L6

Kovel

31 РѕРєС‚СЏР±СЂСЏ 2020 Рі

# Download the data

set.seed(123)  
f\_train <- read.csv2('education\_train.csv', header = TRUE, encoding = 'UNICOD')  
f\_test <- read.csv2('education\_test.csv', header = TRUE, encoding = 'UNICOD')  
f\_train <- f\_train[-1]  
f\_test <- f\_test[-1]

## Висновок: завантажано датасет,який було розподілено на навчальну та тестову вибірки.

# Fitting SVM model

# install.packages('e1071')  
library(e1071)  
class\_svm\_l = svm(school ~ age + traveltime, data = f\_train, kernel = 'linear')  
summary(class\_svm\_l)

##   
## Call:  
## svm(formula = school ~ age + traveltime, data = f\_train, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: eps-regression   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.5   
## epsilon: 0.1   
##   
##   
## Number of Support Vectors: 66

## Висновок: для навчання базової моделі, заснованої на методі опорних векторів, вибрано лінійне ядро.

# Predicting

p <- predict(class\_svm\_l, f\_test[, c('age','traveltime')])  
y <- ifelse(p > 0.5, 1, 0)

## визначено класи об’єктів (вектор у)

## Confusion Matrix

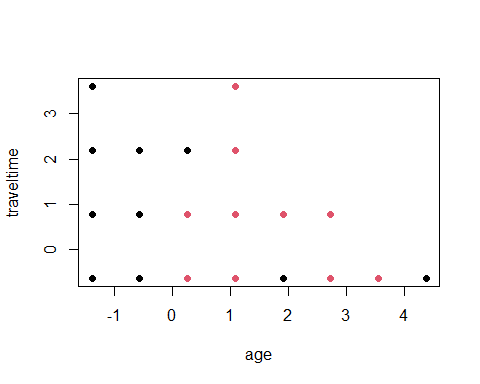
cm = table(f\_test[, 'school'], y)  
print(cm)

## y  
## 0  
## 0 116  
## 1 15

## Висновок: точність моделі погіршалася в порівнянні із логістичною регрессією

# Visualising the Test set results

xgrid = expand.grid(age = f\_test$age, traveltime = f\_test$traveltime)  
ygrid = predict(class\_svm\_l, xgrid)  
  
plot(xgrid, col = as.numeric(ygrid), pch = 10, cex = .9)  
points(f\_test[, c('age','traveltime')], col = as.factor(f\_test$school), pch = 19)

 ##Висновок: на графіку світлим позначені випадки потрапляння в перший тип школи, темним – у другий. Модель описує лінійний варіант розподіляючої кривої. # Fitting RBF-kernel model

# install.packages('e1071')  
library(e1071)  
class\_svm\_r = svm(school ~ age + traveltime, data = f\_train, kernel = 'radial')  
summary(class\_svm\_r)

##   
## Call:  
## svm(formula = school ~ age + traveltime, data = f\_train, kernel = "radial")  
##   
##   
## Parameters:  
## SVM-Type: eps-regression   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.5   
## epsilon: 0.1   
##   
##   
## Number of Support Vectors: 70

## Висновок: для навчання моделі, заснованої на методі опорних векторів, вибрано нелінійне ядро.

# Predicting

p <- predict(class\_svm\_r, f\_test[, c('age','traveltime')])  
y <- ifelse(p > 0.5, 1, 0)

## визначені класи об’єктів (вектор у).

## Confusion Matrix

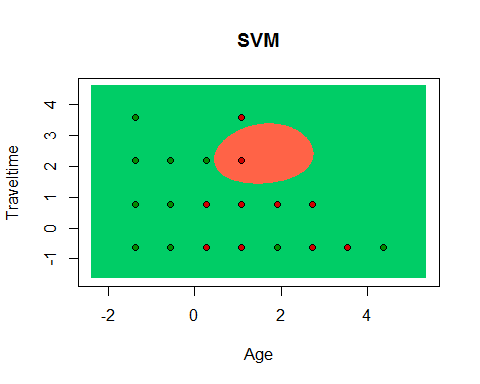
cm = table(f\_test[, 'school'], y)  
print(cm)

## y  
## 0 1  
## 0 116 0  
## 1 14 1

## Висновок: точність моделі стала вищою

# Visualising the Test set results

library(ggplot2)  
set = f\_test[,c('age','traveltime','school')]  
X1 = seq(min(set['age']) - 1, max(set['age']) + 1, by = 0.01)  
X2 = seq(min(set['traveltime']) - 1, max(set['traveltime']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('age', 'traveltime')  
p\_grid = predict(class\_svm\_r, grid\_set)  
y\_grid <- ifelse(p\_grid > 0.5, 1, 0)  
plot(set[, -3],  
 main = 'SVM',  
 xlab = 'Age', ylab = 'Traveltime',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))

 ##Висновок: на графіку червоним позначені випадки потрапляня до школи першого типу, зеленим – ло другого. Червоним виділена зона високої ймовірності потрапляння до першого типу. Модель описує нелінійний варіант розподіляючої кривої.