Module 3: Semi-supervised learning (PheCAP)

Siyue Yang, Jianhui Gao, and Jessica Gronsbell

```
# Load the packages.
packages <- c("tidyverse", "PheCAP", "glmnet", "glmpath", "pROC", "parallel")

# Check if the pacakages are missing or not.
# If missing, install automatically.
# If not missing, load the package.
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.
load("environment_phecap.RData")
```

```
# Load helper functions.
source("../Rscripts/helper_function.R")
```

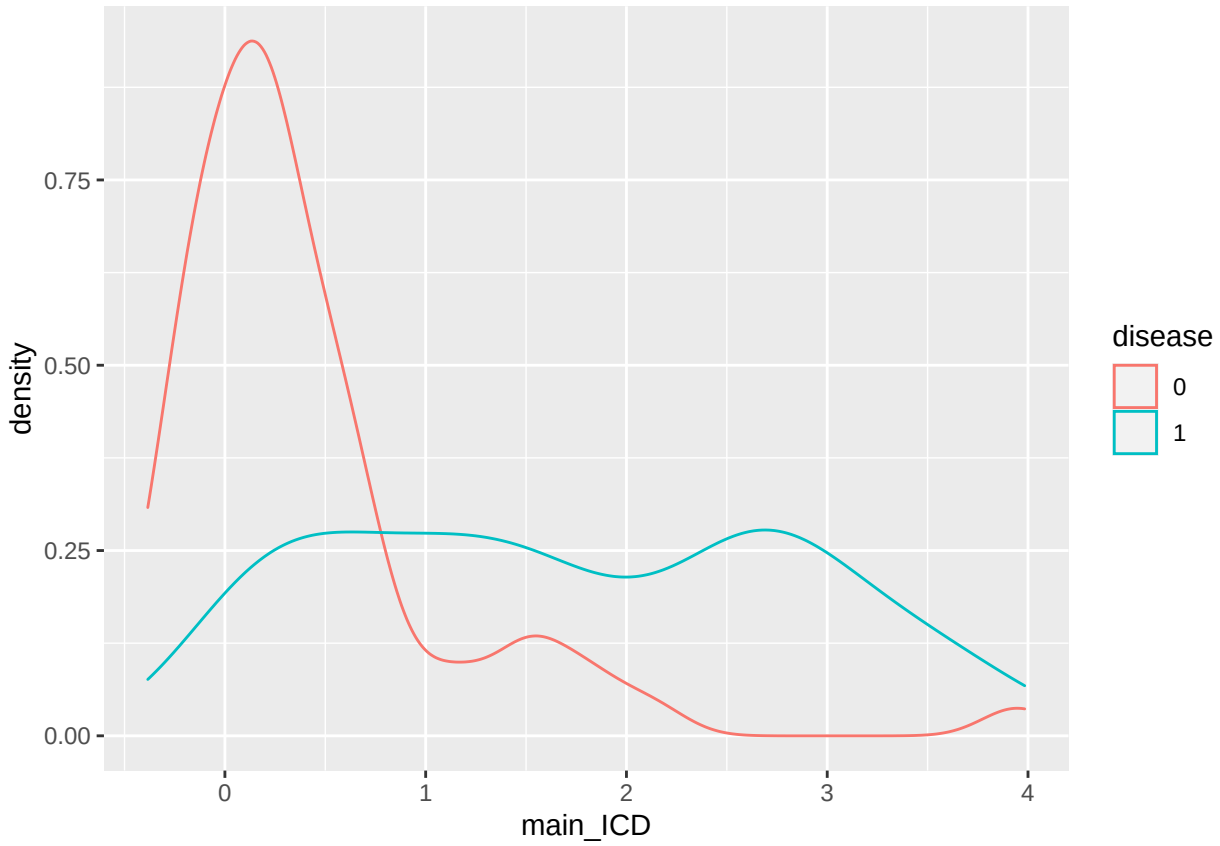
Feature selection

How to select features?

