Module 2: Supervised Learning

Jianhui Gao, Siyue Yang, and Jessica Gronsbell

31/05/2022

```r
# If a package is installed, it will be loaded. If any
## are not, the missing package(s) will be installed
## from CRAN and then loaded.

## First specify the packages of interest
packages <- c(
  "dplyr", "PheCAP", "glmnet", "randomForestSRC", "PheNorm",
  "MAP", "pROC", "mltools", "data.table", "ggplot2", "parallel"
)

## Now load or install&load all
package.check <- lapply(
  packages,
  FUN = function(x) {
    if (!require(x, character.only = TRUE)) {
      install.packages(x, dependencies = TRUE)
      library(x, character.only = TRUE)
    }
  }
)

# load environment from example 1
load("environment.RData")
```

Prepare data for algorithm development

- Split data into training and testing set
- Training 106(60%), Testing 75(40%)

```r
data <- PhecapData(PheCAP::ehr_data, "healthcare_utilization", "label", 75,
  patient_id = "patient_id", seed = 123
)

# Data with non-missing labels
labeled_data <- ehr_data %>% dplyr::filter(!is.na(label))


# All Features
all_x <- ehr_data %>% dplyr::select(
  starts_with("COD"), starts_with("NLP"),
  starts_with("main"), healthcare_utilization
)
```

```r
health_count <- ehr_data$healthcare_utilization

# Training Set
train_data <- ehr_data %>% dplyr::filter(patient_id %in% data$training_set)
train_x <- train_data %>%
  dplyr::select(
    starts_with("COD"), starts_with("NLP"),
    starts_with("main"), healthcare_utilization
  ) %>%
  as.matrix()
train_y <- train_data %>%
  dplyr::select(label) %>%
  pull()

# Testing Set
test_data <- ehr_data %>% dplyr::filter(patient_id %in% data$validation_set)
test_x <- test_data %>%
  dplyr::select(
    starts_with("COD"), starts_with("NLP"),
    starts_with("main"), healthcare_utilization
  ) %>%
  as.matrix()
test_y <- test_data %>%
  dplyr::select(label) %>%
  pull()
```

Penalized logistic regression

- Fit LASSO and Adaptive LASSO(ALASSO)

```r
# Choose best lambda using CV
beta.lasso <- lasso_fit(x = train_x, y = train_y,
                        tuning = "cv", family = "binomial")
```

```r
# Features Selected
names(beta.lasso[abs(beta.lasso)>0])[-1]
```

