WESTERN MICHIGAN UNIVERSITY

STAT 6850

Applied Data Mining

Fall 2014

Instructor: Dr. J.C. Wang

Case Study #4

Using Neural Network and Support Vector Machine to classify Coronary Heart Disease in South Africans Males

Antonio Giraldi

Shoruk Mansour

Joan Martinez

Milton Soto Ferrari

Saurabh Rajratn Kulkarni

Mustafa Yildiz

Description Summary

In this case study we'll study a dataset of males in a heart-disease high-risk region of the Western Cape, South Africa with the objective of developing models using Neural Network and Support Vector Machine techniques to predict the presence of coronary heart disease within this population. Our working dataset is a taken from a larger dataset described in Rousseauw et al, 1983, South African Medical Journal and consist of 462 observations, with 9 attributes and a response variable. Descriptions of males' attributes are as follow:

1. Systolic Blood Pressure
2. Cumulative Tobacco Consumption (kg)
3. Low Density Lipoprotein cholesterol
4. Adiposity
5. Family History of Heart Disease (Present, Absent)
6. Type-A Behavior
7. Obesity
8. Current Alcohol Consumption
9. Age
10. Coronary Heart Disease (Response Variable: Yes, No)

The dataset will be divide into training & test data sets doing an approximately 40%:60% split of chd (Coronary Heart Disease). We also wish to maintain a similar structure in these sets for the attribute famhist (Family History), our goal is to have similar distributions in terms of chd and famhist in both the train set and the test set. Machine learning algorithms will be performed on train sets to then be used to predict responses on test set, accuracy of methods will be discussed.

Exploratory Analysis

First we wanted just to see how the data is presented, we can see the most of the attributes are numerical variables (a mix of discrete and continuous) with famhist being the exception for being categorical; the response variable (chd) shows whether the subject has coronary heart disease or not, and is also categorical, which gives this study a classification task nature. As an example, a male subject, aged 52, with family history of heart disease and respective measurements of each attribute (see table below), shows a positive response of having coronary heart disease. These attributes are complete for each of our 462 subjects.

sbp	tobacco	ldl	adiposity	famhist	typea	obesity	alcohol	age	chd
160	12	5.73	23.11	Present	49	25.3	97.2	52	yes

By creating a frequency bar plot for the response type in our data (Fig 1); we can see that we have almost twice the number of cases without coronary heart disease. This proportion was considered during data split into their respective sets.

Fig 2 shows a mosaic plot of heart disease versus family history. This shows the disproportion of subjects with absent heart disease family history, as explained before this was also taken into consideration when splitting the data.

Fig 3 shows box plots for each attribute by response to look at the variability within each group. Variability of attributes is somewhat similar on each response group and in some cases it looks like there's no significant difference in groups by given attributes (ex. Type-A, Alcohol, Obesity). Other attributes show different behavior when coronary heart disease is present (ex. Age, Adiposity, LDL) with a higher response for this group category.

On these attributes we wanted to determine if we have highly correlated variables. A correlation matrix for all attributes is presented in Table1; by doing an ordered for this correlation matrix we obtain a more useful graphical representation as shown in Fig 4. We see a positive inter-correlated cluster which includes age, tobacco, sbp, obesity, adiposity and ldl. This cluster is highlighted by a high positive correlation between adiposity and obesity/age. Type-A and alcohol show weak correlation with other attributes. We couldn't find strong multicollinearity (using a pair-wise absolute correlation cutoff of 0.9) and proceed the analysis with all attributes.

Data Pre-Process and Variable Importance

In our prediction model we want to use only significant and value-added variables. The first step is to identify near zero variance predictors so we can eliminate these from our input variables. Table 2 shows the results of this analysis; and, as expected, none of the attributes show variability lower than our frequency cut ratio (95/5).

The second step is to analyze variable importance by using several machine learning methods and compare their variable importance rating. Our idea is to look for some agreement between these methods on which variables are significantly important when included in their respective predictor models. As some of these methods require our dataset to be scaled and centered we proceeded to do a pre-processing transformation using our training set as estimator and applying it in both training and test set.

We used the following machine learning models to analyze variable importance:

- Random Forest
- Partial Least Squares (PLS)
- Generalized boosting model (GBM)
- Multivariate Adaptive Regression Splines (MARS)
- Model Free ROC curve analysis (ROC)

Summary of the variable importance for each method is presented in Table 3. Fig 5 shows paired-scatterplot with a linear fit for variable importance in all methods. From here we see

some positive correlation between the methods. Fig 6 shows a more sensitive smooth line fit, here we see that some methods agree more between each other on what variables are more important (ex. ROC and PLS). Fig 7 shows the attribute importance ranking for each method, where we see that there's a general agreement of age and tobacco being important when used as predictors.

Fig 8 might show a better insight on variable importance. As each method has their own importance scaling (as seen in Table 3), we decided to rescale and center each method and group their results by attribute. By doing these we see how the importance of each attribute holds when comparing across methods. We again see there's a general agreement that age and tobacco are to be included as input variables for the classification task. We also see that ldl and famhist are also important for most of the machine learning methods. Alcohol and obesity are generally agreed as not very important predictors.

After analyzing these findings we chose Age, Cumulative Tobacco Consumption, Low Density Lipoprotein cholesterol and Family History as input variables and predictors for Coronary Heart Disease presence.

Neural Network Classifier

A neural network is a powerful computational data model that is able to capture and represent complex input/output relationships. In this case study we will use the training set to build a neural network model, using 5 hidden layers and a decay parameter of 0.05. With these parameters we obtained a 4-5-1 network seen in Fig 9. Table 4 shows positive definite Hessian values (matrix of second derivatives) of this network, confirming we are at a local minimum. By using this network to predict Coronary Heart Disease presence in our test set we obtained results shown in Table 5 and Table 6. Accuracy of this model is around 70% on the test set with a 0.73 Sensitivity and 0.58 Specificity.

For neural network tuning purposes we did a 5-repeat 10-fold cross validation using varying hidden layers from 1 to 7 and decay parameter from 0.5 to .0005. The parameters results are shown in Fig 10, where we see the best decay parameter is usually 0.5 and is not very sensitive to changes in amount of hidden units. The resulting tuned 4-1-1 neural network is shown in Fig 11. When using this network to predict Coronary Heart Disease presence in our test set we obtained results shown in Table 7 and Table 8. Now we see an increase in the model accuracy (approx. 74%) and also increase in both Sensitivity and Specificity (0.77 and 0.7 respectively).

Support Vector Machine Classifier

Support Vector Machines (SVM) are based on the concept of decision planes defining decision boundaries. A decision plane separates a set of objects having different class memberships.

In our case study we will also use this classification learning algorithm to predict coronary heart disease among South African males. The effectiveness of SVM depends on the selection of a kernel transformation function, the kernel's parameters, and a cost parameter. We will evaluate four kernel functions using the training set with default parameters to check internal accuracy on the training set. The results of this can be seen in Table 9. A graphical representation is shown in Fig 12, here we see that a Radial Basis kernel is significantly better compared to others in terms of accuracy, sensitivity and specificity, all these above 90%.

We know want to tune up our SVM model with Radial Kernel as this one showed the best results within training. By using stepwise tuning using varying cost parameters (1:50) and gamma parameters (0.015:2) we obtained the best error reduction result using the following parameters:

```
Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.2102241

Number of Support Vectors:  125
 ( no: 66 yes: 59 )
```

Tuning plot can be seen in Fig 13 and support vector objects are shown in Fig 14 for paired attributes. Using this SVM model to predict Coronary Heart Disease presence in our test set we obtained results shown in Table 10 and Table 11. Accuracy of this model is around 73% on the test set with a 0.75 Sensitivity and 0.65 Specificity.

Concluding Remarks

We were able to predict presence of coronary heart disease in South African males with good accuracy on either method. Results were similar for both methods in terms of accuracy, sensitivity and specificity on the test set, being Neural Network a slightly better classifier for this data set. Chosen input variables were also relevant as predictors; Age, Cumulative Tobacco Consumption, Low Density Lipoprotein cholesterol and Family History resulted in prediction rates above 70% for our test set using both methods.