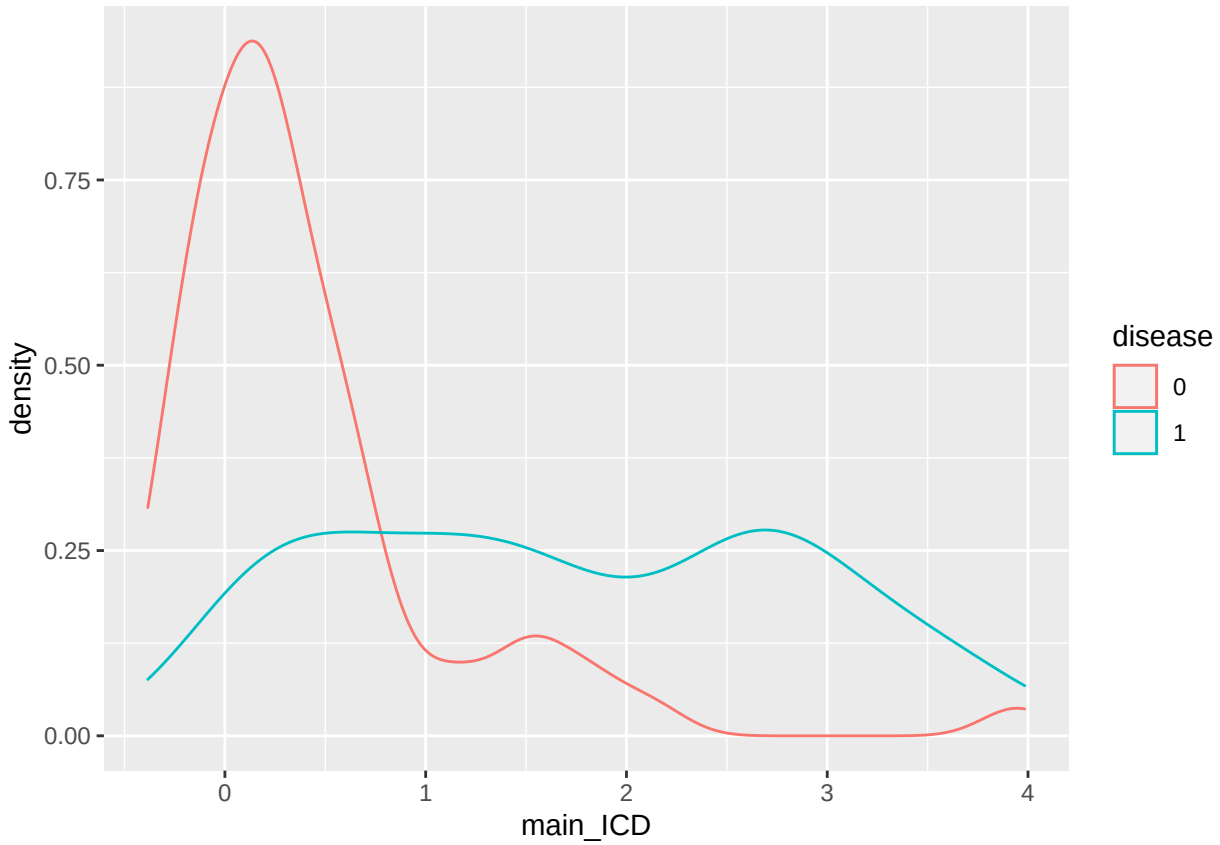
Can leverage some clinical-meaningful features that are related to Y.

e.g. Feature "main_ICD" = the total number of the disease-related billing codes.

```
ehr_data %>%
  filter(!is.na(label)) %>%
  mutate(disease = factor(label)) %>%
  ggplot(aes(x = main_ICD)) +
  geom_density(aes(color = disease))
```

