STA 6106 - Final Project
By: Anmol Sureshkumar Panchal; UID: 4446829
Submitted on 12/04/2018 : 15:25:30 PM

Problem 1:

Explanation with the Code:

Lets Load the required packages to perform hyper tuning of the parameters for the dataset **"pb2.txt"** as **class = V1 =1** for problem 1 and also for performing stepwise regression in forward direction using the same class of the same dataset.

```
library(caret)
library(leaps)
library(MASS)
library("e1071")
```

################### PROBLEM 1 ################################

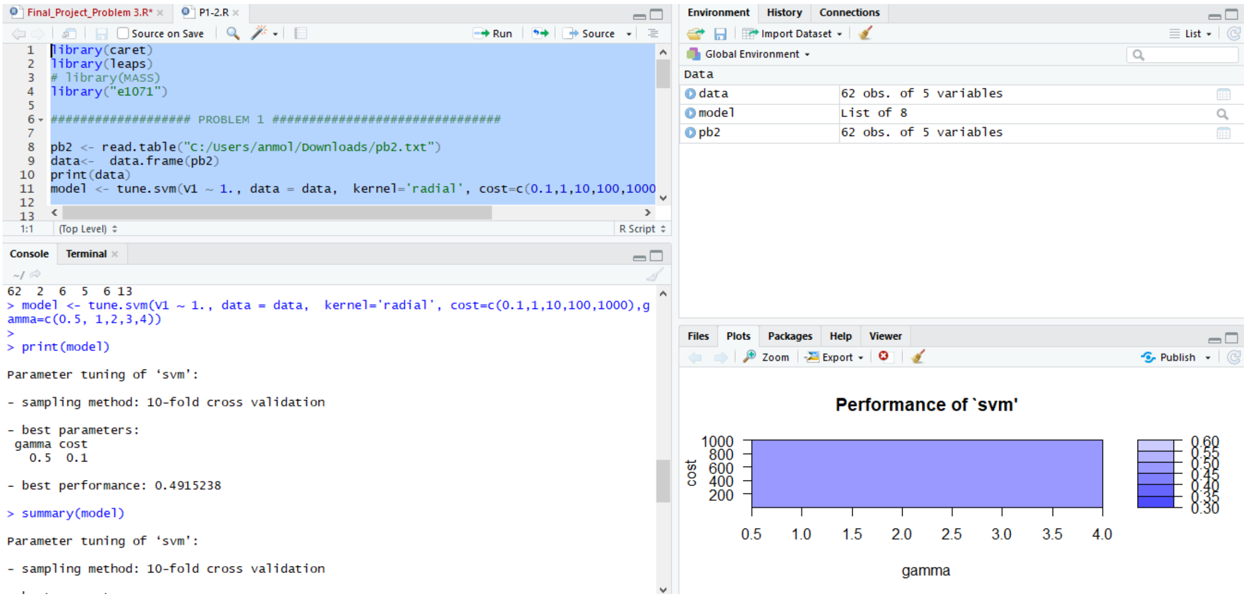
Here below the code lines import the dataset and store it in pb2 as data frame.

```
pb2 <- read.table("C:/Users/anmol/Downloads/pb2.txt")
data<-  data.frame(pb2)
print(data)
```

```
   V1 V2 V3 V4 V5
1   1 15 17 24 14
2   1 17 15 32 26
3   1 15 14 29 23
4   1 13 12 10 16
5   1 20 17 26 28
6   1 15 21 26 21
7   1 15 13 26 22
8   1 13  5 22 22
9   1 14  7 30 17
10  1 17 15 30 27
11  1 17 17 26 20
12  1 17 20 28 24
13  1 15 15 29 24
14  1 18 19 32 28
15  1 18 18 31 27
16  1 15 14 26 21
17  1 10 14 19 17
18  1 18 21 30 29
19  1 18 21 34 26
20  1 13 17 30 24
21  1 16 16 16 16
22  1 11 15 25 23
```

Here below we are going to hyper tune and fit svm model using **tune.svm** with multiple cost parameters and gamma values. The model tunes the svm for each of the value and then selects the best value possible for the model. We will then print the model and also summarize and plot the model using **summary(), plot().**

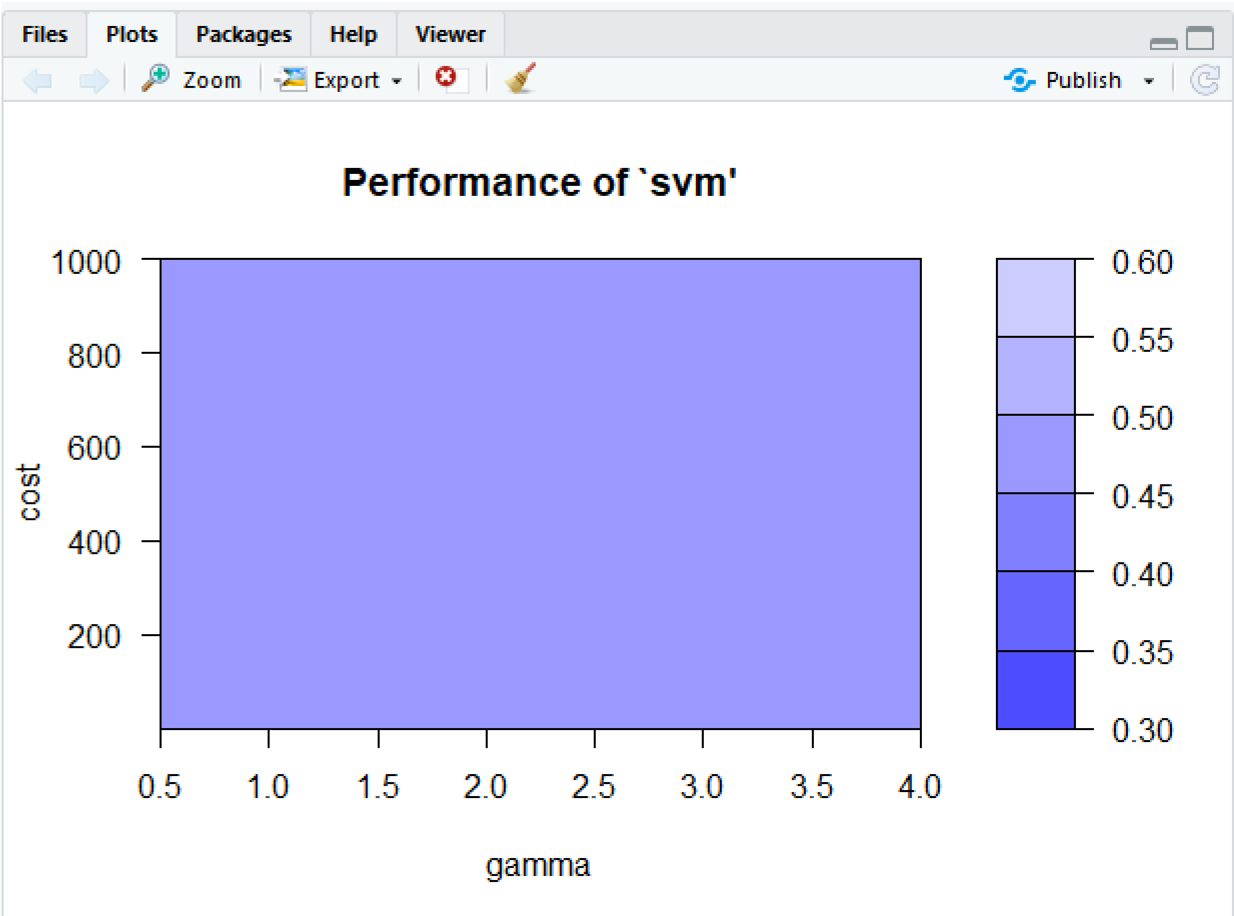
model <- tune.svm(V1 ~ 1., data = data, kernel='radial', cost=c(0.1,1,10,100,1000),gamma=c(0.5, 1,2,3,4))



print(model)
summary(model)

```
> summary(model)

Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 gamma cost
   0.5  0.1

- best performance: 0.4915238

- Detailed performance results:
   gamma  cost     error dispersion
1    0.5 1e-01 0.4915238  0.1800866
2    1.0 1e-01 0.4915238  0.1800866
3    2.0 1e-01 0.4915238  0.1800866
4    3.0 1e-01 0.4915238  0.1800866
5    4.0 1e-01 0.4915238  0.1800866
6    0.5 1e+00 0.4915238  0.1800866
7    1.0 1e+00 0.4915238  0.1800866
8    2.0 1e+00 0.4915238  0.1800866
9    3.0 1e+00 0.4915238  0.1800866
10   4.0 1e+00 0.4915238  0.1800866
11   0.5 1e+01 0.4915238  0.1800866
12   1.0 1e+01 0.4915238  0.1800866
13   2.0 1e+01 0.4915238  0.1800866
14   3.0 1e+01 0.4915238  0.1800866
15   4.0 1e+01 0.4915238  0.1800866
16   0.5 1e+02 0.4915238  0.1800866
17   1.0 1e+02 0.4915238  0.1800866
18   2.0 1e+02 0.4915238  0.1800866
19   3.0 1e+02 0.4915238  0.1800866
20   4.0 1e+02 0.4915238  0.1800866
21   0.5 1e+03 0.4915238  0.1800866
22   1.0 1e+03 0.4915238  0.1800866
23   2.0 1e+03 0.4915238  0.1800866
24   3.0 1e+03 0.4915238  0.1800866
```

plot(model)


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Performance of `svm'
1000
800
600
400
200
cost
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
gamma
0.60
0.55
0.50
0.45
0.40
0.35
0.30

Problem 2:

Explanation with the Code:

Now we will proceed to solve the Problem 2 where we need to perform the step wise regression of our dataset with **class = V1= 1** in forward direction.

In stepwise regression, we pass the full model to step function. It iteratively searches the full scope of variables in backwards directions by default, if scope is not given. It performs multiple iteractions by droping one X variable at a time. In each iteration, multiple models are built by dropping each of the X variables at a time. The AIC of the models is also computed and the model that yields the lowest AIC is retained for the next iteration.

In simpler terms, the variable that gives the minimum AIC when dropped, is dropped for the next iteration, until there is no significant drop in AIC is noticed.

The code below shows how stepwise regression can be done. In forward stepwise, variables will be progressively added.

Here we are declaring a model min.model and performing regression using **lm()** with our class as given "1" and data as (V2,V3,V4,V5).

The stepwise regression (or stepwise selection) consists of iteratively adding and removing predictors, in the predictive model, in order to find the subset of variables in the data set resulting in the best performing model, that is a model that lowers prediction error.

There are three strategies of stepwise regression (James et al. 2014,P. Bruce and Bruce (2017)) one of the Strategy which we are focused on is Forward Selection.

Forward selection, which starts with no predictors in the model, iteratively adds the most contributive predictors, and stops when the improvement is no longer statistically significant.

```
min.model = lm(data$V1 ~ 1)
min.model = lm(data$V1 ~ 1, data=data)
```

```
Start:  AIC=-83.95
data$V1 ~ 1
```

Now we are going to specify the forward stepwise model **fwd.model using step()** giving model , direction and scope. An then print summary of our model and see the Coefficient Intercept.

```
fwd.model = step(min.model, direction='forward', scope=(~ .))
```

summary(fwd.model)

```
Call:
lm(formula = data$V1 ~ 1, data = data)

Residuals:
   Min     1Q Median     3Q    Max 
  -0.5   -0.5    0.0    0.5    0.5 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.50000    0.06402   23.43   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5041 on 61 degrees of freedom
```

Now we are going to do the same thing but with different approach by updating at each step as we go for forward selection taking first V1 then updating subsequent columns/ features V2, V3,V4,V5 as shown below.

```
g <- lm(V1~. , data=data)
summary(g)
```

```
Call:
lm(formula = V1 ~ ., data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.85595 -0.21193 -0.00258  0.26625  0.62049 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.247650   0.244837   9.180 7.85e-13 ***
V2          -0.051552   0.018629  -2.767  0.00761 ** 
V3           0.020424   0.012838   1.591  0.11717    
V4          -0.046625   0.007602  -6.134 8.68e-08 ***
V5           0.031227   0.010066   3.102  0.00299 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.33 on 57 degrees of freedom
Multiple R-squared:  0.5995,	Adjusted R-squared:  0.5714 
F-statistic: 21.33 on 4 and 57 DF,  p-value: 8.515e-11
```

```
g <- update(g, . ~ . - V1)
g
```

```r
g <- update(g, . ~ . - V1)
g
```

```
Call:
lm(formula = V1 ~ V2 + V3 + V4 + V5, data = data)

Coefficients:
(Intercept)           V2           V3           V4           V5  
    2.24765     -0.05155      0.02042     -0.04662      0.03123  
```

g <- update(g, . ~ . - V2)
g

g <- update(g, . ~ . - V3)
g

```
g <- update(g, . ~ . - V2)
g
```

```
Call:
lm(formula = V1 ~ V3 + V4 + V5, data = data)

Coefficients:
(Intercept)           V3           V4           V5
   1.992644     0.004769    -0.056319     0.030002
```

```
g <- update(g, . ~ . - V3)
g
```

```
Call:
lm(formula = V1 ~ V4 + V5, data = data)

Coefficients:
(Intercept)           V4           V5
    2.02722     -0.05532      0.03066
```

g <- update(g, . ~ . - V4)
g

g <- update(g, . ~ . - V5)
g

```
g <- update(g, . ~ . - V4)
g
```

```
Call:
lm(formula = V1 ~ V5, data = data)

Coefficients:
(Intercept)           V5
   1.633584    -0.005967
```

```
g <- update(g, . ~ . - V5)
g
```

```
Call:
lm(formula = V1 ~ 1, data = data)

Coefficients:
(Intercept)
        1.5
```

As we see we remain with one intercept value after updating V5 above which matches the earlier model which we extracted intercept by using step() directly. In Forward selection, which starts with no predictors in the model, iteratively adds the most contributive predictors, and stops when the improvement is no longer statistically significant. Hence remaining with a single intercept coefficient value at the end.

