HW3_Solution

Sara Taheri

2/25/2017

Problem 1

Data preparation and exploration

a) Select the training set: Download the data. Partition the dataset into a training and a validation subsets of equal size, by randomly selecting rows in the training set.

First I will read the data set:

```
library(data.table)
SAheart <- fread('https://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/SAheart.data')
SAheart <- as.data.frame(SAheart)
SAheart_No_row <- nrow(SAheart)
SAheart_No_row
```

```
## [1] 462
```

We have 462 observations, so we separate the data into two equal size parts, with each of them having number of rows equal to 231 (462/231).

```
# set the seed to produce the same result each time running the code
set.seed(231)
# randomly select 231 rows for training set and validation set
train_row <- sample(1:462,231)
validation_row <- c(1:462)[-train_row]
# convert the categorical variable to numeric
SAheart$famhist <- as.character(SAheart$famhist)
SAheart$famhist[which(SAheart$famhist=="Present")] <- "1"
SAheart$famhist[which(SAheart$famhist=="Absent")] <- "0"
SAheart$famhist <- as.numeric(SAheart$famhist)
# create training and validation sets
SAheart_train <- SAheart[train_row,]
SAheart_train <- SAheart_train[,-1]
SAheart_validation <- SAheart[validation_row,]
SAheart_validation <- SAheart_validation[,-1]
```

Note: an alternative way to handle categorical data for regression modeling in R is to specify them as factors. This is particularly convenient when the predictor has more than two possible levels.

b) Data exploration: Consider the training set only. Report one-variable summary statistics, two-variable summary statistics, and discuss your findings (e.g., presence of highly correlated predictors, categorical predictors, missing values, outliers etc).

```
# one variable summary
summary(SAheart_train)
```

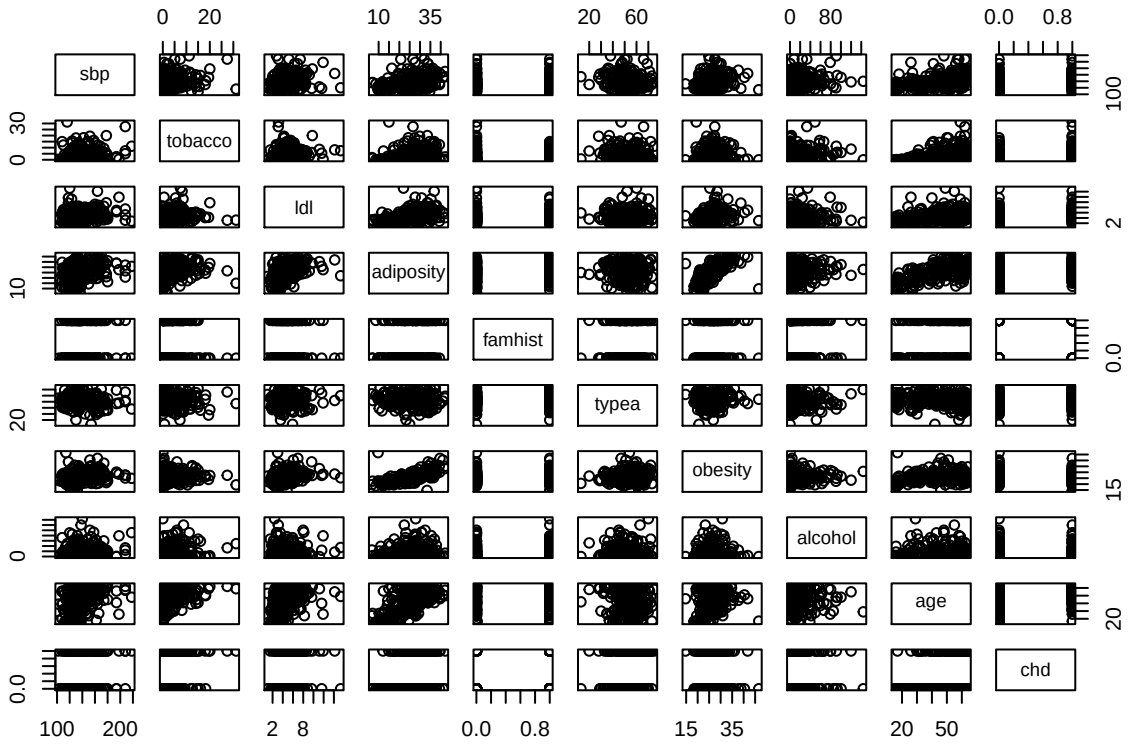
```
##       sbp           tobacco            ldl          adiposity
##  Min.   :102.0   Min.   : 0.000   Min.   : 0.980   Min.   : 6.74
##  1st Qu.:124.0   1st Qu.: 0.050   1st Qu.: 3.255   1st Qu.:19.98
##  Median :134.0   Median : 2.020   Median : 4.370   Median :26.50
##  Mean   :139.3   Mean   : 4.060   Mean   : 4.810   Mean   :25.62
```

```
##  3rd Qu.:151.0   3rd Qu.: 6.135   3rd Qu.: 5.910   3rd Qu.:31.88
##  Max.   :218.0   Max.   :31.200   Max.   :15.330   Max.   :42.17
##     famhist           typea          obesity         alcohol
##  Min.   :0.0000   Min.   :13.00   Min.   :14.70   Min.   :  0.00
##  1st Qu.:0.0000   1st Qu.:46.00   1st Qu.:23.16   1st Qu.:  0.00
##  Median :0.0000   Median :52.00   Median :26.09   Median :  6.43
##  Mean   :0.4026   Mean   :52.46   Mean   :26.25   Mean   : 17.07
##  3rd Qu.:1.0000   3rd Qu.:59.00   3rd Qu.:28.66   3rd Qu.: 24.27
##  Max.   :1.0000   Max.   :75.00   Max.   :46.58   Max.   :144.00
##       age              chd
##  Min.   :15.00   Min.   :0.0000
##  1st Qu.:31.50   1st Qu.:0.0000
##  Median :45.00   Median :0.0000
##  Mean   :43.56   Mean   :0.3593
##  3rd Qu.:57.50   3rd Qu.:1.0000
##  Max.   :64.00   Max.   :1.0000
```

```
# Two variable summary
round(cor(SAheart_train), digits = 2)
```

```
##             sbp tobacco   ldl adiposity famhist typea obesity alcohol
## sbp        1.00    0.20  0.22      0.38    0.10 -0.12    0.23    0.17
## tobacco    0.20    1.00  0.17      0.28    0.07 -0.05    0.09    0.19
## ldl        0.22    0.17  1.00      0.47    0.14  0.05    0.32   -0.01
## adiposity  0.38    0.28  0.47      1.00    0.27 -0.05    0.67    0.12
## famhist    0.10    0.07  0.14      0.27    1.00  0.08    0.17    0.07
## typea     -0.12   -0.05  0.05     -0.05    0.08  1.00    0.09   -0.04
## obesity    0.23    0.09  0.32      0.67    0.17  0.09    1.00    0.05
## alcohol    0.17    0.19 -0.01      0.12    0.07 -0.04    0.05    1.00
## age        0.39    0.47  0.36      0.64    0.27 -0.12    0.29    0.10
## chd        0.20    0.36  0.25      0.28    0.32  0.05    0.07    0.02
##              age  chd
## sbp         0.39 0.20
## tobacco     0.47 0.36
## ldl         0.36 0.25
## adiposity   0.64 0.28
## famhist     0.27 0.32
## typea      -0.12 0.05
## obesity     0.29 0.07
## alcohol     0.10 0.02
## age         1.00 0.41
## chd         0.41 1.00
```

```
pairs(SAheart_train)
```



None of the predictors are highly correlated with chd. Age, famhist, ldl have higher correlation than other predictors. Typea, alcohol and obesity have low correlation. No extremely high correlation is observed.

```
# missing data
sum(is.na(SAheart_train))
```

```
## [1] 0
```

There is no missing data.

