library(frbs)

library(haven)

library(foreign)

zz <- file.path("U:","My Documents")

ANFloc <- file.path(zz,"ANF.sav")

ANF <- read.spss(ANFloc)

ANF=data.frame(ANF)

ANF$BeckAnxiety=as.factor(ANF$BeckAnxiety)

ANFShuffled <- ANF[sample(nrow(ANF)),]

ANFShuffled[,3] <- unclass(ANFShuffled[,3])

tra.ANF <- ANFShuffled[1:500,]

data.fit <- tra.ANF[, 1 : 2]

tst.ANF <- ANFShuffled[501:nrow(ANFShuffled),1:2]

real.ANF <- matrix(ANFShuffled[501:nrow(ANFShuffled),3], ncol = 1)

range.data<-matrix(c(4,14,2,14,2,67), nrow=2)

method.type <- "ANFIS"

control <- list(num.labels = 5, max.iter = 5, step.size = 0.01, type.tnorm = "MIN", typ.mf=4,type.snorm = "MAX", type.implication.func = "ZADEH", name = "ARTIFICIAL SOCIAL PSYCHOLOGIST")

object <- frbs.learn(tra.ANF, range.data, method.type, control)

objPMML <- frbsPMML(object)

write.frbsPMML(objPMML, file = "ANFIS.ARTIFICIAL SOCIAL PSYCHOLOGIST")

object.pmml <- read.frbsPMML("ANFIS.ARTIFICIAL SOCIAL PSYCHOLOGIST.frbsPMML")

res.fit <- predict(object.pmml, data.fit)

res.test <- predict(object.pmml, tst.ANF)

y.pred <- res.test

y.real <- real.ANF

bench <- cbind(y.pred, y.real)

colnames(bench) <- c("pred. val.", "real. val.")

print("Comparison ANFIS Vs Real Value on ANFIS.ARTIFICIAL SOCIAL PSYCHOLOGIST")

print(bench)

residuals <- (y.real - y.pred)

MSE <- mean(residuals^2)

RMSE <- sqrt(mean(residuals^2))

SMAPE <- mean(abs(residuals)/(abs(y.real) + abs(y.pred))/2)\*100

err <- c(MSE, RMSE, SMAPE)

names(err) <- c("MSE", "RMSE", "SMAPE")

print("Error Measurement: ")

print(err)

op <- par(mfrow = c(2, 1))

x1 <- seq(from = 1, to = nrow(res.fit))

result.fit <- cbind(tra.ANF[, 3], res.fit)

plot(x1, result.fit[, 1], col="red", main = "(( the training data(red) Vs Sim. result(blue))", type = "l", ylab = "Anxiety")

lines(x1, result.fit[, 2], col="blue")

result.test <- cbind(y.real, y.pred)

x2 <- seq(from = 1, to = nrow(result.test))

plot(x2 , result.test[, 1], col="red", main = " ((Real Data(red) Vs Sim. result(blue))", type = "l", ylab = "Anxiety")

lines(x2, result.test[, 2], col="blue", type = "l")

par(op)

par(op)

plotMF(object)

summary(object)