Title::- Prepare a classification model using SVM for salary data

# Data(Train)

> 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 : chr " State-gov" " Self-emp-not-inc" " Private" " Private" ...

$ education : chr " Bachelors" " Bachelors" " HS-grad" " 11th" ...

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

$ maritalstatus: chr " Never-married" " Married-civ-spouse" " Divorced" " Married-civ-spouse" ...

$ occupation : chr " Adm-clerical" " Exec-managerial" " Handlers-cleaners" " Handlers-cleaners" ...

$ relationship : chr " Not-in-family" " Husband" " Not-in-family" " Husband" ...

$ race : chr " White" " White" " White" " Black" ...

$ sex : chr " Male" " Male" " Male" " Male" ...

$ 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 : chr " United-States" " United-States" " United-States" " United-States" ...

$ Salary : chr " <=50K" " <=50K" " <=50K" " <=50K" ...

> salary <- rbind(train,test)

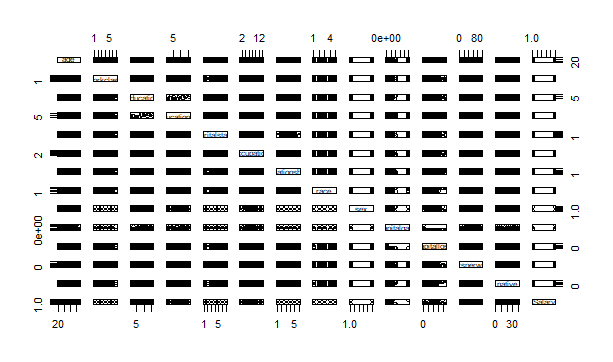
> View(salary)

> table(salary$Salary)

<=50K >50K

34013 11208

> plot(salary)



#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$workclass <- as.integer(as.factor(salary$workclass))

salary$education <- as.integer(factor(salary$education))

salary$maritalstatus <- as.integer(factor(salary$maritalstatus))

salary$occupation <- as.integer(factor(salary$occupation))

salary$relationship <- as.integer(factor(salary$relationship))

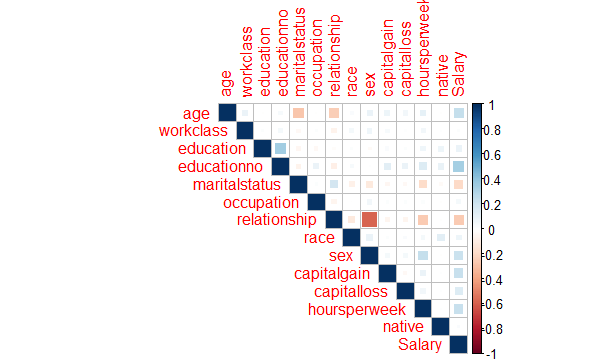
salary$race <- as.integer(factor(salary$race))

salary$sex <- as.integer(factor(salary$sex))

salary$native <- as.integer(factor(salary$native))

salary$Salary <- as.integer(factor(salary$Salary))

corrplot(cor(salary),method = c("square"),type = "upper)



# normalising df

> normal <- function(x){

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

+ }

norm\_salary <- normal(salary)

>

> View(norm\_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

1. 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%

cor(predrbfdot,test\_norm$Salary)

[,1]

[1,] 0.6005968

######Model 2 polydot

modelpolydot <- ksvm(train\_norm$Salary~.,data=train\_norm,kernel="polydot")

> predpolydot <- predict(modelpolydot,newdata=test\_norm)

> cor(predpolydot,test\_norm$Salary)

[,1]

[1,] 0.3166736

#####Model 3: vanilladot

>

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

Setting default kernel parameters

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

> cor(predvanilladot,test\_norm$Salary)

[,1]

[1,] 0.316676

\*\*\*\*\*\*\*\*\*\*Kernel rbfdot has the highest accuracy with 60.05%\*\*\*\*\*\*\*\*\*

kernels <- c("rbfdot","vanilladot","polydot")

> acc\_bag <- list()

> pred\_info <- list()

> table\_info <- list()

> for(i in kernels){

+ model\_bag <- ksvm(Salary~.,data=train\_norm,kernel=i)

+ pred\_bag <- predict(model\_bag,test\_norm)

+ pred\_info[[i]] <- (pred\_bag)

+ acc\_bag[[i]] <- cor(pred\_bag,test\_norm$Salary)

+ table\_info[[i]] <- table(pred\_bag,test\_norm$Salary)

+ }

Setting default kernel parameters

Setting default kernel parameters

> acc\_bag

$rbfdot

[,1]

[1,] 0.6002733

$vanilladot

[,1]

[1,] 0.316676

$polydot

[,1]

[1,] 0.3166736

>

9.71282364360841e-06 1 0

9.7128621729984e-06 1 0

9.71324706209769e-06 1 0

9.71334615298173e-06 1 0

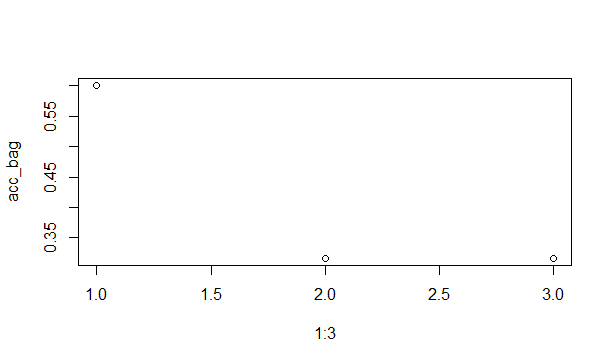
9.71334629117039e-06 1 0

9.7135596971974e-06 1 0

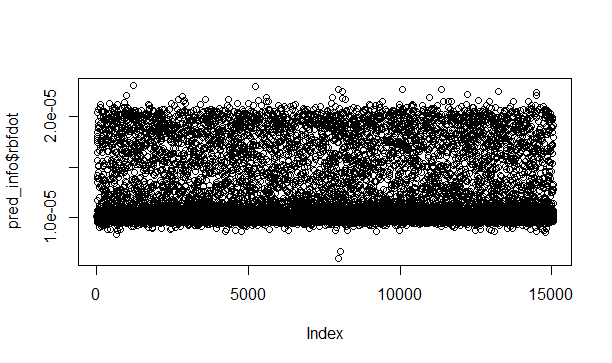
9.71358315770402e-06 1 0

[ reached getOption("max.print") -- omitted 13406 rows ]

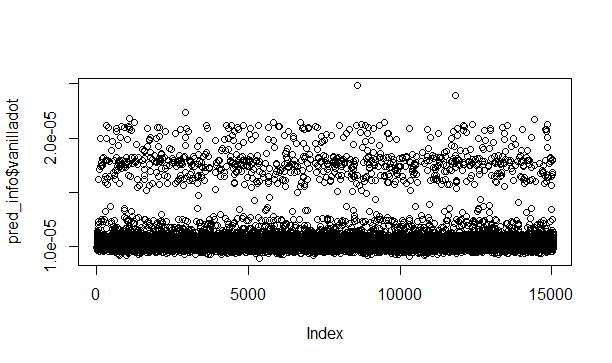
> plot(1:3,acc\_bag)



> plot(pred\_info$rbfdot)



> plot(pred\_info$vanilladot)



> plot(pred\_info$polydot)