I will show you again our above g model using step and its stats.

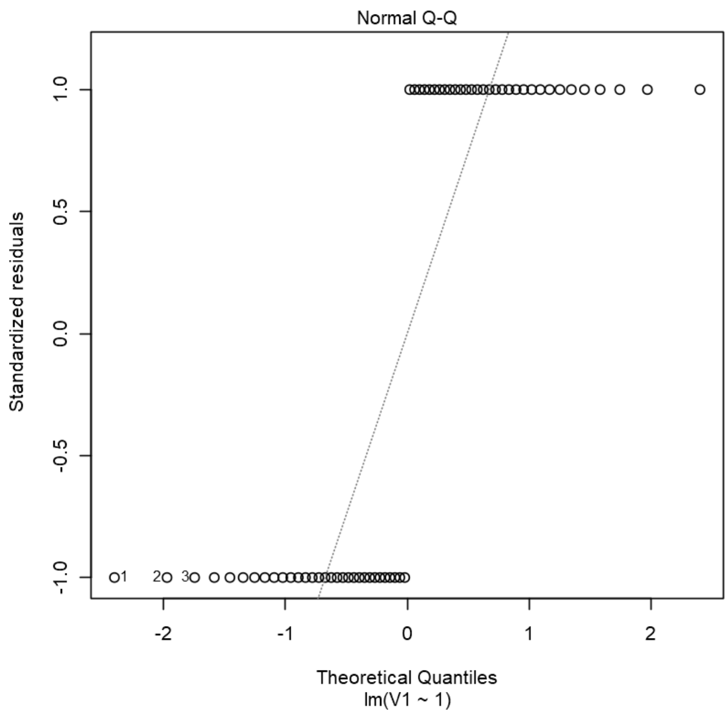
step(g , direction = "forward")

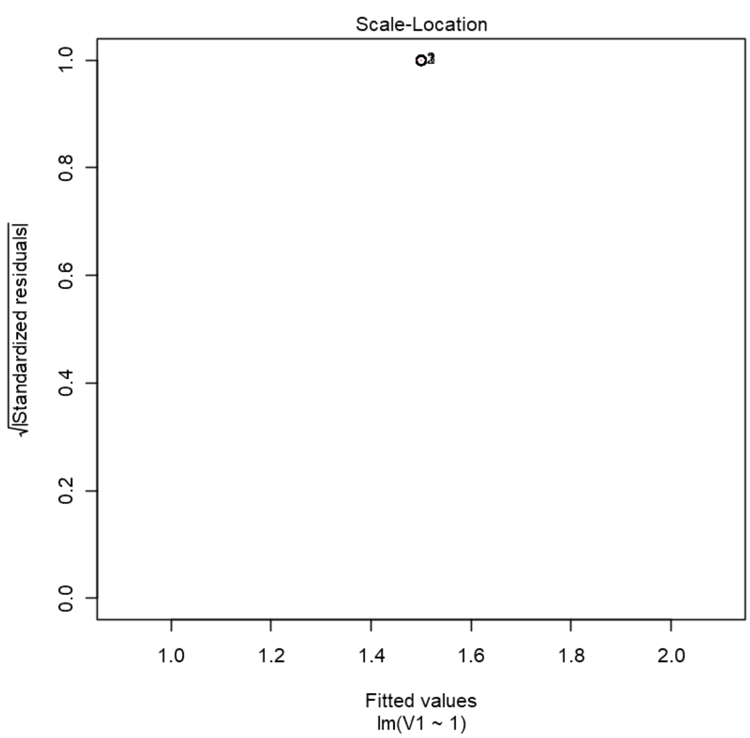
```
step(g , direction = "forward")
```

```
Start:  AIC=-83.95
V1 ~ 1



Call:
lm(formula = V1 ~ 1, data = data)

Coefficients:
(Intercept)
        1.5
```

Now we will Plot the g model we created. The first plot shows the the theoretical quantiles against std. residuals. Second plot shows the scaling of the location of the class against sqrt of residuals.

plot(g)





Problem 3: PART b) Deep Learning SVDD.

Modeling For modeling, I am using R’s H2O implementation with the h2o package. For more details and other examples, see posts of machine learning webinar (https://shiring.github.io/machine_learning/2017/03/31/webinar_code), on building neural nets with h2o (https://shiring.github.io/machine_learning/2017/02/27/h2o).

First load the packages we will require tidyverse, h20 , anomaly.

library(tidyverse)

library(h2o)

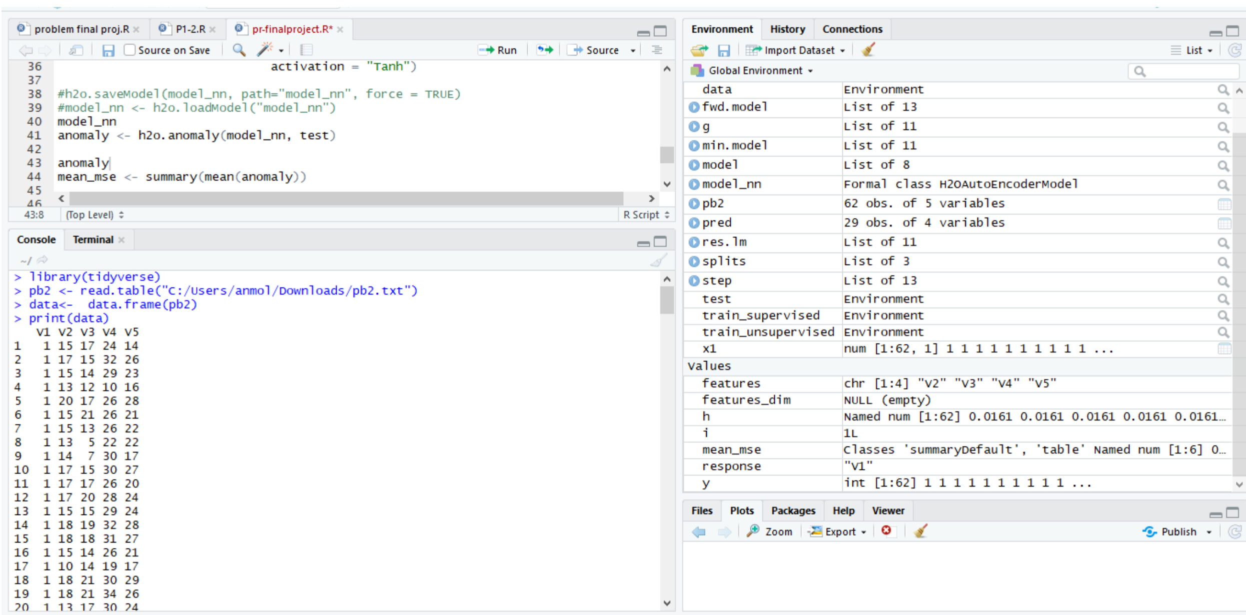
library(anomaly)

Now we need to import our dataset.

pb2 <- read.table("C:/Users/anmol/Downloads/pb2.txt")

data<- data.frame(pb2)

print(data)



Printing summary of our dataset and also separately for class 1.

summary(data)

summary(data[data$V1 == "1",])

```
Console   Terminal ×
~/
60  2  7  7 19 18
61  2 12 15  7 28
62  2  6  5  6 13
>
> summary(data)
       V1              V2               V3               V4               V5       
 Min.   :1.0   Min.   : 2.00   Min.   : 5.00   Min.   : 6.00   Min.   : 9.00  
 1st Qu.:1.0   1st Qu.:12.25   1st Qu.:13.00   1st Qu.:16.00   1st Qu.:20.25  
 Median :1.5   Median :14.00   Median :16.00   Median :23.00   Median :23.00  
 Mean   :1.5   Mean   :14.13   Mean   :14.90   Mean   :21.94   Mean   :22.39  
 3rd Qu.:2.0   3rd Qu.:16.00   3rd Qu.:17.75   3rd Qu.:28.00   3rd Qu.:26.00  
 Max.   :2.0   Max.   :20.00   Max.   :21.00   Max.   :34.00   Max.   :29.00  
>
> summary(data[data$V1 == "1", ])
       V1          V2              V3               V4              V5       
 Min.   :1   Min.   :10.0   Min.   : 5.00   Min.   :10.0   Min.   :14.00  
 1st Qu.:1   1st Qu.:15.0   1st Qu.:14.00   1st Qu.:25.5   1st Qu.:21.00  
 Median :1   Median :16.0   Median :16.00   Median :28.0   Median :23.00  
 Mean   :1   Mean   :15.9   Mean   :15.87   Mean   :27.0   Mean   :22.65  
 3rd Qu.:1   3rd Qu.:18.0   3rd Qu.:18.50   3rd Qu.:30.0   3rd Qu.:26.00  
 Max.   :1   Max.   :20.0   Max.   :21.00   Max.   :34.0   Max.   :29.00  
```

Below code shows how we will get connected to h2o cluster .

h2o.init(nthreads = -1)

```
>
> library(h2o)
> h2o.init(nthreads = -1)
 Connection successful!

R is connected to the H2O cluster: 
    H2O cluster uptime:         1 days 10 minutes 
    H2O cluster timezone:       -05:00 
    H2O data parsing timezone:  UTC 
    H2O cluster version:        3.20.0.8 
    H2O cluster version age:    2 months and 11 days  
    H2O cluster name:           H2O_started_from_R_anmol_ehz555 
    H2O cluster total nodes:    1 
    H2O cluster total memory:   2.48 GB 
    H2O cluster total cores:    4 
    H2O cluster allowed cores:  4 
    H2O cluster healthy:        TRUE 
    H2O Connection ip:          localhost 
    H2O Connection port:        54321 
    H2O Connection proxy:       NA 
    H2O Internal Security:      FALSE 
    H2O API Extensions:         Algos, AutoML, Core V3, Core V4 
    R Version:                  R version 3.4.3 (2017-11-30) 
```

Below code shows how we will save our data to h2o cluster we created above **(H2O_started_from_R_anmol_ehz555)**.

Now we will save this data in data frame format to h2o cluster.

convert data to H2OFrame

data <- as.h2o(data)

Now we will split the data into supervised training set and unsupervised training set and rest of it as testing set.

