MachineLearning_Lab3Block1

Sreenand.S

15/12/2019

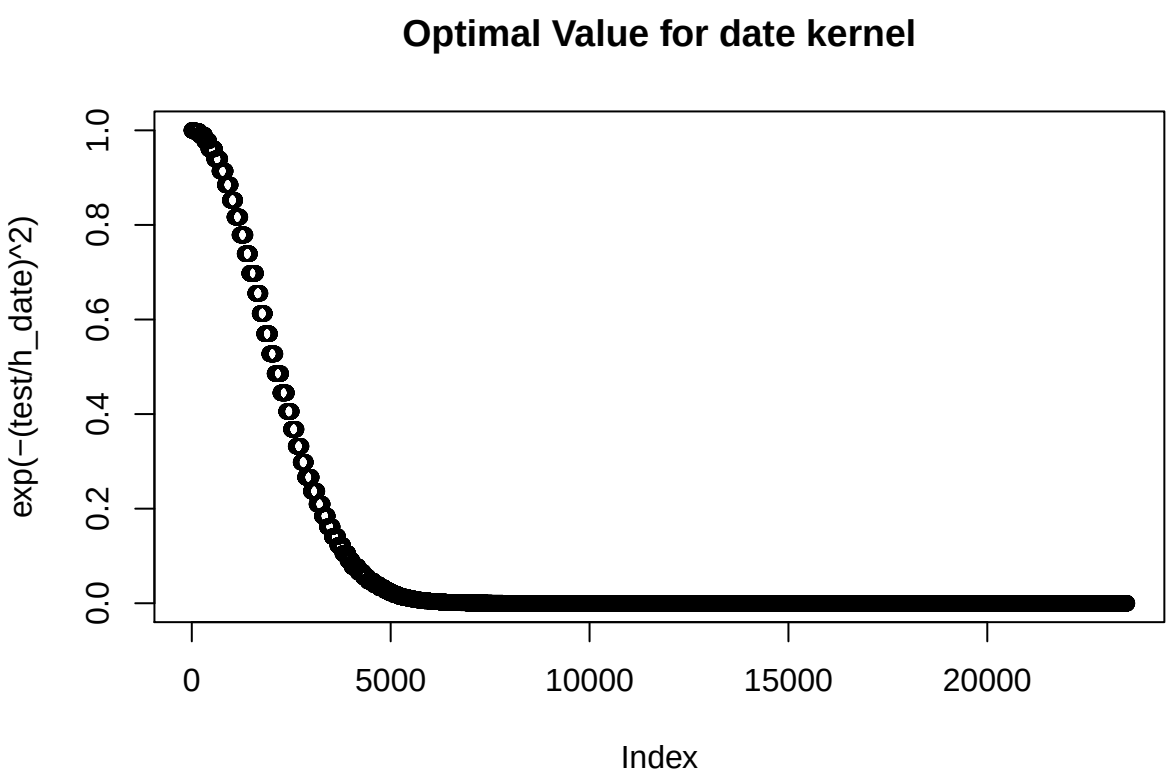
Assignment 1 :Kernel Methods

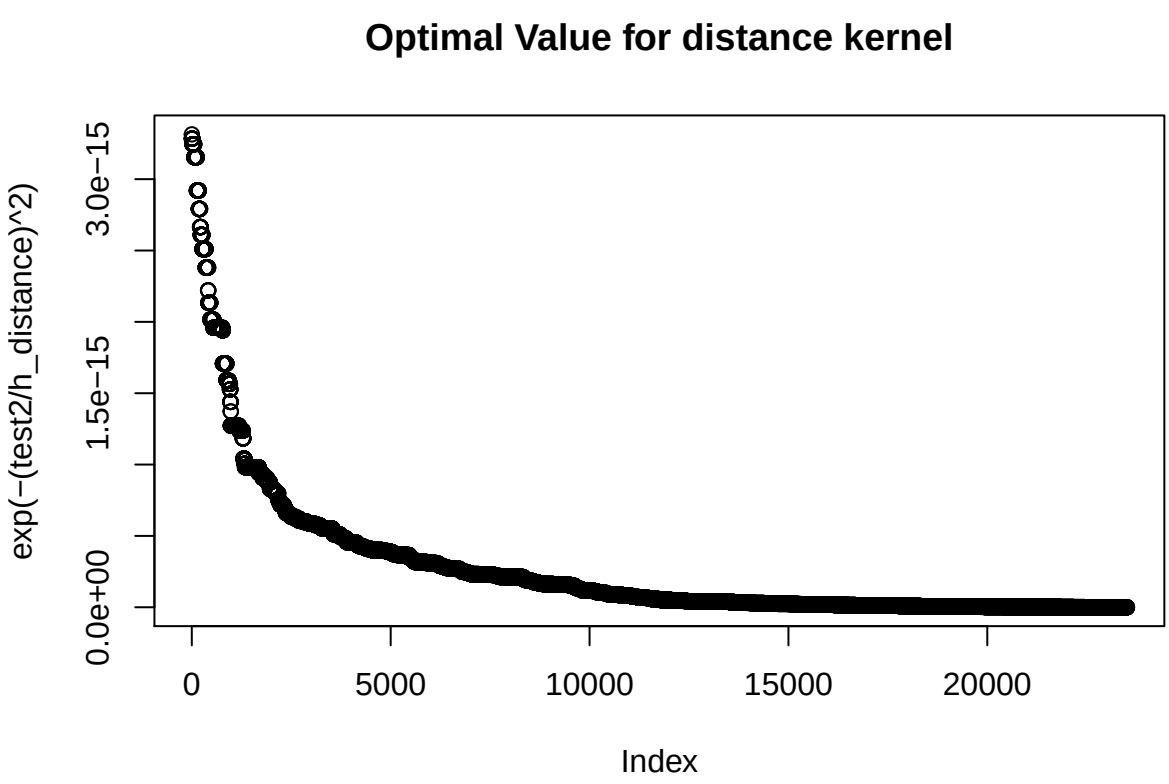
```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```

1.1

We take the point of interest (a,b) as 58.4274 and 14.826 which was given in the pseudocode and the station coordinates are derived from the csv file. We take the day of interest to be 1996-01-04 since there is data ranging from 1980 to 2016 and any day could be taken within this range. Hours of interest are taken from the pseudocode.

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```


Optimal Value for date kernel
exp(−(test/h_date)^2)
0.0
0.2
0.4
0.6
0.8
1.0
0
5000
10000
15000
20000
Index




