MSDS_600_Week_5_MLinR.R

SeanOMalley1

Sat Jul 30 10:21:14 2016

```
# MSDS Week 5
# Sean O'Malley

library("e1071")
```

```
## Warning: package 'e1071' was built under R version 3.1.3
```

```
library(rpart)

# 4x2 array with the input values
x <- array(data=c(0,0,1,1,0,1,0,1),dim=c(4,2))
x
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    0    1
## [3,]    1    0
## [4,]    1    1
```

```
# Vector of factors
y <- factor(c(0,1,1,0))
y
```

```
## [1] 0 1 1 0
## Levels: 0 1
```

```
# SVM Practice Model 1
# Support vector machine is a discriminative classifier formally defined by a separat
ing hyperplane
# Given labeled training data, the algorithm outputs an optimal hyperplane which cate
gorizes new examples
?svm
model1 <- svm(x,y,type="C-classification")
summary(model1)
```

```
##
## Call:
## svm.default(x = x, y = y, type = "C-classification")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.5
##
## Number of Support Vectors:  4
##
##  ( 2 2 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
print(model1)
```

```
##
## Call:
## svm.default(x = x, y = y, type = "C-classification")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.5
##
## Number of Support Vectors:  4
```

```
predict(model1,x)
```

```
## 1 2 3 4
## 0 1 1 0
## Levels: 0 1
```

```
##Classification Model in rpart and svm

# Attain dataset, set test and train groups
data(Glass,package="mlbench")
index2 <- 1:nrow(Glass)
testindex2 <- sample(index2, trunc(length(index2)/3))
testset2 <- Glass[testindex2,]
trainset2 <- Glass[-testindex2,]

# predict with svm
svm_model2 <- svm(Type~., data=trainset2, cost=100, gamma=1)
svm_model2_predict <- predict(svm_model2, testset2[,-10])

# predict with rpart
rpart_model2 <- rpart(Type~., data=trainset2)
rpart_model2_predict <- predict(rpart_model2, testset2[,-10], type="class")

# svm confusion matrix
table(pred=svm_model2_predict, true=testset2[,10])
```

```
##     true
## pred  1  2  3  5  6  7
##    1 13  5  3  0  1  0
##    2  6 19  3  4  4  4
##    3  2  1  0  0  0  0
##    5  0  0  0  1  0  0
##    6  0  0  0  0  0  0
##    7  0  0  0  0  0  5
```

```
# rpart confusion matrix
table(pred=rpart_model2_predict, true=testset2[,10])
```

```
##     true
## pred  1  2  3  5  6  7
##    1 14  4  2  0  1  1
##    2  5 20  2  5  3  1
##    3  1  1  2  0  1  0
##    5  0  0  0  0  0  0
##    6  0  0  0  0  0  0
##    7  1  0  0  0  0  7
```

```
## Nonlinear e-Regression with svm and rpart

# Attain dataset, set test and train groups
data(Ozone, package="mlbench")
index3 <- 1:nrow(Ozone)
testindex3 <- sample(index3, trunc(length(index3)/3))
testset3 <- na.omit(Ozone[testindex3,-3])
trainset3 <- na.omit(Ozone[-testindex3,-3])

# SVM e-Regression
svm_model3 <- svm(V4~., data=trainset3, cost=1000, gamma=0.0001)
svm_model3_predict <- predict(svm_model3, testset3[,-3])
crossprod(svm_model3_predict - testset3[,3]) / length(testindex3)
```

```
##          [,1]
## [1,] 14.13241
```

```
# Rpart e-Regression
rpart_model3 <- rpart(V4~., data=trainset3)
rpart_model3_predict <- predict(rpart_model3, testset3[,-3])
crossprod(rpart_model3_predict - testset3[,3]) / length(testindex3)
```

```
##          [,1]
## [1,] 27.85237
```

```
## SVM Iris Example
attach(iris)

# divide into x and y classes
x <- subset(iris, select=-Species)
y <- Species

# create SVM model and show sumary
svm_model4 <- svm(Species~., data=iris)
summary(svm_model3)
```

```
##
## Call:
## svm(formula = V4 ~ ., data = trainset3, cost = 1000, gamma = 1e-04)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1000
##       gamma:  1e-04
##     epsilon:  0.1
##
##
## Number of Support Vectors:  116
```

```
# predict svm model 4
svm_model4_predict <- predict(svm_model4,x)

# view in table
table(svm_model4_predict,y)
```

```
##                   y
## svm_model4_predict setosa versicolor virginica
##         setosa         50          0         0
##         versicolor      0         48         2
##         virginica       0          2        48
```