Problem 2

Fit logistic regression on the training set. Perform variable selection using all subsets selection and AIC or BIC criteria. [Hint: it may be interesting to also consider statistical interactions]

```
# Logistic regression on all the training data
glm.fit <- glm(chd~., data = SAheart_train, family = "binomial")
summary(glm.fit)
```

```
##
## Call:
## glm(formula = chd ~ ., family = "binomial", data = SAheart_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8451  -0.7823  -0.3407   0.7801   2.4859
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -4.595030   1.906806  -2.410 0.015961 *
## sbp          0.007073   0.008262   0.856 0.391971
## tobacco      0.108416   0.036372   2.981 0.002875 **
## ldl          0.130180   0.081045   1.606 0.108216
## adiposity    0.043521   0.042813   1.017 0.309369
## famhist      1.256766   0.338415   3.714 0.000204 ***
## typea        0.028249   0.017657   1.600 0.109624
## obesity     -0.119647   0.065852  -1.817 0.069231 .
## alcohol     -0.005334   0.006960  -0.766 0.443460
## age          0.041406   0.017048   2.429 0.015150 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 301.69  on 230  degrees of freedom
## Residual deviance: 224.44  on 221  degrees of freedom
## AIC: 244.44
##
## Number of Fisher Scoring iterations: 5
```

Now I will perform all (best) subset selection, to select only the predictors which have significant effect on *chd*:

```
library(bestglm)
```

```
## Loading required package: leaps
```

```
##
## Attaching package: 'bestglm'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     SAheart
```

```
SAheart_train_bestglm <- SAheart_train
dimnames(SAheart_train_bestglm)[[2]][10] <- "y"
glm.fit.best <- bestglm(Xy = SAheart_train_bestglm, family = binomial,
                        IC = "AIC", method = "exhaustive")
```

```
## Morgan-Tatar search since family is non-gaussian.
```

```
summary(glm.fit.best)
```

```
## Fitting algorithm:  AIC-glm
## Best Model:
##             df deviance
## Null Model 224 226.7552
## Full Model 230 301.6946
##
##  likelihood-ratio test - GLM
##
## data:  H0: Null Model vs. H1: Best Fit AIC-glm
## X = 74.939, df = 6, p-value = 3.952e-14
```

```
names(glm.fit.best)
```

```
## [1] "BestModel"   "BestModels"  "Bestq"       "qTable"      "Subsets"
```

```
## [6] "Title"       "ModelReport"
```

We want the best model, so I will call the BestModel from *glm.fit.best* :

```
a <- glm.fit.best$BestModel
a$coefficients
```

```
## (Intercept)     tobacco         ldl     famhist       typea     obesity
## -4.45030886  0.10651044  0.15350966  1.26484043  0.02545650 -0.06583343
##         age
##  0.05283433
```

As you can see by looking at the coefficients, the best model, is the one with predictors *tobacco, ldl, famhist, typea, obesity* and *age*.

Problem 3

Fit LDA on the training set, using the standard workflow.

Since LDA assumes multivariate Normal distributions and cannot be used with categorical variables, I will exclude the categorical variables from the training set.

```
# Fit LDA on training set using all the predictors
library(MASS)
lda.fit <- lda(chd ~ sbp + tobacco + ldl + adiposity + typea + obesity + alcohol + age,
               data = SAheart_train)
lda.fit
```

```
## Call:
## lda(chd ~ sbp + tobacco + ldl + adiposity + typea + obesity +
##     alcohol + age, data = SAheart_train)
##
## Prior probabilities of groups:
##         0         1
## 0.6406926 0.3593074
##
## Group means:
##        sbp  tobacco      ldl adiposity    typea  obesity  alcohol      age
## 0 136.2095 2.694257 4.409122  23.95000 52.11486 26.03358 16.62459 38.94595
## 1 144.8434 6.495060 5.523735  28.58434 53.07229 26.64675 17.87759 51.79518
##
## Coefficients of linear discriminants:
##                    LD1
## sbp        0.005553314
## tobacco    0.098459436
## ldl        0.109225238
## adiposity  0.032417977
## typea      0.024626419
## obesity   -0.078230219
## alcohol   -0.004761039
## age        0.038248900
```

I want to compare the coefficients of LDA with the one we got in problem 2 for the best subset selection in logistic regression:

If you look at the coefficients, *sbp* and *alcohol* has very close to zero coefficient and they were not selected by best model in logistic regression.

Also, *tobacco*,*ldl*,*typea* and *obesity* coefficients in both models are very close.

The remaining coefficients is *adiposity* which is somewhat close to zero in LDA and has not been selected by best subset selection in Logistic regression. The other remaining coefficient is *age* which the coefficient in LDA differes from best subset selection in logistic by 0.02.

Note that LDA does not do dimension reduction and we lost one predictor.

Problem 4

Fit logistic regression with Lasso regularization on the training set.

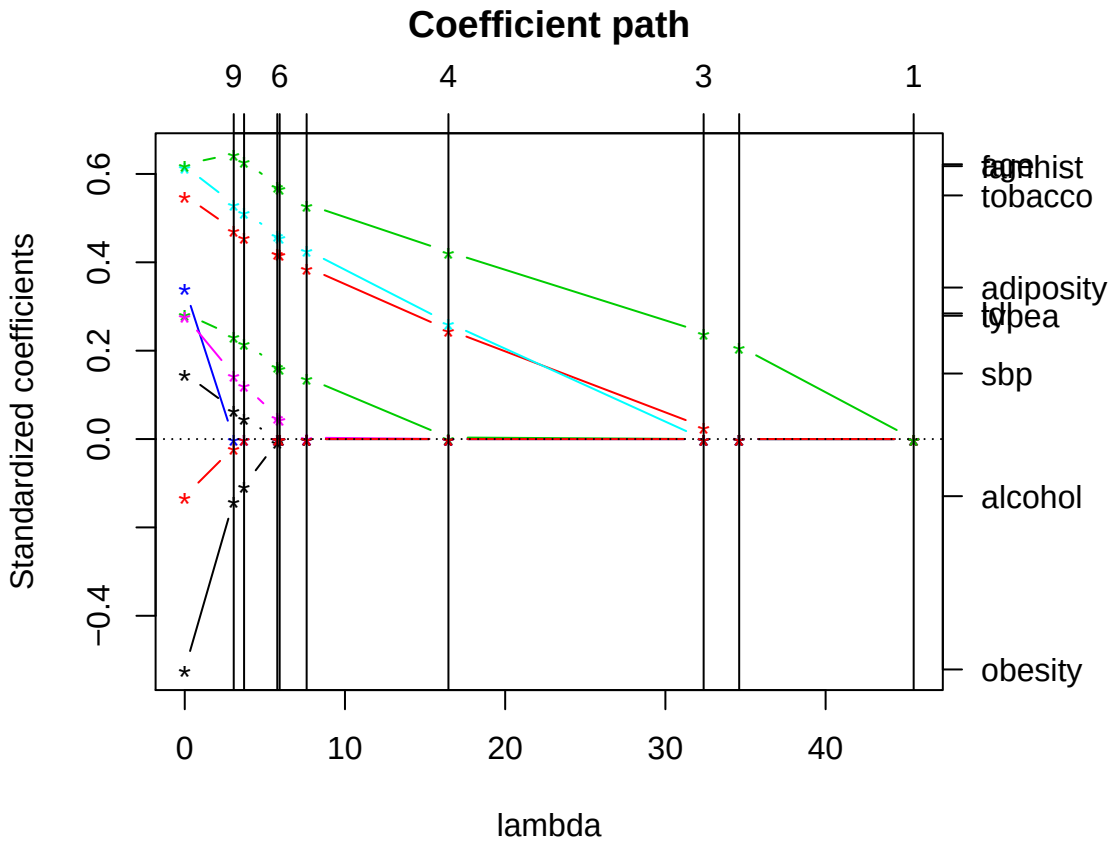
a) Produce and interpret the plot of paths of the individual coefficients

First, we try many different values of regularization parameter λ, and find λ minimizing predictive error estimated by cross-validation. By ploting the estimated coefficients for differenct values of λ, we can see the effect of regularization, and the predictors selected for each value. For instance, when λ is close to zero we have more non-zero coefficients. This is due to milder regularization. On the other hand, as λ increases, the coefficients of many of the predictors are assigned to 0.

```
library(glmpath)
```

```
## Loading required package: survival
```

```
fit.glmpath <- glmpath(x = as.matrix(SAheart_train[,-10]),
                       y=SAheart_train[,10],family=binomial)
par(mfrow=c(1,1), mar=c(4,4,4,8))
plot(fit.glmpath, xvar="lambda")
```

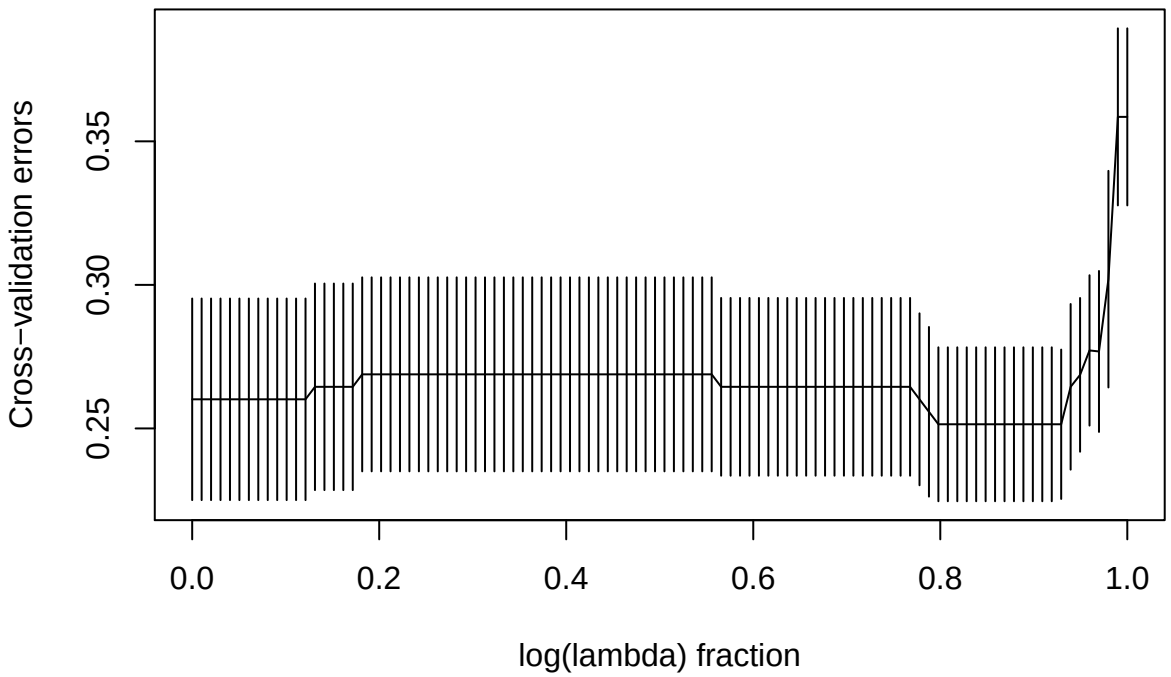


b) Produce the plot of regularized parameter versus cross-validated predicted error.

