTest[,1:10])

predTest = res2$net.result

predTest = ifelse(predTest>=0.5,1,0)

table(predTest, shuttleTest$use)

which(predTest==1 & shuttleTest$use==0)

which(predTest==0 & shuttleTest$use==1)

##### Deep Learning

data(Pima.tr)

data(Pima.te)

pima = rbind(Pima.tr, Pima.te)

pima.scale = as.data.frame(scale(pima[,-8]))

pima.scale$type = pima$type

str(pima.scale)

getwd()

write.csv(pima.scale, file="pimaScale.csv", row.names=FALSE)

library(h2o)

localH2O = h2o.init()

h2o.getConnection()

path = "C:/Users/clesmeister/chap7 NN/pimaScale.csv"

pima.hex = h2o.uploadFile(path=path, destination\_frame="pima.hex")

class(pima.hex)

head(pima.hex)

summary(pima.hex)

rand = h2o.runif(pima.hex, seed = 123)

train = pima.hex[rand <= 0.7, ]

train = h2o.assign(train, key = "train")

test = pima.hex[rand > 0.7, ]

test <- h2o.assign(test, key = "test")

h2o.table(train[,8])

h2o.table(test[,8])

args(h2o.deeplearning)

demo(h2o.deeplearning)

dlmodel <- h2o.deeplearning(x=1:7, y=8, training\_frame = train,

validation\_frame = test,

seed=123,

variable\_importances = TRUE,

hidden=c(100,100))

dlmodel

dlmodel@allparameters

dlmodel@model$variable\_importances

dlPredict = h2o.predict(dlmodel,newdata=test)

dlPredict

dlPred = as.data.frame(dlPredict)

head(dlPred)