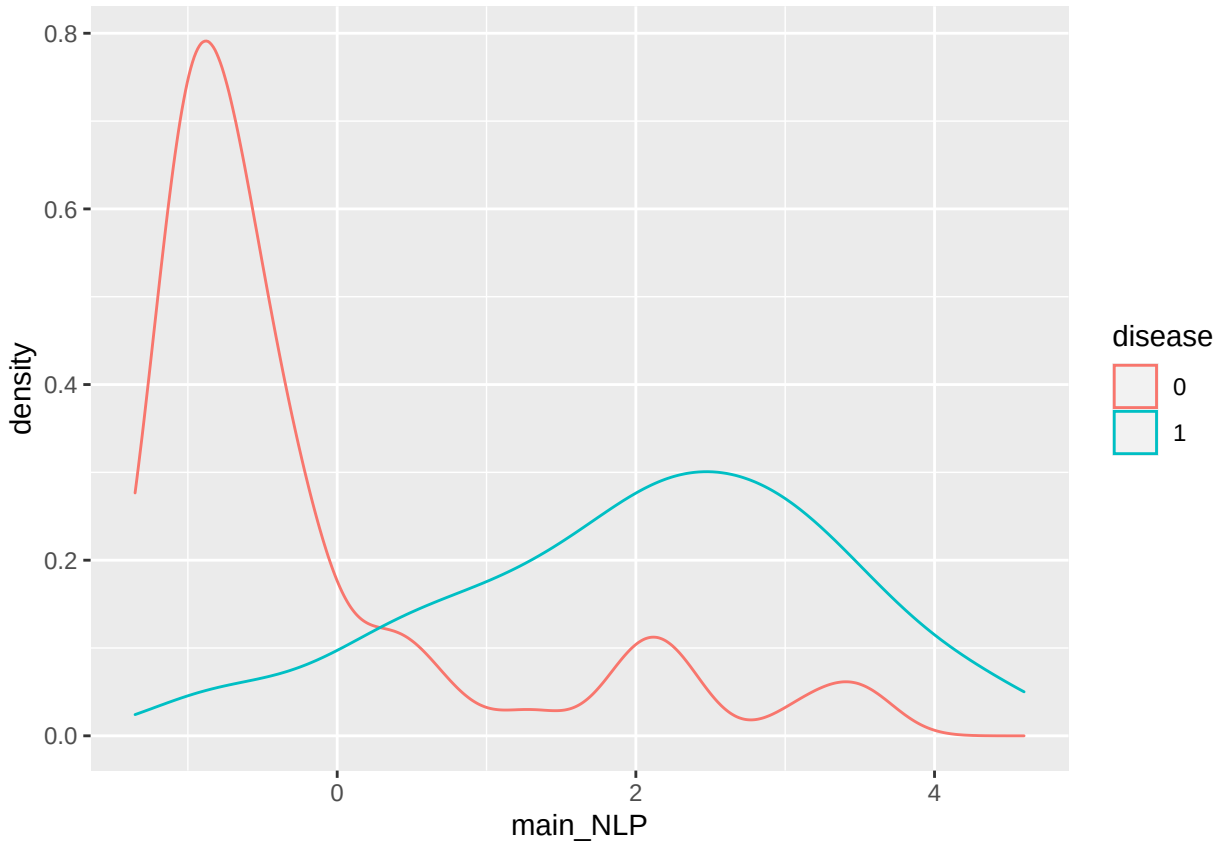



```
# With log transformation.
ehr_data %>%
  filter(!is.na(label)) %>%
  mutate(disease = factor(label)) %>%
  ggplot(aes(x = main_ICD)) +
  geom_density(aes(color = disease))
```