We obtained the plot by using different values of lambda(by default), and the error achieved by cross validation.

```
fit.cv.glmpath <- cv.glmpath(x=(as.matrix(SAheart_train[,-10])),
                  y=SAheart_train[,10],
                  family=binomial, nfold=10, plot.it=T, type = "response", mode = "lambda")
```

```
## CV Fold 1
## CV Fold 2
## CV Fold 3
## CV Fold 4
## CV Fold 5
## CV Fold 6
## CV Fold 7
## CV Fold 8
## CV Fold 9
## CV Fold 10
```



c) Select regularization parameter, and refit the model with this parameter.

The value of the shrinkage parameter λ which minimizes cross validated predicted error is

```
cv.s <- fit.cv.glmpath$fraction[which.min(fit.cv.glmpath$cv.error)]
# log lambda
cv.s
```

```
## [1] 0.7979798
```

```
# lambda
exp(cv.s)
```

```
## [1] 2.221049
```

d) Fit the model with the selected predictors only on the full training set.

```
pred.coef <- predict(fit.glmpath, s=cv.s, mode="norm.fraction", type="coefficients")
pred.coef
```

```
##                   Intercept         sbp  tobacco       ldl  adiposity
## 0.797979797979798 -4.462763 0.004870786 0.099754 0.1169728 0.01876937
##                    famhist      typea     obesity  alcohol        age
## 0.797979797979798 1.157656 0.02054138 -0.06965044 -0.00274 0.04229766
## attr(,"s")
## [1] 0.7979798
## attr(,"fraction")
##         1
## 0.7979798
## attr(,"mode")
## [1] "norm.fraction"
```

Let's compare the coefficients with the one we obtained by best subset selection in Logistic regression:

sbp, *adiposity* and *alcohol* have near zero cofficients here and are also excluded from best subset selection for logistic regression.

The coefficient of *tobacco*,*typea* and *obesity* are close to the ones we calculated for best subset selection for logistic regression.

The coefficient of *ldl* differs by a factor of 0.4, *famhist* differes by a factor of 0.3, *age* differs by a factor of 0.01.

Problem 5

Fit the nearest shrunken centroids model on the training set.

a) Use cross-validation to select the best regularization parameter.

We first reformat the training and the validation sets to the format expected by `pamr`. We fit the classifier for a range of thresholds specified, use cross validation in `pamr.cv`, and obtain the threshold that minimizes the cross-validated prediction error. Then, we refit the model with the selected threshold, and print the centroids based on that threshold.

```
library(pamr)
```

```
## Loading required package: cluster
```

```
# Reformat the dataset for pamr
pamrTrain <- list(x=t(as.matrix(SAheart_train[,-10])), y=SAheart_train[,10])
pamrValid <- list(x=t(as.matrix(SAheart_validation[,-10])),
                  y=SAheart_validation[,10])
# Fit the classifier on the entire training set
fit.pamr <- pamr.train(pamrTrain)
```

```
## 123456789101112131415161718192021222324252627282930
```

```
fit.pamr$centroids
```

```
##                     0          1
## sbp       136.2094595 144.8433735
## tobacco     2.6942568   6.4950602
## ldl         4.4091216   5.5237349
## adiposity  23.9500000  28.5843373
## famhist     0.2837838   0.6144578
## typea      52.1148649  53.0722892
## obesity    26.0335811  26.6467470
## alcohol    16.6245946  17.8775904
## age        38.9459459  51.7951807
## attr(,"scaled:scale")
## y
##   0   1
## 148  83
```

```
fit.cv.pamr <- pamr.cv(fit.pamr, pamrTrain)
```

```
## 12Fold 1 :123456789101112131415161718192021222324252627282930
## Fold 2 :123456789101112131415161718192021222324252627282930
## Fold 3 :123456789101112131415161718192021222324252627282930
## Fold 4 :123456789101112131415161718192021222324252627282930
## Fold 5 :123456789101112131415161718192021222324252627282930
```

```
## Fold 6 :12345678910111213141516171819202122324252627282930
## Fold 7 :12345678910111213141516171819202122324252627282930
## Fold 8 :12345678910111213141516171819202122324252627282930
## Fold 9 :12345678910111213141516171819202122324252627282930
## Fold 10 :12345678910111213141516171819202122324252627282930
```

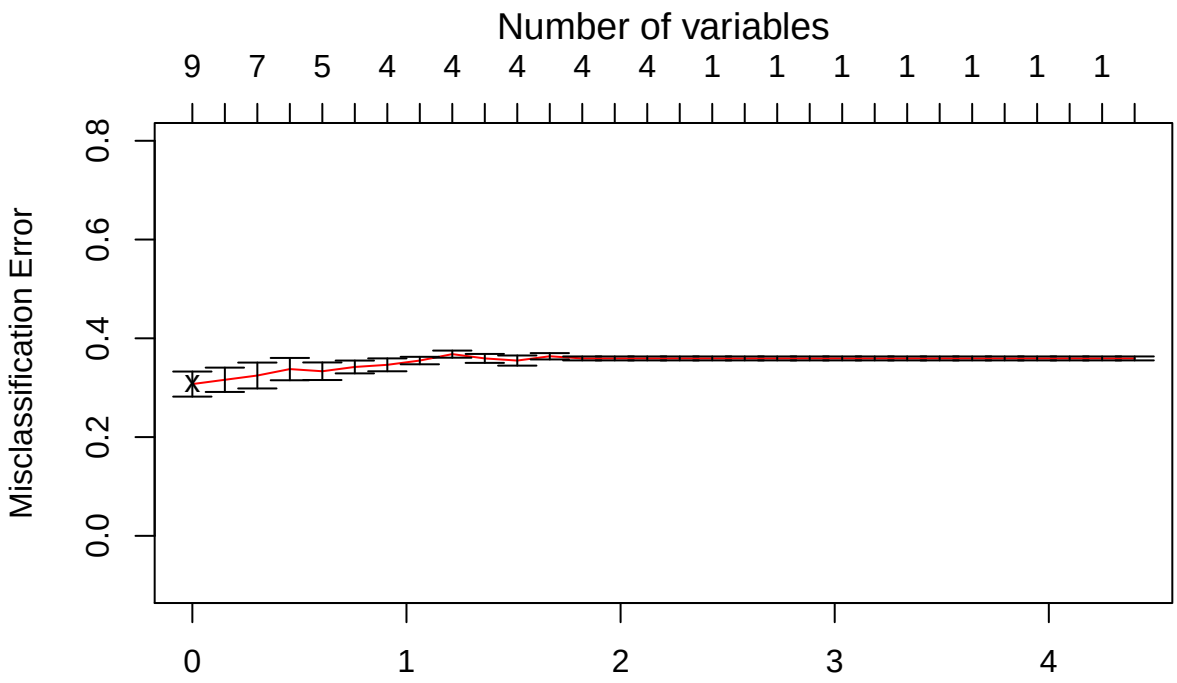
```
fit.cv.pamr
```

```
## Call:
## pamr.cv(fit = fit.pamr, data = pamrTrain)
##    threshold nonzero errors
## 1  0.000     9       71
## 2  0.152     9       73
## 3  0.304     7       75
## 4  0.455     5       78
## 5  0.607     5       77
## 6  0.759     5       79
## 7  0.911     4       80
## 8  1.062     4       82
## 9  1.214     4       85
## 10 1.366     4       83
## 11 1.518     4       82
## 12 1.669     4       84
## 13 1.821     4       83
## 14 1.973     4       83
## 15 2.125     4       83
## 16 2.276     1       83
## 17 2.428     1       83
## 18 2.580     1       83
## 19 2.732     1       83
## 20 2.883     1       83
## 21 3.035     1       83
## 22 3.187     1       83
## 23 3.339     1       83
## 24 3.490     1       83
## 25 3.642     1       83
## 26 3.794     1       83
## 27 3.946     1       83
## 28 4.097     1       83
## 29 4.249     1       83
## 30 4.401     0       83
```