The ratio c(0.25,0.25) should be given less than 1 for splitting the data frame. Here 0.25 refers to 25% of dataset. So total first 50% of our dataset will go under training set(includes 25% supervised and 25% unsupervised) and rest of it as testing set.

```
splits <- h2o.splitFrame(data,
                         ratios = c(0.25, 0.25),
                         seed = 62)
```

Then we will allocate this split parts respectively to supervised , unsupervised and testing sets.

```
train_unsupervised  <- splits[[1]]
train_supervised  <- splits[[2]]
test <- splits[[3]]
```

Then we declare response for our class as V1 and rest columns as our features.

```
response <- "V1"
features <- setdiff(colnames(train_unsupervised), response)
```

Then we create our deep learning model using h2o.deeplearning with suitable parameters as shown below.

```
model_nn <- h2o.deeplearning(x = features,
                             training_frame = train_unsupervised,
                             model_id = "model_nn",
                             autoencoder = TRUE,
                             reproducible = TRUE, #slow - turn off for real problems
                             ignore_const_cols = FALSE,
                             seed = 42,
                             hidden = c(10, 2, 10),
                             epochs = 100,
                             activation = "Tanh")
model_nn
```

```
Model Details:
==============

H2OAutoEncoderModel: deeplearning
Model ID:  model_nn
Status of Neuron Layers: auto-encoder, gaussian distribution, Quadratic loss, 146 weigh
ts/biases, 6.0 KB, 1,700 training samples, mini-batch size 1
  layer units  type dropout       l1       l2 mean_rate rate_rms momentum
1     1     4 Input  0.00 %       NA       NA        NA       NA       NA
2     2    10  Tanh  0.00 % 0.000000 0.000000  0.055073 0.032179 0.000000
3     3     2  Tanh  0.00 % 0.000000 0.000000  0.050587 0.042692 0.000000
4     4    10  Tanh  0.00 % 0.000000 0.000000  0.013229 0.006946 0.000000
5     5     4  Tanh      NA 0.000000 0.000000  0.015977 0.011171 0.000000
  mean_weight weight_rms mean_bias bias_rms
1          NA         NA        NA       NA
2   -0.110606   0.406723  0.003262 0.058956
3   -0.033283   0.463853  0.012037 0.041905
4    0.055432   0.408690 -0.001309 0.049260
5    0.013903   0.407261 -0.074322 0.013025


H2OAutoEncoderMetrics: deeplearning
** Reported on training data. **

Training Set Metrics:
=====================

MSE: (Extract with `h2o.mse`) 0.02607657
RMSE: (Extract with `h2o.rmse`) 0.1614824
```

pred <- as.data.frame(h2o.predict(object = model_nn, newdata = test))

pred

```
> pred <- as.data.frame(h2o.predict(object = model_nn, newdata = test))
  |============================================================================| 100%
> pred
   reconstr_V2 reconstr_V3 reconstr_V4 reconstr_V5
1    16.400402   17.954015    28.14696    25.57341
2    14.353768   16.155506    26.40398    23.28497
3    11.780462    8.147822    11.26361    16.82663
4    19.490523   18.383070    25.21076    27.53755
5    14.132387   17.037151    28.58179    23.69351
6    12.626558   10.291875    15.27204    18.67344
7    16.314832   18.122866    28.62518    25.62899
8    14.682529   16.707463    27.26564    23.81681
9    17.423608   18.754798    28.77864    26.63472
10   17.871613   19.374697    29.67115    27.25766
11   16.670876   19.371086    31.06050    26.60016
12   16.452898   17.382956    26.79949    25.25296
13   14.689031   16.983376    27.86847    23.99002
14   13.607299   10.779812    15.34014    19.53998
15   12.227722   14.785254    25.71333    21.17262
16   10.358022   11.955852    21.61009    18.35811
17   11.121650    8.333973    12.37034    16.59368
18   13.069004   15.225897    25.75624    21.95092
19   14.994742   16.355838    26.14558    23.78412
20   11.958109   10.353716    16.12590    18.33308
21   16.756647   13.840540    18.62624    23.27658
22   13.731359   15.364356    25.33873    22.43276
23   15.335191   15.919434    24.79967    23.71539
24   16.384099   15.044862    21.67508    23.79057
25    8.923257    6.538418    10.47181    14.29848
26   14.139125   15.855367    25.97578    22.97452
```

Now we will check for anomalies using h20.anomaly for our test data with the model we created above.

anomaly <- h2o.anomaly(model_nn, test)

anomaly

mean_mse <- summary(mean(anomaly))

mean_mse

```
> anomaly <- h2o.anomaly(model_nn, test)
>
> anomaly
  Reconstruction.MSE
1        0.014346901
2        0.007556481
3        0.017998400
4        0.002703263
5        0.026796628
6        0.055372575

[29 rows x 1 column]
> mean_mse <- summary(mean(anomaly))
>
> mean_mse
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02475 0.02475 0.02475 0.02475 0.02475 0.02475
>
```

Its able to detect 29 rows x 1 col as anomalies which shows it has great accuracy and efficiency as we have total 31 anomaly/outliers points but this model was able to detect 29 of those.So I think its pretty good.

Problem 3: PART a) SVDD.

Code with Explanation:

First we are loading the required packages.

```
library(e1071)

library(quadprog)
```

Then we are going to load our Dataset.

```
data <- read.table("C:/Users/anmol/Downloads/pb2.txt")

#Features


X = as.matrix(data[,2:5])

y = as.matrix(data[, 1])

n <- length(y)
```

Then we will replace all class values by "-1" which are not "1".

```
for (i in 1:n){

  if (y[i] > 1){

    y[i]<--1

  }

}
```

Then we are going to define our gaussian kernel.

```
gaussianKern <- function(x, y, sigma){

  exp(-(t(x-y)%*%(x-y))/(2*sigma^2))

}
```

```
> gaussianKern(c(5, 10), c(1, 7), sigma=0.5)
              [,1]
[1,] 1.92875e-22
```

Then we will define the function for calculating gram matrix for our dataset so that we can apply further equations which you can refer from the Question paper.

```
gram_mat <- function(mydat, sigma){

  N <- dim(mydat)[1]

  if (!is.matrix(mydat)) mydat <- as.matrix(mydat) #change class of mydat to matrix

  gram_matrix <- matrix(0, N, N)

  for(i in 1:N){

    for(k in 1:N){
```

```
      gram_matrix[i,k] <- gaussianKern(mydat[i,], mydat[k,], sigma=sigma)
    }
  }
  print(gram_matrix)
}
```

You can see the output below for our training data of class 1 data with sigma = 1.5 and its respective generated gram matrix.

```
> trainingData <- dataTrain(31, 5)
> sig <- 1.5
> gm <- gram_mat(trainingData, sig)
              [,1]         [,2]         [,3]         [,4]        [,5]         [,6]
 [1,] 1.000000e+00 0.0563566595 4.498005e-05 7.976390e-05 0.007921912 2.300353e-02
 [2,] 5.635666e-02 1.0000000000 1.551939e-02 6.334805e-03 0.047310818 1.221121e-02
 [3,] 4.498005e-05 0.0155193858 1.000000e+00 2.044467e-04 0.023730433 3.519834e-03
 [4,] 7.976390e-05 0.0063348052 2.044467e-04 1.000000e+00 0.034012729 5.708144e-06
 [5,] 7.921912e-03 0.0473108181 2.373043e-02 3.401273e-02 1.000000000 2.084771e-02
 [6,] 2.300353e-02 0.0122112095 3.519834e-03 5.708144e-06 0.020847706 1.000000e+00
 [7,] 2.678406e-03 0.0174513858 7.132731e-02 4.498639e-03 0.444507223 6.931481e-03
 [8,] 9.588512e-05 0.0050548294 3.467709e-03 5.962054e-04 0.060356389 5.152105e-04
 [9,] 2.055413e-04 0.0066286414 1.096495e-02 2.056372e-07 0.001252052 6.811580e-02
[10,] 1.041680e-01 0.0293094107 5.958292e-03 3.799637e-04 0.093913894 2.636767e-02
[11,] 1.352273e-02 0.2885677389 1.138287e-02 5.341232e-03 0.119074561 5.062708e-03
[12,] 9.106412e-05 0.0251174663 3.467717e-01 8.590268e-03 0.207732435 3.534152e-03
[13,] 8.916797e-03 0.0816019924 9.343537e-03 2.226577e-03 0.213000940 1.254290e-01
[14,] 3.913771e-02 0.3631021165 1.135982e-02 2.155718e-02 0.342291895 3.830792e-02
[15,] 6.628123e-03 0.2364688415 1.234856e-01 2.839249e-03 0.028733337 7.510646e-03
[16,] 9.210415e-04 0.0392968152 8.582698e-03 3.054864e-03 0.158996301 8.058820e-03
[17,] 3.204755e-03 0.0190441882 4.250858e-03 3.170931e-07 0.001963932 3.757994e-02
[18,] 8.107114e-03 0.3660060410 5.550151e-02 6.056755e-04 0.011917637 2.734745e-02
[19,] 1.521608e-01 0.3133053405 1.405764e-02 1.959369e-03 0.211774186 5.743245e-02
[20,] 7.416183e-04 0.1172763693 2.072887e-01 3.903650e-05 0.003114821 5.702446e-03
[21,] 1.403986e-03 0.0088873845 3.855704e-03 4.829063e-02 0.095337362 2.207450e-03
[22,] 2.206391e-02 0.3310101899 8.374154e-03 1.371130e-03 0.035572851 9.110031e-02
[23,] 1.486895e-04 0.0072989744 4.780333e-03 2.961026e-01 0.208229627 4.702885e-05
[24,] 3.883578e-04 0.1319421170 3.978850e-03 3.558616e-02 0.031949460 3.391855e-04
[25,] 7.782999e-03 0.0009235249 8.522288e-04 1.722166e-04 0.048939827 2.424992e-03
[26,] 8.703985e-04 0.0286253927 2.869492e-03 1.173331e-01 0.349478405 1.996715e-03
[27,] 1.000406e-04 0.0025770651 3.101587e-02 4.106747e-03 0.243943201 4.493129e-04
[28,] 1.033143e-03 0.0068414398 9.544915e-02 2.476699e-03 0.296271172 1.247792e-02
[29,] 5.619964e-02 0.0711248285 3.613265e-02 4.838144e-05 0.074689372 1.853116e-01
[30,] 1.904131e-02 0.0624199807 5.254441e-03 8.137245e-03 0.485656793 8.959187e-02
[31,] 2.552920e-04 0.0196389524 7.178037e-02 1.982288e-03 0.048875099 1.432935e-04
```

```
[15,] 3.878040e-03 0.0039742510 0.0035843596 0.0503967274 5.449273e-02 0.010210190
[16,] 2.108531e-04 0.3015666868 0.0377248826 0.0075019087 2.133720e-02 0.307263652
[17,] 5.202412e-05 0.0003125823 0.0003842375 0.0004382070 1.468434e-01 0.007667146
[18,] 2.233009e-04 0.0041625860 0.0004210963 0.0072125926 4.532098e-02 0.017566340
[19,] 2.491899e-02 0.0424780934 0.0313489710 0.0422603183 4.447280e-01 0.215364976
[20,] 1.122361e-04 0.0004543634 0.0007534446 0.0054691047 5.040752e-02 0.002045780
[21,] 1.370169e-02 0.0247354171 0.0045258073 0.1049924930 2.481125e-03 0.027244668
[22,] 1.426331e-04 0.0382419462 0.0004846220 0.0038194243 3.941650e-02 0.160631634
[23,] 6.888768e-03 0.1466834949 0.1827660151 0.0601144081 1.268162e-03 0.022324335
[24,] 1.054528e-05 0.2068837829 0.0033520138 0.0008637815 1.566684e-03 0.050631013
[25,] 1.000000e+00 0.0006539132 0.0260030494 0.1558371332 3.865110e-02 0.005647761
[26,] 6.539132e-04 1.0000000000 0.0397457267 0.0168345968 3.589812e-03 0.367217684
[27,] 2.600305e-02 0.0397457267 1.0000000000 0.2043847046 1.710240e-02 0.026298487
[28,] 1.558371e-01 0.0168345968 0.2043847046 1.0000000000 7.301050e-02 0.045728294
[29,] 3.865110e-02 0.0035898120 0.0171024041 0.0730105021 1.000000e+00 0.065122717
[30,] 5.647761e-03 0.3672176841 0.0262984866 0.0457282943 6.512272e-02 1.000000000
[31,] 7.086020e-03 0.0065263227 0.2499438263 0.0477655735 2.795708e-02 0.004339534
             [,31]
 [1,] 0.0002552920
 [2,] 0.0196389524
 [3,] 0.0717803735
 [4,] 0.0019822879
 [5,] 0.0488750987
 [6,] 0.0001432935
 [7,] 0.2391657722
 [8,] 0.0379744161
 [9,] 0.0002256946
[10,] 0.0238476063
[11,] 0.0891574335
[12,] 0.0981004169
[13,] 0.0026725725
[14,] 0.0175966382
[15,] 0.0254482220
[16,] 0.0143045708
[17,] 0.0018381232
```

Then we are going to calculate distance of each data point and calculate the mean of that distance which you can see the output below the kernelDistance function.

```
kernelDistance <- function(point, data, alphas, gramMat, sigma){

 #calculate the distance for a single data point in the gaussian kernel space

 t1 <- gaussianKern(point, point, sigma)

 t2 <- -2*sum(sapply(1:length(alphas), function(m) alphas[m]*gaussianKern(point, data[m,], sigma)))

 t3 <- t(alphas) %*% gramMat %*% alphas

 sqrt(t1+t2+t3)

}
```

```
> kernelDistance(trainingData[1,], trainingData, alphaValues, gm, sig)
          [,1]
[1,] 0.9711759
```