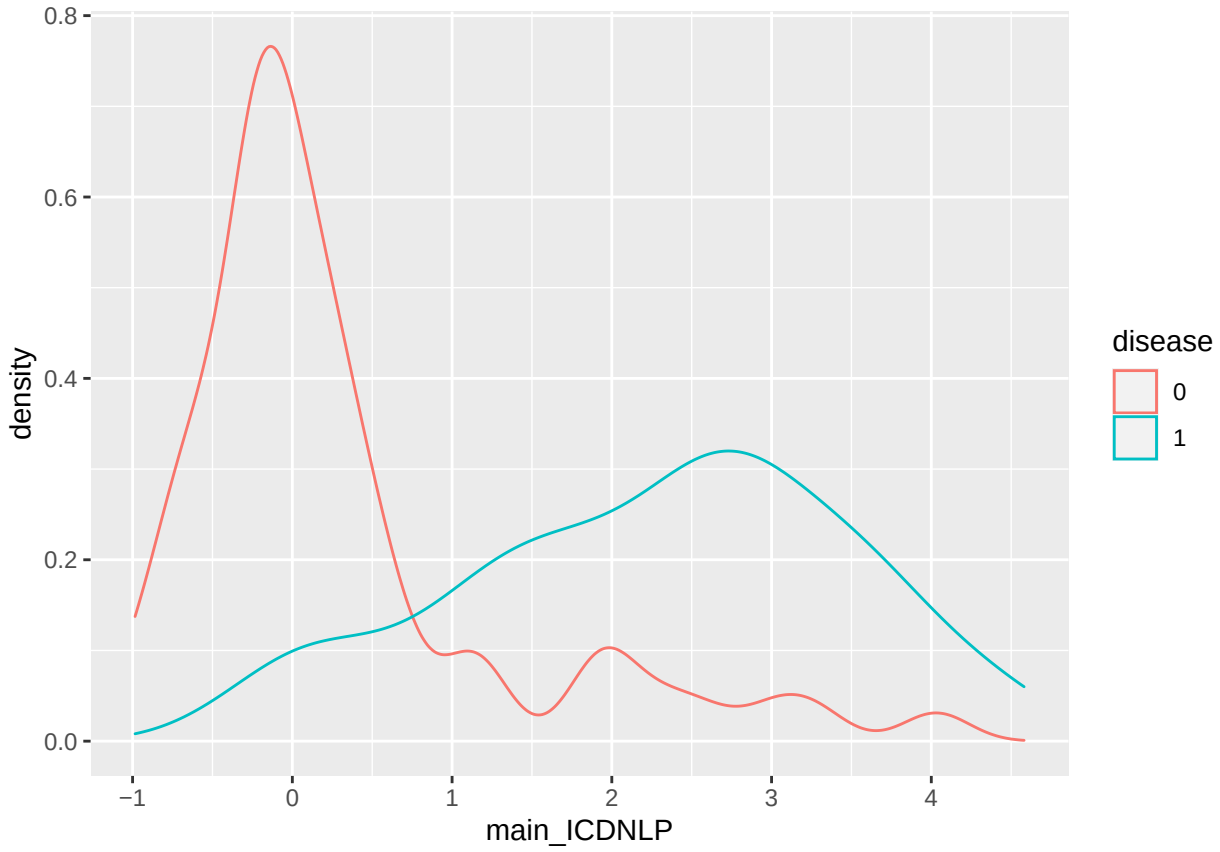
The more the disease-related codes, the more **likely** the patient has the disease.

```
ehr_data %>%
  filter(!is.na(label)) %>%
  mutate(disease = factor(label)) %>%
  ggplot(aes(x = main_NLP)) +
  geom_density(aes(color = disease))
```



Other options? - Combine them.

```
ehr_data %>%
  filter(!is.na(label)) %>%
  mutate(disease = factor(label)) %>%
  ggplot(aes(x = main_ICDNLP)) +
  geom_density(aes(color = disease))
```




```
nonmissing_index <- which(!is.na(ehr_data$label))
y <- ehr_data$label[nonmissing_index]
sicd <- ehr_data$main_ICD
snlp <- ehr_data$main_NLP
sicdnlp <- ehr_data$main_ICDNLP

get_auc(y, sicd[nonmissing_index])
```

```
## [1] 0.8394551
```

```
get_auc(y, snlp[nonmissing_index])
```

```
## [1] 0.8841149
```

```
get_auc(y, sicdnlp[nonmissing_index])
```

```
## [1] 0.8875034
```

We call these highly predictive features of the true disease status "surrogates".