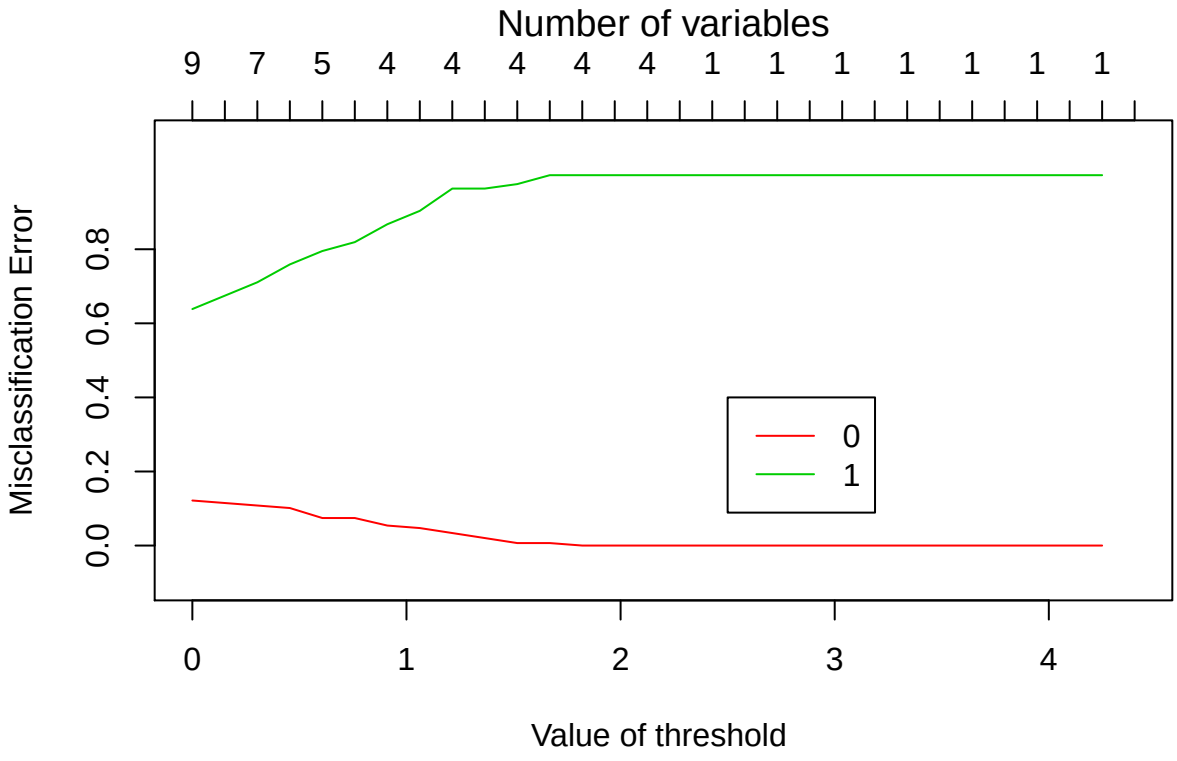
```
names(fit.cv.pamr)
```

```
##  [1] "threshold"       "error"           "loglik"
##  [4] "size"            "yhat"            "y"
##  [7] "prob"            "folds"           "cv.objects"
## [10] "pvalue.survival" "call"            "sample.subset"
```

```
pamr.plotcv(fit.cv.pamr)
```



Nearest Shrunken Centroids



The misclassifcation error is lower for small values of the threshold. There is not much change in the error for chd = 0 samples as the threshold increases, but the misclassification of chd = 1 samples increases rapidly. You would choose a threshold that has the largest amount of shrinkage without resulting in an increase in error as compared to no shrinkage. I will choose the threshold as 0.7.

Let's see the confusion matrix:

```
pamr.confusion(fit.cv.pamr, threshold=.7)
```

```
##     0  1 Class Error rate
## 0 137 11       0.07432432
## 1  68 15       0.81927711
## Overall error rate= 0.341
```

b) Refit the model with the selected regularization parameter

```
# Refit the classifier on the full dataset, but using the threshold
fit.pamr <- pamr.train(pamrTrain, threshold=0.7)
```

```
## 1
```

```
fit.pamr
```

```
## Call:
## pamr.train(data = pamrTrain, threshold = 0.7)
##   threshold nonzero errors
## 1 0.7       5       74
```

c) Visualize the centroids of the selected mode

The visualization is presented here:

```
#pamr.plotcen(fit.pamr, pamrTrain, threshold=0.7)
```

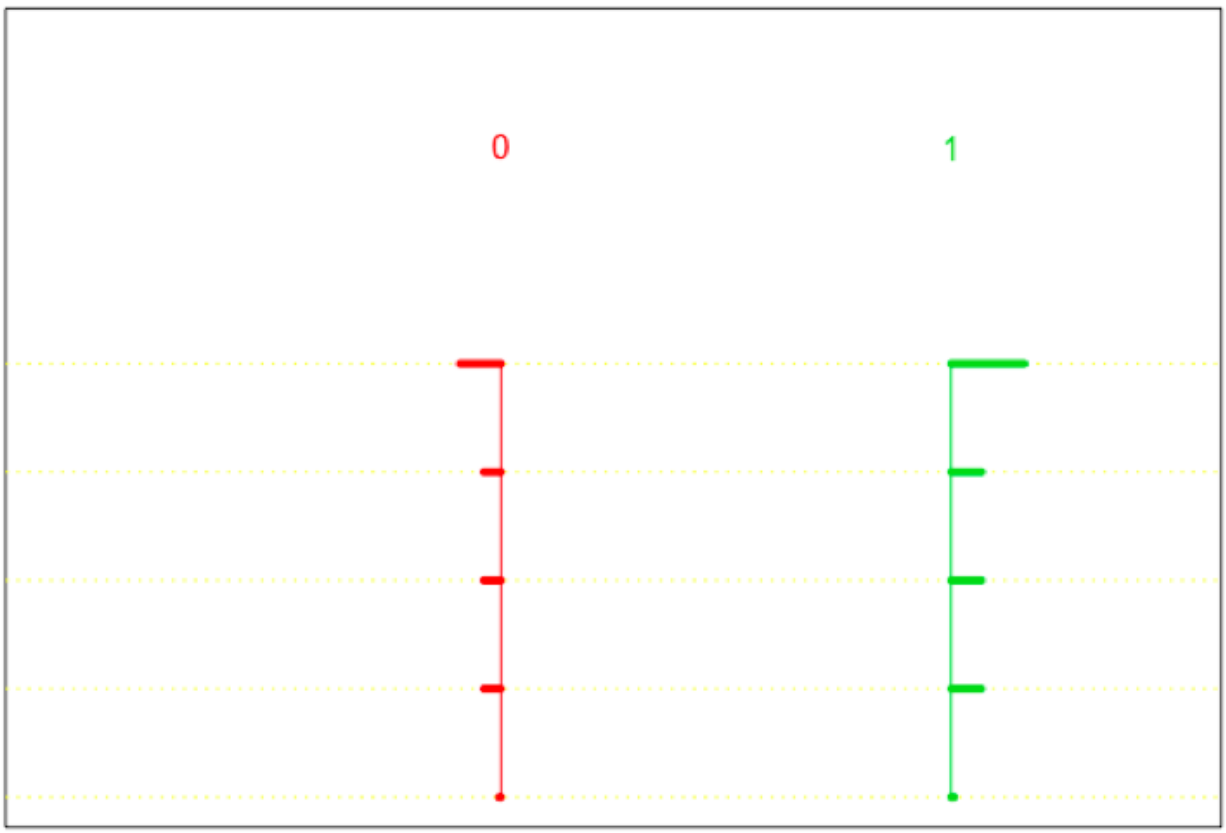


Figure 1: figure 1

Problem 6

Evaluate the performance of the classifiers

a) Evaluate the performance of the classifiers

For each classifier, we first derive the probabilities for each observation (i.e., the probability that the reponse value is 1), the use that to find the accuracy and then use it to achieve the performance of the classifier. Then we plot the ROC curve, and print the area under the curve, so that we can compare the classifiers. We will use this value to compare classifiers on validation set, and the one with higher area is better.

For logistic regression with all predictors:

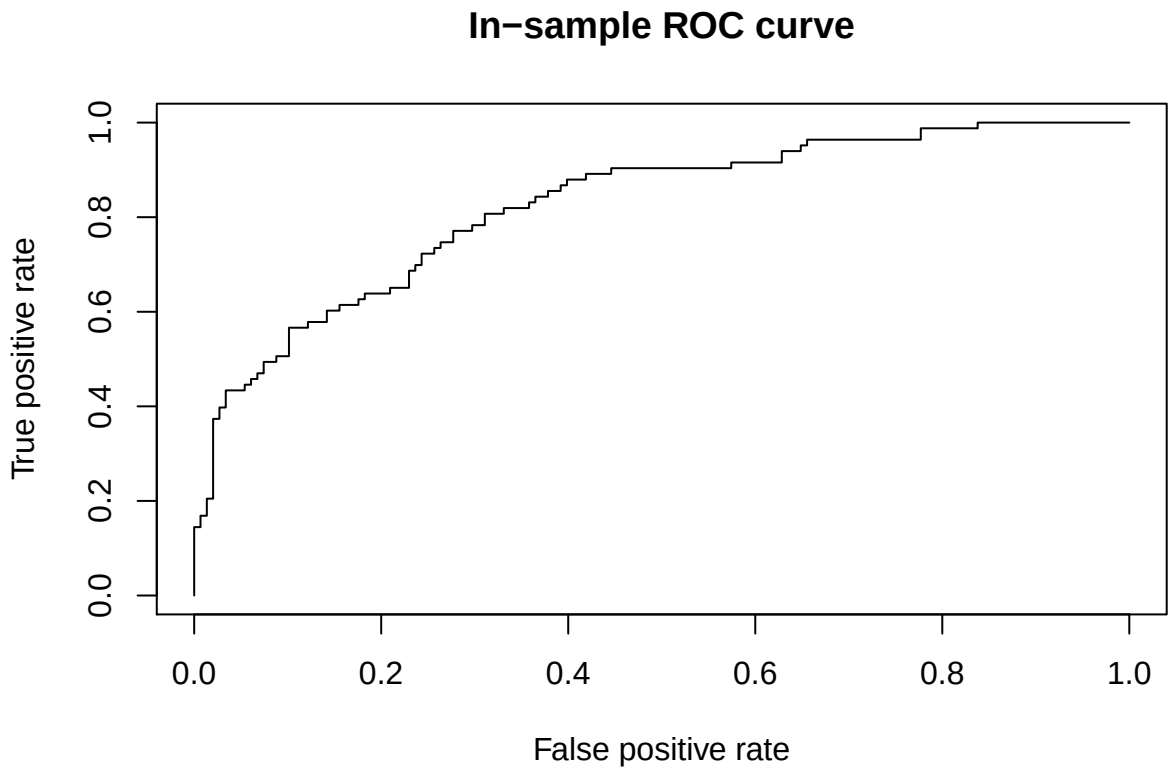
```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

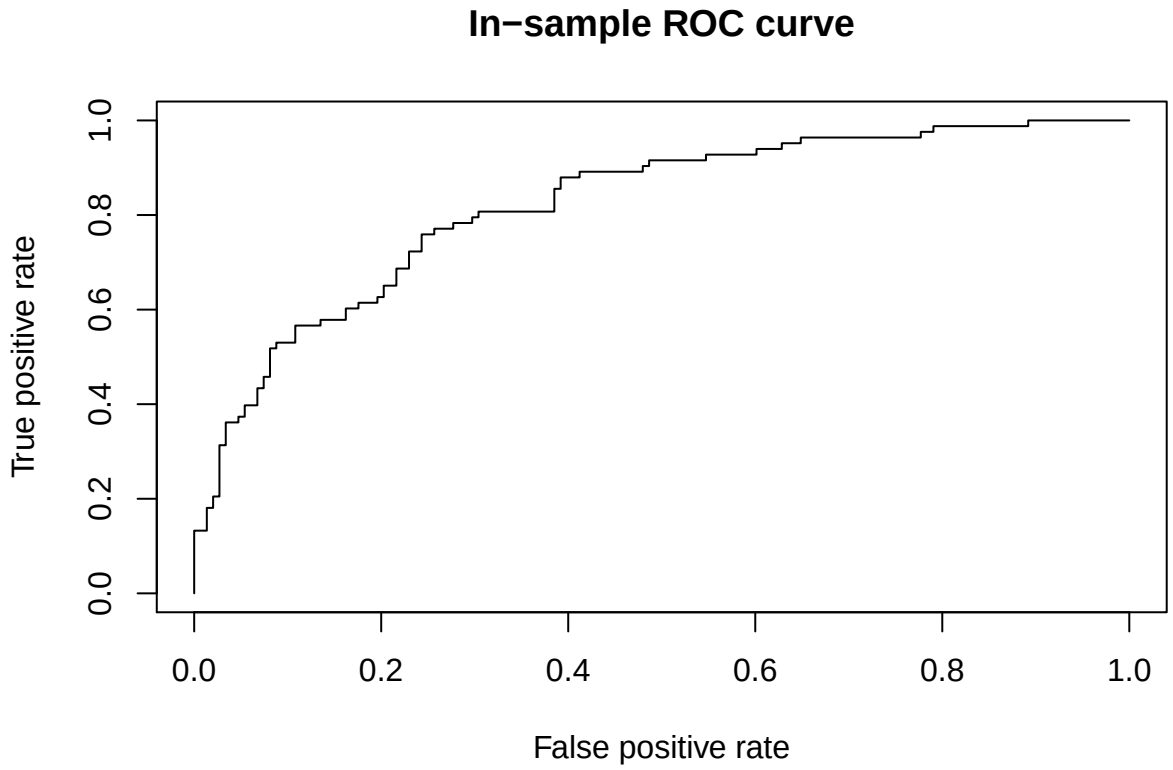
```
logit.prob = predict(glm.fit, SAheart_train[,-10], type="response")
logit.pred = prediction(logit.prob, SAheart_train[,10])
logit.perf = performance(logit.pred, 'tpr', 'fpr')
plot(logit.perf, colorize=F, main="In-sample ROC curve")
```



```
logit.area.train <- unlist(attributes(performance(logit.pred, "auc"))$y.values)
```

For logistic regression with subset selection:

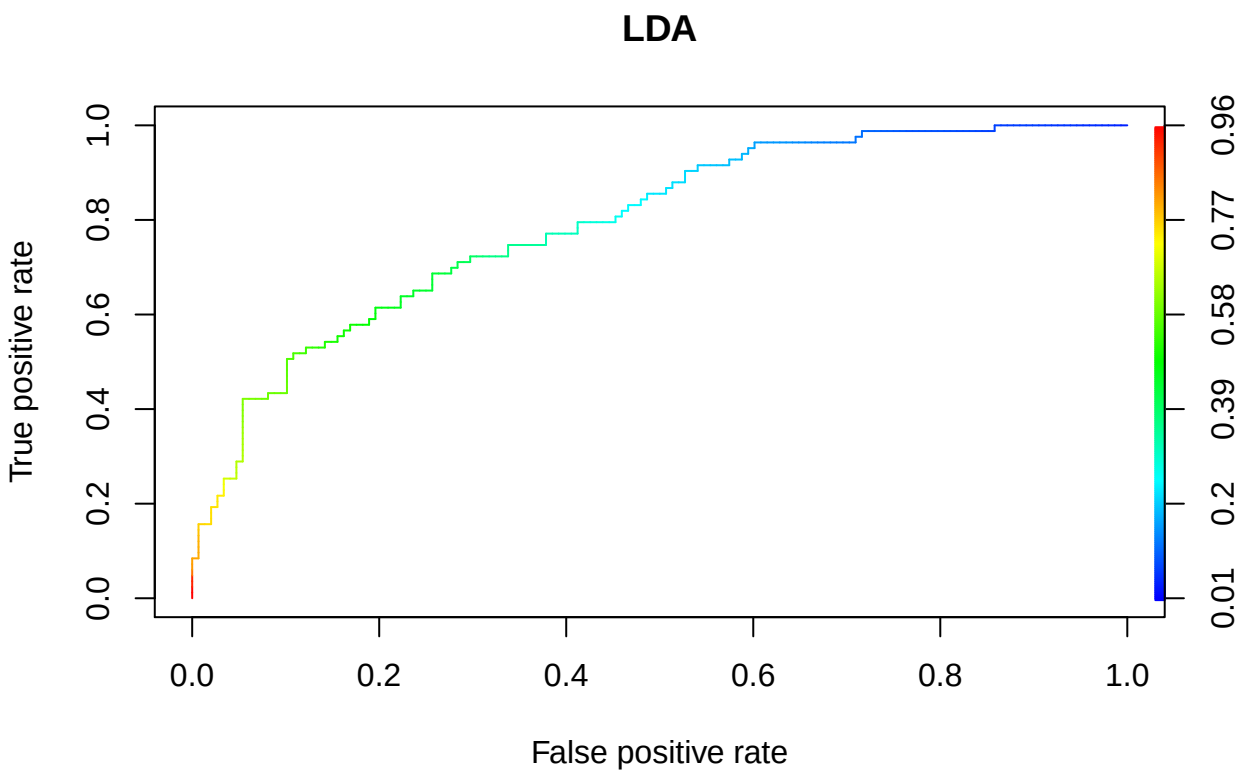
```
logit.best.prob <- predict(glm.fit.best$BestModel, SAheart_train[,-10],
                           type="response")
logit.best.pred <- prediction(logit.best.prob, SAheart_train[,10])
logit.best.perf <- performance(logit.best.pred, 'tpr', 'fpr')
plot(logit.best.perf, colorize=F, main="In-sample ROC curve")
```