Then we are going to store this a vector list of distance make.d.vec as shown below:

```
make.d.vec <- function(mydat, sigma){

 #creates the d vector for quadprog


 d <- sapply(1:dim(mydat)[1], function(m) gaussianKern(mydat[m,], mydat[m,], sigma=sigma))
```

```
  print(d)
}
```

Now we will train our dataset for class1 data and show the plot for its respective data points after training. This will return all class1 datapoints.

```
dataTrain <- function(n, p, negativeProportion=0){
  numNegative <- round(negativeProportion*n)
  numPositive <- n-numNegative
  positiveMean <- rnorm(p, mean=4, sd=1)
  negativeMean <- rnorm(p, mean=-4, sd=1)
  Mat <- matrix(0, p, p)
  for(i in 1:p){
    for(j in 1:p){
      if(i==j){Mat[i, j] <- 2}
      else{
        Mat[i, j] <- 0.1 ^ abs(i-j)}
    }
  }
  sigma <- Mat
  positiveData <- mvrnorm(numPositive, positiveMean, sigma)
  if(numNegative > 0) {
    negativeData <- mvrnorm(numNegative, negativeMean, sigma)
    return(rbind(positiveData, negativeData))
  }
  else return(positiveData)
}
```

```
> dataTrain(31, 5, 0.2)
           [,1]       [,2]      [,3]      [,4]        [,5]
 [1,]  3.613697  6.4106805  3.508484  5.038901  2.10878444
 [2,]  4.404582  2.5390523  3.562087  6.598613  3.37729584
 [3,]  4.489537  5.6301670  4.243939  4.823457  3.35308154
 [4,]  4.169102  3.5948120  5.480779  6.626076  1.40334116
 [5,]  1.455932  4.5114567  5.725625  6.422430  1.03117738
 [6,]  2.032707  4.6223550  4.220467  6.966330  1.93801175
 [7,]  5.567808  3.6110019  3.106678  6.314397  4.04207615
 [8,]  3.788002  5.3519879  3.655099  6.431344  3.53763394
 [9,]  2.023585  3.9628681  6.867409  6.299948  3.04322378
[10,]  1.381936  2.2728589  3.650975  8.951605  3.73442547
[11,]  3.850311  5.6679231  4.838118  3.509204  2.07015453
[12,]  1.490891  1.7274810  3.464555  4.739973  1.56422572
[13,]  3.174189  4.3291053  4.851582  6.621133  0.52207822
[14,]  4.249751  2.0791764  3.744992  5.483598  1.00922313
[15,]  3.650942  2.2630585  3.839940  7.721521  2.63462912
[16,]  4.992859  2.0099689  4.582876  5.547778  4.20538280
[17,]  5.552282  6.0706376  2.009404  6.572026  2.15183682
[18,]  2.816877  4.0182239  3.640791  4.758830  4.90901951
[19,]  1.601029  5.5211954  3.802419  3.433840  4.76659232
[20,]  1.342090  5.2415482  3.665378  6.662303  4.29524414
[21,]  3.085520  3.3092393  3.531508  7.381407  0.91842660
[22,]  3.795330  4.5205781  3.105072  9.827880  6.29791977
[23,]  2.633773  3.1638666  3.033717  6.916706  0.03133517
[24,]  2.862843  7.1332677  5.592685  7.403250  3.89388567
[25,]  2.647861  4.5010906  6.301277  4.950981  1.30880381
[26,] -4.428188 -3.0013765 -7.638683 -2.299671 -3.11093536
[27,] -4.735069 -0.3710194 -2.910293 -5.355502 -3.83641225
[28,] -3.199763  0.6067856 -3.569291 -1.709475 -1.81276760
[29,] -4.498749  0.3586353 -3.331098 -2.763715  0.70895780
[30,] -4.356033 -2.0583478 -4.209705 -2.070813 -4.11100541
[31,] -7.678227 -1.7633708 -2.766896 -4.131038 -4.53291839
> 
```

Now we are going to create a function name svddTrain to to detect all support vectors and outliers for above generated trained data and apply to it.

```
svddTrain <- function(X, Gram_Matrix, sigma, C1, C2=0, negativeProportion=0){
 if (!is.matrix(X)) X <- as.matrix(X)
 N <- dim(X)[1]
 numNegative <- round(negativeProportion*N) #number of negative rows in training data
 numPositive <- N-numNegative #number of positive rows in training data
 d <- make.d.vec(X, sigma)
 D <- gram_mat(X, sigma)
 D <- 2*D
 D <- D + diag(dim(D)[1])*1e-12
```

Then creating b, the first and second row makes sure alphas sum to 1 and others guarantee they are greater than 0.

```
bv <- c(1,
        rep(0, N),
        rep(-C1, numPositive),
        rep(-C2, numNegative))
```

Initialize the designed A matrix to go along with bv:

```
A  <- cbind(rep(1, N), diag(N), -diag(N))
alpha <- solve.QP(D, d, A, bv, meq=1)$solution #the alphas
non_zero_alphas <- alpha[round(alpha, digits=4) > 0]
locations <- which(round(alpha, digits=4) > 0)
support_vectors <- X[locations,]
num_SVs <- length(locations)
center <- t(alpha) %*% X
return(list(num_SVs=num_SVs,
           locations=locations,
           alpha=alpha,
           nza=non_zero_alphas,
           sv=support_vectors,
           ctr=center))
}
```

```
> alphaValues <- svdd$alpha
> alphaValues
 [1]  5.609992e-02  4.637145e-02  1.310323e-02  3.350254e-02  0.000000e+00
 [6]  4.173378e-02  4.319670e-02  4.621810e-02  2.966826e-02  6.240512e-02
[11] -3.249747e-18  5.282196e-02  4.007884e-04  6.093297e-02  1.152881e-02
[16]  3.983577e-02  4.845762e-02  8.408624e-03  2.518833e-02  5.334694e-02
[21]  4.750722e-02  4.378185e-02  9.613720e-03  4.581417e-02  2.007690e-02
[26] -2.568416e-19  3.423334e-02  3.746832e-04  4.305743e-02  3.278342e-02
[31]  4.953635e-02
```