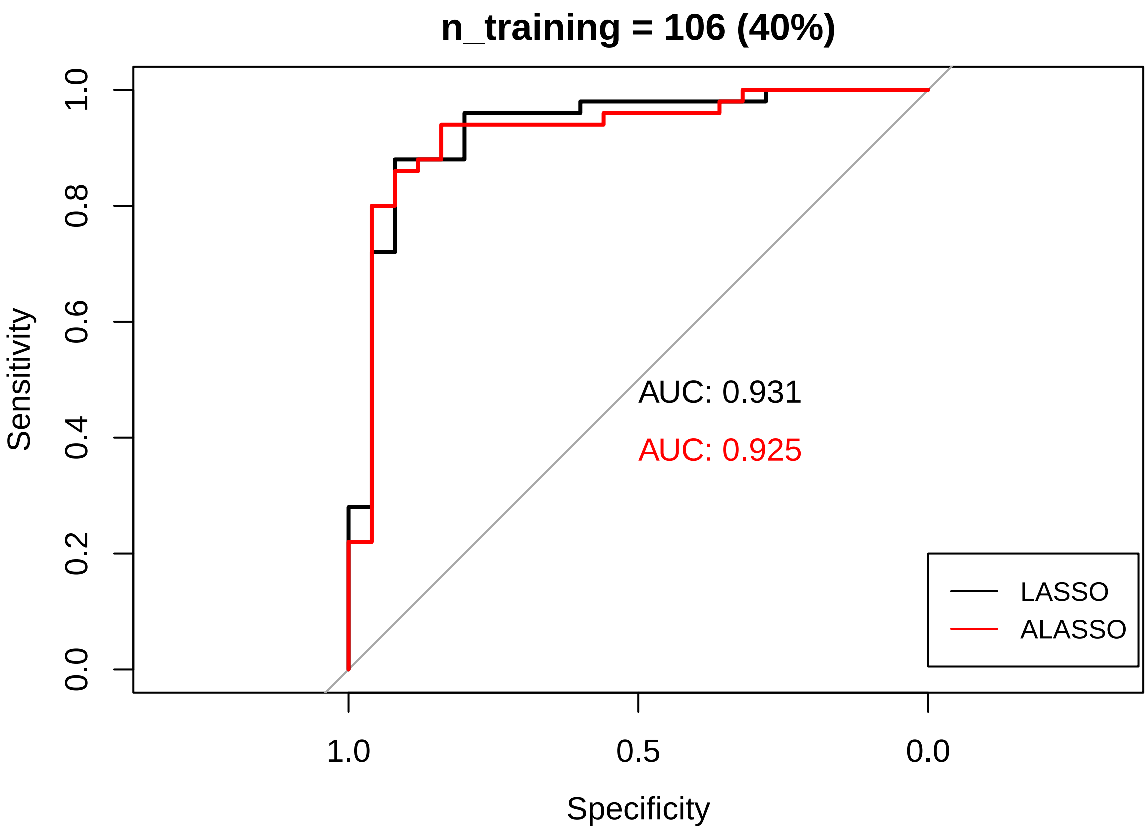
```
## [1] "NLP93"                  "NLP104"                 "NLP304"
## [4] "main_NLP"               "main_ICDNLP"            "healthcare_utilization"
```

```r
# prediction on testing set
y_hat.lasso <- linear_model_predict(beta = beta.lasso, x = test_x,
                                    probability = TRUE)
```

```r
# Fit Adaptive LASSO
beta.alasso <- adaptive_lasso_fit(x = train_x, y = train_y,
                                  tuning = "cv", family = "binomial")
y_hat.alasso <- linear_model_predict(beta = beta.alasso, x = test_x,
                                    probability = TRUE)



# Features Selected
names(beta.alasso[abs(beta.alasso)>0])[-1]
```

```
## [1] "NLP304"                 "main_NLP"               "healthcare_utilization"
```

```r
roc.lasso <- roc(test_y, y_hat.lasso)
roc.alasso <- roc(test_y, y_hat.alasso)

plot(roc.lasso,
  print.auc = TRUE, main = "n_training = 106 (40%)"
)
plot(roc.alasso,
  print.auc = TRUE, col = 'red', add = TRUE, print.auc.y = 0.4
)
legend(0, 0.2, legend = c("LASSO", "ALASSO"), col = c("black","red"),
       lty = 1, cex = 0.8)
```




```r
roc_full.lasso <- get_roc(y_true = test_y, y_score = y_hat.lasso)
head(roc_full.lasso,10)
```

```
##          cutoff    pos.rate  FPR       TPR       PPV       NPV        F1
##  [1,] 0.9468465 0.006666667 0.00 0.1306667 1.0000000 0.3651412 0.2311321
##  [2,] 0.9140868 0.066666667 0.00 0.2053333 1.0000000 0.3861998 0.3407080
##  [3,] 0.8813270 0.193333333 0.02 0.2800000 0.9655172 0.4049587 0.4341085
##  [4,] 0.8781599 0.200000000 0.04 0.3900000 0.9512195 0.4403670 0.5531915
##  [5,] 0.8749928 0.206666667 0.04 0.5000000 0.9615385 0.4897959 0.6578947
##  [6,] 0.8130579 0.413333333 0.04 0.6100000 0.9682540 0.5517241 0.7484663
##  [7,] 0.7511229 0.500000000 0.06 0.7200000 0.9600000 0.6266667 0.8228571
##  [8,] 0.7445305 0.506666667 0.08 0.7600000 0.9500000 0.6571429 0.8444444
##  [9,] 0.7379380 0.513333333 0.08 0.8000000 0.9523810 0.6969697 0.8695652
## [10,] 0.6905611 0.586666667 0.08 0.8400000 0.9545455 0.7419355 0.8936170
```

```
roc_full.alasso <- get_roc(y_true = test_y, y_score = y_hat.alasso)
head(roc_full.lasso,10)
```

```
##          cutoff    pos.rate  FPR       TPR       PPV       NPV        F1
##  [1,] 0.9468465 0.006666667 0.00 0.1306667 1.0000000 0.3651412 0.2311321
##  [2,] 0.9140868 0.066666667 0.00 0.2053333 1.0000000 0.3861998 0.3407080
##  [3,] 0.8813270 0.193333333 0.02 0.2800000 0.9655172 0.4049587 0.4341085
##  [4,] 0.8781599 0.200000000 0.04 0.3900000 0.9512195 0.4403670 0.5531915
##  [5,] 0.8749928 0.206666667 0.04 0.5000000 0.9615385 0.4897959 0.6578947
##  [6,] 0.8130579 0.413333333 0.04 0.6100000 0.9682540 0.5517241 0.7484663
##  [7,] 0.7511229 0.500000000 0.06 0.7200000 0.9600000 0.6266667 0.8228571
##  [8,] 0.7445305 0.506666667 0.08 0.7600000 0.9500000 0.6571429 0.8444444
##  [9,] 0.7379380 0.513333333 0.08 0.8000000 0.9523810 0.6969697 0.8695652
## [10,] 0.6905611 0.586666667 0.08 0.8400000 0.9545455 0.7419355 0.8936170
```