APPENDIX

1. Tables

Table1. ***Correlation Matrix***

Correlation Matrix								
	sbp	**tobacco**	**ldl**	**adiposity**	**typea**	**obesity**	**alcohol**	**age**
sbp	1	0.212247	0.158296	0.3565	-0.05745	0.238067	0.140096	0.388771
tobacco	-	1	0.158905	0.28664	-0.01461	0.124529	0.200813	0.45033
ldl	-	-	1	0.440432	0.044048	0.330506	-0.0334	0.311799
adiposity	-	-	-	1	-0.04314	0.716556	0.10033	0.625954
typea	-	-	-	-	1	0.074006	0.039498	-0.10261
obesity	-	-	-	-	-	1	0.05162	0.291777
alcohol	-	-	-	-	-	-	1	0.101125
age	-	-	-	-	-	-	-	1

Table2: **NearZeroVariance Table**				
	freqRatio	**percentUnique**	**zeroVar**	**nzv**
sbp	1	13.41991	FALSE	FALSE
tobacco	9.727273	46.32035	FALSE	FALSE
ldl	1	71.21212	FALSE	FALSE
adiposity	1	88.31169	FALSE	FALSE
famhist	1.40625	0.4329	FALSE	FALSE
typea	1.086957	11.68831	FALSE	FALSE
obesity	1	86.58009	FALSE	FALSE
alcohol	6.875	53.8961	FALSE	FALSE
age	1.176471	10.60606	FALSE	FALSE

Table3: **Variable Importance Table**					
	Random Forest	**PLS**	**GBM**	**MARS**	**ROC**
adiposity	0.450714	0.037968	7.295677	4.510762	0.57513
age	15.09847	0.048732	22.09248	100	0.660156
alcohol	3.832095	0.014955	13.72259	0	0.489909
famhist	3.665108	0.039669	10.05043	76.82337	0.638542
ldl	6.280549	0.053065	18.78634	28.9512	0.664779
obesity	-2.46754	0.021526	10.19283	0	0.520703
sbp	-0.16615	0.024829	9.024486	43.46633	0.561068
tobacco	10.91252	0.05564	28.99314	58.45489	0.684701
typea	4.712358	0.032936	20.38716	13.66977	0.568294

Table4: **Neural Network Hessian Evaluation**			
106.1644	3.404329	0.883345	0.27712
25.2007	3.262692	0.75196	0.236626
16.30212	2.814436	0.702326	0.220337
9.518365	2.328221	0.579	0.208497
6.689868	1.865172	0.443119	0.131698
5.919026	1.504002	0.363527	0.113554
4.462778	1.326338	0.341001	0.101123
3.806633	1.058002	0.314245	

Table5. Neural Network Prediction Matrix

	No	**Yes**
No	154	28
Yes	56	40

Table6. Neural Network Test Prediction Results

	Accuracy	**Sensitivity**	**Specificity**
Results	0.697842	0.733333	0.588235

Table7. X-Val Neural Network Prediction Matrix

	No	**Yes**
No	161	21
Yes	47	49

Table8. X-Val Neural Network Test Prediction Results

	Accuracy	**Sensitivity**	**Specificity**
Results	0.755396	0.774038	0.7

Table9: **Support Vector Machine Models**				
	Radial	Sigmoid	Linear	Polynomial
Accuracy	0.945652	0.619565	0.744565	0.766304
Sensitivity	0.943548	0.708333	0.76259	0.776978
Specificity	0.95	0.453125	0.688889	0.733333