Now we are going to run svddTrain to to detect all support vectors and outliers for for our training data and apply to it. This svddTrain returns locations,alpha values and support vectors. We also see its output below with the plot of its data points.

```
> svddTrain(trainingData, gm, sig, 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]         [,2]         [,3]         [,4]         [,5]
[,6]
```

```
 [1,] 1.000000e+00 0.0563566595 4.498005e-05 7.976390e-05 0.007921912 2.30035
3e-02
 [2,] 5.635666e-02 1.0000000000 1.551939e-02 6.334805e-03 0.047310818 1.22112
1e-02
 [3,] 4.498005e-05 0.0155193858 1.000000e+00 2.044467e-04 0.023730433 3.51983
4e-03
 [4,] 7.976390e-05 0.0063348052 2.044467e-04 1.000000e+00 0.034012729 5.70814
4e-06
 [5,] 7.921912e-03 0.0473108181 2.373043e-02 3.401273e-02 1.000000000 2.08477
1e-02
 [6,] 2.300353e-02 0.0122112095 3.519834e-03 5.708144e-06 0.020847706 1.00000
0e+00
 [7,] 2.678406e-03 0.0174513858 7.132731e-02 4.498639e-03 0.444507223 6.93148
1e-03
 [8,] 9.588512e-05 0.0050548294 3.467709e-03 5.962054e-04 0.060356389 5.15210
5e-04
 [9,] 2.055413e-04 0.0066286414 1.096495e-02 2.056372e-07 0.001252052 6.81158
0e-02
[10,] 1.041680e-01 0.0293094107 5.958292e-03 3.799637e-04 0.093913894 2.63676
7e-02
[11,] 1.352273e-02 0.2885677389 1.138287e-02 5.341232e-03 0.119074561 5.06270
8e-03
[12,] 9.106412e-05 0.0251174663 3.467717e-01 8.590268e-03 0.207732435 3.53415
2e-03
[13,] 8.916797e-03 0.0816019924 9.343537e-03 2.226577e-03 0.213000940 1.25429
0e-01
[14,] 3.913771e-02 0.3631021165 1.135982e-02 2.155718e-02 0.342291895 3.83079
2e-02
[15,] 6.628123e-03 0.2364688415 1.234856e-01 2.839249e-03 0.028733337 7.51064
6e-03
[16,] 9.210415e-04 0.0392968152 8.582698e-03 3.054864e-03 0.158996301 8.05882
0e-03
[17,] 3.204755e-03 0.0190441882 4.250858e-03 3.170931e-07 0.001963932 3.75799
4e-02
[18,] 8.107114e-03 0.3660060410 5.550151e-02 6.056755e-04 0.011917637 2.73474
5e-02
[19,] 1.521608e-01 0.3133053405 1.405764e-02 1.959369e-03 0.211774186 5.74324
5e-02
[20,] 7.416183e-04 0.1172763693 2.072887e-01 3.903650e-05 0.003114821 5.70244
6e-03
[21,] 1.403986e-03 0.0088873845 3.855704e-03 4.829063e-02 0.095337362 2.20745
0e-03
[22,] 2.206391e-02 0.3310101899 8.374154e-03 1.371130e-03 0.035572851 9.11003
1e-02
[23,] 1.486895e-04 0.0072989744 4.780333e-03 2.961026e-01 0.208229627 4.70288
5e-05
[24,] 3.883578e-04 0.1319421170 3.978850e-03 3.558616e-02 0.031949460 3.39185
5e-04
[25,] 7.782999e-03 0.0009235249 8.522288e-04 1.722166e-04 0.048939827 2.42499
2e-03
[26,] 8.703985e-04 0.0286253927 2.869492e-03 1.173331e-01 0.349478405 1.99671
5e-03
[27,] 1.000406e-04 0.0025770651 3.101587e-02 4.106747e-03 0.243943201 4.49312
9e-04
[28,] 1.033143e-03 0.0068414398 9.544915e-02 2.476699e-03 0.296271172 1.24779
2e-02
[29,] 5.619964e-02 0.0711248285 3.613265e-02 4.838144e-05 0.074689372 1.85311
6e-01
[30,] 1.904131e-02 0.0624199807 5.254441e-03 8.137245e-03 0.485656793 8.95918
7e-02
```

```
[31,] 2.552920e-04 0.0196389524 7.178037e-02 1.982288e-03 0.048875099 1.43293
5e-04
              [,7]         [,8]         [,9]        [,10]       [,11]
[,12]
 [1,] 0.0026784064 9.588512e-05 2.055413e-04 0.1041679756 0.013522734 9.10641
2e-05
 [2,] 0.0174513858 5.054829e-03 6.628641e-03 0.0293094107 0.288567739 2.51174
7e-02
 [3,] 0.0713273116 3.467709e-03 1.096495e-02 0.0059582923 0.011382872 3.46771
7e-01
 [4,] 0.0044986395 5.962054e-04 2.056372e-07 0.0003799637 0.005341232 8.59026
8e-03
 [5,] 0.4445072233 6.035639e-02 1.252052e-03 0.0939138941 0.119074561 2.07732
4e-01
 [6,] 0.0069314813 5.152105e-04 6.811580e-02 0.0263676699 0.005062708 3.53415
2e-03
 [7,] 1.0000000000 3.090549e-02 6.112935e-04 0.2116499156 0.053247903 1.77864
5e-01
 [8,] 0.0309054863 1.000000e+00 2.784073e-03 0.0009007756 0.203316834 5.34612
8e-02
 [9,] 0.0006112935 2.784073e-03 1.000000e+00 0.0003022169 0.008701951 5.25759
7e-03
[10,] 0.2116499156 9.007756e-04 3.022169e-04 1.0000000000 0.017788890 5.75608
3e-03
[11,] 0.0532479029 2.033168e-01 8.701951e-03 0.0177888895 1.000000000 5.75983
2e-02
[12,] 0.1778644678 5.346128e-02 5.257597e-03 0.0057560832 0.057598316 1.00000
0e+00
[13,] 0.0266124314 5.457920e-02 4.835197e-02 0.0086942225 0.147428451 7.43201
8e-02
[14,] 0.0631652820 6.067468e-02 9.924823e-03 0.0331860697 0.457531806 8.44562
7e-02
[15,] 0.0368520291 2.795171e-04 1.644995e-03 0.0493647015 0.021382253 4.80807
8e-02
[16,] 0.0304065038 4.686361e-01 1.894726e-02 0.0021358379 0.341081877 1.18726
5e-01
[17,] 0.0019662982 8.992508e-03 3.265900e-01 0.0028707808 0.047026234 2.08759
6e-03
[18,] 0.0054309054 3.199384e-04 1.953272e-02 0.0093580522 0.028341153 2.49271
6e-02
[19,] 0.1457647395 4.073855e-02 1.134727e-02 0.2360369717 0.444992519 3.30777
7e-02
[20,] 0.0061426472 4.321480e-04 2.917714e-02 0.0050090622 0.018615479 2.87879
0e-02
[21,] 0.0301555210 5.538649e-05 8.519618e-06 0.0187893037 0.001202053 1.82491
9e-02
[22,] 0.0040930943 2.435306e-03 4.157503e-02 0.0055758307 0.075683787 2.02695
6e-02
[23,] 0.1327338314 1.114127e-02 3.117463e-06 0.0068148062 0.022042929 8.28561
8e-02
[24,] 0.0035453457 3.975289e-02 1.298149e-03 0.0002865783 0.226555027 5.24355
3e-02
[25,] 0.1724966089 1.617025e-04 4.794432e-06 0.4485969470 0.001091929 1.46410
2e-03
[26,] 0.0360169563 7.300429e-02 3.830376e-04 0.0024779266 0.102970076 1.04925
6e-01
[27,] 0.5571822539 9.182120e-02 1.246153e-04 0.0210524367 0.032247541 1.82625
0e-01
[28,] 0.5970084062 3.348726e-03 4.272065e-04 0.1426406609 0.007805329 1.57702
3e-01
```

```
[29,] 0.1306039031 8.256243e-03 3.836418e-02 0.3050258146 0.073556671 2.14028
2e-02
[30,] 0.0699719091 5.033785e-02 7.836077e-03 0.0281398913 0.130792633 6.49440
1e-02
[31,] 0.2391657722 3.797442e-02 2.256946e-04 0.0238476063 0.089157433 9.81004
2e-02
             [,13]       [,14]        [,15]        [,16]        [,17]
[,18]
 [1,] 0.0089167974 0.039137713 0.0066281235 0.0009210415 3.204755e-03 0.00810
71141
 [2,] 0.0816019924 0.363102116 0.2364688415 0.0392968152 1.904419e-02 0.36600
60410
 [3,] 0.0093435374 0.011359815 0.1234856056 0.0085826981 4.250858e-03 0.05550
15071
 [4,] 0.0022265771 0.021557181 0.0028392491 0.0030548635 3.170931e-07 0.00060
56755
 [5,] 0.2130009403 0.342291895 0.0287333368 0.1589963014 1.963932e-03 0.01191
76373
 [6,] 0.1254290052 0.038307919 0.0075106462 0.0080588196 3.757994e-02 0.02734
74471
 [7,] 0.0266124314 0.063165282 0.0368520291 0.0304065038 1.966298e-03 0.00543
09054
 [8,] 0.0545792006 0.060674683 0.0002795171 0.4686360660 8.992508e-03 0.00031
99384
 [9,] 0.0483519666 0.009924823 0.0016449955 0.0189472599 3.265900e-01 0.01953
27233
[10,] 0.0086942225 0.033186070 0.0493647015 0.0021358379 2.870781e-03 0.00935
80522
[11,] 0.1474284513 0.457531806 0.0213822528 0.3410818765 4.702623e-02 0.02834
11532
[12,] 0.0743201841 0.084456273 0.0480807751 0.1187265274 2.087596e-03 0.02492
71624
[13,] 1.0000000000 0.549291950 0.0107648914 0.4324683060 2.722401e-02 0.03925
53710
[14,] 0.5492919502 1.000000000 0.0476199459 0.3461128797 1.711253e-02 0.08109
19895
[15,] 0.0107648914 0.047619946 1.0000000000 0.0025704231 1.899624e-03 0.46078
69995
[16,] 0.4324683060 0.346112880 0.0025704231 1.0000000000 2.105187e-02 0.00650
93950
[17,] 0.0272240065 0.017112533 0.0018996238 0.0210518719 1.000000e+00 0.01093
51434
[18,] 0.0392553710 0.081091990 0.4607869995 0.0065093950 1.093514e-02 1.00000
00000
[19,] 0.1627434741 0.434368944 0.0626566975 0.1055049817 7.240132e-02 0.05643
89385
[20,] 0.0062809489 0.014216033 0.2999274335 0.0027414382 2.477523e-02 0.39713
48515
[21,] 0.0079051391 0.022399345 0.0511193083 0.0009060003 4.020418e-06 0.01074
69702
[22,] 0.3327010221 0.346894256 0.0671850985 0.0590088140 2.074952e-02 0.36112
01519
[23,] 0.0055844010 0.035847364 0.0076542195 0.0142936661 8.155217e-06 0.00077
72953
[24,] 0.0846771136 0.237158688 0.0090158418 0.1985801499 1.297102e-03 0.02254
20076
[25,] 0.0007696002 0.002949783 0.0038780401 0.0002108531 5.202412e-05 0.00022
33009
[26,] 0.2284060373 0.341369377 0.0039742510 0.3015666868 3.125823e-04 0.00416
25860
```

```
[27,] 0.0097366841 0.021272999 0.0035843596 0.0377248826 3.842375e-04 0.00042
10963
[28,] 0.0176156108 0.026100906 0.0503967274 0.0075019087 4.382070e-04 0.00721
25926
[29,] 0.0622771853 0.081998231 0.0544927309 0.0213371995 1.468434e-01 0.04532
09833
[30,] 0.7301894225 0.589538074 0.0102101899 0.3072636521 7.667146e-03 0.01756
63404
[31,] 0.0026725725 0.017596638 0.0254482220 0.0143045708 1.838123e-03 0.00320
60488
            [,19]        [,20]        [,21]        [,22]        [,23]
[,24]
 [1,] 0.152160768 0.0007416183 1.403986e-03 0.0220639090 1.486895e-04 3.88357
8e-04
 [2,] 0.313305341 0.1172763693 8.887385e-03 0.3310101899 7.298974e-03 1.31942
1e-01
 [3,] 0.014057636 0.2072886650 3.855704e-03 0.0083741545 4.780333e-03 3.97885
0e-03
 [4,] 0.001959369 0.0000390365 4.829063e-02 0.0013711299 2.961026e-01 3.55861
6e-02
 [5,] 0.211774186 0.0031148210 9.533736e-02 0.0355728510 2.082296e-01 3.19494
6e-02
 [6,] 0.057432452 0.0057024457 2.207450e-03 0.0911003122 4.702885e-05 3.39185
5e-04
 [7,] 0.145764739 0.0061426472 3.015552e-02 0.0040930943 1.327338e-01 3.54534
6e-03
 [8,] 0.040738551 0.0004321480 5.538649e-05 0.0024353064 1.114127e-02 3.97528
9e-02
 [9,] 0.011347271 0.0291771414 8.519618e-06 0.0415750347 3.117463e-06 1.29814
9e-03
[10,] 0.236036972 0.0050090622 1.878930e-02 0.0055758307 6.814806e-03 2.86578
3e-04
[11,] 0.444992519 0.0186154788 1.202053e-03 0.0756837871 2.204293e-02 2.26555
0e-01
[12,] 0.033077768 0.0287879030 1.824919e-02 0.0202695643 8.285618e-02 5.24355
3e-02
[13,] 0.162743474 0.0062809489 7.905139e-03 0.3327010221 5.584401e-03 8.46771
1e-02
[14,] 0.434368944 0.0142160333 2.239934e-02 0.3468942563 3.584736e-02 2.37158
7e-01
[15,] 0.062656697 0.2999274335 5.111931e-02 0.0671850985 7.654220e-03 9.01584
2e-03
[16,] 0.105504982 0.0027414382 9.060003e-04 0.0590088140 1.429367e-02 1.98580
1e-01
[17,] 0.072401321 0.0247752284 4.020418e-06 0.0207495224 8.155217e-06 1.29710
2e-03
[18,] 0.056438938 0.3971348515 1.074697e-02 0.3611201519 7.772953e-04 2.25420
1e-02
[19,] 1.000000000 0.0272657355 6.589637e-03 0.1060170182 1.322194e-02 2.87737
8e-02
[20,] 0.027265736 1.0000000000 7.341909e-04 0.0454006079 2.579984e-04 5.45903
3e-03
[21,] 0.006589637 0.0007341909 1.000000e+00 0.0088192176 5.170017e-02 1.54194
6e-03
[22,] 0.106017018 0.0454006079 8.819218e-03 1.0000000000 9.902117e-04 8.25317
0e-02
[23,] 0.013221944 0.0002579984 5.170017e-02 0.0009902117 1.000000e+00 2.00282
7e-02
[24,] 0.028773783 0.0054590326 1.541946e-03 0.0825317021 2.002827e-02 1.00000
0e+00
```

```
[25,] 0.024918994 0.0001122361 1.370169e-02 0.0001426331 6.888768e-03 1.05452
8e-05
[26,] 0.042478093 0.0004543634 2.473542e-02 0.0382419462 1.466835e-01 2.06883
8e-01
[27,] 0.031348971 0.0007534446 4.525807e-03 0.0004846220 1.827660e-01 3.35201
4e-03
[28,] 0.042260318 0.0054691047 1.049925e-01 0.0038194243 6.011441e-02 8.63781
5e-04
[29,] 0.444727954 0.0504075205 2.481125e-03 0.0394165017 1.268162e-03 1.56668
4e-03
[30,] 0.215364976 0.0020457800 2.724467e-02 0.1606316336 2.232433e-02 5.06310
1e-02
[31,] 0.058229308 0.0150242713 1.017259e-03 0.0008364091 6.543484e-02 7.39190
9e-03
             [,25]        [,26]        [,27]        [,28]        [,29]
[,30]
 [1,] 7.782999e-03 0.0008703985 0.0001000406 0.0010331432 5.619964e-02 0.0190
41311
 [2,] 9.235249e-04 0.0286253927 0.0025770651 0.0068414398 7.112483e-02 0.0624
19981
 [3,] 8.522288e-04 0.0028694921 0.0310158706 0.0954491486 3.613265e-02 0.0052
54441
 [4,] 1.722166e-04 0.1173331156 0.0041067465 0.0024766985 4.838144e-05 0.0081
37245
 [5,] 4.893983e-02 0.3494784047 0.2439432012 0.2962711722 7.468937e-02 0.4856
56793
 [6,] 2.424992e-03 0.0019967148 0.0004493129 0.0124779172 1.853116e-01 0.0895
91872
 [7,] 1.724966e-01 0.0360169563 0.5571822539 0.5970084062 1.306039e-01 0.0699
71909
 [8,] 1.617025e-04 0.0730042860 0.0918212030 0.0033487262 8.256243e-03 0.0503
37852
 [9,] 4.794432e-06 0.0003830376 0.0001246153 0.0004272065 3.836418e-02 0.0078
36077
[10,] 4.485969e-01 0.0024779266 0.0210524367 0.1426406609 3.050258e-01 0.0281
39891
[11,] 1.091929e-03 0.1029700760 0.0322475414 0.0078053293 7.355667e-02 0.1307
92633
[12,] 1.464102e-03 0.1049255811 0.1826250274 0.1577022811 2.140282e-02 0.0649
44008
[13,] 7.696002e-04 0.2284060373 0.0097366841 0.0176156108 6.227719e-02 0.7301
89423
[14,] 2.949783e-03 0.3413693769 0.0212729986 0.0261009065 8.199823e-02 0.5895
38074
[15,] 3.878040e-03 0.0039742510 0.0035843596 0.0503967274 5.449273e-02 0.0102
10190
[16,] 2.108531e-04 0.3015666868 0.0377248826 0.0075019087 2.133720e-02 0.3072
63652
[17,] 5.202412e-05 0.0003125823 0.0003842375 0.0004382070 1.468434e-01 0.0076
67146
[18,] 2.233009e-04 0.0041625860 0.0004210963 0.0072125926 4.532098e-02 0.0175
66340
[19,] 2.491899e-02 0.0424780934 0.0313489710 0.0422603183 4.447280e-01 0.2153
64976
[20,] 1.122361e-04 0.0004543634 0.0007534446 0.0054691047 5.040752e-02 0.0020
45780
[21,] 1.370169e-02 0.0247354171 0.0045258073 0.1049924930 2.481125e-03 0.0272
44668
[22,] 1.426331e-04 0.0382419462 0.0004846220 0.0038194243 3.941650e-02 0.1606
31634
```