Different train size

- randomly sample traning size = 50, 70, 90
- rest as testing set
- repeat 600 times

```
start<- Sys.time()
auc_supervised <- validate_supervised(dat = labeled_data, nsim = 600,
                                      n.train = c(50, 70, 90))
end <- Sys.time()
end - start
```

```
## Time difference of 3.296876 mins
```

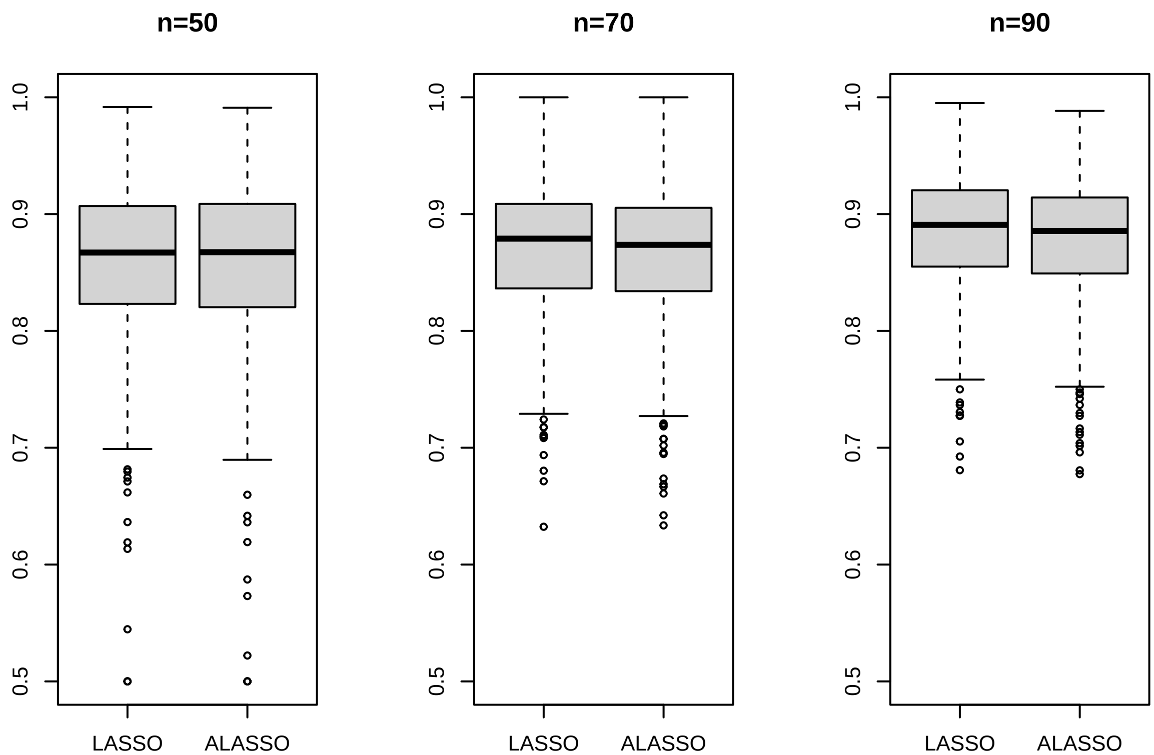
```
# median AUC
apply(auc_supervised, 2, median)
```

```
##  n=50,LASSO  n=70,LASSO  n=90,LASSO n=50,ALASSO n=70,ALASSO n=90,ALASSO
##   0.8670982   0.8789683   0.8907670   0.8673935   0.8736602   0.8855655
```