Opportunities of using surrogate features

1. Feature selection to reduce p

2. Algorithm development with limited Y
3. Algorithm validation with limited Y

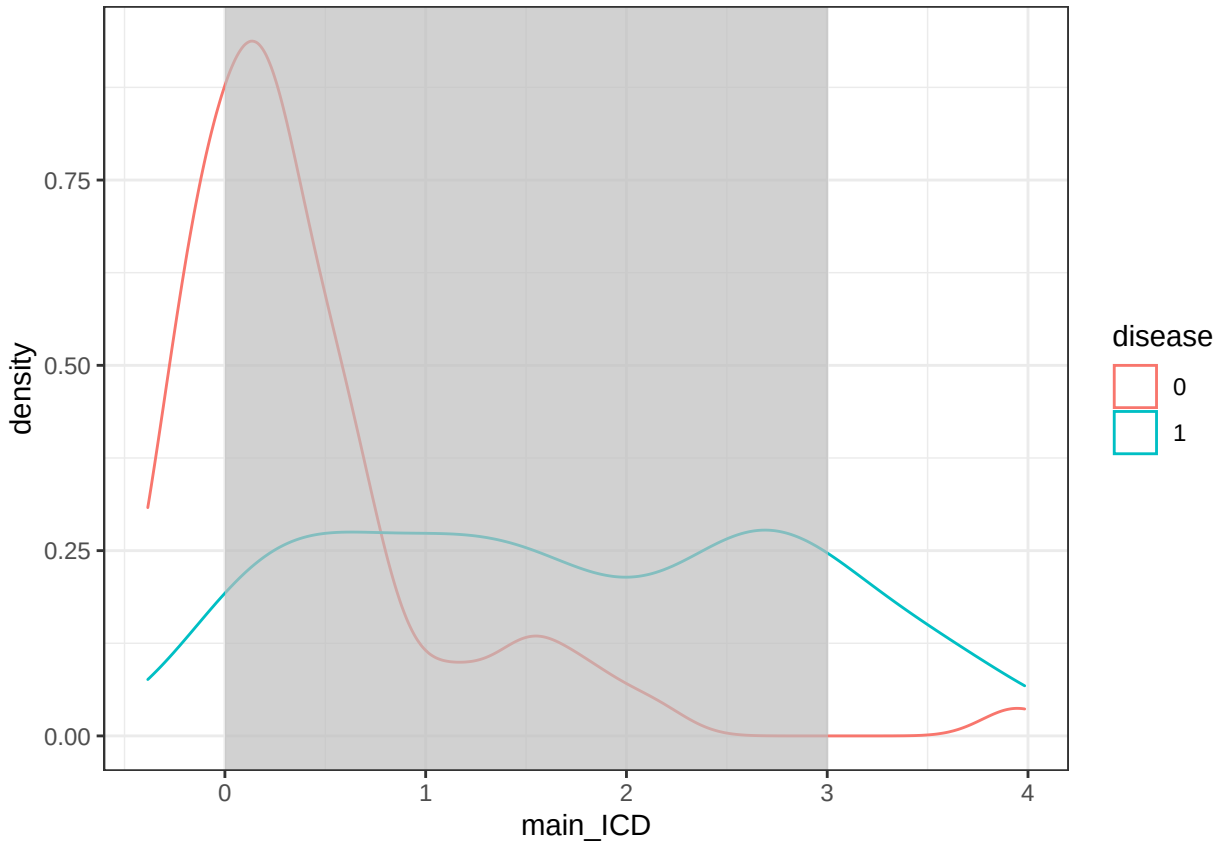
Opportunity 2 and 3 will be covered in the next module!