```
##  [1] "2020-03-12 04:00:00 CET" "2020-03-12 06:00:00 CET"
##  [3] "2020-03-12 08:00:00 CET" "2020-03-12 10:00:00 CET"
##  [5] "2020-03-12 12:00:00 CET" "2020-03-12 14:00:00 CET"
##  [7] "2020-03-12 16:00:00 CET" "2020-03-12 18:00:00 CET"
##  [9] "2020-03-12 20:00:00 CET" "2020-03-12 22:00:00 CET"
## [11] "2020-03-13 00:00:00 CET"
```

1.2

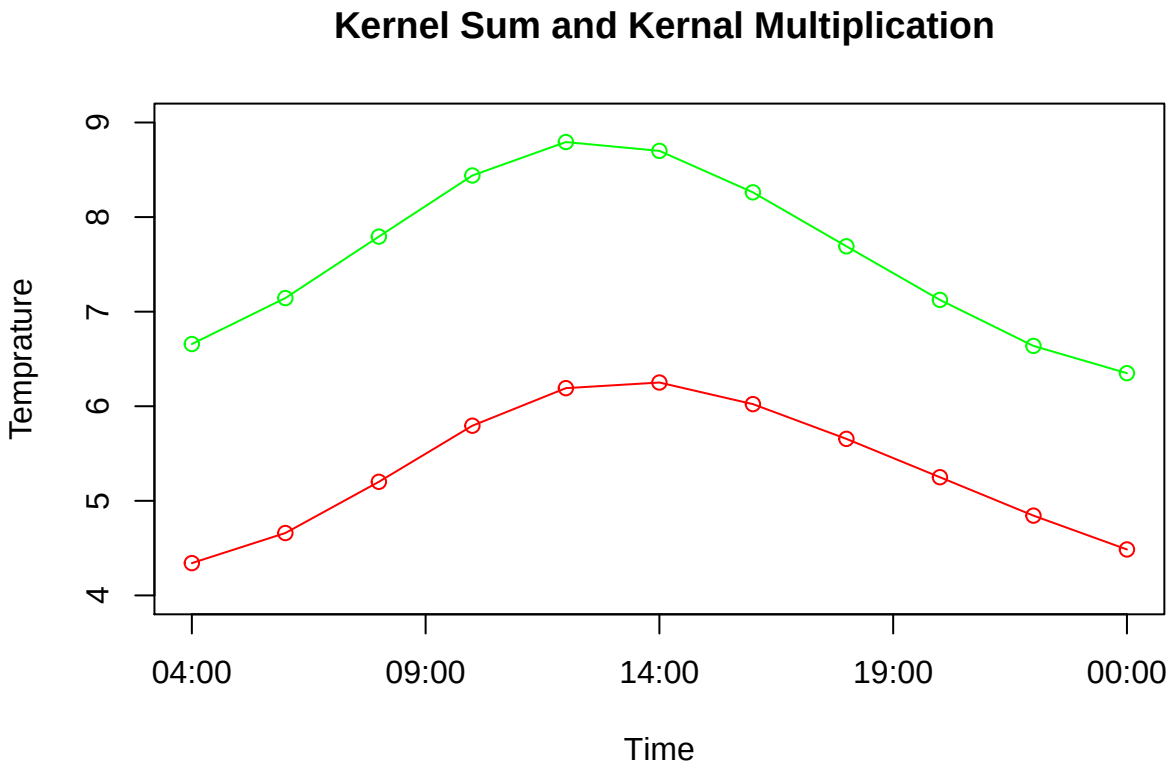
Kernel Distance- After sorting the station distances and plotting the graph applying the kernel function where h = 10^6 we get the above graph. The graph represents a gaussian kernel giving weightage to the points nearer to the station. Since the equation involves minus exp we need to take a higher value as our width to get a higher output therefore we take the distance as 10^6.

Kernel Date - After sorting the date and plotting the graph for date applying the kernel function we get the above graph.After several tries when the kernel width is chosen as 20 we get the above graph.This graph is represents a gaussian kernel giving weightage to the days which are nearer to the date chosen.

Kernel Hour- The h parameter for hour is taken as 5 since a lesser hour gives a more accurate prediction.

```
##  [1] "Temperature values for the kernel addition are  4.34135317234258"
##  [2] "Temperature values for the kernel addition are  4.65946099751498"
##  [3] "Temperature values for the kernel addition are  5.20103882619056"
##  [4] "Temperature values for the kernel addition are  5.79424119769127"
##  [5] "Temperature values for the kernel addition are  6.19129782801284"
##  [6] "Temperature values for the kernel addition are  6.25062615048725"
```

```
##  [7] "Temperature values for the kernel addition are  6.02239033917218"
##  [8] "Temperature values for the kernel addition are  5.65508231372065"
##  [9] "Temperature values for the kernel addition are  5.2495150963101"
## [10] "Temperature values for the kernel addition are  4.84343665952954"
## [11] "Temperature values for the kernel addition are  4.48564718190922"
```