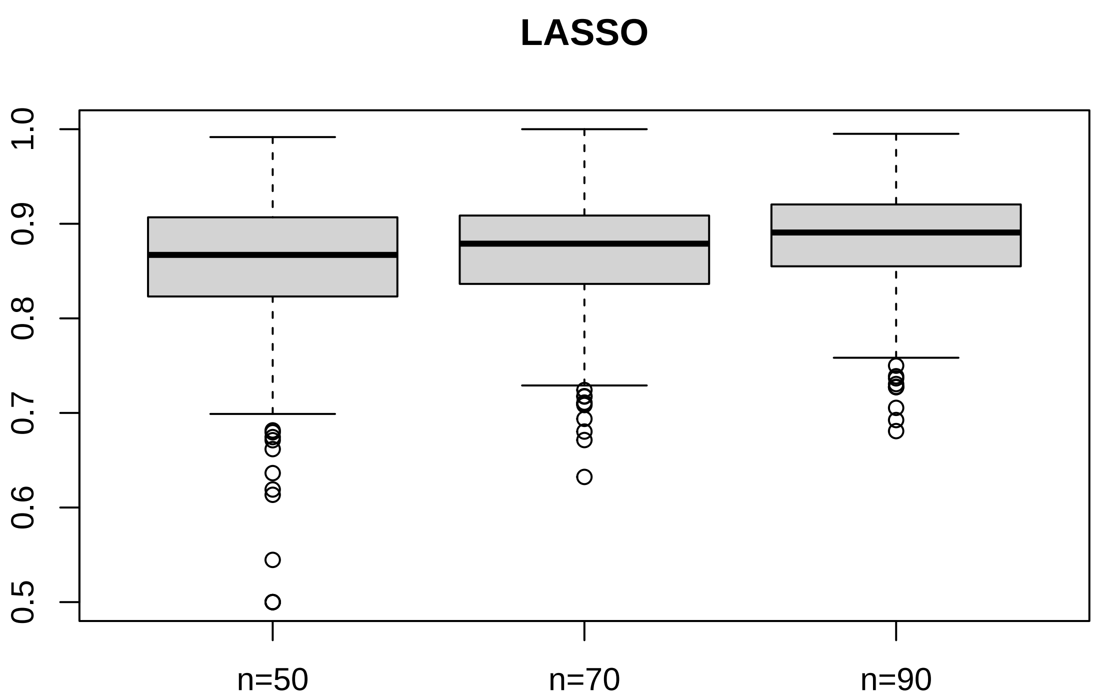
```
# se
apply(auc_supervised, 2, sd)
```

```
##  n=50,LASSO  n=70,LASSO  n=90,LASSO n=50,ALASSO n=70,ALASSO n=90,ALASSO
##  0.07197811  0.05588511  0.05184181  0.07300341  0.05871336  0.05415953
```

