Prepare a classification model using SVM for salary data

> library(ggplot2)

> library(kernlab)

> doParallel::registerDoParallel(cores = 2)

> train <- read.csv(file.choose())

> View(train)

> test <- read.csv(file.choose())

> View(test)

> str(train)

'data.frame': 30161 obs. of 14 variables:

$ age : int 39 50 38 53 28 37 49 52 31 42 ...

$ workclass : Factor w/ 7 levels " Federal-gov",..: 6 5 3 3 3 3 3 5 3 3 ...

$ education : Factor w/ 16 levels " 10th"," 11th",..: 10 10 12 2 10 13 7 12 13 10 ...

$ educationno : int 13 13 9 7 13 14 5 9 14 13 ...

$ maritalstatus: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...

$ occupation : Factor w/ 14 levels " Adm-clerical",..: 1 4 6 6 10 4 8 4 10 4 ...

$ relationship : Factor w/ 6 levels " Husband"," Not-in-family",..: 2 1 2 1 6 6 2 1 2 1 ...

$ race : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...

$ sex : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...

$ capitalgain : int 2174 0 0 0 0 0 0 0 14084 5178 ...

$ capitalloss : int 0 0 0 0 0 0 0 0 0 0 ...

$ hoursperweek : int 40 13 40 40 40 40 16 45 50 40 ...

$ native : Factor w/ 40 levels " Cambodia"," Canada",..: 38 38 38 38 5 38 22 38 38 38 ...

$ Salary : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...

> salary <- rbind(train,test)

> View(salary)

> table(salary$Salary)

<=50K >50K

34013 11208

#Creating dummies

> level\_work <- levels(salary$workclass)

> leveledu <- levels(salary$education)

> level\_mari <- levels(salary$maritalstatus)

> level\_occ <- levels(salary$occupation)

> level\_rel <- levels(salary$relationship)

> level\_race <- levels(salary$race)

> level\_sex <- levels(salary$sex)

> level\_native <- levels(salary$native)

> level\_salary <- levels(salary$Salary)

> salary$workclass <- as.integer(factor(salary$workclass,levels =c(level\_work),labels=c(1,2,3,4,5,6,7)))

> salary$education <- as.integer(factor(salary$education,levels = c(leveledu),labels = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)))

> salary$maritalstatus <- as.integer(factor(salary$maritalstatus,levels = c(level\_mari),labels =c(1,2,3,4,5,6,7) ))

> salary$occupation <- as.integer(factor(salary$occupation,levels = c(level\_occ),labels = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14)))

> salary$relationship <- as.integer(factor(salary$relationship,levels = c(level\_rel),labels = c(1,2,3,4,5,6)))

> salary$race <- as.integer(factor(salary$race,levels = c(level\_race),labels = c(1,2,3,4,5)))

> salary$sex <- as.integer(factor(salary$sex,levels = c(level\_sex),labels = c(1,2)))

> salary$native <- as.integer(factor(salary$native,levels = c(level\_native),labels = c(seq(1,40,1))))

> salary$Salary <- as.integer(factor(salary$Salary,levels = c(level\_salary),labels = c(1,2)))

#Normalising df

> normal <- function(x){

return((x-min(x))/(max(x)-min(x)))

}

> norm\_salary <- normal(salary)

#Splitting of data to test and train

> train\_norm <- norm\_salary[1:30161,]

> View(train\_norm)

> test\_norm <- norm\_salary[30162:45221,]

#Model Building

#Model 1 : rbfdot

> modelrbfdot <- ksvm(Salary~.,data=train\_norm,kernel="rbfdot")

> predrbfdot <- predict(modelrbfdot,newdata=test\_norm)

> cor(predrbfdot,test\_norm$Salary)

[,1]

[1,] 0.6005603

Accuracy is 60.05%

#Model 2 : besseldot

> modelbesseldot <- ksvm(Salary~.,data=train\_norm,kernel=" besseldot ")

> predbesseldot t <- predict(modelbesseldot,newdata=test\_norm)

> cor(predbesseldot,test\_norm$Salary)

[,1]

[1,] 0.1666603

Accuracy is 16.66%

#Model 3: vanilladot

> modelvanilladot <- ksvm(Salary~.,data=train\_norm,kernel=" vanilladot ")

> predvanilladot <- predict(modelvanilladot,newdata=test\_norm)

> cor(predvanilladot,test\_norm$Salary)

[,1]

[1,] 0.3166603

Accuracy is 31.66%

Kernel rbfdot has the highest accuracy with 60.05%(Model 1)