Table10. Tuned SVM Test Prediction Matrix

	No	**Yes**
No	159	23
Yes	52	44

Table11. Tuned SVM Test Prediction Results

	Accuracy	**Sensitivity**	**Specificity**
Results	0.730216	0.753555	0.656716

2. Plots

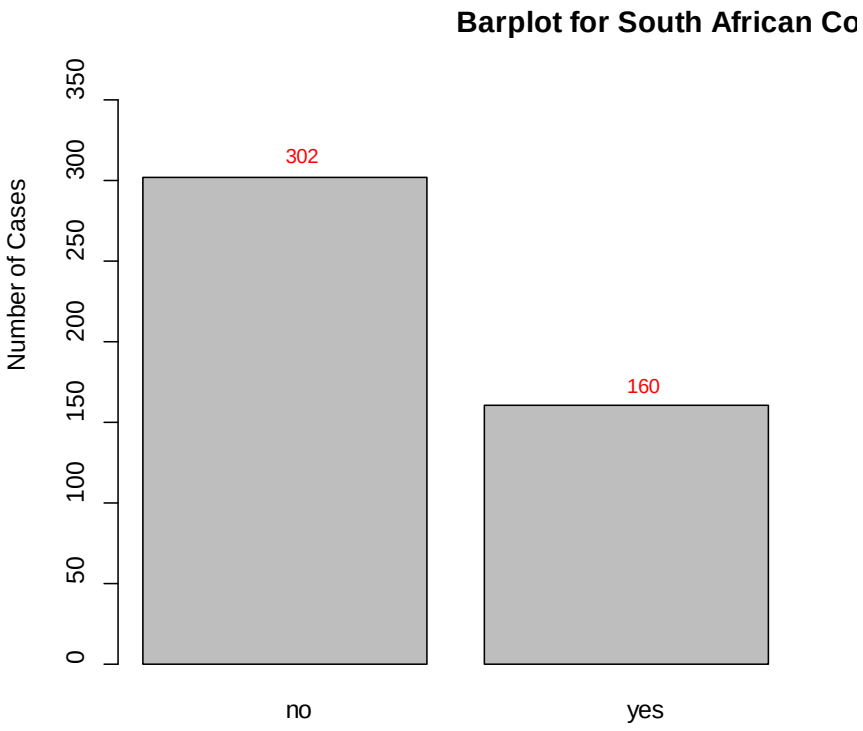


Fig 1

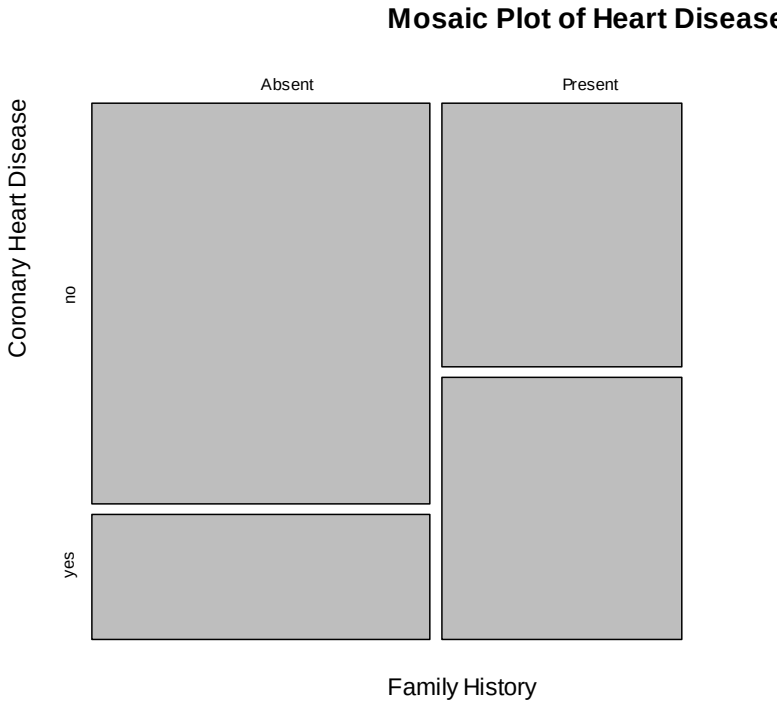


Fig 2

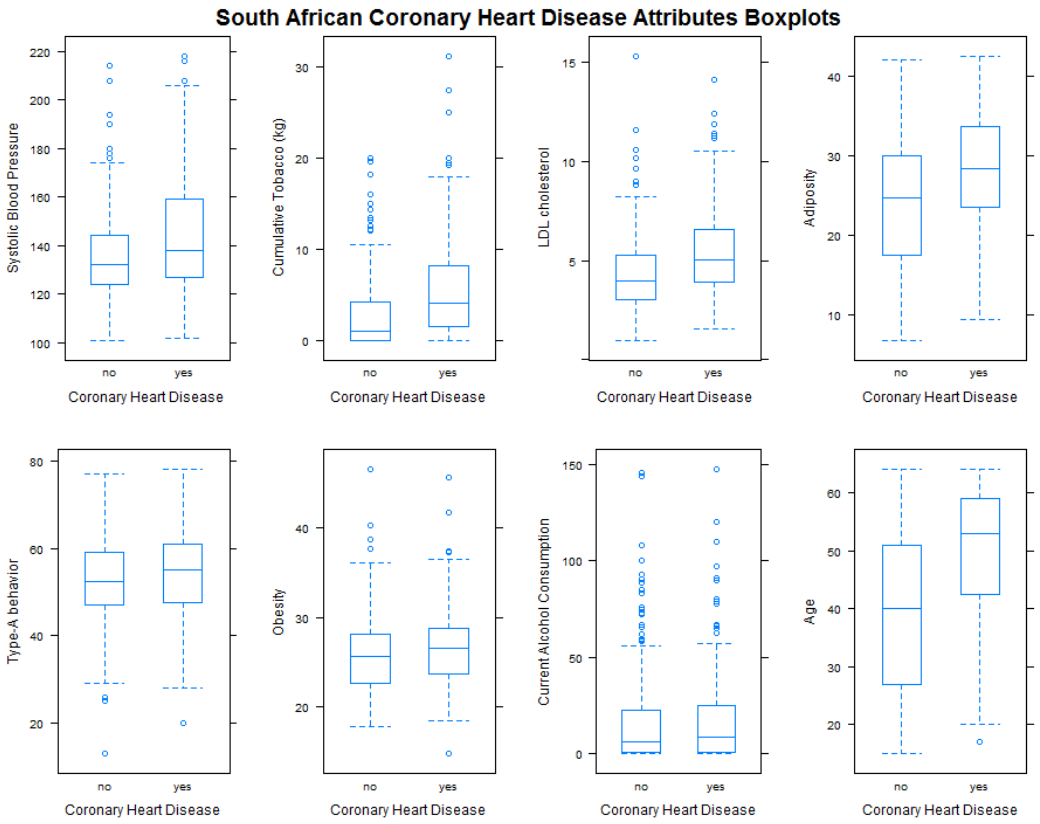

South African Coronary Heart Disease Attributes Boxplots
Systolic Blood Pressure
Cumulative Tobacco (kg)
LDL cholesterol
Adiposity
Type-A behavior
Obesity
Current Alcohol Consumption
Age
no
yes
Coronary Heart Disease

Fig3

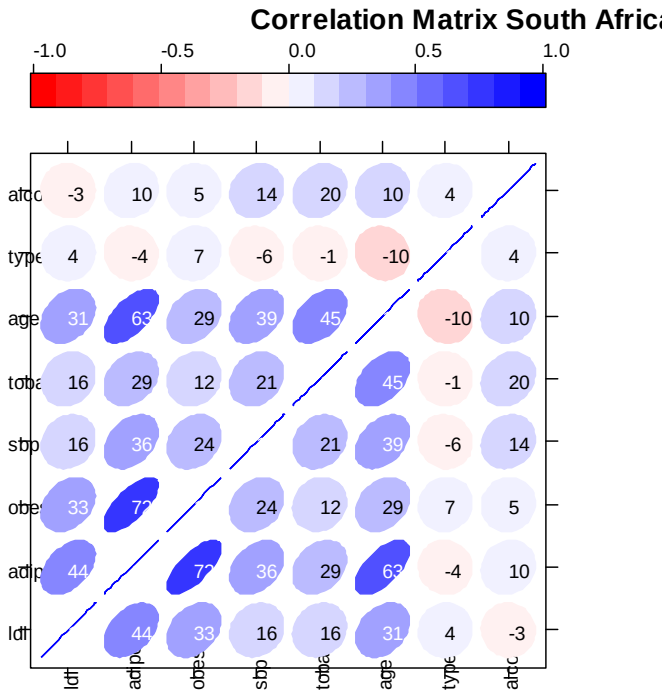

Correlation Matrix South Afric
-1.0
-0.5
0.0
0.5
1.0
-3 10 5 14 20 10 4
4 -4 7 -6 -1 -10 4
31 63 29 39 45 -10 10
16 29 12 21 45 -1 20
16 36 24 21 39 -6 14
33 24 12 29 7 5
44 36 29 63 -4 10
44 33 16 16 31 4 -3

Fig 4

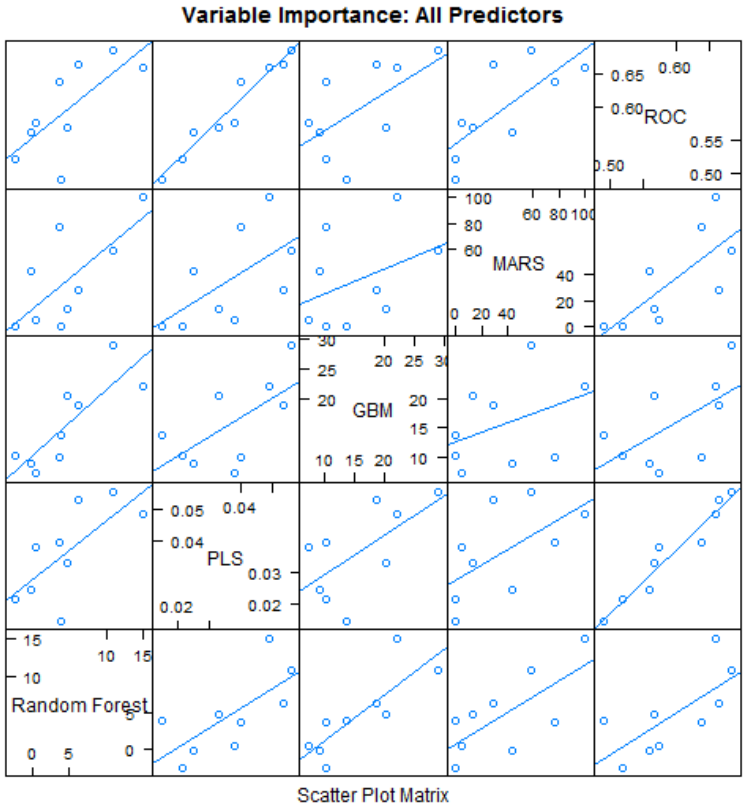

Variable Importance: All Predictors
ROC
MARS
GBM
PLS
Random Forest
Scatter Plot Matrix

Fig 5

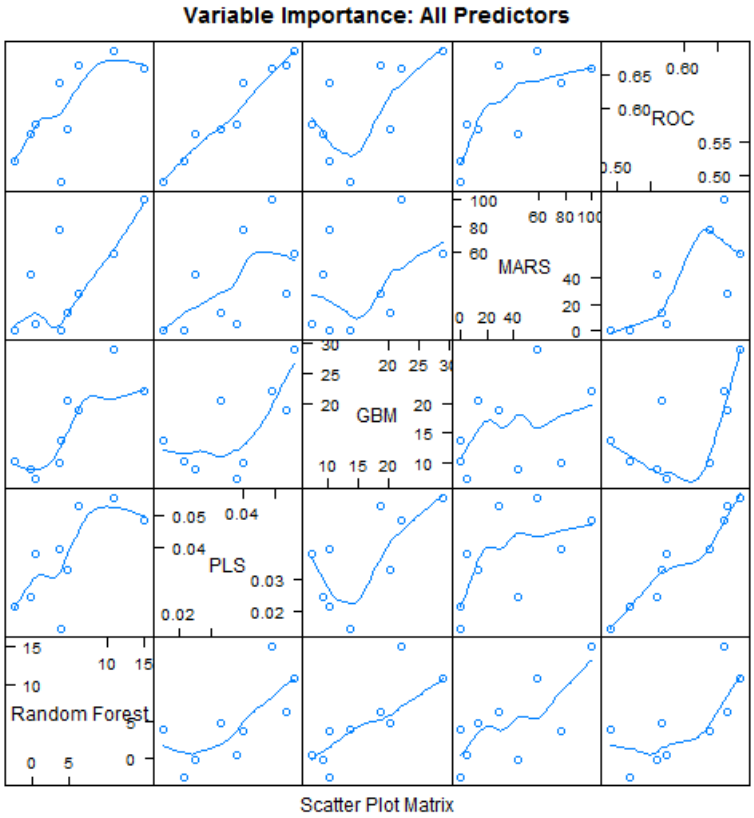

Variable Importance: All Predictors
ROC
MARS
GBM
PLS
Random Forest
Scatter Plot Matrix