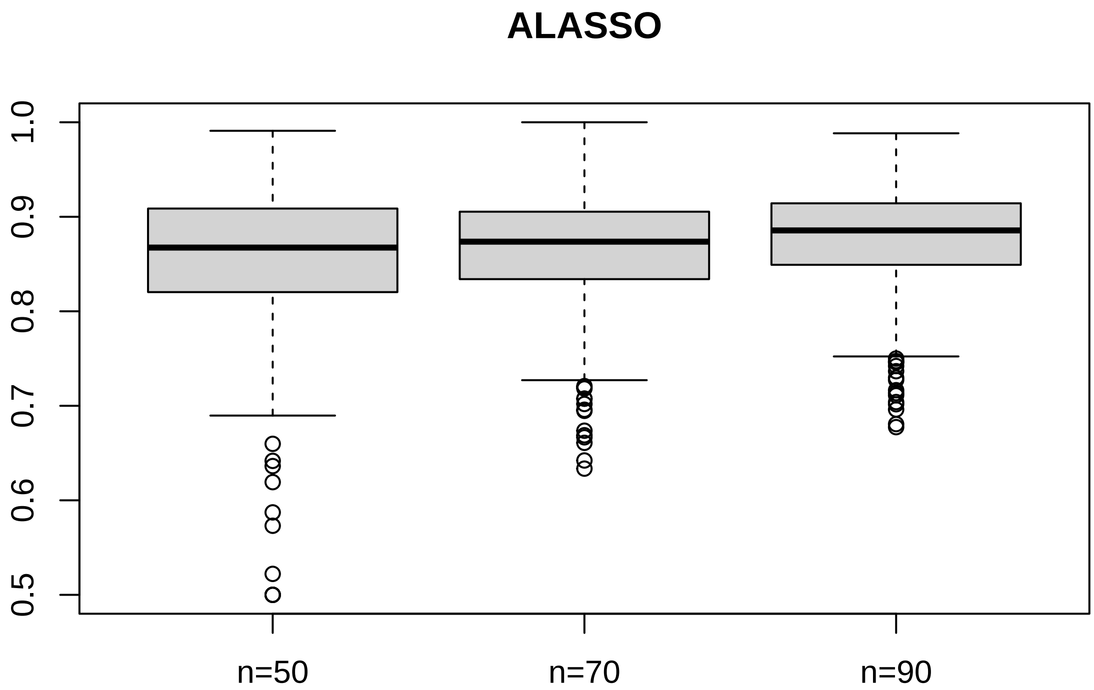
```
par(mfrow =c(1,3))
boxplot(auc_supervised %>% select(starts_with("n=50")) , ylim = c(0.5, 1),
        names = c("LASSO", "ALASSO"), main = "n=50")
boxplot(auc_supervised %>% select(starts_with("n=70")) , ylim = c(0.5, 1),
        names = c("LASSO", "ALASSO"), main = "n=70")
boxplot(auc_supervised %>% select(starts_with("n=90")) , ylim = c(0.5, 1),
        names = c("LASSO", "ALASSO"), main = "n=90")
```




```r
boxplot(auc_supervised[,1:3], ylim = c(0.5, 1),
        names = c("n=50", "n=70", "n=90"), main = "LASSO")
```




```r
boxplot(auc_supervised[,4:6], ylim = c(0.5, 1),
        names = c("n=50", "n=70", "n=90"), main = "ALASSO")
```