```
logit.best.area.train <- unlist(attributes(performance(logit.best.pred, "auc"))$y.values)
```

For LDA classifier:

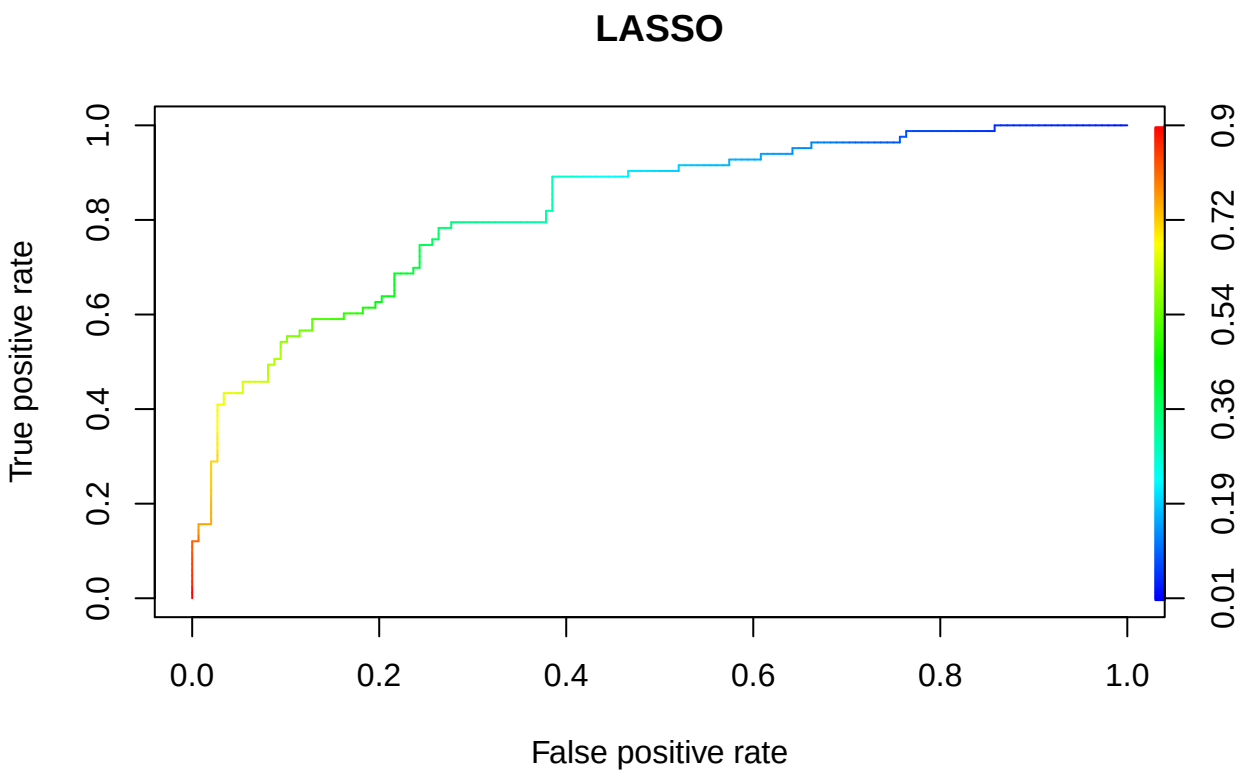
```
lda.prob = predict(lda.fit, SAheart_train[,-10])$posterior[,2]
lda.pred = prediction(lda.prob, SAheart_train[,10])
lda.perf = performance(lda.pred, "tpr", "fpr")
plot(lda.perf, colorize=T, main="LDA")
```




```
lda.area.train <- unlist(attributes(performance(lda.pred, "auc"))$y.values)
```

For Lasso regression:

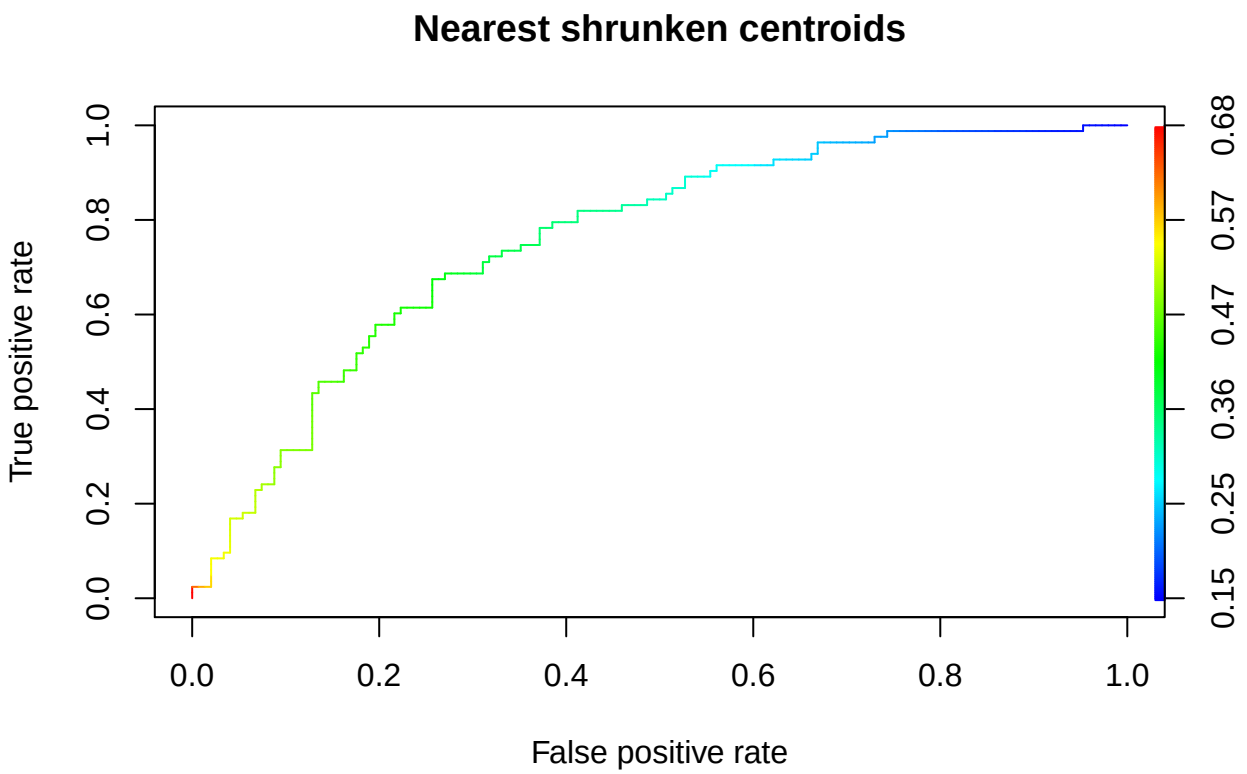
```
glmpath.prob = predict(fit.glmpath,newx=as.matrix(SAheart_train[,-10]),s=cv.s,
                       mode="norm.fraction",type="response")
glmpath.pred = prediction( predictions=glmpath.prob, labels=SAheart_train[,10])
glmpath.perf = performance(glmpath.pred, "tpr", "fpr")
plot(glmpath.perf, colorize=T, main="LASSO")
```




```
lasso.area.train <- unlist(attributes(performance(glmpath.pred, "auc"))$y.values)
```

For nearest shrunken centroids:

```
pamr.prob = pamr.predict(fit.pamr, newx=pamrTrain$x, threshold=0.7,
                         type="posterior")[,2]
pamr.pred = prediction(predictions=pamr.prob, labels= SAheart_train[,10])
pamr.perf <- performance(pamr.pred, "tpr", "fpr")
plot(pamr.perf, colorize=T, main="Nearest shrunken centroids")
```

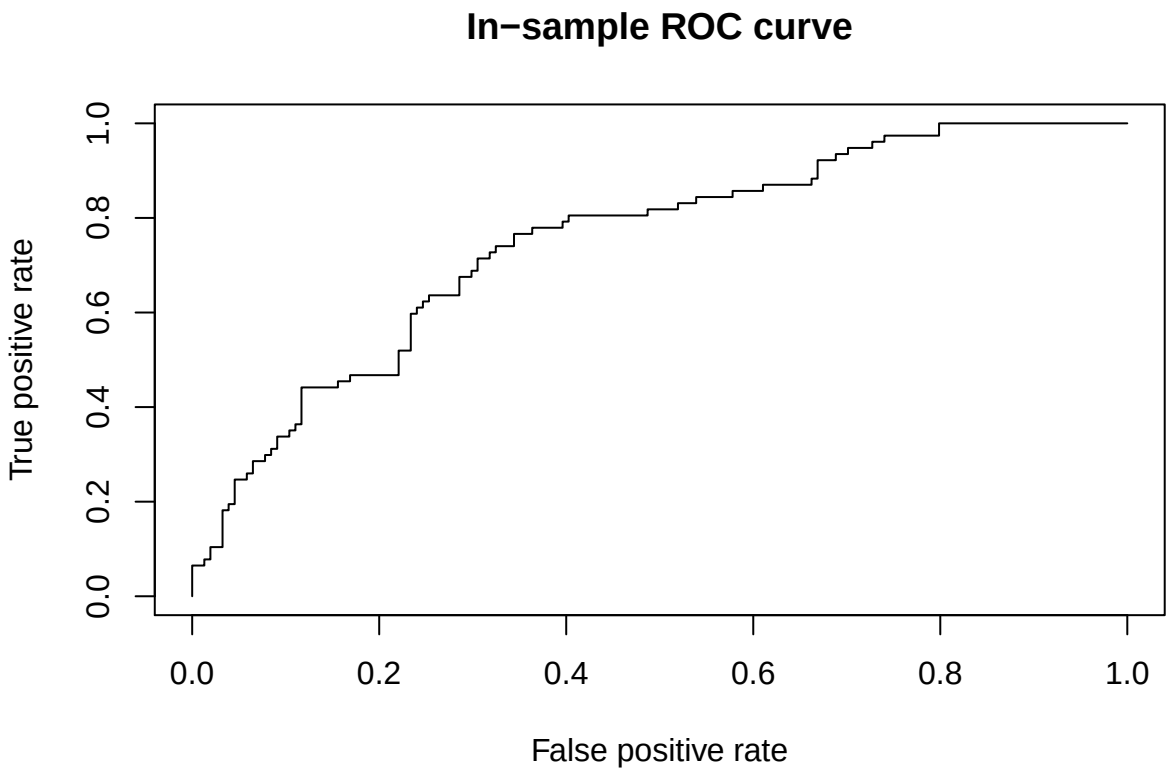



