R Notebook

library('kernlab')  
library('RCurl')

## Loading required package: bitops

library('ggplot2')

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:kernlab':  
##   
## alpha

library('GGally')  
  
file <- getURL('https://d37djvu3ytnwxt.cloudfront.net/assets/courseware/v1/e39a3df780dacd5503df6a8322d72cd2/asset-v1:GTx+ISYE6501x+2T2017+type@asset+block/credit\_card\_data-headers.txt', ssl.verifyhost=FALSE, ssl.verifypeer=FALSE)  
  
data <- read.csv(textConnection(file), header=T, sep = "\t")  
  
head(data)

## A1 A2 A3 A8 A9 A10 A11 A12 A14 A15 R1  
## 1 1 30.83 0.000 1.25 1 0 1 1 202 0 1  
## 2 0 58.67 4.460 3.04 1 0 6 1 43 560 1  
## 3 0 24.50 0.500 1.50 1 1 0 1 280 824 1  
## 4 1 27.83 1.540 3.75 1 0 5 0 100 3 1  
## 5 1 20.17 5.625 1.71 1 1 0 1 120 0 1  
## 6 1 32.08 4.000 2.50 1 1 0 0 360 0 1

x <- as.matrix(data[,1:10])  
y <- data[,11]  
  
model <- ksvm(x, y, type = "C-svc", kernal = "polydot", C=100, scaled = TRUE, cross = 3)  
  
a <- colSums(data[model@SVindex,1:10]\*model@coef[[1]])  
a

## A1 A2 A3 A8 A9   
## -8.694337 -427.403564 -41.225700 187.057685 25.550495   
## A10 A11 A12 A14 A15   
## -13.027721 98.556233 -12.234817 -9673.427935 282594.235826

a0 <- sum(a\*data[1,1:10]) - model@b  
a0

## [1] -1966872

pred <- predict(model,data[,1:10])  
pred

## [1] 1 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [71] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0  
## [106] 1 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [141] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1  
## [176] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [211] 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [246] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [281] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0  
## [316] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [351] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [386] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [421] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [456] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [491] 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1  
## [526] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [561] 1 1 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0  
## [596] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0  
## [631] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

sum(pred == data[,11]) / nrow(data)

## [1] 0.9541284

coef(model)

## [[1]]  
## [1] 100.00000000 12.05054441 21.21123562 100.00000000 18.28384789  
## [6] 100.00000000 83.26041550 47.36307942 41.42059767 50.30885869  
## [11] 5.50471090 100.00000000 8.75753753 3.35462502 21.21477751  
## [16] 23.22954085 0.20534121 58.20787167 0.86865702 0.27143818  
## [21] 100.00000000 100.00000000 100.00000000 100.00000000 100.00000000  
## [26] 78.84830426 92.25050677 31.65908498 0.75425067 100.00000000  
## [31] 29.09425727 0.22368897 3.54688917 -0.93790376 -13.13701426  
## [36] -100.00000000 -95.44376140 -100.00000000 -100.00000000 -21.13260214  
## [41] -61.62480901 -41.36785185 -98.84073228 -100.00000000 -100.00000000  
## [46] -100.00000000 -36.61535844 -3.23733424 -14.77879184 -100.00000000  
## [51] -100.00000000 -60.15732523 -32.19409884 -100.00000000 -100.00000000  
## [56] -33.24051107 -100.00000000 -100.00000000 -100.00000000 -4.36054690  
## [61] -100.00000000 -69.98087685 -70.66776668 -65.23808323 -100.00000000  
## [66] -100.00000000 -3.58531215 -100.00000000 -95.64657524 -47.19589821  
## [71] 13.27833181 2.66230830 0.27382381 2.27473610 100.00000000  
## [76] 60.74940173 0.22151160 8.83939813 19.30593861 2.31750947  
## [81] 1.17244758 100.00000000 13.45564313 8.22920166 39.60397199  
## [86] 100.00000000 100.00000000 98.79721671 0.13480022 4.47547796  
## [91] 64.65612413 100.00000000 83.31132758 1.33598819 31.07044989  
## [96] 30.56427174 29.74172149 60.08838680 1.41272963 4.89014653  
## [101] 39.26833066 2.66876194 14.27740835 65.13896397 100.00000000  
## [106] 99.70326757 100.00000000 21.06109263 100.00000000 3.97077301  
## [111] 9.94596168 100.00000000 19.58839146 15.12343465 100.00000000  
## [116] 1.64922516 0.12020357 27.82353190 91.34208549 -7.86959396  
## [121] -42.42562919 100.00000000 -12.98504389 -100.00000000 -1.26072940  
## [126] -0.52015229 -0.09000659 -0.60642030 -10.21944340 -1.78623407  
## [131] -1.09164631 -5.37543169 -59.44920512 -12.82038552 -100.00000000  
## [136] -100.00000000 0.27382631 72.73205063 100.00000000 100.00000000  
## [141] 13.64103572 100.00000000 -18.81470391 -5.61172810 -14.58770792  
## [146] -13.96105417 -70.85648154 -0.71420336 -40.55295321 -14.38023521  
## [151] -5.94838338 -2.38655673 -37.32924637 -11.05153059 -96.43262106  
## [156] -28.71793468 -100.00000000 -1.96304288 -9.31190117 -4.73935897  
## [161] -1.72153597 -7.22331236 -62.44534974 -51.85973645 -79.48820075  
## [166] -30.35278074 -2.34661298 -8.92323673 -14.11949602 -60.77628379  
## [171] -2.30579003 -100.00000000 -1.60454686 -18.89841084 0.65297461  
## [176] 24.47367684 5.98331946 23.91764021 83.67880830 5.99071148  
## [181] 28.03385570 2.63139802 61.44323737 10.38928508 18.86567733  
## [186] 12.08096358 69.16353241 7.45536855 -100.00000000 -59.03142290  
## [191] -89.07173294 -13.98762893 -17.89633969 -2.73478341 -100.00000000  
## [196] -16.68496616 -7.87952578 -87.49906426 -100.00000000 -13.59082057  
## [201] -3.70591534 -100.00000000 -76.66858363 -100.00000000 -100.00000000  
## [206] -69.20090292 -0.71120056 -100.00000000 1.83554980 45.93312080  
## [211] 24.82503410 100.00000000 9.87414615 34.89076977 100.00000000  
## [216] 1.36792332 39.03348503 100.00000000 8.77792618 100.00000000  
## [221] 10.28763419 1.02174342 100.00000000 83.01969180 100.00000000  
## [226] 45.26317951 100.00000000 -11.94692009 -100.00000000 100.00000000  
## [231] -7.54206162 -8.41784373 -4.12437846 -100.00000000 -2.69320372  
## [236] -100.00000000 -5.03060844 -2.69145331 -13.86932067 -86.87226228  
## [241] -6.17660373 -20.63636865