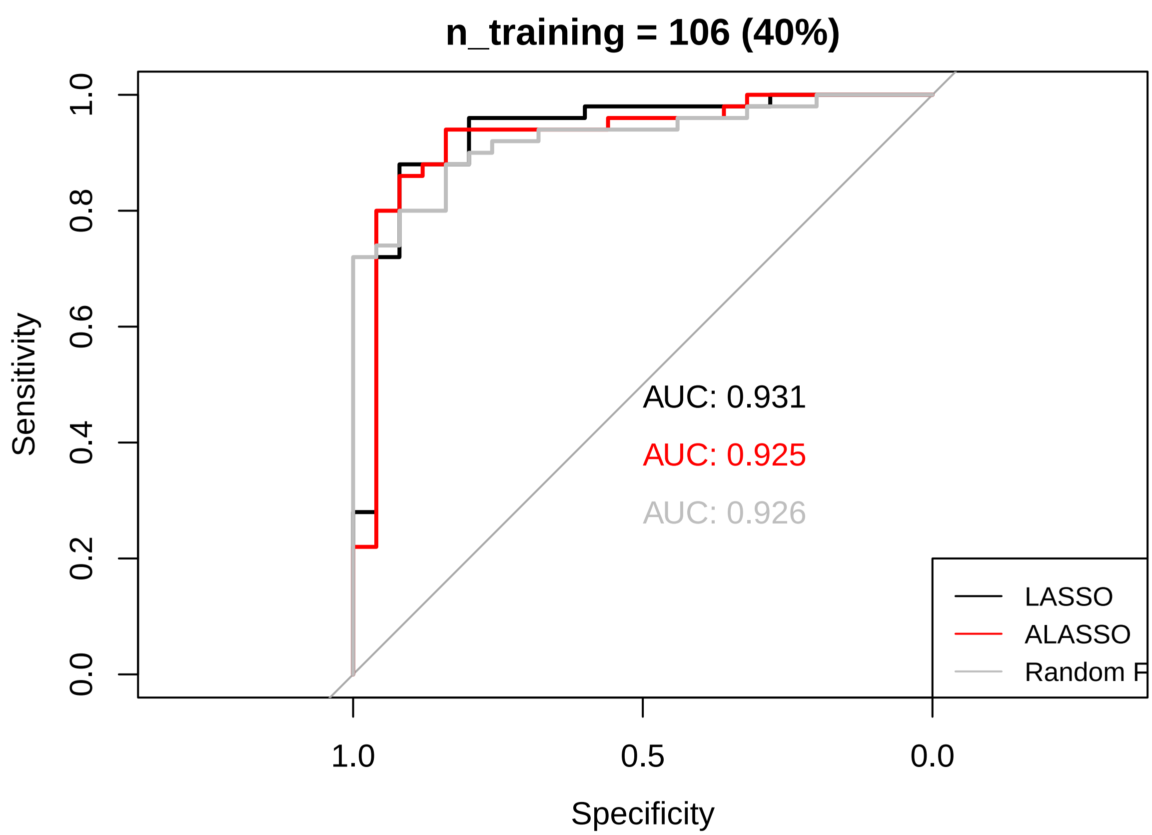


Appendix

Random Forest

```
model_rf <- rfsrc(y ~., data = data.frame(y =train_y, x = train_x))
y_hat.rf <- predict(model_rf, newdata=data.frame(x = test_x))$predicted
```