```
##  [1] "Temperature values for the kernel multiplication are 6.65764815683984"
##  [2] "Temperature values for the kernel multiplication are 7.14365039391322"
##  [3] "Temperature values for the kernel multiplication are 7.79364295393641"
##  [4] "Temperature values for the kernel multiplication are 8.4399260539887"
##  [5] "Temperature values for the kernel multiplication are 8.79355067900343"
##  [6] "Temperature values for the kernel multiplication are 8.70020574511566"
##  [7] "Temperature values for the kernel multiplication are 8.26158408468688"
##  [8] "Temperature values for the kernel multiplication are 7.69104933029637"
##  [9] "Temperature values for the kernel multiplication are 7.12437874316827"
## [10] "Temperature values for the kernel multiplication are 6.63800890707069"
## [11] "Temperature values for the kernel multiplication are 6.34975380375313"
```

1.3

Here it is observed that the kernel multiplication prediction is much higher than that of the sum kernel. The predictions are much higher than sum kernel and this is because the product of small values in turn gives an even smaller value. Similarly product of higher values give a higher value.

Assignment 2: Kernel SVM

Model selection

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 0.5
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.05
##
## Number of Support Vectors : 1077
##
## Objective Function Value : -313.3263
## Training error : 0.056087
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.05
##
## Number of Support Vectors : 1007
##
## Objective Function Value : -467.7423
## Training error : 0.04
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 5
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.05
##
## Number of Support Vectors : 966
##
## Objective Function Value : -1109.212
## Training error : 0.021304
```

```
##
## svm_pred1 nonspam spam
##   nonspam     681   77
##   spam         22  370
```

```
## The misclassification rate  for model 1 is  0.08608696
```

```
##
## svm_pred2 nonspam spam
##   nonspam     680   57
##   spam         23  390

##
## The misclassification rate for model 2 is  0.06956522

##
## svm_pred3 nonspam spam
##   nonspam     676   59
##   spam         27  388

##
## The misclassification rate for model 3 is  0.07478261
```

2.1

Here C is the slack parameter and more the value of C wider is the margin. Here it is obseverved that with the increase in C value the no. of support vectors are reducing. When, C = 0.5 :total no. of support vectors are 1077 and misclassification rate is 8.6% C = 1 :total no. of support vectors are 1007 and misclassification rate is 6.9% C = 5 :total no. of support vectors are 966 and misclassification rate is 7.4% Here,infact the slack reduces with the increase in C parameter.The model with the C=1 value seems to be have a better accuracy than the rest of the models of the and is observed to be the best fit for the data.

2.1:Generalization error

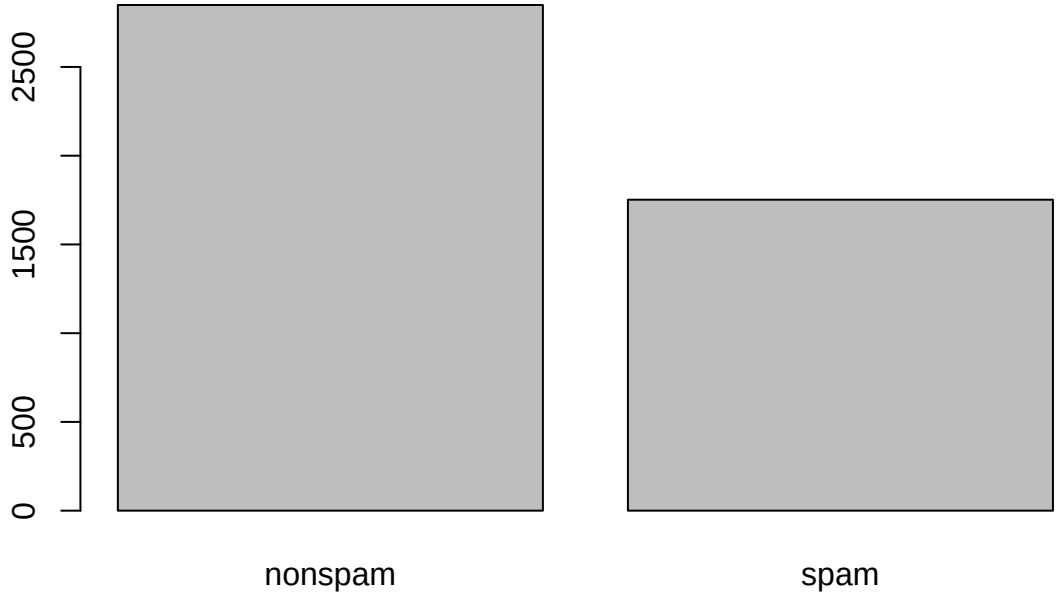
```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used

## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.05
##
## Number of Support Vectors : 1361
##
## Objective Function Value : -633.7041
## Training error : 0.038841

##
## svm_pred2 nonspam spam
##   nonspam     641   62
##   spam         33  415

##
## The misclassification rate for model 2 is  0.08253692
```