Fig 6

Fig 7

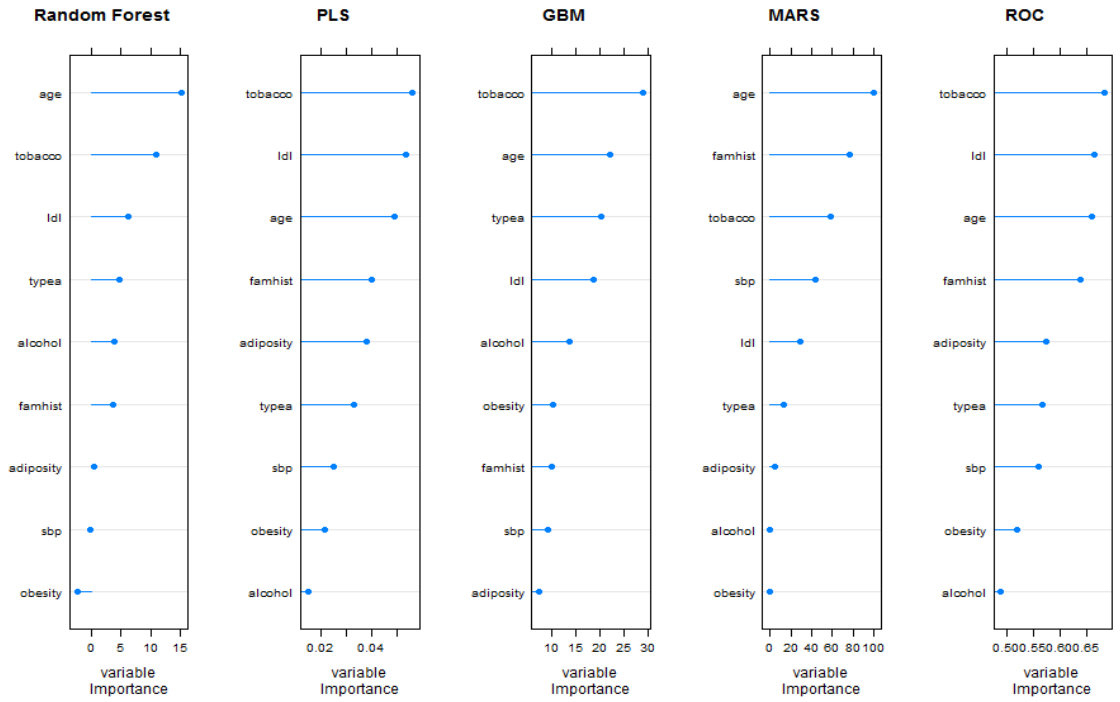
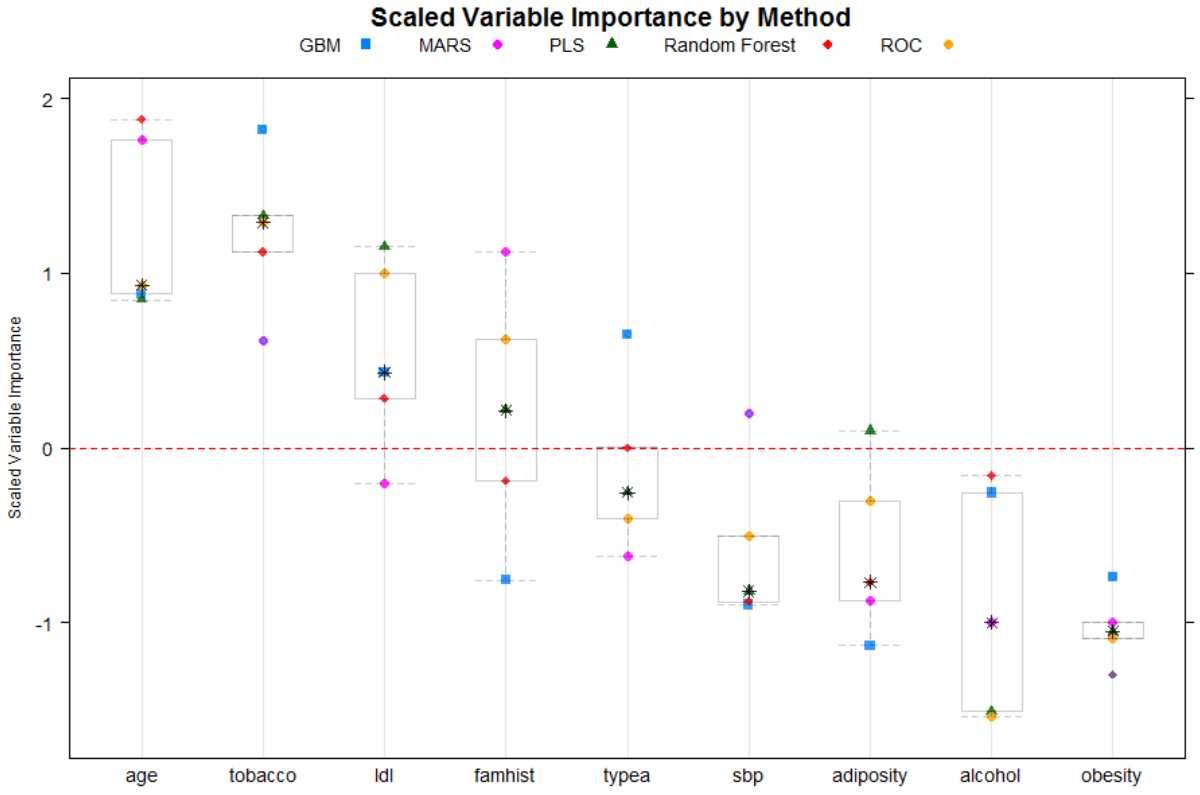
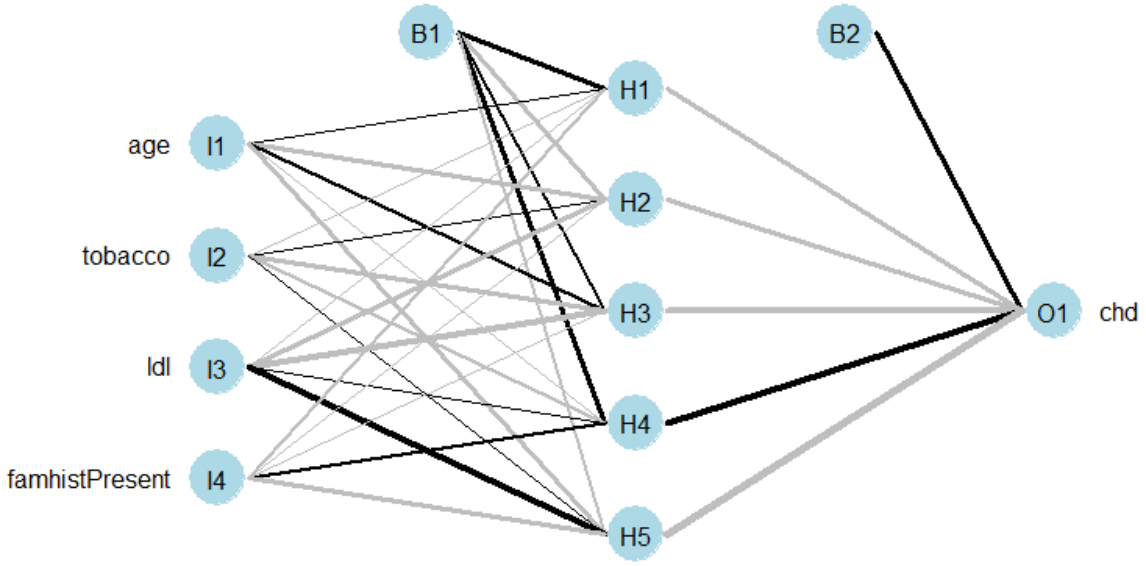

Random Forest
age
tobacco
ldl
typea
alcohol
famhist
adiposity
sbp
obesity
0 5 10 15
variable Importance
PLS
tobacco
ldl
age
famhist
adiposity
typea
sbp
obesity
alcohol
0.02 0.04
variable Importance
GBM
tobacco
age
typea
ldl
alcohol
obesity
famhist
sbp
adiposity
10 15 20 25 30
variable Importance
MARS
age
famhist
tobacco
sbp
ldl
typea
adiposity
alcohol
obesity
0 20 40 60 80 100
variable Importance
ROC
tobacco
ldl
age
famhist
adiposity
typea
sbp
obesity
alcohol
0.50 0.55 0.60 0.65
variable Importance

Fig 8


Scaled Variable Importance by Method
GBM
MARS
PLS
Random Forest
ROC
Scaled Variable Importance
2
1
0
-1
age
tobacco
ldl
famhist
typea
sbp
adiposity
alcohol
obesity