b(model)

## [1] -0.7262857

error(model)

## [1] 0.04587156

kernelf(model)

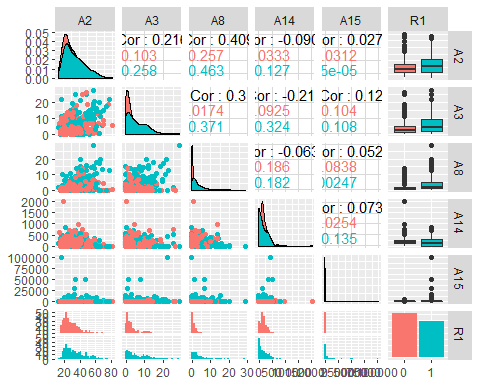
## Gaussian Radial Basis kernel function.   
## Hyperparameter : sigma = 0.0974816256340452

kernals <- c('rbfdot','polydot','vanilladot','tanhdot','laplacedot','besseldot','anovadot','splinedot','stringdot')  
  
for(kernal in kernals){  
model <- ksvm(x, y, type = "C-svc", kernal = kernal, C=100, scaled = TRUE, cross = 5)  
pred <- predict(model,data[,1:10])  
cat('\n',kernal,'pred: ', sum(pred == data[,11]) / nrow(data))  
}

##   
## rbfdot pred: 0.9480122  
## polydot pred: 0.9571865  
## vanilladot pred: 0.9571865  
## tanhdot pred: 0.9571865  
## laplacedot pred: 0.9525994  
## besseldot pred: 0.9525994  
## anovadot pred: 0.9571865  
## splinedot pred: 0.9525994  
## stringdot pred: 0.9525994

data[,11] <- as.factor(data[,11])  
GGally::ggpairs(data[, c(2:4,9:11)], aes(colour=R1))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



results<-c()   
n=1   
results < NULL

## logical(0)

for(i in seq(from=1, to=100, by=10))  
{   
 model <- ksvm(x, y, scaled =TRUE, C=i, type ="C-svc", kernel="vanilladot",cross=5)   
   
pred <- predict(model, data[,1:10])   
results[n] <- sum(pred == data[,11])/nrow(data)   
n=n+1   
}

## Setting default kernel parameters   
## Setting default kernel parameters   
## Setting default kernel parameters   
## Setting default kernel parameters   
## Setting default kernel parameters   
## Setting default kernel parameters   
## Setting default kernel parameters   
## Setting default kernel parameters   
## Setting default kernel parameters   
## Setting default kernel parameters

results

## [1] 0.8639144 0.8639144 0.8639144 0.8639144 0.8639144 0.8639144 0.8639144  
## [8] 0.8639144 0.8639144 0.8639144