We have chosen C=1 as the slack parameter from the previos results and trained it on the combined table of training and validation and predict using the test data which has been untouched. This produces a miscalssification rate of 8.2% which is almost close to the previous result hence we

2.2

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-uniform
## 'Rounding' sampler used
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.05
##
## Number of Support Vectors : 1655
##
## Objective Function Value : -802.313
## Training error : 0.039774
```




```
##
## svm_pred4 nonspam spam
##   nonspam    2727  122
##   spam         61 1691
```

```
## The misclassification rate  when whole dataframe is used is  0.03977396
```

When the C value after finding the miscalculation rates and applied here over the whole dataset then the it is found to be only 3.9% and the no. of vectors 1655. The error here can be comapared as less than that of the generaliation error.

2.3:Return to the user

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.05
##
## Number of Support Vectors : 1655
##
## Objective Function Value : -802.313
## Training error : 0.039774
```

2.4:Purpose of the C parameter

C is the slack parameter and more the value of C wider is the margin.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(readxl)
library(readr)
library(kernlab)
library(geosphere)
RNGversion('3.5.1')
set.seed(1234567890)
stations <- read.csv(file.choose())
temps <- read.csv(file.choose())
st <- merge(stations,temps,by="station_number")

h_distance <- 1000000# These three values are up to the students
h_date <-20
h_time <-5
a <- 58.4274 # The point to predict (up to the students)
b <- 14.826
date <- "1996-01-04" # The date to predict (up to the students)
times <- c("04:00:00", "06:00:00", "08:00:00","10:00:00",
           "12:00:00" ,"14:00:00", "16:00:00","18:00:00",
           "20:00:00","22:00:00","24:00:00")
sum_vector <- vector(length=length(times))
mult_vector <- vector(length=length(times))

RNGversion('3.5.1')
#Date
st$date_diff = as.numeric(difftime(date,st$date,units = c("days")))
```

```
#daydel<-which(date_diff<0)
st<-st[st$date_diff>0,]
for (i in 1:length(st$date_diff)) {
st$date_diff[i] = min(365 - st$date_diff[i]%%365,st$date_diff[i]%%365)
}
test<-sort(st$date_diff)
plot(exp(-(test/ h_date) ^ 2), main = "Optimal Value for date kernel")
st$date_diff = exp(-(test / h_date) ^ 2)

#distance calculation
st$dist_vec = abs(distHaversine(st[,c(5,4)], c(a,b)))
test2<-sort(st$dist_vec)
plot(exp(-(test2 / h_distance) ^ 2), main = "Optimal Value for distance kernel")
st$dist_vec = exp(-(test2 / h_distance) ^ 2)



for (i in 1:length(times))
{
   st[times[i]]<- as.numeric(abs(difftime(strptime(times[i],"%H"),strptime(as.character(st[,'time']),"%

   st[,(i+14)] <- sapply(st[,(i+14)], function(x) min(x,(24-x)) )



}
  time_vec = exp(-(st[,15:25]/h_time)^2)
#for (k in 1:length(times)) {
  sum_k = st$dist_vec + st$date_diff + time_vec
  sumk = st$air_temperature * sum_k
  sum_vector = colSums(sumk) / colSums(sum_k)


  mul_k = st$dist_vec * st$date_diff *  time_vec
  mulk = st$air_temperature * mul_k
  mult_vector = colSums(mulk) / colSums(mul_k)
#}