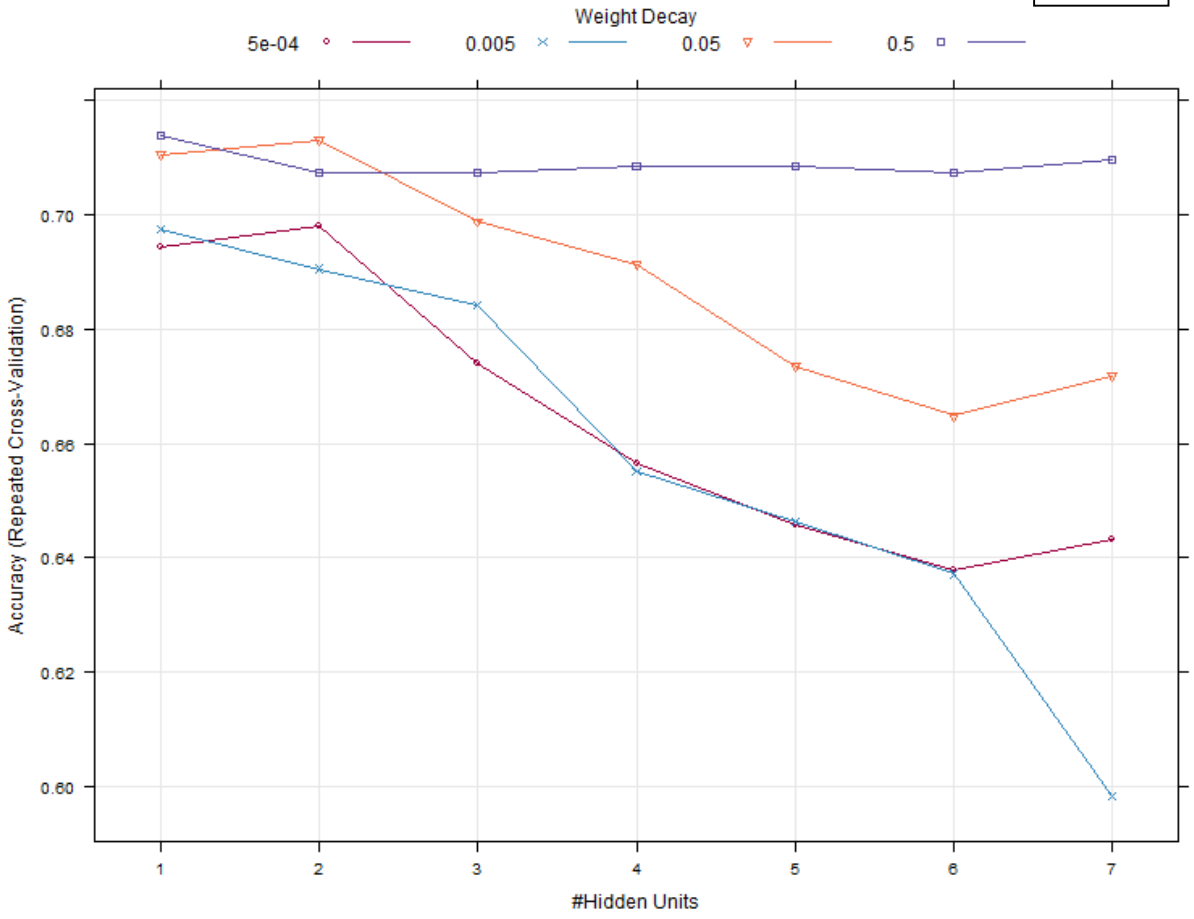
South African Heart Disease Neural Network

Fig 9



10-fold Cross-Validated Neural Network Parameters

Fig 10



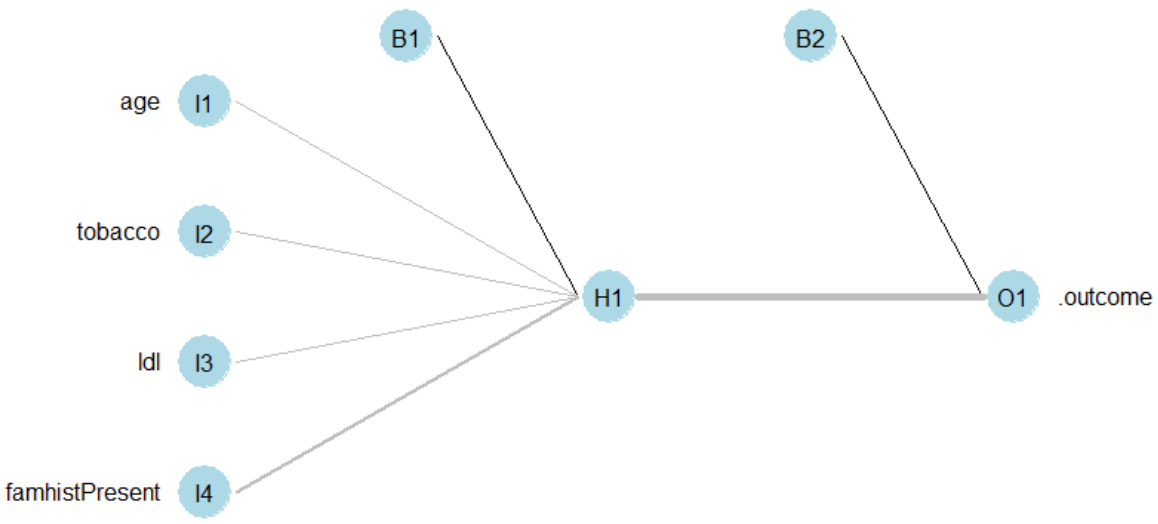

SA Heart Disease X-Val Neural Network
B1
B2
age
I1
tobacco
I2
ldl
I3
famhistPresent
I4
H1
O1
.outcome

Fig 11

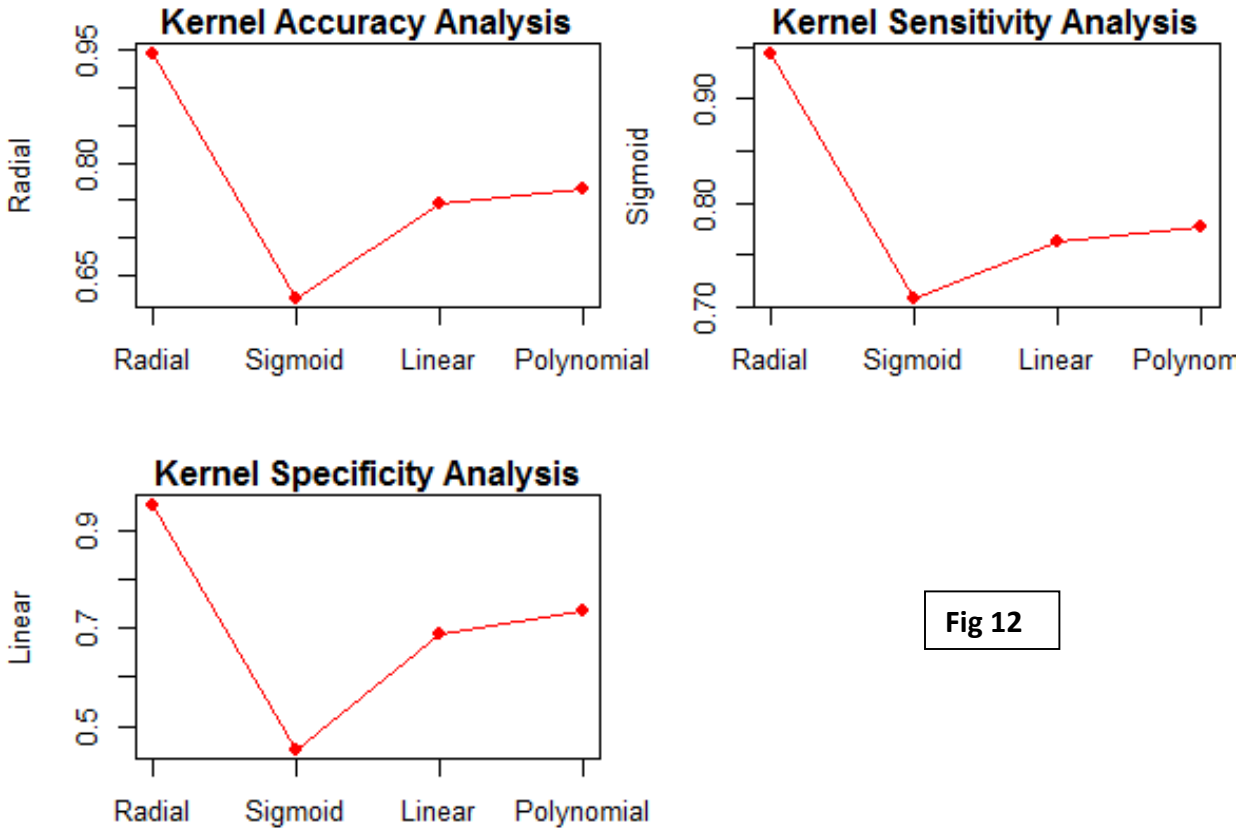

Kernel Accuracy Analysis
Radial
0.95
0.80
0.65
Radial
Sigmoid
Linear
Polynomial
Kernel Sensitivity Analysis
Sigmoid
0.90
0.80
0.70
Radial
Sigmoid
Linear
Polynom
Kernel Specificity Analysis
Linear
0.9
0.7
0.5
Radial
Sigmoid
Linear
Polynomial