Feature selection method

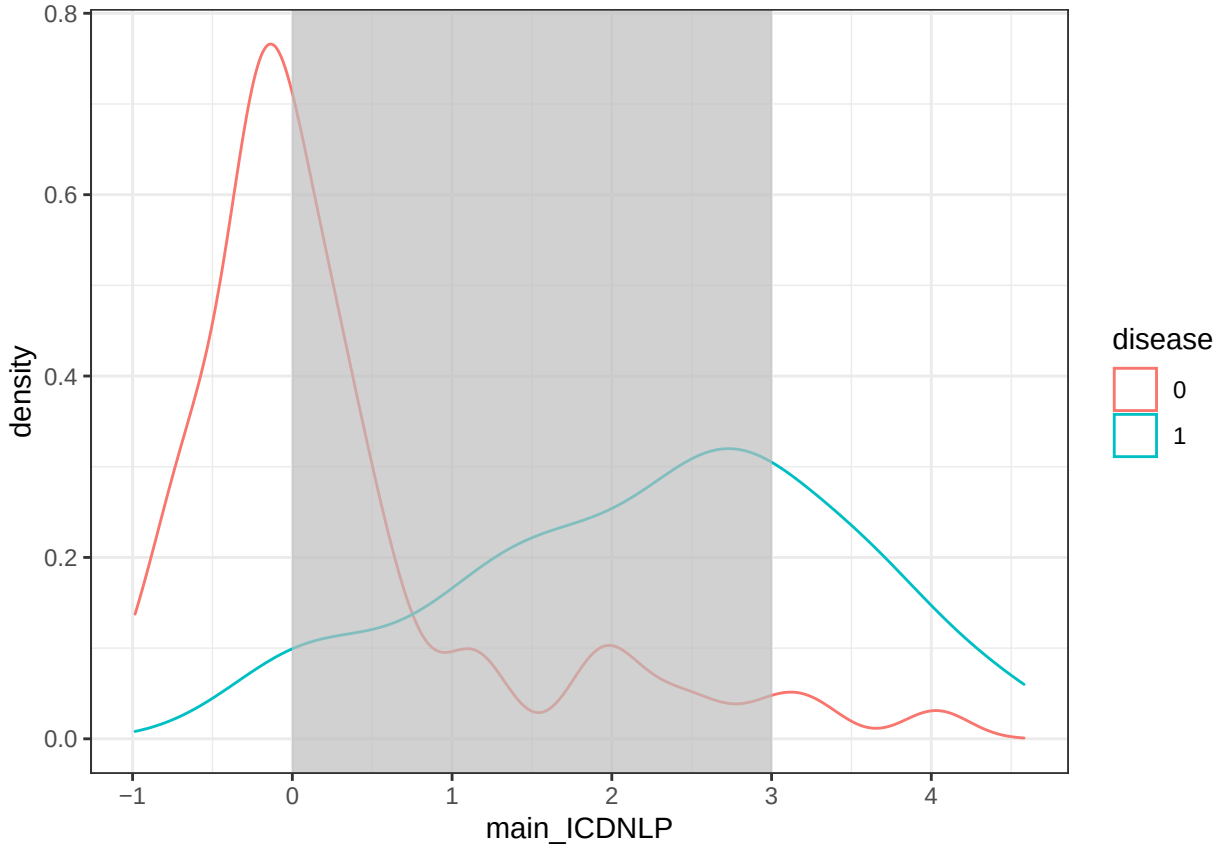
Motivation (Extreme assumption):

- Patients with **high** main ICD or NLP mentions generally have the phenotype.
- Patients with **extremely** low counts are unlikely to have the phenotype.

```
ehr_data %>%
  filter(!is.na(label)) %>%
  mutate(disease = factor(label)) %>%
  ggplot(aes(x = main_ICD)) +
  geom_density(aes(color = disease)) +
  theme_bw() +
  annotate("rect", fill = "grey", alpha = 0.7, xmin = 0, xmax = 3,
           ymin = -Inf, ymax = Inf)
```



```r
ehr_data %>%
  filter(!is.na(label)) %>%
  mutate(disease = factor(label)) %>%
  ggplot(aes(x = main_ICDNLP)) +
  geom_density(aes(color = disease)) +
  theme_bw() +
  annotate("rect", fill = "grey", alpha = 0.7, xmin = 0, xmax = 3,
           ymin = -Inf, ymax = Inf)
```



- Left white rect: patients not having the disease.
- Right white rect: patients having the disease.

