Q37-WilliamKennedy-300015367

William Kennedy

2023-11-21

1. Create a binary variable that takes on a 1 for cars with gas mileage above the median, and a 0 for cars with gas mileage below the median.

```
Auto = read.csv("Auto.csv")
mpg.med = median(Auto$mpg)
Auto$mpg01 = ifelse(Auto$mpg>mpg.med,1,0)
Auto = subset(Auto, select=(-name))
```

2. Fit a support vector classifier to the data with various values of cost, in order to predict whether a car gets high or low gas mileage. Report the cross-validation errors associated with different values of this parameter. Comment on your results.

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.2
```

```
set.seed(0)
ind.train = sample(nrow(Auto),
round(0.7*nrow(Auto)),
replace=FALSE)

Auto.tr = Auto[ind.train,]
Auto.te = Auto[-ind.train,]
y = Auto.tr[,c("mpg01")]
Auto.tr.1 = subset(Auto.tr, select = -c(mpg))
dat = data.frame(Auto.tr.1,y)

svm.lin = tune(svm, y~., data = dat, kernel = "linear",
ranges = list(cost = c(0.001, 0.01, 0.1,1, 5, 10, 100)))
best.svm.lin = svm.lin$best.model

results = predict(best.svm.lin,dat)
summary(best.svm.lin)
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = y ~ ., data = dat, ranges = list(cost = c(0.001,
##     0.01, 0.1, 1, 5, 10, 100)), kernel = "linear")
##
##
```

```
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  1
##       gamma:  0.125
##     epsilon:  0.1
##
##
## Number of Support Vectors:  25
```

When the cost is greater than or to 1, the CVR for K=10 stays the same. So to reduce complexity it would be best to keep the cost function at C=1.

3.

```
library(kernlab)

svm.rad = tune(svm , y~., data = dat,
kernel = "radial",
ranges = list (cost = c(0.01, .01, 1, 10),
gamma = c(0.5, 1, 2, 3)))

svm.poly = tune(svm , y~., data = dat,
kernel = "polynomial",
ranges = list (cost = c(0.01, .01, 1, 10),
gamma = c(0.5, 1, 2, 3),
degree = c(1,2,3)))

best.svm.rad = svm.rad$best.model
best.svm.poly = svm.poly$bestmodel

summary(best.svm.rad)
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = y ~ ., data = dat, ranges = list(cost = c(0.01,
##     0.01, 1, 10), gamma = c(0.5, 1, 2, 3)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  10
##       gamma:  0.5
##     epsilon:  0.1
##
##
## Number of Support Vectors:  96
```

```
summary(best.svm.poly)
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

When the kernel=radial, the smallest errors occur when cost=10 and gamma=0.5, however the difference between cost=1 and cost=10 for gamma=0.5 is a difference less than 0.001. So for the sake of simplicity it is best to go with cost=1 and gamma=0.5.

When kernel=polynomial, the having when degree=1, cost = 0.01, and gamma=2 this is the smallest error in which these parameters are the most optimal 4.

```
plot(best.svm.lin,data=Auto, mpg01~Lag1)
plot(best.svm.rad,data=Auto, mpg01~Lag1)
```

The functions run without errors but my graphs won't plot anything, I'm not sure what's wrong with them.