```
nearest.area.train <- unlist(attributes(performance(pamr.pred, "auc"))$y.values)
```

b) Evaluate the performance of the classifiers using ROC curves on the validation set.

For logistic regression with all predictors:

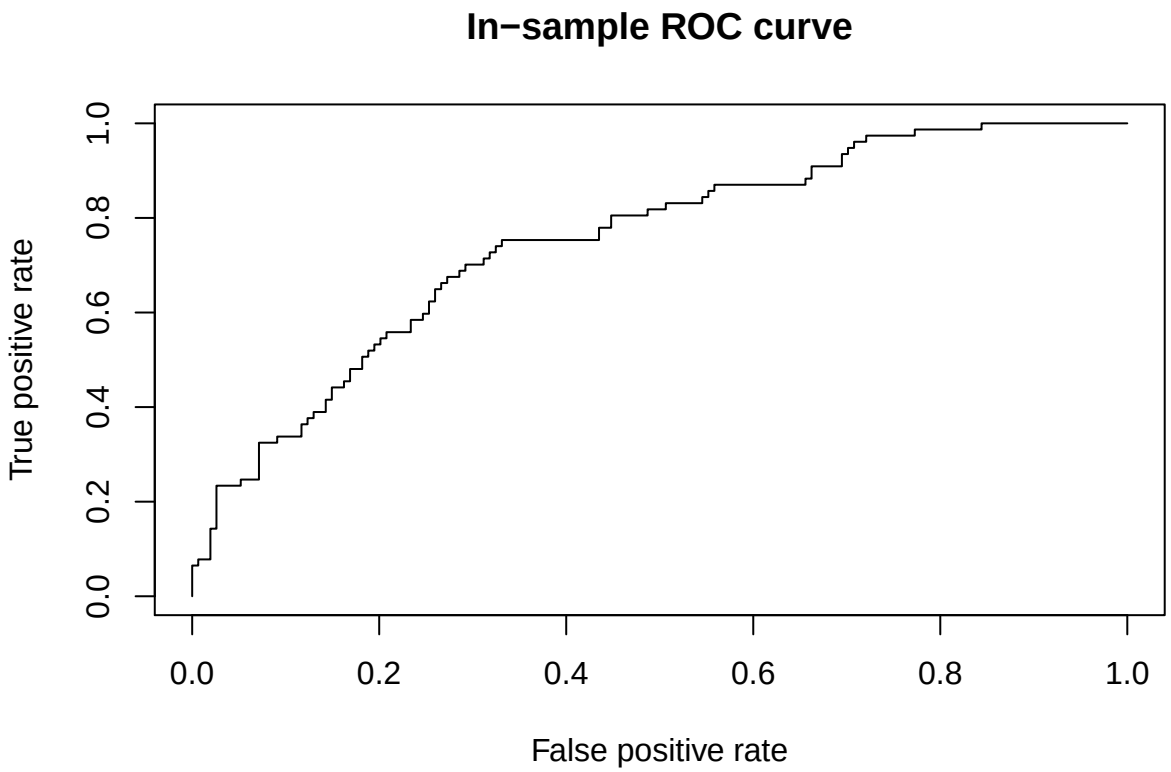
```
library(ROCR)
logit.prob.valid = predict(glm.fit, SAheart_validation[,-10],
                           type="response")
logit.pred.valid = prediction(logit.prob.valid, SAheart_validation[,10])
logit.perf.valid = performance(logit.pred.valid, 'tpr', 'fpr')
plot(logit.perf.valid, colorize=F, main="In-sample ROC curve")
```




```
logit.area.valid <- unlist(attributes(performance(logit.pred.valid, "auc"))$y.values)
```

For logistic regression with subset selection:

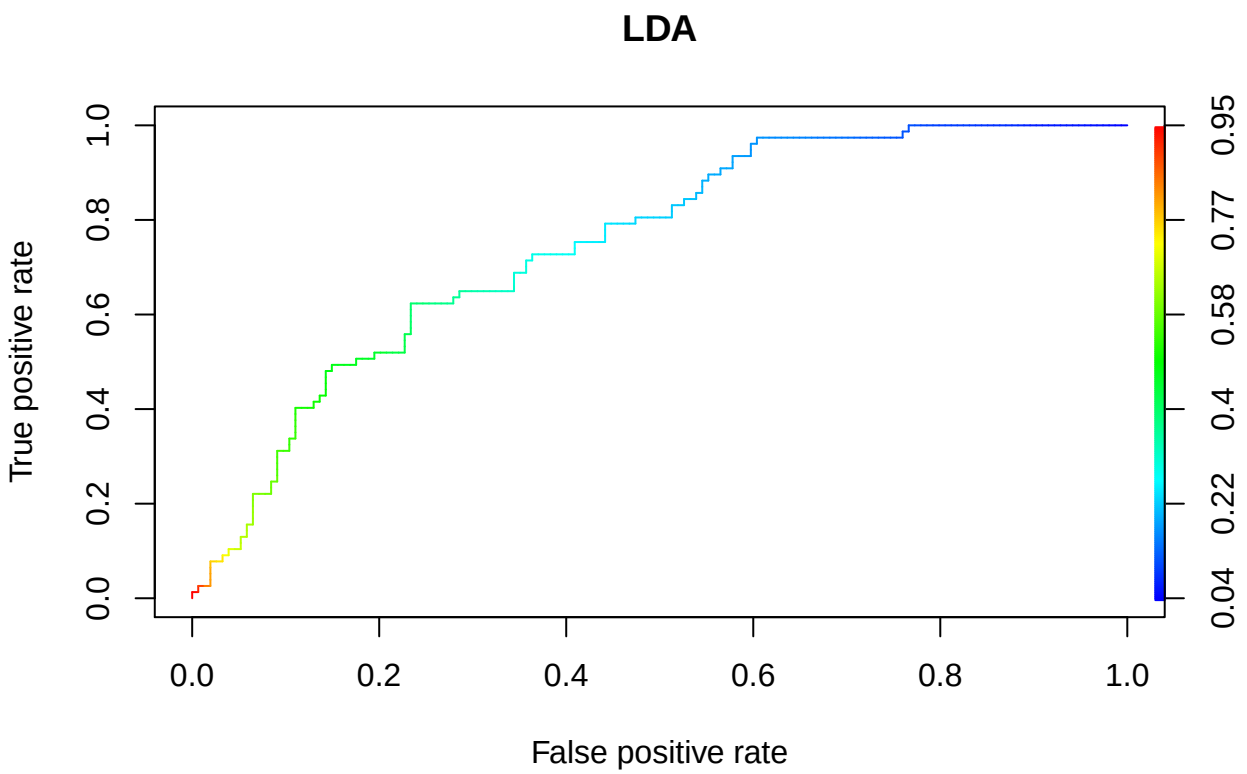
```
logit.best.prob.valid <- predict(glm.fit.best$BestModel, SAheart_validation[,-10],
                                 type="response")
logit.best.pred.valid <- prediction(logit.best.prob.valid, SAheart_validation[,10])
logit.best.perf.valid <- performance(logit.best.pred.valid, 'tpr', 'fpr')
plot(logit.best.perf.valid, colorize=F, main="In-sample ROC curve")
```