# Convert to Standard time
conv_time = as.POSIXlt(paste(Sys.Date(), times), format="%Y-%m-%d %H:%M:%S")
print(conv_time)

# testsist = distHaversine(c(55.38360,12.82030),c(58.4274,14.826))
 # std_time = as.POSIXlt(paste(Sys.time(), "04:00:00"), format="%Y-%m-%d %H:%M:%S")
plot(x = conv_time,y = sum_vector,type = 'o', main = ' Kernel Sum and Kernal Multiplication',xlab='Time
       ylab = 'Temprature',col = 'red',ylim = c(4,9)  )

  paste("Temperature values for the kernel addition are ",sum_vector)

points(x = conv_time,y = mult_vector,type = 'o', main = ' Kernel Multiplication',xlab='Time', ylab = 'T
paste0("Temperature values for the kernel multiplication are ",mult_vector)
RNGversion('3.5.1')
data("spam")
```

```r
df<- spam
n = dim(df)[1]
set.seed(12345)
id = sample(1:n, floor(n*0.5))
train = df[id,]
id1 = setdiff(1:n, id)
set.seed(12345)
id2 = sample(id1, floor(n*0.25))
valid = df[id2,]
id3 = setdiff(id1,id2)
test = df[id3,]
combined = rbind(train,valid)

svm_model1<-ksvm(type~.,data = train,C=0.5,kernel = "rbfdot", kpar = list(sigma = 0.05))
svm_model2<-ksvm(type~.,data = train,C=1,kernel = "rbfdot",kpar = list(sigma = 0.05))
svm_model3<-ksvm(type~.,data = train,C=5,kernel = "rbfdot",kpar = list(sigma = 0.05))
svm_pred1<-predict(svm_model1,valid,type = "response")
svm_pred2<-predict(svm_model2,valid,type = "response")
svm_pred3<-predict(svm_model3,valid,type = "response")
print(svm_model1)
print(svm_model2)
print(svm_model3)
par(mfrow = c(2,2))

# plot(df$type)
# plot(svm_pred1)
# plot(svm_pred2)
# plot(svm_pred3)

confusionmatrix1<-table(svm_pred1,valid$type)
confusionmatrix1
misclas1 = 1- sum(diag(confusionmatrix1))/sum(confusionmatrix1)
cat("The misclassification rate  for model 1 is ",misclas1)

confusionmatrix2<-table(svm_pred2,valid$type)
confusionmatrix2
misclas2 = 1- sum(diag(confusionmatrix2))/sum(confusionmatrix2)
cat("\nThe misclassification rate for model 2 is ",misclas2)

confusionmatrix3<-table(svm_pred3,valid$type)
confusionmatrix3
misclas3 = 1- sum(diag(confusionmatrix3))/sum(confusionmatrix3)
cat("\nThe misclassification rate for model 3 is ",misclas3)

## Including Plots


RNGversion('3.5.1')
svm_model2<-ksvm(type~.,data = combined,C=1,kernel = "rbfdot",kpar = list(sigma = 0.05))###combined = v

svm_pred2<-predict(svm_model2,test,type = "response")

print(svm_model2)
```

```
confusionmatrix2<-table(svm_pred2,test$type)
confusionmatrix2
misclas2 = 1- sum(diag(confusionmatrix2))/sum(confusionmatrix2)
cat("\nThe misclassification rate for model 2 is ",misclas2)


#Using the whole dataframe
RNGversion('3.5.1')
svm_model4<-ksvm(type~.,data=df,C=1,kernel = "rbfdot",kpar = list(sigma = 0.05))
print(svm_model4)
svm_pred4<-predict(svm_model4,df)
plot(svm_pred4)

confusionmatrix4<-table(svm_pred4,df$type)
confusionmatrix4
misclas4 = 1- sum(diag(confusionmatrix4))/sum(confusionmatrix4)
cat("The misclassification rate  when whole dataframe is used is ",misclas4)
svm_model4<-ksvm(type~.,data=df,C=1,kernel = "rbfdot",kpar = list(sigma = 0.05))
svm_model4
```