Prepare data for feature selection

Prepare surrogates

Surrogates are available for all the patients!

```r
# Prepare 3 surrogates.
sicd <- ehr_data$main_ICD
snlp <- ehr_data$main_NLP
sicdnlp <- ehr_data$main_ICDNLP

# Prepare features to be selected.
x <- data.matrix(ehr_data %>% select(starts_with("COD") | starts_with("NLP")))
```

Run surrogate-assisted feature extraction (SAFE) and show result.

```
# Truncated using 3 and 1.
SAFE_icd <- extreme_method(sicd, x, u_bound = 3, l_bound = 0)
SAFE_nlp <- extreme_method(snlp, x, u_bound = 3, l_bound = 0)
SAFE_both <- extreme_method(sicdnlp, x, u_bound = 3, l_bound = 0)

# Majority voting.
beta <- rbind(SAFE_icd$beta_all, SAFE_nlp$beta_all, SAFE_both$beta_all)
SAFE_select <- which(colMeans(beta, na.rm = T) >= 0.5)
SAFE_feature <- colnames(x)[SAFE_select]
SAFE_feature
```

```
##  [1] "NLP6"   "NLP56"  "NLP93"  "NLP160" "NLP161" "NLP231" "NLP306" "NLP309" "NLP321" "NLP349"
## [11] "NLP403" "NLP434" "NLP446" "NLP495"
```

We select features that occur 50% among the three different surrogate-selected feature sets. This is the idea of *majority voting*.

Train phenotyping model and show the AUC on the testing set.

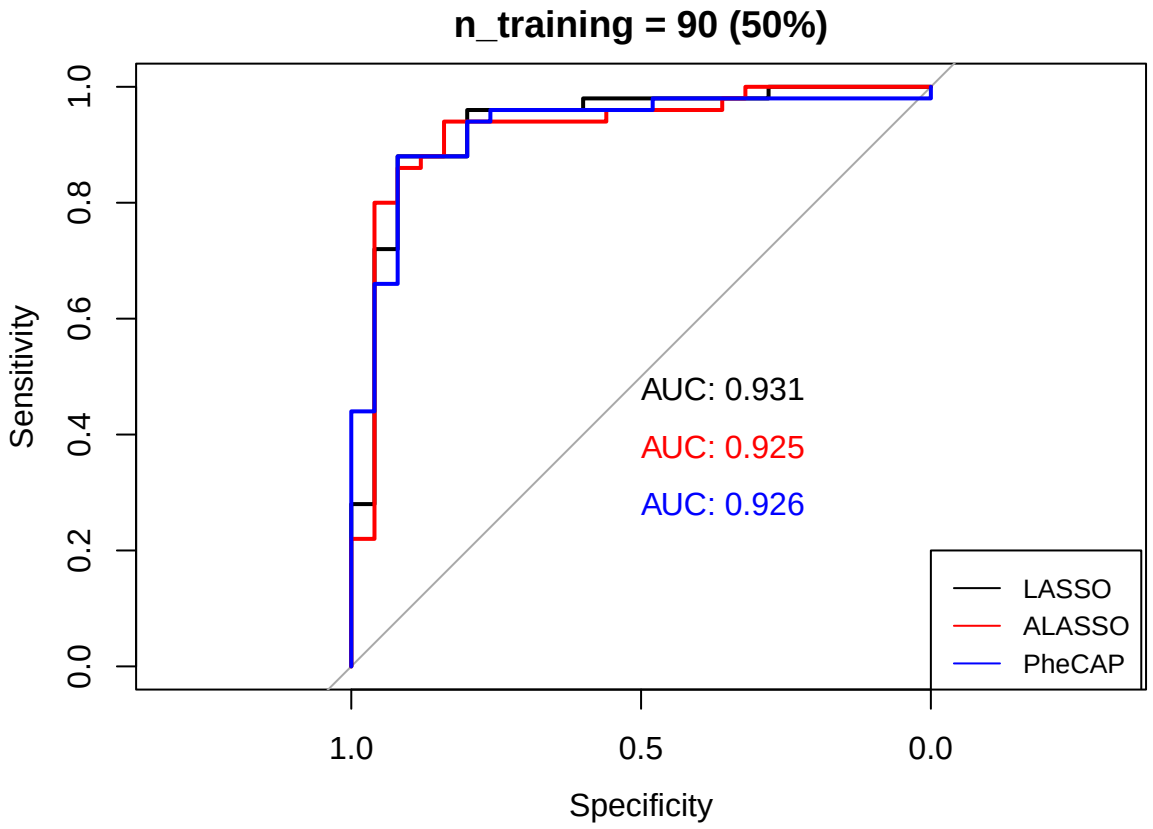
```
selected_features <- c("main_ICD", "main_NLP", "main_ICDNLP", "healthcare_utilization", SAFE_feature)

# Fit PheCAP.
beta.phecap <- adaptive_lasso_fit(x = train_x[, selected_features],
                                  y = train_y,
                                  tuning = "cv", family = "binomial")

y_hat.phecap <- linear_model_predict(beta = beta.phecap,
                                     x = test_x[, selected_features] ,
                                     probability = TRUE)
```