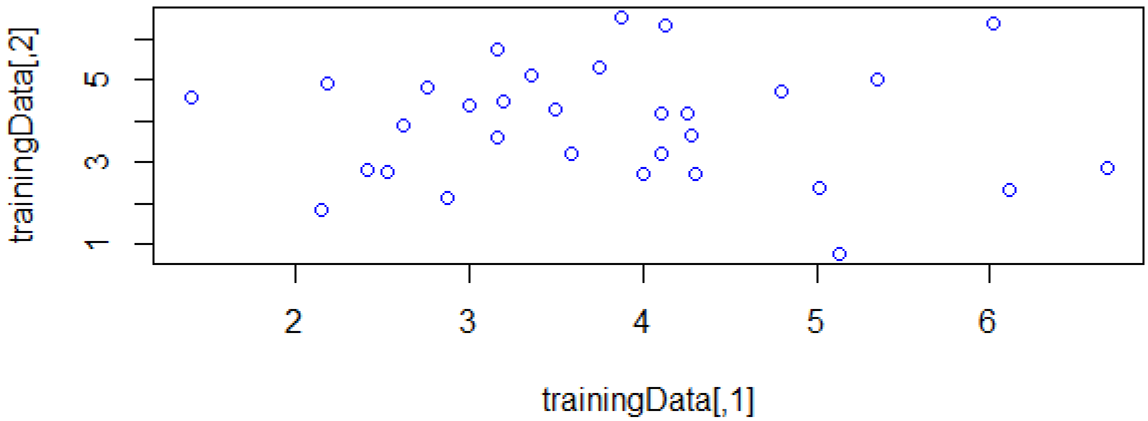
```
[23,] 6.888768e-03 0.1466834949 0.1827660151 0.0601144081 1.268162e-03 0.0223
24335
[24,] 1.054528e-05 0.2068837829 0.0033520138 0.0008637815 1.566684e-03 0.0506
31013
[25,] 1.000000e+00 0.0006539132 0.0260030494 0.1558371332 3.865110e-02 0.0056
47761
[26,] 6.539132e-04 1.0000000000 0.0397457267 0.0168345968 3.589812e-03 0.3672
17684
[27,] 2.600305e-02 0.0397457267 1.0000000000 0.2043847046 1.710240e-02 0.0262
98487
[28,] 1.558371e-01 0.0168345968 0.2043847046 1.0000000000 7.301050e-02 0.0457
28294
[29,] 3.865110e-02 0.0035898120 0.0171024041 0.0730105021 1.000000e+00 0.0651
22717
[30,] 5.647761e-03 0.3672176841 0.0262984866 0.0457282943 6.512272e-02 1.0000
00000
[31,] 7.086020e-03 0.0065263227 0.2499438263 0.0477655735 2.795708e-02 0.0043
39534
             [,31]
 [1,] 0.0002552920
 [2,] 0.0196389524
 [3,] 0.0717803735
 [4,] 0.0019822879
 [5,] 0.0488750987
 [6,] 0.0001432935
 [7,] 0.2391657722
 [8,] 0.0379744161
 [9,] 0.0002256946
[10,] 0.0238476063
[11,] 0.0891574335
[12,] 0.0981004169
[13,] 0.0026725725
[14,] 0.0175966382
[15,] 0.0254482220
[16,] 0.0143045708
[17,] 0.0018381232
[18,] 0.0032060488
[19,] 0.0582293084
[20,] 0.0150242713
[21,] 0.0010172588
[22,] 0.0008364091
[23,] 0.0654348421
[24,] 0.0073919094
[25,] 0.0070860201
[26,] 0.0065263227
[27,] 0.2499438263
[28,] 0.0477655735
[29,] 0.0279570773
[30,] 0.0043395344
[31,] 1.0000000000
$num_SVs
[1] 26
```

No of support vectors shown is 26/32 at respective locations as shown below with its alpha values and support vector values base d on kernelDistance.

```
$locations
 [1]  1  2  3  4  6  8  9 10 11 12 13 15 17 18 20 21 22 23 24 25 26 27 28 29
30 31
```

```
$alpha
 [1]  6.761905e-02  1.947124e-02  4.351868e-02  5.697130e-02  9.795463e-19
 [6]  5.536575e-02 -7.993914e-18  5.854808e-02  5.102040e-02  2.188574e-02
[11]  2.030230e-02  2.296042e-02  7.616995e-03  3.753987e-18  3.266129e-02
[16]  0.000000e+00  4.902980e-02  1.364846e-02 -3.007983e-20  4.149972e-02
[21]  6.250726e-02  3.456631e-02  3.513547e-02  5.080233e-02  5.739913e-02
[26]  2.680857e-02  3.437704e-02  3.123471e-02  2.618228e-02  2.866640e-02
[31]  5.020126e-02

$nza
 [1] 0.067619047 0.019471241 0.043518678 0.056971303 0.055365753 0.058548082
 [7] 0.051020405 0.021885739 0.020302300 0.022960425 0.007616995 0.032661291
[13] 0.049029799 0.013648461 0.041499719 0.062507258 0.034566311 0.035135467
[19] 0.050802331 0.057399129 0.026808567 0.034377039 0.031234714 0.026182280
[25] 0.028666403 0.050201263

$sv
            [,1]      [,2]     [,3]     [,4]     [,5]
 [1,] 0.09023472 2.3894565 1.691397 6.950158 3.921643
 [2,] 1.37906241 3.2362167 4.059906 4.922655 3.003185
 [3,] 4.08807071 2.4537805 6.041264 4.198302 5.521934
 [4,] 1.04372676 7.5862766 4.719663 4.811571 4.818272
 [5,] 1.45614807 0.9926762 4.961180 8.064773 5.027954
 [6,] 5.19812290 4.9735594 4.536947 7.360258 3.145646
 [7,] 3.55898083 0.4370824 6.291612 7.154523 2.846151
 [8,] 2.09500785 2.6548284 2.154129 6.068758 6.179375
 [9,] 3.20655238 4.1304952 3.771711 6.053447 2.700670
[10,] 3.86118205 4.2695774 6.151408 5.337687 5.194869
[11,] 2.09940684 3.5855494 5.583740 7.713858 3.726197
[12,] 1.46388190 2.6015290 4.626006 3.670441 5.050181
[13,] 3.90567830 0.6485456 4.132889 7.016784 2.408257
[14,] 1.03443603 1.9694635 5.460826 4.426547 3.771847
[15,] 2.75039135 1.2640273 5.278501 3.602526 3.807367
[16,] 0.23267060 4.8180714 5.068749 5.351637 7.032550
[17,] 0.76200613 2.7240181 5.627410 6.288271 3.103929
[18,] 2.91152657 6.7297211 4.176564 5.119391 5.748137
[19,] 2.32361332 5.3841989 5.669042 5.421390 2.125571
[20,] 2.47322090 3.4638244 1.807154 6.480799 7.766715
[21,] 2.23454300 6.0199686 5.416464 7.010586 4.147629
[22,] 4.89423005 5.1064033 4.030712 6.085554 6.104440
[23,] 3.07996149 3.6295221 4.517652 5.858962 7.280738
[24,] 2.98040495 1.5677684 3.372086 6.520638 4.879424
[25,] 1.90833417 4.1937057 4.884421 7.847907 4.434470
[26,] 4.87492043 4.2686959 3.307956 4.238787 4.837835

$ctr
         [,1]     [,2]     [,3]     [,4]     [,5]
[1,] 2.525036 3.525415 4.410406 5.943771 4.625282
```

```
> plot(trainingData,col="blue")
```