Fig 12

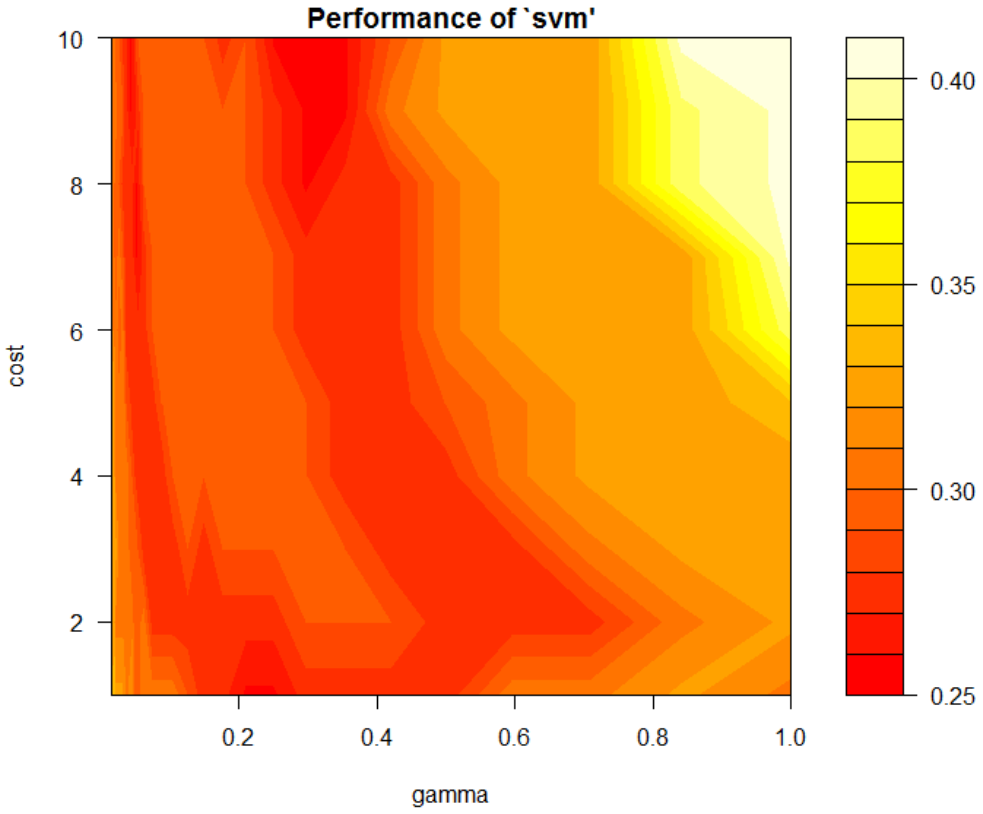

Performance of `svm'
cost
gamma
10
8
6
4
2
0.2
0.4
0.6
0.8
1.0
0.40
0.35
0.30
0.25