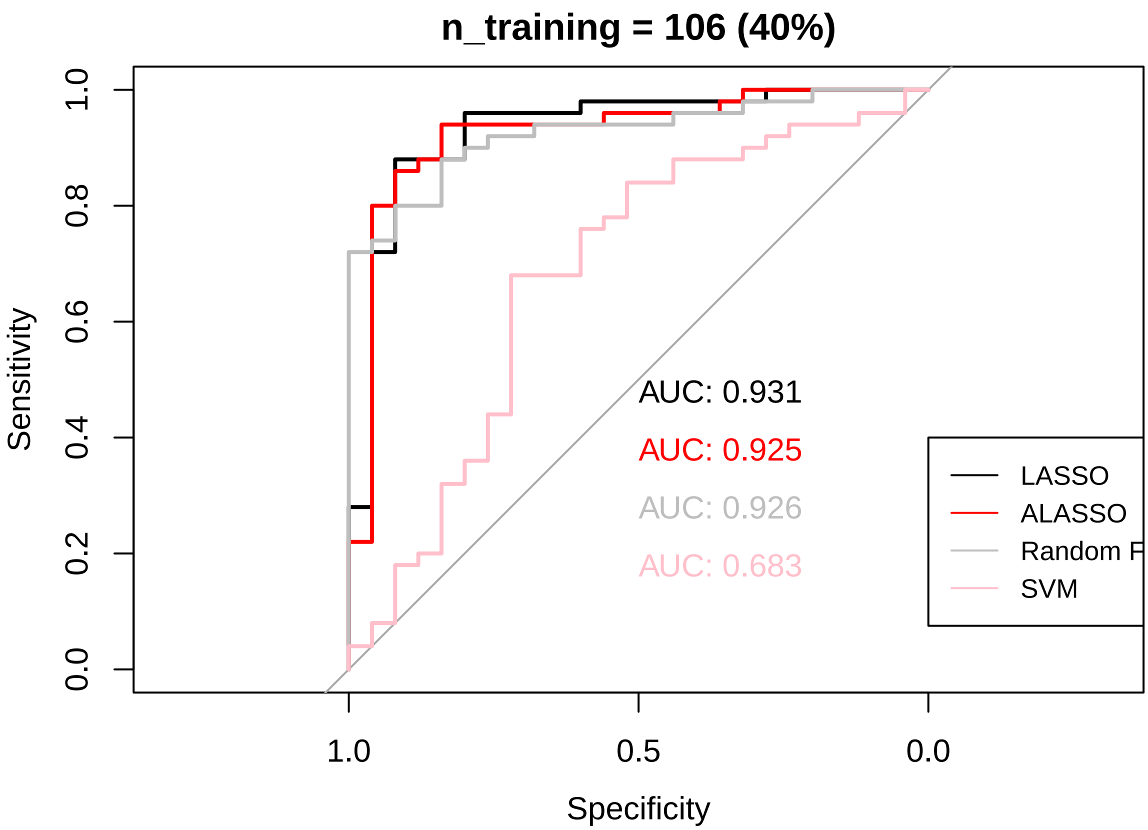
```
roc.rf <- roc(test_y, y_hat.rf)

plot(roc.lasso,
  print.auc = TRUE, main = "n_training = 106 (40%)"
)
plot(roc.alasso,
  print.auc = TRUE, col = 'red', add = TRUE, print.auc.y = 0.4
)
plot(roc.rf,
  print.auc = TRUE, col = 'grey', add = TRUE, print.auc.y = 0.3
)
legend(0, 0.2, legend = c("LASSO", "ALASSO","Random Forest"), col = c("black","red", "grey"),
       lty = 1, cex = 0.8)
```



SVM

```
model_svm <- fit_svm(x = train_x, y = train_y)
y_hat.svm <- predict_svm(model_svm, test_x)
```

```
roc.svm <- roc(test_y, y_hat.svm)

plot(roc.lasso,
  print.auc = TRUE, main = "n_training = 106 (40%)"
)
plot(roc.alasso,
  print.auc = TRUE, col = 'red', add = TRUE, print.auc.y = 0.4
)
plot(roc.rf,
  print.auc = TRUE, col = 'grey', add = TRUE, print.auc.y = 0.3
)
plot(roc.svm,
  print.auc = TRUE, col = 'pink', add = TRUE, print.auc.y = 0.2
)
legend(0, 0.4, legend = c("LASSO", "ALASSO","Random Forest", "SVM"),
       col = c("black","red", "grey", "pink"),
       lty = 1, cex = 0.8)
```


n_training = 106 (40%)
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
Specificity
1.0
0.5
0.0
AUC: 0.931
AUC: 0.925
AUC: 0.926
AUC: 0.683
LASSO
ALASSO
Random F
SVM

Validation