```
logit.best.area.valid <- unlist(attributes(performance(logit.best.pred.valid, "auc"))$y.values)
```

For LDA classifier:

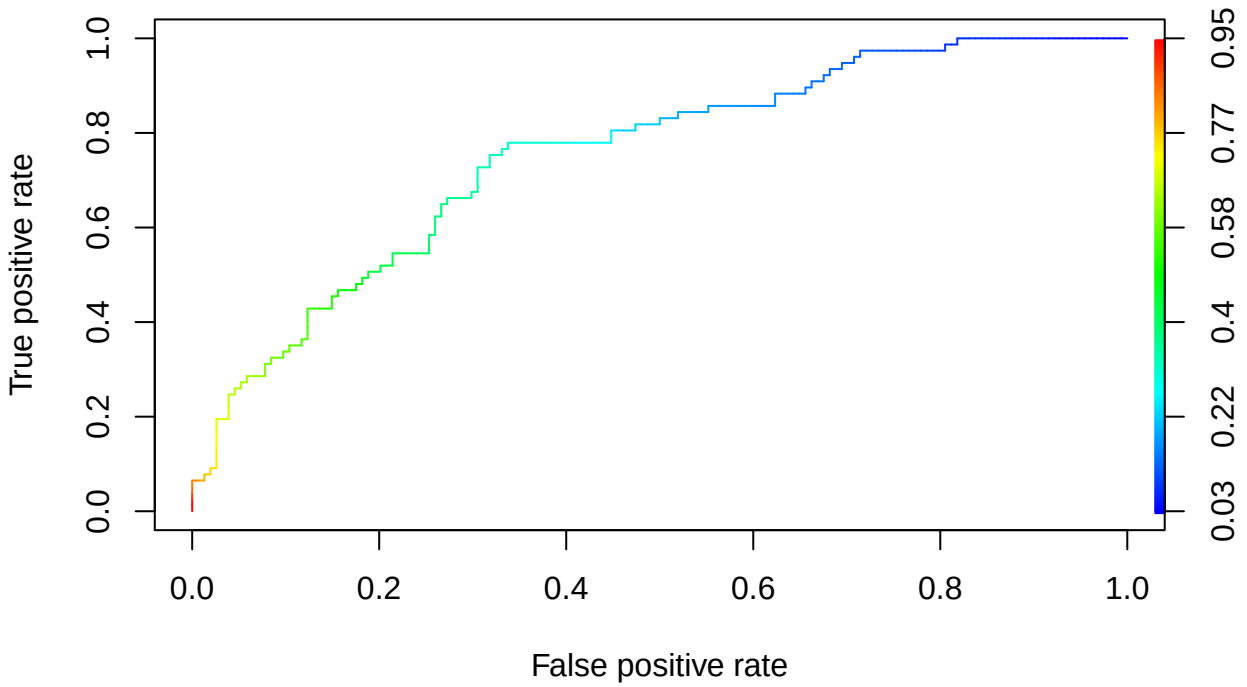
```
lda.prob.valid = predict(lda.fit, SAheart_validation[,-10])$posterior[,2]
lda.pred.valid = prediction(lda.prob.valid, SAheart_validation[,10])
lda.perf.valid = performance(lda.pred.valid, "tpr", "fpr")
plot(lda.perf.valid, colorize=T, main="LDA")
```




```
lda.area.vali <- unlist(attributes(performance(lda.pred.valid, "auc"))$y.values)
```

For Lasso regression:

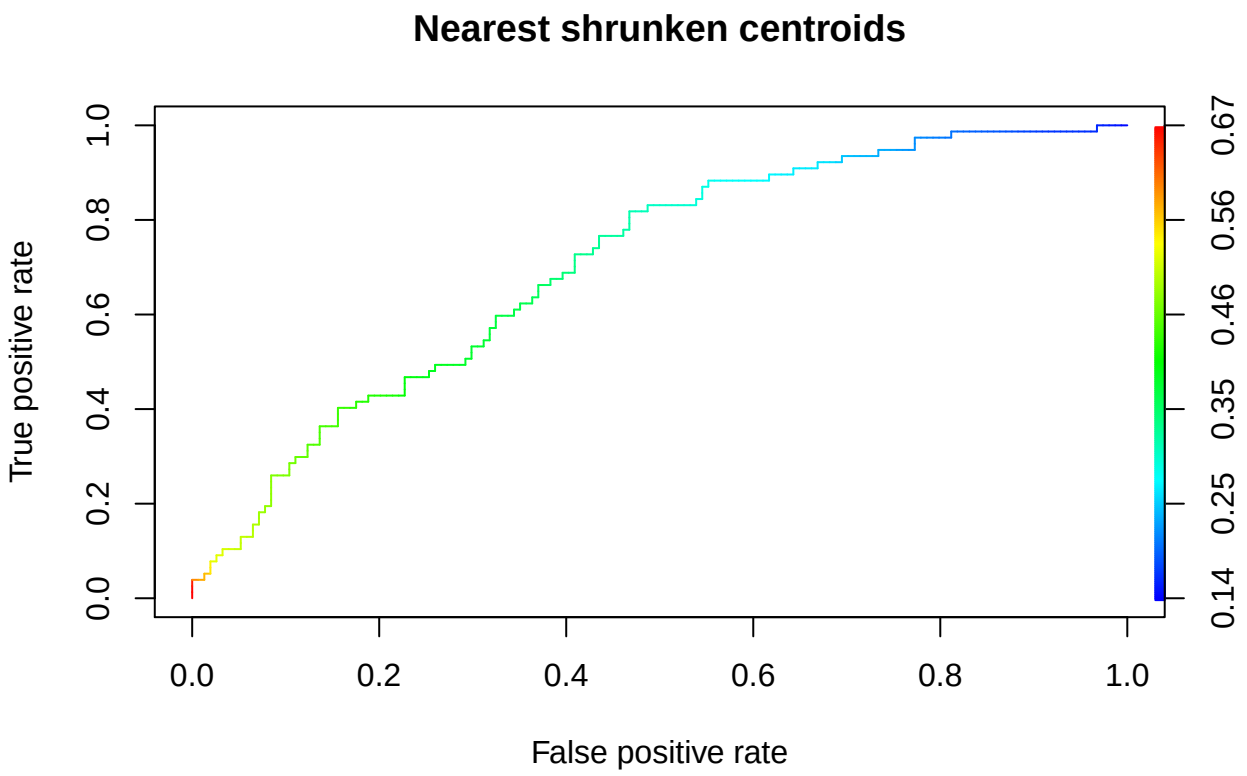
```
glmpath.prob.valid = predict(fit.glmpath,newx=as.matrix(SAheart_validation[,-10]),
                             s=cv.s,mode="norm.fraction",type="response")
glmpath.pred.valid = prediction( predictions=glmpath.prob.valid,
                                 labels=SAheart_validation[,10])
glmpath.perf.valid = performance(glmpath.pred.valid, "tpr", "fpr")
plot(glmpath.perf.valid, colorize=T, main="LASSO")
```




```
lasso.area.valid <- unlist(attributes(performance(glmpath.pred.valid, "auc"))$y.values)
```

For nearest shrunken centroids:

```
pamr.prob.valid = pamr.predict(fit.pamr, newx=pamrValid$x, threshold=0.7, type="posterior")[,2]
pamr.pred.valid = prediction(predictions=pamr.prob.valid, labels= SAheart_validation[,10])
pamr.perf.valid <- performance(pamr.pred.valid, "tpr", "fpr")
plot(pamr.perf.valid, colorize=T, main="Nearest shrunken centroids")
```




```
nearest.area.valid <- unlist(attributes(performance(pamr.pred.valid, "auc"))$y.values)
```

Here is a data frame that shows the areas under the ROC curve for train and test set:

```
##             method train.area validation.area
## 1         logistic  0.8244871       0.7482712
## 2    logistic.best  0.8213937       0.7498735
## 3              LDA  0.7916802       0.7485242
## 4            lasso  0.8244057       0.7515601
## 5 nearest shrunken  0.7587919       0.7039973
```

c) Summarize your findings. How do the results differ between the training and the validation set? Which approach(es) perform(s) better on the validation set? What is are the reasons for this difference in performance? Which models are more interpretable?

Based on the ROC curves seen in part a and b, it is clear that for each model, the area under the curve for validation set is higher than for training set. This is expected, because the model is optimized for prediction in the training set, and therefore will have a higher area under the curve for training set compared to validation set.

For comparing the models, we should consider areas under the ROC curves on the validation set. It can be seen that lasso worked best in this dataset. The reason for this difference is that, each of these models work accurate under some assumptions. For instance, `lda` assumes multivariate Normal distribution, which may not be verified. Depending on the dataset, and the distribution the predictors have, we might end up with the model which works best for that particular dataset.

The number of predictors in logistic best subset selection is less than in other methods and we can see that this method also works better than other methods.

Problem 7

KM problem 4.20

a) GaussI $\leq$ LinLog. Both have logistic (sigmoid) posteriors $p(y|x,w) = \sigma(yw^T x)$, but LinLog is the logistic model which is trained to maximize $p(y|x,w)$. (GaussI may have high joint $p(y,x)$, but this does not necessarily mean $p(y|x)$ is high; LinLog can achieve the maximum of $p(y|x)$, so will necessarily do at least as well as GaussI.)

b) GaussX $\leq$ QuadLog. Both have logistic posteriors with quadratic features, but QuadLog is the model of this class maximizing the average log probabilities.

c) LinLog $\leq$ QuadLog. Logistic regression models with linear features are a subclass of logistic regression models with quadratic functions. The maximum from the superclass is at least as high as the maximum from the subclass.

d) GaussI $\leq$ QuadLog. Follows from above inequalities.

e) Although one might expect that higher log likelihood results in better classification performance, in general, having higher average $logp(y|x)$ does not necessarily translate to higher or lower classification error. For example, consider linearly separable data. We have $L(LinLog) > L(GaussI)$, since maximum likelihood logistic regression will set the weights to infinity, to maximize the probability of the correct labels (hence $p(y_i|x_i,\hat{w}) = 1$ for all i). However, we have $R(linLog) = R(gaussI)$, since the data is linearly separable. (The GaussI model may or may not set σ very small, sesulting in possibly very large class conditional pdfs; however, the posterior over y is a discrete pmf, and can never exceed I.)

As another example, suppose the true label is always 1 (as opposed to 0), but model M always predicts $p(y = 1|x, M) = 0.49$. It will always misclassify, but it is at least close to the decision boundary. By contrast, there might be another model M' that predicts $p(y = 1|x, M') = 1$ on even-numbered inputs, and $p(y = 1|x, M') = 0$ on odd-numbered inputs. Clearly $R(M') = 0.5 < R(M) = 1$, but $L(M') = -inf < L(M) = log(0.49)$.