Fig 13

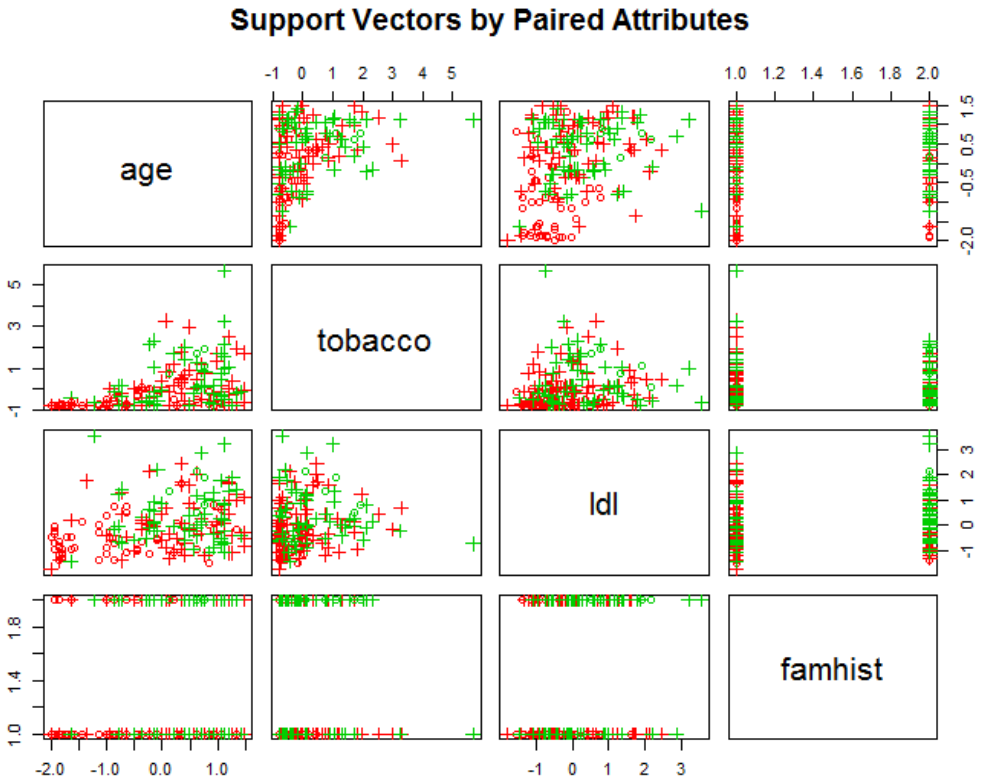

Support Vectors by Paired Attributes
age
tobacco
ldl
famhist

Fig 14

R Code

```
require(latticeExtra)
require(corrgram)
require(caret)
require(pROC)
require(randomForest)
require(nnet)
require(e1071)

library(devtools)
source_url('https://gist.githubusercontent.com/fawda123/7471137/raw/e297a212033087021bdd770625a0f
09024a22882/nnet_plot_update.r')
source_url('http://www.stat.wmich.edu/wang/685/Rcodes/pnlcorg.R')

### South African Heart Disease Data (SAheart.data)
##A retrospective sample of males in a heart-disease high-risk region
##of the Western Cape, South Africa. There are roughly two controls per
##case of CHD. Many of the CHD positive men have undergone blood
##pressure reduction treatment and other programs to reduce their risk
##factors after their CHD event. In some cases the measurements were
##made after these treatments

##sbp                systolic blood pressure
##tobacco                cumulative tobacco (kg)
##ldl                low densiity lipoprotein cholesterol
##adiposity
##famhist                family history of heart disease (Present, Absent)
##typea                type-A behavior
##obesity
##alcohol                current alcohol consumption
##age                age at onset
##chd                response, coronary heart disease

#seed = 12345
#set.seed(seed)

## READ DATA
ESLdata <- "http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets"
SAheart <- read.csv(paste(ESLdata,"SAheart.data",sep="/"),
                    row.names=1)
SAheart$chd <- factor(SAheart$chd,labels=c("no","yes"))
rm(ESLdata)

##SPLIT
nrow(SAheart)
with(SAheart, table(chd))
with(SAheart, table(famhist,chd))
(sizes <- round(with(SAheart,table(famhist,chd))*.4))

train <- NULL
for(f in levels(SAheart$famhist))
  for(c in levels(SAheart$chd)){
    s <- with(SAheart,which((famhist==f)&(chd==c)))
    train <- c(train,sample(s,sizes[f,c]))
  }
heart.train <- SAheart[train,]
heart.test <- SAheart[-train,]
rm(sizes,train,f,c,s)

######## DESCRIPTIVE STATISTICS #########

#Coronary Heart Disease Barplot
ylim <- c(0, 1.2*max(as.numeric(table(SAheart[10]))))
xx <- barplot(table(SAheart[10]), width = 0.85, ylim = ylim, ylab ="Number of Cases")
text(x =xx, y = as.numeric(table(SAheart[10])), label = as.numeric(table(SAheart[10])),
     pos = 3, cex = 0.8, col = "red")
title("Barplot for South African Coronary Heart Disease")
rm(xx,ylim)
```

```
#Mosaic plots for Categorical Attributes
mosaicplot(~ famhist + chd, data=SAheart, main="Mosaic Plot of Heart Disease vs Family History",
           xlab = "Family History", ylab="Coronary Heart Disease")


#Boxplots for other attributes
plot.new()
title(main = "\nSouth African Coronary Heart Disease Attributes Boxplots", outer=T)

plot(bwplot(sbp ~ chd, data = SAheart, pch="|",
            xlab = "Coronary Heart Disease", ylab="Systolic Blood Pressure"),
     split = c(1, 1, 4, 2), newpage = FALSE)

plot(bwplot(tobacco  ~ chd, data = SAheart, pch="|",
            xlab = "Coronary Heart Disease", ylab="Cumulative Tobacco (kg)"),
     split = c(2, 1, 4, 2), newpage = FALSE)

plot(bwplot(ldl ~ chd, data = SAheart, pch="|",
            xlab = "Coronary Heart Disease", ylab="LDL cholesterol"),
     split = c(3, 1, 4, 2), newpage = FALSE)

plot(bwplot(adiposity ~ chd, data = SAheart, pch="|",
            xlab = "Coronary Heart Disease", ylab="Adiposity"),
     split = c(4, 1, 4, 2), newpage = FALSE)

plot(bwplot(typea ~ chd, data = SAheart, pch="|",
            xlab = "Coronary Heart Disease", ylab="Type-A behavior"),
     split = c(1, 2, 4, 2), newpage = FALSE)

plot(bwplot(obesity ~ chd, data = SAheart, pch="|",
            xlab = "Coronary Heart Disease", ylab="Obesity"),
     split = c(2, 2, 4, 2), newpage = FALSE)

plot(bwplot(alcohol ~ chd, data = SAheart, pch="|",
            xlab = "Coronary Heart Disease", ylab="Current Alcohol Consumption"),
     split = c(3, 2, 4, 2), newpage = FALSE)

plot(bwplot(age ~ chd, data = SAheart, pch="|",
            xlab = "Coronary Heart Disease", ylab="Age"),
     split = c(4, 2, 4, 2), newpage = FALSE)


#Correlation Tests
(cormatrix <- cor(SAheart[-c(5,10)],SAheart[-c(5,10)], use="p"))

ord <- order.dendrogram(as.dendrogram(hclust(dist(cormatrix))))

levelplot(cormatrix[ord,ord], at=do.breaks(c(-1.01,1.01),20),
          xlab=NULL, ylab=NULL,  # no axes labels
          main="Correlation Matrix South African Coronary Heart Disease",
          scales=list(x=list(rot=90)),
          panel=panel.corrgram,   # use correlogram panel function
          label=TRUE,  # with labels of correlations x 100
          col.regions=colorRampPalette(c("red","white","blue")),
          colorkey=list(space="top"))  # using color key on the top
rm(ord)

cat("Correlation Matrix\n",file = "exploratory.csv")
cat( "\t", colnames(cormatrix), "\n", file = "exploratory.csv", sep = ",",append=T)
write.table(cormatrix, file = "exploratory.csv", sep = ",",append=T,
            qmethod = "double",col.names = F)

(findCorrelation(cor(SAheart[-c(5,10)]), 0.9))  #Multicollinearity not significantly present

######## DATA PRE-PROCESSING ########

#Near Zero Variance
dim(SAheart)

(nzv <- nearZeroVar(SAheart[-10], saveMetrics = TRUE))
```

```
cat("\n" , "NearZeroVariance Table\n",file = "exploratory.csv", append=T)
cat("\t",  colnames(nzv), "\n", file = "exploratory.csv", sep = ",",append=T)
write.table(nzv, file = "exploratory.csv", sep = ",",append=T,
            qmethod = "double",col.names = F)


#Scaling and Centering
str(heart.train)

preproc <- function( x, z, excludeClasses=c("factor"), ... ) {
  whichToExclude <- sapply( x, function(y) any(sapply(excludeClasses, function(excludeClass)
is(y,excludeClass) )) )
  processedMat <- predict( preProcess( x[!whichToExclude], ...), newdata=z[!whichToExclude] )
  z[!whichToExclude] <- processedMat
  z
}

proc.full <- preproc(heart.train, SAheart)
proc.train <- preproc(heart.train, heart.train)
proc.test <- preproc(heart.train, heart.test)


#Variable Importance
ctrl <- trainControl(verboseIter = FALSE, classProbs = TRUE)


#Random Forest
#set.seed(seed)
rfFit <- randomForest(chd~., data=proc.train, ntree= 2000, importance=T)
(rfimp <- varImp(rfFit))
rfimp <- rfimp[order(row.names(rfimp)),][1]
colnames(rfimp) <- "Random Forest"
print(rfimp)


#Partial Least Squares
#set.seed(seed)
plsFit <- train(chd~., data=proc.train, method = "pls",
                tuneGrid = data.frame(.ncomp = 1:9), trControl = ctrl)
(plsimp <- varImp(plsFit$finalModel))
plsimp <- data.frame(plsimp[order(rownames(plsimp)),])
rownames(plsimp) <- row.names(rfimp)
colnames(plsimp) <- "PLS"
print(plsimp)

#Generalized boosting model
#set.seed(seed)
gbmGrid <-  expand.grid(interaction.depth = c(1, 3, 5, 7),
                        n.trees = c(50,500, 1000),
                        shrinkage = 0.1)
gbmFit <- train(chd~.,data=proc.train, method = "gbm",
                tuneGrid = gbmGrid, verbose = FALSE)
(gbmimp <- varImp(gbmFit$finalModel))
gbmimp <- data.frame(gbmimp[order(rownames(gbmimp)),])
rownames(gbmimp) <- row.names(rfimp)
colnames(gbmimp) <- "GBM"
print(gbmimp)

#MARS model
eaFit <- bagEarth(chd~., data=proc.train, glm=list(family=binomial))
(eaimp <- varImp(eaFit, value = "gcv"))
eaimp["famhistPresent",] = sum(eaimp[c("famhistAbsent","famhistPresent"),])
eaimp <- data.frame(eaimp[order(rownames(eaimp)),])
eaimp <- data.frame(eaimp[-5,])
rownames(eaimp) <- row.names(rfimp)
colnames(eaimp) <- "MARS"
print(eaimp)
```

```
#Model Free
(ROCFit <- filterVarImp(proc.train[-10], proc.train[[10]]))
ROCimp <- ROCFit[order(row.names(ROCFit)),][1]
colnames(ROCimp) <- "ROC"
print(ROCimp)


(Imp <- cbind(rfimp, plsimp, gbmimp, eaimp, ROCimp))

cat("\n" , "Variable Importance Table\n",file = "exploratory.csv", append=T)
cat("\t",  colnames(Imp), "\n", file = "exploratory.csv", sep = ",",append=T)
write.table(Imp, file = "exploratory.csv", sep = ",",append=T,
            qmethod = "double",col.names = F)


#Var Importance plots
splom(Imp, type = c("p", "r"),
      main = "Variable Importance: All Predictors")
splom(Imp, type = c("p", "smooth"),
      main = "Variable Importance: All Predictors")

ntop <- 9
n <- 5
vars <- NULL  # vector to collect 3 list of variable names, initialized
hasMore <- c(rep(T,n-1),F)
for(i in 1:n){   # for each method
  ord <- order(Imp[,i], decreasing=TRUE)[ntop:1]
  x <- Imp[ord,i]
  y <- rownames(Imp)[ord]
  vars <- c(vars, y)
  y <- factor(y, levels=y)
  plot(dotplot(y ~ x, type=c("p","h"), xlab="variable\nImportance",
               main=names(Imp)[i]),
       split=c(i,1,n,1), more=hasMore[i])