Now similarly,we are going to run svddTrain to to detect all support vectors and outliers for for entire dataset to check anomalies which are not class =1, this is nothing but out testing data and apply to it. This svddTrain returns locations,alpha values and support vectors. We also see its output below with the plot of its data points.

```
> svddTrain(testingData, gm, sig, 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 1 1 1
1 1 1 1
[41] 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]        [,2]         [,3]         [,4]         [,5]
[,6]
 [1,] 1.000000e+00 0.080682095 4.092260e-03 0.0316790291 2.546673e-04 0.41961
85834
 [2,] 8.068209e-02 1.000000000 1.622451e-03 0.5842273645 8.569092e-03 0.44682
72954
 [3,] 4.092260e-03 0.001622451 1.000000e+00 0.0008143734 4.902086e-02 0.00405
19901
 [4,] 3.167903e-02 0.584227364 8.143734e-04 1.0000000000 2.254226e-02 0.15368
96774
 [5,] 2.546673e-04 0.008569092 4.902086e-02 0.0225422640 1.000000e+00 0.00140
43553
 [6,] 4.196186e-01 0.446827295 4.051990e-03 0.1536896774 1.404355e-03 1.00000
00000
 [7,] 1.607969e-01 0.078615411 8.105450e-02 0.0308074904 8.906221e-03 0.33069
84792
 [8,] 3.290242e-02 0.060176001 1.509466e-01 0.0477055175 6.004669e-02 0.10059
23563
 [9,] 2.397473e-03 0.012202228 1.115957e-02 0.0035645846 7.741130e-03 0.01747
54297
[10,] 2.308118e-02 0.174549073 1.609956e-02 0.0484002987 7.574113e-03 0.23979
10685
[11,] 7.356333e-02 0.284296787 3.077421e-03 0.2416240720 8.301903e-03 0.38351
64056
[12,] 7.691849e-03 0.079540640 5.207161e-03 0.0355163050 1.292176e-02 0.07675
10322
[13,] 4.250620e-03 0.057807130 2.543937e-04 0.0066428312 1.933628e-04 0.01663
57188
[14,] 3.126414e-02 0.082466845 8.671618e-04 0.0350357481 1.333271e-03 0.12649
96245
```

```
[15,] 1.998874e-03 0.159113975 1.483786e-03 0.2087400757 3.664697e-02 0.04756
05155
[16,] 5.655104e-02 0.123019474 8.111455e-04 0.1300475915 4.525708e-03 0.04867
08000
              [,7]         [,8]         [,9]        [,10]        [,11]
[,12]
 [1,] 1.607969e-01 3.290242e-02 2.397473e-03 0.0230811831 0.0735633300 7.6918
49e-03
 [2,] 7.861541e-02 6.017600e-02 1.220223e-02 0.1745490729 0.2842967865 7.9540
64e-02
 [3,] 8.105450e-02 1.509466e-01 1.115957e-02 0.0160995554 0.0030774209 5.2071
61e-03
 [4,] 3.080749e-02 4.770552e-02 3.564585e-03 0.0484002987 0.2416240720 3.5516
30e-02
 [5,] 8.906221e-03 6.004669e-02 7.741130e-03 0.0075741129 0.0083019030 1.2921
76e-02
 [6,] 3.306985e-01 1.005924e-01 1.747543e-02 0.2397910685 0.3835164056 7.6751
03e-02
 [7,] 1.000000e+00 3.397384e-01 1.056451e-01 0.2615769389 0.3163062658 1.7129
90e-01
 [8,] 3.397384e-01 1.000000e+00 1.067746e-02 0.3262367371 0.1104056821 2.7891
11e-02
 [9,] 1.056451e-01 1.067746e-02 1.000000e+00 0.0240356186 0.0653780897 6.3644
46e-01
[10,] 2.615769e-01 3.262367e-01 2.403562e-02 1.0000000000 0.1644297928 7.8709
52e-02
[11,] 3.163063e-01 1.104057e-01 6.537809e-02 0.1644297928 1.0000000000 3.1231
49e-01
[12,] 1.712990e-01 2.789111e-02 6.364446e-01 0.0787095212 0.3123149205 1.0000
00e+00
[13,] 1.290943e-03 1.014429e-03 3.627733e-04 0.0088748846 0.0008546747 9.5013
91e-04
[14,] 1.133356e-01 6.283434e-03 2.668443e-01 0.0196932469 0.2982036371 5.2239
71e-01
[15,] 3.751869e-02 1.363452e-01 6.275514e-03 0.2226694937 0.1911870691 5.2562
45e-02
[16,] 1.040403e-02 2.924884e-03 4.722108e-03 0.0018946954 0.0301844313 1.4982
79e-02
             [,13]        [,14]        [,15]        [,16]        [,17]
[,18]
 [1,] 4.250620e-03 3.126414e-02 1.998874e-03 5.655104e-02 1.432731e-04 1.6235
30e-02
 [2,] 5.780713e-02 8.246684e-02 1.591140e-01 1.230195e-01 1.478035e-02 9.3158
97e-02
 [3,] 2.543937e-04 8.671618e-04 1.483786e-03 8.111455e-04 6.030471e-04 3.1273
04e-06
 [4,] 6.642831e-03 3.503575e-02 2.087401e-01 1.300476e-01 1.572611e-02 2.3253
95e-01
 [5,] 1.933628e-04 1.333271e-03 3.664697e-02 4.525708e-03 7.478405e-03 1.1988
63e-04
 [6,] 1.663572e-02 1.264996e-01 4.756052e-02 4.867080e-02 1.935651e-03 2.6745
68e-02
 [7,] 1.290943e-03 1.133356e-01 3.751869e-02 1.040403e-02 6.684773e-04 7.4659
58e-04
 [8,] 1.014429e-03 6.283434e-03 1.363452e-01 2.924884e-03 2.974665e-02 8.7745
83e-04
 [9,] 3.627733e-04 2.668443e-01 6.275514e-03 4.722108e-03 9.054582e-06 1.1742
08e-05
[10,] 8.874885e-03 1.969325e-02 2.226695e-01 1.894695e-03 2.526823e-02 8.2077
71e-04
```

```
[11,] 8.546747e-04 2.982036e-01 1.911871e-01 3.018443e-02 9.052750e-04 1.5896
47e-02
[12,] 9.501391e-04 5.223971e-01 5.256245e-02 1.498279e-02 9.345623e-05 3.3289
49e-04
[13,] 1.000000e+00 9.057124e-04 6.805995e-04 1.318962e-02 2.614547e-03 8.2703
95e-04
[14,] 9.057124e-04 1.000000e+00 9.504935e-03 5.382390e-02 8.148539e-06 1.5990
56e-03
[15,] 6.805995e-04 9.504935e-03 1.000000e+00 2.010636e-03 5.092675e-02 4.8090
67e-03
[16,] 1.318962e-02 5.382390e-02 2.010636e-03 1.000000e+00 1.124412e-04 5.3247
58e-02
             [,19]        [,20]        [,21]        [,22]        [,23]
[,24]
 [1,] 0.0746894161 0.0063363464 0.2483127823 3.180175e-03 1.300854e-04 5.5045
68e-03
 [2,] 0.7215678658 0.2128956113 0.1392515702 5.843639e-02 1.174144e-03 3.3210
72e-02
 [3,] 0.0062602180 0.0184274820 0.0102175187 3.799282e-02 9.449994e-03 3.8937
16e-03
 [4,] 0.3258456864 0.0837403204 0.0305547891 3.467442e-02 6.043487e-04 1.6494
38e-02
 [5,] 0.0119096749 0.0434236952 0.0008619446 1.278254e-01 2.327564e-02 4.5793
07e-03
 [6,] 0.4406085304 0.0994486015 0.2573515788 1.060570e-02 4.035314e-04 6.9443
88e-02
 [7,] 0.1267881374 0.1055619328 0.0679890217 1.037719e-02 3.058958e-03 2.5424
51e-01
 [8,] 0.1823628741 0.2444795083 0.0607135139 3.301353e-02 6.753099e-04 6.1963
73e-02
 [9,] 0.0088892420 0.0279482531 0.0010586018 5.783711e-03 1.144191e-01 2.2297
71e-01
[10,] 0.4492090760 0.6506360026 0.0862271163 2.056102e-02 3.968713e-04 1.4477
41e-01
[11,] 0.1969252180 0.1007637708 0.0180236759 5.017436e-03 1.242459e-03 3.8372
86e-01
[12,] 0.0492854840 0.0875626907 0.0032143847 8.792120e-03 3.589159e-02 4.9201
94e-01
[13,] 0.0667363685 0.0194001023 0.1209784228 7.636809e-02 3.779698e-04 7.3480
27e-05
[14,] 0.0282662610 0.0140998519 0.0041103950 2.306502e-03 1.315321e-02 1.7607
05e-01
[15,] 0.2201115987 0.3510535796 0.0048780927 1.115124e-02 1.859488e-04 9.3109
83e-02
[16,] 0.0339260597 0.0038200268 0.0242304675 3.971065e-02 9.455794e-03 1.1485
51e-03
             [,25]        [,26]       [,27]        [,28]        [,29]
[,30]
 [1,] 4.106368e-05 0.0895240536 0.008226585 1.470751e-03 0.0044960068 5.26556
8e-05
 [2,] 5.632939e-03 0.2317308144 0.490897793 1.915144e-02 0.1869016786 4.49806
1e-02
 [3,] 1.714968e-02 0.0252306002 0.002219466 2.651381e-02 0.0011781219 6.06372
4e-07
 [4,] 3.093100e-03 0.0435171880 0.520123698 3.970178e-03 0.0363154639 7.80541
0e-03
 [5,] 8.114653e-02 0.0053128142 0.073600831 9.306136e-03 0.0042514171 2.71949
2e-05
 [6,] 8.694404e-04 0.4362876209 0.085797098 7.904109e-03 0.0537890448 3.60049
4e-03
```

```
 [7,] 2.622437e-03 0.4500170097 0.034634450 7.411252e-03 0.0180305418 1.08620
8e-04
 [8,] 4.512590e-02 0.1384536943 0.040651336 4.570341e-02 0.0052677727 8.39757
0e-05
 [9,] 9.490635e-04 0.1350093905 0.032167685 6.568091e-04 0.0699913266 1.91582
1e-04
[10,] 4.177897e-02 0.4887442635 0.075470216 6.821446e-02 0.0514657537 6.04111
8e-03
[11,] 9.593564e-04 0.2113217541 0.195372083 1.080220e-03 0.0320754957 1.45390
0e-03
[12,] 1.782117e-03 0.2378954698 0.155957039 1.093181e-03 0.1322430290 1.82772
1e-03
[13,] 2.610514e-03 0.0305732263 0.015880642 8.368672e-02 0.1376998134 4.01149
5e-02
[14,] 6.094295e-05 0.1413489964 0.077319231 1.411050e-04 0.0788665269 9.27604
5e-04
[15,] 3.378440e-02 0.0497323247 0.245410618 6.857194e-03 0.0158216742 5.79237
1e-03
[16,] 9.754470e-05 0.0166827680 0.103858183 8.037518e-04 0.0381494952 5.78798
3e-04
             [,31]        [,32]        [,33]        [,34]        [,35]        
[,36]
 [1,] 1.007871e-01 0.0095111237 7.657390e-02 5.540731e-03 8.318379e-04 1.3497
39e-01
 [2,] 5.296528e-01 0.1782878312 1.339063e-02 2.217400e-01 9.382231e-02 2.0055
25e-02
 [3,] 2.707753e-04 0.0086715804 1.218663e-01 3.349724e-04 2.654729e-04 7.7727
09e-03
 [4,] 1.606704e-01 0.0290180278 1.522886e-03 7.026083e-01 2.415042e-01 1.2955
53e-02
 [5,] 3.354485e-04 0.0084738245 1.000334e-03 4.143305e-02 6.230301e-02 4.6421
33e-04
 [6,] 4.198766e-01 0.0916710804 9.161400e-02 2.877040e-02 7.505568e-03 1.0179
70e-01
 [7,] 2.925148e-02 0.0610665411 3.021324e-01 7.147794e-03 3.262691e-03 7.9938
05e-02
 [8,] 2.122833e-02 0.0258210021 8.867669e-02 1.352646e-02 3.617267e-03 1.8685
42e-01
 [9,] 9.096002e-04 0.1014032287 2.593561e-02 2.189292e-03 9.611656e-03 9.8237
76e-05
[10,] 1.178980e-01 0.1675704393 1.045938e-01 8.142895e-03 3.282108e-03 5.3966
49e-02
[11,] 7.107583e-02 0.0427547730 1.551793e-02 9.738000e-02 4.671465e-02 2.0686
97e-02
[12,] 8.264850e-03 0.1493793123 1.811244e-02 2.094227e-02 4.905497e-02 5.5552
47e-04
[13,] 8.851766e-02 0.1414448225 6.241638e-03 1.542079e-03 1.131980e-03 4.0314
22e-04
[14,] 1.300506e-02 0.0596839645 1.065114e-02 1.867160e-02 3.099613e-02 4.9382
65e-04
[15,] 3.369013e-02 0.0261624501 1.541315e-03 1.034533e-01 4.659392e-02 7.7749
38e-03
[16,] 2.617008e-02 0.0199205904 1.974636e-03 1.329853e-01 1.028401e-01 9.4999
86e-04
             [,37]        [,38]        [,39]       [,40]        [,41]        
[,42]
 [1,] 2.919132e-05 5.235494e-04 3.253216e-04 0.088434888 0.0286483621 4.38119
4e-04
 [2,] 2.576779e-03 2.080926e-02 3.127436e-02 0.539027386 0.1967413030 3.06608
9e-02
```

```
 [3,] 2.060320e-04 1.481783e-02 6.955852e-04 0.033005146 0.1028132135 1.86055
5e-03
 [4,] 1.318737e-04 6.022231e-02 1.216981e-02 0.378781843 0.1888293765 5.87114
5e-02
 [5,] 5.779197e-05 3.413247e-01 9.555835e-03 0.067266704 0.2888786266 1.38302
1e-01
 [6,] 2.066537e-03 5.863371e-03 8.738569e-03 0.297786028 0.1112810209 4.58850
0e-03
 [7,] 2.249115e-03 2.506840e-02 1.406243e-02 0.172413252 0.2486244161 9.78093
9e-03
 [8,] 2.463245e-03 1.972263e-01 2.835901e-03 0.318387638 0.3255342132 6.68225
8e-03
 [9,] 1.858439e-03 5.185220e-03 3.490582e-01 0.015019742 0.1019378698 8.80489
7e-02
[10,] 8.632670e-02 4.488721e-02 1.795629e-02 0.275025225 0.1583775820 4.86563
5e-03
[11,] 7.538912e-04 5.007483e-02 4.677825e-02 0.181353292 0.2211228654 5.38365
3e-02
[12,] 3.142746e-03 2.129909e-02 4.897282e-01 0.057490845 0.2063281851 1.84822
4e-01
[13,] 2.894998e-03 5.429654e-05 1.289413e-03 0.045252812 0.0059255874 2.55754
0e-04
[14,] 2.215586e-04 2.080598e-03 1.542650e-01 0.029786692 0.0702290167 6.43324
5e-02
[15,] 4.473718e-03 3.252822e-01 2.158114e-02 0.177890736 0.1595332959 3.67368
2e-02
[16,] 2.865832e-06 1.136744e-03 5.949764e-03 0.083535184 0.0625572662 3.13609
3e-02
             [,43]        [,44]        [,45]        [,46]        [,47]
[,48]
 [1,] 6.974247e-04 2.220697e-01 5.392337e-05 1.290946e-02 0.0090740504 3.0912
96e-01
 [2,] 1.648904e-01 1.212592e-01 1.593986e-02 1.930887e-02 0.2180448766 2.5781
43e-02
 [3,] 6.677560e-06 2.569189e-04 1.335606e-07 1.300300e-04 0.0047943307 5.7086
61e-04
 [4,] 5.409358e-02 2.063219e-02 3.600052e-02 4.648998e-03 0.1103775231 3.7025
94e-02
 [5,] 1.774203e-04 4.581684e-05 3.549057e-05 1.231934e-04 0.0341770435 2.2825
07e-04
 [6,] 1.827275e-02 1.931265e-01 2.416237e-03 1.351838e-02 0.0783490460 1.2103
80e-01
 [7,] 5.506144e-04 1.481338e-02 1.177111e-04 2.992643e-03 0.0699502021 6.6220
11e-02
 [8,] 1.287735e-03 2.019741e-03 7.327266e-04 1.223074e-04 0.0204518918 1.2457
92e-02
 [9,] 8.386318e-05 1.137221e-03 2.304216e-06 1.127732e-02 0.2863192841 1.0679
86e-03
[10,] 2.344995e-02 8.457515e-03 3.620064e-03 4.022057e-04 0.0576522685 3.2081
10e-03
[11,] 4.854607e-03 1.284673e-02 7.049524e-03 4.162590e-03 0.1739506149 1.1325
14e-01
[12,] 1.228549e-03 3.787143e-03 1.547154e-04 1.339650e-02 0.5429277484 5.7939
58e-03
[13,] 8.066188e-02 1.250651e-01 1.554934e-05 1.711969e-02 0.0133577288 4.6169
64e-05
[14,] 6.829195e-04 1.986870e-02 6.974822e-05 7.092708e-02 0.3244613738 3.2091
46e-02
[15,] 2.731158e-02 5.155821e-04 7.786203e-02 6.158259e-05 0.0547307924 1.9566
40e-03
```

```
[16,] 2.295732e-03 1.048668e-01 3.864400e-05 2.289588e-01 0.0937304211 3.5546
86e-02
             [,49]       [,50]        [,51]        [,52]        [,53]
[,54]
 [1,] 0.0122913536 0.045591656 1.129955e-02 9.500513e-04 0.0007269035 1.06836
7e-01
 [2,] 0.1367197791 0.224061757 1.696318e-01 4.011412e-02 0.0819034205 6.85068
1e-02
 [3,] 0.0329237882 0.129133565 4.606127e-03 3.275075e-03 0.0162798248 1.00120
1e-01
 [4,] 0.0689795205 0.143500611 4.025959e-01 1.902277e-02 0.0511474068 4.70238
4e-02
 [5,] 0.0587452149 0.126472995 1.124442e-01 2.239922e-02 0.1633392149 2.50094
2e-02
 [6,] 0.1239050593 0.173396802 6.499041e-02 1.925844e-02 0.0173837435 2.17493
0e-01
 [7,] 0.3444181711 0.279318790 7.922390e-02 5.452744e-02 0.0338699863 8.53881
1e-01
 [8,] 0.1982000123 0.639601799 9.159075e-02 1.423203e-02 0.1089642443 4.60736
0e-01
 [9,] 0.3327042193 0.027680160 3.265911e-02 5.423101e-01 0.0383256960 8.36572
1e-02
[10,] 0.3328835812 0.395119492 4.363689e-02 4.890531e-02 0.1978965944 1.91739
8e-01
[11,] 0.3557554946 0.141806476 4.076807e-01 1.204858e-01 0.0445383301 3.59388
0e-01
[12,] 0.6129999703 0.068882999 1.609803e-01 7.633751e-01 0.0920086859 1.55980
6e-01
[13,] 0.0022910438 0.015323307 3.859144e-04 6.691662e-04 0.0057475881 4.84791
7e-04
[14,] 0.1658385391 0.019334174 8.543720e-02 2.069808e-01 0.0086785112 8.59745
9e-02
[15,] 0.2083036528 0.183768478 2.618400e-01 5.377243e-02 0.2818917566 6.11467
3e-02
[16,] 0.0100070236 0.022200752 5.064264e-02 5.398135e-03 0.0027060745 1.13912
1e-02
             [,55]        [,56]        [,57]        [,58]        [,59]
[,60]
 [1,] 7.296462e-02 0.0235695524 1.562100e-01 1.360612e-04 3.091503e-03 1.3903
34e-04
 [2,] 1.593510e-01 0.2808102421 4.872739e-02 5.074080e-02 3.154638e-02 1.7875
10e-02
 [3,] 1.563150e-05 0.0007682442 2.327323e-01 5.408391e-07 2.028871e-05 3.4279
85e-04
 [4,] 4.544448e-02 0.0396403597 1.263125e-02 8.822302e-03 5.215997e-03 1.0522
05e-02
 [5,] 1.286088e-05 0.0005092094 9.211019e-03 1.720158e-05 4.057495e-05 1.3642
44e-02
 [6,] 3.626421e-01 0.1669277120 1.390550e-01 3.974706e-03 3.420806e-02 2.2827
08e-03
 [7,] 2.727543e-02 0.0193846156 2.768527e-01 6.956028e-05 1.097477e-02 2.6402
91e-03
 [8,] 4.310290e-03 0.0158405984 2.172972e-01 4.442673e-05 2.789935e-04 6.0847
83e-04
 [9,] 1.022850e-03 0.0021131289 1.371561e-02 8.316940e-05 9.033642e-02 1.2730
82e-01
[10,] 4.388277e-02 0.1594400871 1.116862e-01 2.325803e-03 6.123281e-03 2.2388
39e-03
[11,] 1.363483e-01 0.0191207410 2.608172e-02 7.852289e-04 4.949582e-02 1.1632
39e-02
```

```
[12,] 1.183485e-02 0.0091714994 1.524910e-02 7.627175e-04 1.920417e-01 1.6089
78e-01
[13,] 2.114660e-03 0.3828900073 2.071170e-02 1.217279e-01 1.411613e-03 1.7001
48e-03
[14,] 3.784216e-02 0.0066522122 9.479366e-03 7.463334e-04 4.135657e-01 6.9615
71e-02
[15,] 1.298821e-02 0.0181096514 5.416899e-03 1.613944e-03 1.670717e-03 4.8145
73e-03
[16,] 4.727343e-03 0.0097394871 1.608459e-02 2.373413e-03 8.098807e-03 2.0001
90e-02
             [,61]        [,62]
 [1,] 0.0005958152 1.580492e-02
 [2,] 0.0376922091 1.589088e-02
 [3,] 0.0337916877 6.632075e-03
 [4,] 0.0366617646 5.692081e-03
 [5,] 0.2975788563 1.036858e-03
 [6,] 0.0112021088 9.523570e-02
 [7,] 0.0495629148 4.374123e-01
 [8,] 0.1837058868 9.116924e-02
 [9,] 0.0434980411 8.625374e-02
[10,] 0.1464772939 1.548522e-01
[11,] 0.0540366507 2.557428e-01
[12,] 0.0934805091 1.694773e-01
[13,] 0.0007154473 3.773380e-05
[14,] 0.0074040573 7.858185e-02
[15,] 0.2952552948 3.227781e-02
[16,] 0.0014669888 4.181877e-04
 [ reached getOption("max.print") -- omitted 46 rows ]
$num_SVs
[1] 44
```

