**Assignment-Support Vector Machines**

1. Prepare a classification model using SVM for salary data

Soln: #Salary data

SalaryData

SalaryData\_train<-SalaryData[1:12000,]

SalaryData\_test<-SalaryData[12001:15060,]

#Training Model on the data

library(kernlab)

SalaryData\_classifier<-ksvm(Salary ~.,data=SalaryData\_train,kernel="vanilladot")

#evaluating model performance

SalaryData\_predictions<-predict(SalaryData\_classifier,SalaryData\_test)

head(SalaryData\_predictions)

table(SalaryData\_predictions,SalaryData\_test$Salary)

agreement<-SalaryData\_predictions==SalaryData\_test$Salary

prop.table(table(agreement))

Results in R:

library(kernlab)

> SalaryData\_classifier<-ksvm(Salary ~.,data=SalaryData\_train,kernel="vanilladot")

Setting default kernel parameters

> #evaluating model performance

> SalaryData\_predictions<-predict(SalaryData\_classifier,SalaryData\_test)

> head(SalaryData\_predictions)

[1] <=50K <=50K <=50K <=50K >50K <=50K

Levels: <=50K >50K

> table(SalaryData\_predictions,SalaryData\_test$Salary)

SalaryData\_predictions <=50K >50K

<=50K 2114 309

>50K 171 466

> agreement<-SalaryData\_predictions==SalaryData\_test$Salary

> prop.table(table(agreement))

agreement

FALSE TRUE

0.1568627 0.8431373

Improving the Model of the performance further:

SalaryData\_classifier\_rbf<-ksvm(Salary~.,data=SalaryData\_train,kernel="rbfdot")

SalaryData\_predictions\_rbf<-predict(SalaryData\_classifier\_rbf,SalaryData\_test)

head(letter\_predictions\_rbf)

agreement\_rbf<-SalaryData\_predictions\_rbf==SalaryData\_test$Salary

table(agreement\_rbf)

prop.table(table(agreement\_rbf))

Results in R:

SalaryData\_classifier\_rbf<-ksvm(Salary~.,data=SalaryData\_train,kernel="rbfdot")

> SalaryData\_predictions\_rbf<-predict(SalaryData\_classifier\_rbf,SalaryData\_test)

> agreement\_rbf<-SalaryData\_predictions\_rbf==SalaryData\_test$Salary

> table(agreement\_rbf)

agreement\_rbf

FALSE TRUE

456 2604

> prop.table(table(agreement\_rbf))

agreement\_rbf

FALSE TRUE

0.1490196 0.8509804

Efficiency of model increased by 1% only. Hence further iterations to be made.

1. ForestFires: classify the Size\_Categorie using SVM

forestfires

forestfires[,-(10:30)]

forestfires\_train<-forestfires[1:400,]

forestfires\_test<-forestfires[401:517,]

#Training Model on the data

library(kernlab)

forestfires\_classifier<-ksvm(size\_category~.,data=forestfires\_train[,-(10:30)],kernel="vanilladot")

#evaluating model performance

forestfires\_predictions<-predict(forestfires\_classifier,forestfires\_test)

head(forestfires\_predictions)

table(forestfires\_predictions,forestfires\_test$size\_category)

agreement<-forestfires\_predictions==forestfires\_test$size\_category

prop.table(table(agreement))

Results in R:

forestfires\_predictions<-predict(forestfires\_classifier,forestfires\_test)

> head(forestfires\_predictions)

[1] small small small small small small

Levels: large small

> table(forestfires\_predictions,forestfires\_test$size\_category)

forestfires\_predictions large small

large 0 0

small 34 83

> prop.table(table(agreement))

agreement

FALSE TRUE

0.1568627 0.8431373