}
rm(ntop,vars,hasMore,i,x,y,ord, n)

#Subsetting by Var Importance
(Impsc <- preproc(Imp, Imp))

x <- as.data.frame(c(rep("Random
Forest",9),rep("PLS",9),rep("GBM",9),rep("MARS",9),rep("ROC",9)))
x[2] <- factor(rep(rownames(Impsc),5))
x[3] <- c(Impsc[[1]],Impsc[[2]],Impsc[[3]],Impsc[[4]],Impsc[[5]])
colnames(x) <- c("Method", "Predictor", "varImp")

(y <- as.data.frame(sort(rowMeans(-Impsc))))
x$Predictor <- factor(x$Predictor, levels=rownames(y))

dotplot(varImp ~ Predictor, data = x, groups = Method,
        main = list("Scaled Variable Importance by Method", cex=1.6),
        par.settings = list(superpose.symbol = list(pch = 15:19, cex=1.2),
                            box.rectangle = list(col="black",alpha=0.2),
                            box.umbrella = list(col="black",alpha=0.2)),
        auto.key=list(columns = 5, cex=1.2, pch = 15:19),
        cex=1.2, scales=list(cex=1.2), alpha=0.8,
        ylab="Scaled Variable Importance",
        panel = function(...) {
          panel.dotplot(...)
          panel.abline(h=0, col=2, lty=2)
          panel.bwplot(...,pch=8, col="black")
        })

(include <- rownames(which(y < 0, arr.ind=TRUE)))

imp.full <- proc.full[c(include,"chd")]
imp.train <- proc.train[c(include,"chd")]
imp.test <- proc.test[c(include,"chd")]

rm(x,y,include)
```

```
################## NEURAL NETWORK ####################

#set.seed(seed)
NN <- nnet(chd ~ ., data=imp.train, size=5, Hess=T,
           decay=0.05, rang=0.1, maxit=200)

summary(NN)

NNCM <- confusionMatrix(imp.test[["chd"]], predict(NN, imp.test, type="class"))
NNCM$table
NNCMres <-
(cbind(NNCM$overall["Accuracy"],cbind(sensitivity(NNCM$table),specificity(NNCM$table))))
colnames(NNCMres) = c("Accuracy","Sensitivity","Specificity")
rownames(NNCMres) = "Results"
print(NNCMres)

(NNHess <- eigen(NN$Hess, only.values=T)$values)

cat("\n", "Neural Network Prediction Matrix\n", file = "NN.csv")
cat( "\t", colnames(NNCM$table), "\n", file = "NN.csv", sep = ",", append=T)
write.table(NNCM$table, file = "NN.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Neural Network Test Prediction Results\n", file = "NN.csv", append = T)
cat( "\t", colnames(NNCMres), "\n", file = "NN.csv", sep = ",", append=T)
write.table(NNCMres, file = "NN.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Neural Network Hessian Evaluation\n", file = "NN.csv", append = T)
write.table(NNHess, file = "NN.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

#Neural Network Plot
plot(NN, main ="South African Heart Disease Neural Network")


#Neural Network X-Val
fitControl <- trainControl(method = "repeatedcv",
                           number = 10, repeats = 5,
                           classProbs = TRUE)
nnGrid <- expand.grid(size=1:7, decay=5*10^(-(4:1)))

#set.seed(seed)
NNXV <- train(chd ~ ., data=imp.train, method="nnet",
              trace=F,
              tuneGrid=nnGrid,
              trControl=fitControl)
summary(NNXV)
trellis.par.set(caretTheme())
plot(NNXV,  main ="10-fold Cross-Validated Neural Network Parameters")

NNXVCM <- confusionMatrix(imp.test[["chd"]], predict(NNXV, imp.test))
NNXVCM$table
NNXVCMres <-
(cbind(NNXVCM$overall["Accuracy"],cbind(sensitivity(NNXVCM$table),specificity(NNXVCM$table))))
colnames(NNXVCMres) = c("Accuracy","Sensitivity","Specificity")
rownames(NNXVCMres) = "Results"
print(NNXVCMres)

cat("\n", "X-Val Neural Network Prediction Matrix\n", file = "NN.csv", append = T)
cat( "\t", colnames(NNXVCM$table), "\n", file = "NN.csv", sep = ",", append=T)
write.table(NNXVCM$table, file = "NN.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "X-Val Neural Network Test Prediction Results\n", file = "NN.csv", append = T)
cat( "\t", colnames(NNXVCMres), "\n", file = "NN.csv", sep = ",", append=T)
write.table(NNXVCMres, file = "NN.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

#NN X-val Plot
plot.nnet(NNXV, main ="SA Heart Disease X-Val Neural Network")
```

```
################## Support Vector Machines ####################

svm1   <- svm(chd ~ ., data=imp.train, cost=100, gamma=1)

SVMCM <- confusionMatrix(imp.train[["chd"]], predict(svm1, imp.train))
SVMCM$table
SVMCMres <-
(cbind(SVMCM$overall["Accuracy"],cbind(sensitivity(SVMCM$table),specificity(SVMCM$table))))
colnames(SVMCMres) = c("Accuracy","Sensitivity","Specificity")
rownames(SVMCMres) = "Radial"

svm2   <- svm(chd ~ ., data=imp.train, cost=100, gamma=1,
              kernel='sigmoid', type='C-classification')

SVMCM2 <- confusionMatrix(imp.train[["chd"]], predict(svm2, imp.train))
SVMCM2$table
SVMCM2res <-
(cbind(SVMCM2$overall["Accuracy"],cbind(sensitivity(SVMCM2$table),specificity(SVMCM2$table))))
colnames(SVMCM2res) = c("Accuracy","Sensitivity","Specificity")
rownames(SVMCM2res) = "Sigmoid"

svm3   <- svm(chd ~ ., data=imp.train, cost=100, gamma=1,
              kernel='linear', type='C-classification')

SVMCM3 <- confusionMatrix(imp.train[["chd"]], predict(svm3, imp.train))
SVMCM3$table
SVMCM3res <-
(cbind(SVMCM3$overall["Accuracy"],cbind(sensitivity(SVMCM3$table),specificity(SVMCM3$table))))
colnames(SVMCM3res) = c("Accuracy","Sensitivity","Specificity")
rownames(SVMCM3res) = "Linear"

svm4   <- svm(chd ~ ., data=imp.train, cost=100, gamma=1,
              kernel='polynomial', type='C-classification')

SVMCM4 <- confusionMatrix(imp.train[["chd"]], predict(svm4, imp.train))
SVMCM4$table
SVMCM4res <-
(cbind(SVMCM4$overall["Accuracy"],cbind(sensitivity(SVMCM4$table),specificity(SVMCM4$table))))
colnames(SVMCM4res) = c("Accuracy","Sensitivity","Specificity")
rownames(SVMCM4res) = "Polynomial"

(allsvm <- cbind(t(SVMCMres), t(SVMCM2res), t(SVMCM3res), t(SVMCM4res)))


cat("\n" , "Support Vector Machine Models\n",file = "SVM.csv")
cat("\t",  colnames(allsvm), "\n", file = "SVM.csv", sep = ",",append=T)
write.table(allsvm, file = "SVM.csv", sep = ",",append=T,
            qmethod = "double",col.names = F)


#PLOT KERNEL MODELS
xx <- colnames(allsvm)
oldpar <- par(mfrow=c(2,2), mar=c(4.1,4.1,1.5,0.5))
plot(allsvm[1,], pch=16, type="o", col="red", ylab=colnames(allsvm)[1],
     xaxt = "n", xlab = "", main = "Kernel Accuracy Analysis")
axis(1, 1:4, labels= xx)
plot(allsvm[2,], pch=16, type="o", col="red", ylab=colnames(allsvm)[2],
     xaxt = "n", xlab = "", main = "Kernel Sensitivity Analysis")
axis(1, 1:4, labels= xx)
plot(allsvm[3,], pch=16, type="o", col="red", ylab=colnames(allsvm)[3],
     xaxt = "n", xlab = "", main = "Kernel Specificity Analysis")
axis(1, 1:4, labels= xx)
oldpar <- par(mfrow=c(1,1), mar=c(4.1,4.1,1.5,0.5))
```

```
#####Tuning SVM

#Radial
svmkernel <- "radial"

set.seed(12345)
svmTuned <- tune(svm, chd ~ ., data=imp.train, kernel=svmkernel,
                 ranges = list(gamma=2^(-1:1), cost = 1:50),
                 tunecontrol = tune.control(sampling = "fix"))
head(summary(svmTuned))
plot(svmTuned, color.palette=heat.colors)

set.seed(12345)
svmTuned <- tune(svm, chd ~ ., data=imp.train, kernel=svmkernel,
                 ranges = list(gamma=2^(seq(-6, 0, .25)), cost = 1:10),
                 tunecontrol = tune.control(sampling = "fix"))
head(summary(svmTuned))
plot(svmTuned, color.palette=heat.colors)
plot(svmTuned,type='p',theta=240,phi=0)


#Best Tune
set.seed(12345)
svmopt <- best.tune(svm, chd ~ ., data=imp.train, kernel=svmkernel,
           ranges = list(gamma=2^(seq(-6, 0, .25)), cost = 1:10),
           tunecontrol = tune.control(sampling = "fix"))

summary(svmopt)

SV <- 1:184 %in% svmopt$index + 1  # 1=non Support Vector, 2=Support Vector
pairs(imp.train[-5], col=as.integer(imp.train[[5]])+1,
      pch=c(1,3)[SV],cex=c(1,1.25)[SV], main = "Support Vectors by Paired Attributes")

legend(locator(1), xpd = TRUE, horiz = F, lwd=1, pch = c(1,1), bty='n', text.width=0.2,
       cex=0.9, legend=c("No","Yes"))

# SVM Test Set Prediction
SVMBest <- confusionMatrix(imp.test[["chd"]], predict(svmopt, imp.test))
SVMBest$table
SVMBestres <-
(cbind(SVMBest$overall["Accuracy"],cbind(sensitivity(SVMBest$table),specificity(SVMBest$table))))
colnames(SVMBestres) = c("Accuracy","Sensitivity","Specificity")
rownames(SVMBestres) = "Radial"
print(SVMBestres)

cat("\n", "Tuned SVM Test Prediction Matrix\n", file = "SVM.csv", append = T)
cat( "\t", colnames(SVMBest$table), "\n", file = "SVM.csv", sep = ",", append=T)
write.table(SVMBest$table, file = "SVM.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Tuned SVM Test Prediction Results\n", file = "SVM.csv", append = T)
cat( "\t", colnames(SVMBestres), "\n", file = "SVM.csv", sep = ",", append=T)
write.table(SVMBestres, file = "SVM.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)
```