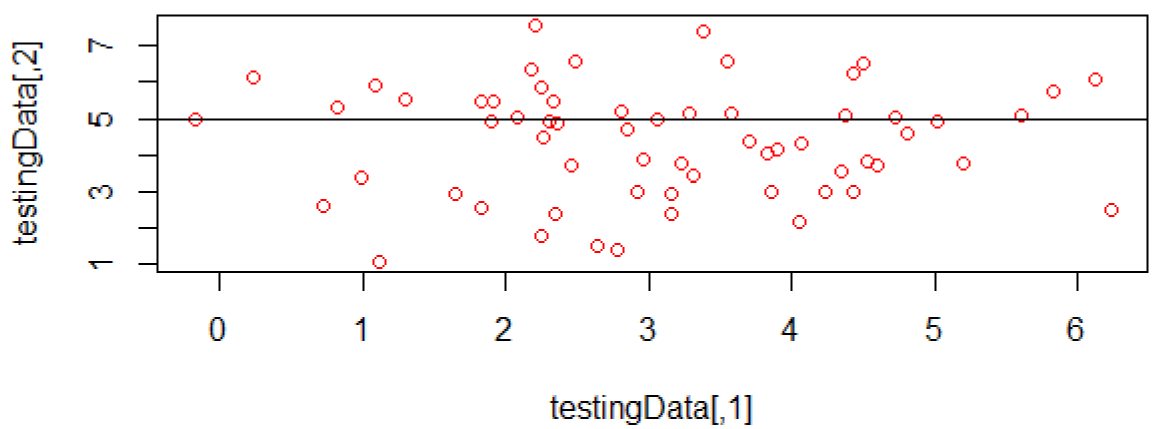
The number of support vectors shown is 44/62 whereas for our tra
ining data it was 26/32 which shows the anomalies are all the da
ta points except the first 26 support vector points from our tra
ining data. Hence we will plot the testing data and separate tho
se in the plot to show all the outliers.

```
$locations
 [1]  1  3  5  8  9 10 11 13 14 15 16 17 18 21 22 23 24 25 28 29 30 31 32 33
34 35
[27] 36 37 38 39 42 43 44 45 46 48 51 53 54 55 58 59 60 62

$alpha
 [1]  1.260001e-02  7.928980e-18  3.937868e-02  3.303867e-19  2.725124e-02
 [6]  4.629264e-18  2.392634e-17  8.525005e-03  2.287001e-02  1.091905e-02
[11]  9.543937e-04 -2.405955e-17  3.226877e-02  9.438790e-03  2.027697e-02
[16]  2.582101e-02  4.006145e-02  4.184614e-02  5.002803e-18  2.766776e-18
[21]  1.553552e-02  2.370886e-02  3.917276e-02  1.175315e-02  2.853808e-02
[26] -8.896782e-19  6.450108e-19  2.378126e-02  8.925478e-03  2.437416e-02
[31]  6.384823e-03  5.987622e-03  3.335825e-02  1.741404e-02  1.909865e-02
[36]  3.938422e-02  4.695585e-02  2.215575e-02  1.067065e-02 -7.598505e-19
[41]  2.633815e-18  1.163496e-02  2.206666e-02  2.460702e-02  4.672418e-02
[46]  3.506823e-02  3.382477e-18  3.908869e-02  1.344536e-17 -4.845969e-18
[51]  5.666360e-03 -5.370913e-19  3.164602e-03  2.366259e-03  3.811470e-02
[56]  0.000000e+00 -1.269227e-19  1.751086e-02  3.520231e-02  1.865363e-02
[61]  8.228452e-20  3.072087e-02

$nza
 [1] 0.0126000121 0.0393786841 0.0272512363 0.0085250047 0.0228700101 0.01091
90533
```

```
 [7] 0.0009543937 0.0322687721 0.0094387900 0.0202769712 0.0258210105 0.04006
14516
[13] 0.0418461406 0.0155355164 0.0237088599 0.0391727639 0.0117531511 0.02853
80785
[19] 0.0237812625 0.0089254779 0.0243741640 0.0063848230 0.0059876216 0.03335
82487
[25] 0.0174140446 0.0190986518 0.0393842206 0.0469558533 0.0221557506 0.01067
06535
[31] 0.0116349626 0.0220666597 0.0246070241 0.0467241752 0.0350682306 0.03908
86898
[37] 0.0056663604 0.0031646016 0.0023662588 0.0381146954 0.0175108649 0.03520
23061
[43] 0.0186536317 0.0307208669

$sv
            [,1]     [,2]       [,3]     [,4]     [,5]
 [1,]  5.5977334 5.094358  5.2633796 2.513084 5.016300
 [2,]  2.6457069 1.534220  6.7260307 2.810419 3.954622
 [3,]  0.7286382 2.610299  4.1238814 2.674235 2.559285
 [4,]  4.0580673 2.149746  4.4068183 3.458399 3.374560
 [5,]  2.2989033 4.931869  6.6891653 5.401523 2.594440
 [6,]  4.2309416 2.992986  4.0164414 5.376912 4.057617
 [7,]  4.3702725 5.066545  4.1463761 4.178995 2.523742
 [8,]  2.3584635 4.841319  3.8018753 4.602155 7.757832
 [9,]  3.5505210 6.551883  5.6362775 4.649769 2.877646
[10,]  3.3132569 3.430487  2.3200121 4.740112 2.494812
[11,]  2.4853288 6.588919  4.4606731 1.976617 4.744562
[12,]  2.7834024 1.431723  1.1413700 3.736567 5.088274
[13,]  4.4952705 6.494561  1.4605743 1.773990 4.417158
[14,]  4.5346656 3.804138  4.4251649 3.312775 6.476406
[15,]  0.9858909 3.382568  4.2383970 3.180430 5.444588
[16,] -0.1758969 4.972045  7.0224179 3.770126 3.520038
[17,]  4.0692150 4.306756  5.1483638 5.570418 1.685746
[18,]  1.1190452 1.082353  3.3339879 4.808951 4.465527
[19,]  2.2449960 1.784582  4.0040054 4.390092 6.446080
[20,]  1.8300092 5.480035  4.4633098 5.566171 5.137143
[21,]  2.2555975 5.867759  1.6915839 6.888984 5.827820
[22,]  4.7245437 5.036609  2.8394372 4.309368 5.693370
[23,]  2.2616143 4.477295  4.8412990 5.552059 5.173903
[24,]  4.4275420 3.002772  6.7839421 4.369660 5.260596
[25,]  2.3352804 5.489914  2.3283723 2.565376 3.025506
[26,]  1.0841955 5.885880  2.9817378 3.221792 2.734472
[27,]  6.2314498 2.488920  3.9805455 2.447046 4.601445
[28,]  3.1639862 2.382196  4.2154337 8.219323 5.237836
[29,]  2.3411871 2.394067  3.1488101 3.413932 1.727325
[30,]  1.3031890 5.544871  4.9710386 6.024822 2.763682
[31,]  0.8220926 5.298072  4.3167483 3.796896 1.949370
[32,]  3.0549208 4.952992  1.1812313 5.456960 5.959858
[33,]  4.4257392 6.224635  4.7275833 3.507785 6.702668
[34,]  4.8091230 4.571131 -0.4897508 4.747629 2.727548
[35,]  2.2032633 7.564829  6.0541782 3.489383 5.624492
[36,]  5.8276872 5.755722  4.8603164 1.645434 3.047184
[37,]  2.8440467 4.721104  3.5521443 3.194873 2.006617
[38,]  1.6482637 2.940950  3.6009948 5.158391 3.427722
[39,]  4.3520335 3.530744  5.4750707 3.537293 2.786150
[40,]  6.1269080 6.103999  3.2773348 4.985897 4.367119
[41,]  2.1859400 6.348908  1.8464495 6.052829 6.641536
[42,]  3.3768471 7.400949  5.3825904 6.242989 3.664693
[43,]  0.2316643 6.119836  4.9121767 5.076668 3.166442
[44,]  5.1943439 3.792195  5.6753659 5.302207 2.040370
```

```
$ctr
         [,1]    [,2]     [,3]     [,4]     [,5]
[1,] 3.178484 4.39783 3.980788 4.204238 4.286701
```

```
> plot(testingData,col="red")
> abline(h=5)
```



The points above the black horizontal line are detected as outliers by average KernelDistance and training Dataset. The above plot is tested on entire dataset and 6 points of even class =1 are detected as outliers which needs further rimprovements.