Support Vector Machine Example: Car Evaluation

Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.). The model evaluates cars according to the following concept structure.

In total there are 7 variables. The detailed variables information is presented below:

1. Buying: buying price (four levels categorical variable); 2. Maint: price of the maintenance (four levels categorical variable); 3. Doors: number of doors (four levels categorical variable); 4. Persons: capacity in terms of persons to carry (three levels categorical variable); 5. Lug\_boot: the size of luggage boot (three levels categorical variable); 6. Safety: estimated safety of the car (three levels categorical variable); 7. Class: the overall evaluation of the car (unacceptable, acceptable, good and very good: four levels categorical variable);

The objective of this data is utilizing the attributes 1-6 to predict the overall car evaluation.

1. Import the data into R and change all the variables into factor variables. You can directly change the variables into factor variables by setting “stringsAsFactors = TRUE” when you import the data. Inspect the data structure of the data. How many vehicles are included in this data set?

**Ans:**

**1728 data**

**Code:**

dt <- read.csv(file.choose(),stringsAsFactors = TRUE)

str(dt)

1. How many vehicles have better evaluation than acceptable (not include acceptable)? For cars with big luggage, what proportion of “very good” evaluations do these cars have? For all the vehicles having “very good” evaluations, what proportion of median safety do these cars have? How many vehicles were bought in very high price and have good evaluation? How many vehicles have high maintenance price and have very good overall evaluation?

**Ans:**

134(69+65) vehicles have better evaluation than acceptable.

For cars with big luggage, what proportion of “very good” evaluations do these cars have= 40 cars

For all the vehicles having “very good” evaluations, what proportion of median safety do these cars have = 0.

How many vehicles were bought in very high price and have good evaluation= 0

How many vehicles have high maintenance price and have very good overall evaluation = 13

**Code:**

summary(dt)

table(dt$lug\_boot,dt$class)

table(dt$safety,dt$class)

table(dt$buying,dt$class)

table(dt$maint,dt$class)

1. Randomize the data set and generate training and testing samples on the randomized dataset. Select the first 70% of the vehicles as training sample (use floor () function or other functions to round the number) and the rest 30% of the vehicles as testing sample.

**Ans:**

data.frame': 1209 obs. of 7 variables:

$ buying : Factor w/ 4 levels "high","low","med",..: 2 3 3 3 1 3 1 3 4 3 ...

$ maint : Factor w/ 4 levels "high","low","med",..: 4 1 3 1 1 4 1 4 2 4 ...

$ doors : Factor w/ 4 levels "2","3","4","5more": 1 2 2 2 4 2 4 3 3 2 ...

$ persons : Factor w/ 3 levels "2","4","more": 2 3 2 1 1 2 3 2 3 1 ...

$ lug\_boot: Factor w/ 3 levels "big","med","small": 3 3 1 2 3 2 2 1 2 1 ...

$ safety : Factor w/ 3 levels "high","low","med": 1 2 1 3 3 3 1 2 2 1 ...

$ class : Factor w/ 4 levels "acc","good","unacc",..: 1 3 4 3 3 3 1 3 3 3 ...

data.frame': 519 obs. of 7 variables:

$ buying : Factor w/ 4 levels "high","low","med",..: 4 4 4 4 4 4 4 4 4 4 ...

$ maint : Factor w/ 4 levels "high","low","med",..: 4 4 4 4 4 4 4 4 4 4 ...

$ doors : Factor w/ 4 levels "2","3","4","5more": 1 1 1 1 1 1 1 1 1 1 ...

$ persons : Factor w/ 3 levels "2","4","more": 1 1 1 1 1 1 1 2 2 2 ...

$ lug\_boot: Factor w/ 3 levels "big","med","small": 3 3 3 2 1 1 1 3 2 2 ...

$ safety : Factor w/ 3 levels "high","low","med": 2 3 1 3 2 3 1 1 3 1 ...

$ class : Factor w/ 4 levels "acc","good","unacc",..: 3 3 3 3 3 3 3 3 3 3 ...

**Code:**

set.seed(1234)

train.index <- sample(1:nrow(dt), floor(0.7\*nrow(dt)), replace = FALSE)

dt.train <- dt[train.index,]

dt.test <- dt[-train.index,]

str(dt.train)

str(dt.test)

1. Build a Support Vector Machine model on training sample with class variable as outcome variable and the rest attributes as predictors. Build this model using kernel = “vanilladot” and other default parameters. Look at the basic information about the model. How many support vectors does this model generate? What is the training error of this model?

**Ans:**

Number of Support Vectors : **369**

Training error : **0.065343**

SV type: C-svc (classification)

parameter : cost C = 1

Linear (vanilla) kernel function.

Number of Support Vectors : 369

Objective Function Value : -47.9099 -185.3264 -27.2007 -27.2727 -16 -11.2727

Training error : 0.065343

**Code:**

library(e1071)

model.svm.car <- ksvm(class~., data=dt.train, kernel="vanilladot")

model.svm.car

1. Evaluate the model. Use the model to predict the class labels of testing sample. Then use table or cross table to evaluate the performance. How many vehicles in total have been misclassified? How many vehicles have been classified as unacceptable are actually good?

**Ans:**

How many vehicles in total have been misclassified= 32

How many vehicles have been classified as unacceptable are actually good=0

**Code:**

model.svm.car.test <- ksvm(class~., data=dt.test, kernel="vanilladot")

predicted.car.test <- predict(model.svm.car.test, dt.test[-7])

table(predicted.car.test)

library(gmodels)

CrossTable(predicted.car.test,dt.test$class,dnn=c("Predicted","Actual"))

1. Evaluate the model in another way. Look only at agreement vs. non-agreement and construct a vector of TRUE/FALSE indicating correct/incorrect predictions. How many “TRUE” agreements do the model generate for testing sample? What percentage does the “TRUE” agreements have among all agreements?

**Ans:**

How many “TRUE” agreements do the model generate for testing sample=**487**

What percentage does the “TRUE” agreements have among all agreements=**93.834297%**

**Code:**

accuracy <- ifelse(predicted==dt.test$class,TRUE,FALSE)

table(accuracy)

prop.table(table(accuracy))\*100

1. Build a new Support Vector Machine model on training sample with class variables as outcome variable and the rest attributes as predictors. Build this model using kernel = "rbfdot" and other default parameters. How many support vectors does this model generate? What is the training error of this model?

**Ans:**

Number of Support Vectors : **301**

Training error : **0.078998**

**Code:**

model.svm.new <- ksvm(class~., data=dt.test, kernel="rbfdot")

model.svm.new

1. Evaluate the new model. Use the model to predict the class labels of testing sample. Then use table or cross table to evaluate the performance. How many vehicles in total have been misclassified? How many vehicles have been classified as good are actually unacceptable?

**Ans:**

How many vehicles in total have been misclassified = 13

How many vehicles have been classified as good are actually unacceptable = 0

**Code:**

predicted.car.test <- predict(model.svm.new, dt.test[-7])

table(predicted.car.test)

library(gmodels)

CrossTable(predicted.car.test,dt.test$class,dnn=c("Predicted","Actual"))