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

plot(roc.lasso,
  print.auc = TRUE, main = "n_training = 90 (50%)"
)
plot(roc.alasso,
  print.auc = TRUE, col = 'red', add = TRUE, print.auc.y = 0.4
)
plot(roc.phecap,
  print.auc = TRUE, col = 'blue', add = TRUE, print.auc.y = 0.3
)
legend(0, 0.2, legend = c("LASSO", "ALASSO", "PheCAP"), col = c("black","red", "blue"),
       lty = 1, cex = 0.8)
```




```
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
```

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

```
##          cutoff    pos.rate  FPR       TPR       PPV       NPV        F1
##  [1,] 0.9980230 0.006666667 0.00 0.2104348 1.0000000 0.3877276 0.3477011
##  [2,] 0.9651339 0.160000000 0.00 0.3252174 1.0000000 0.4256107 0.4908136
##  [3,] 0.9322447 0.300000000 0.02 0.4400000 0.9777778 0.4666667 0.6068966
##  [4,] 0.9320450 0.306666667 0.04 0.4950000 0.9611650 0.4873096 0.6534653
##  [5,] 0.9318453 0.313333333 0.04 0.5500000 0.9649123 0.5161290 0.7006369
##  [6,] 0.8991719 0.440000000 0.04 0.6050000 0.9680000 0.5485714 0.7446154
##  [7,] 0.8664985 0.460000000 0.06 0.6600000 0.9565217 0.5802469 0.7810651
##  [8,] 0.8601956 0.466666667 0.08 0.7150000 0.9470199 0.6174497 0.8148148
##  [9,] 0.8538928 0.473333333 0.08 0.7700000 0.9506173 0.6666667 0.8508287
## [10,] 0.7916090 0.586666667 0.08 0.8250000 0.9537572 0.7244094 0.8847185
```

Different training size

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

```
selected_index <- which(colnames(ehr_data) %in% selected_features == TRUE)
```

```
start<- Sys.time()
auc_phecap <- validate_phecap(dat = labeled_data, nsim = 600,
                              n.train = c(50, 70, 90),
                              selected_features = selected_index)
end <- Sys.time()
end - start
```

```
## Time difference of 1.384746 mins
```

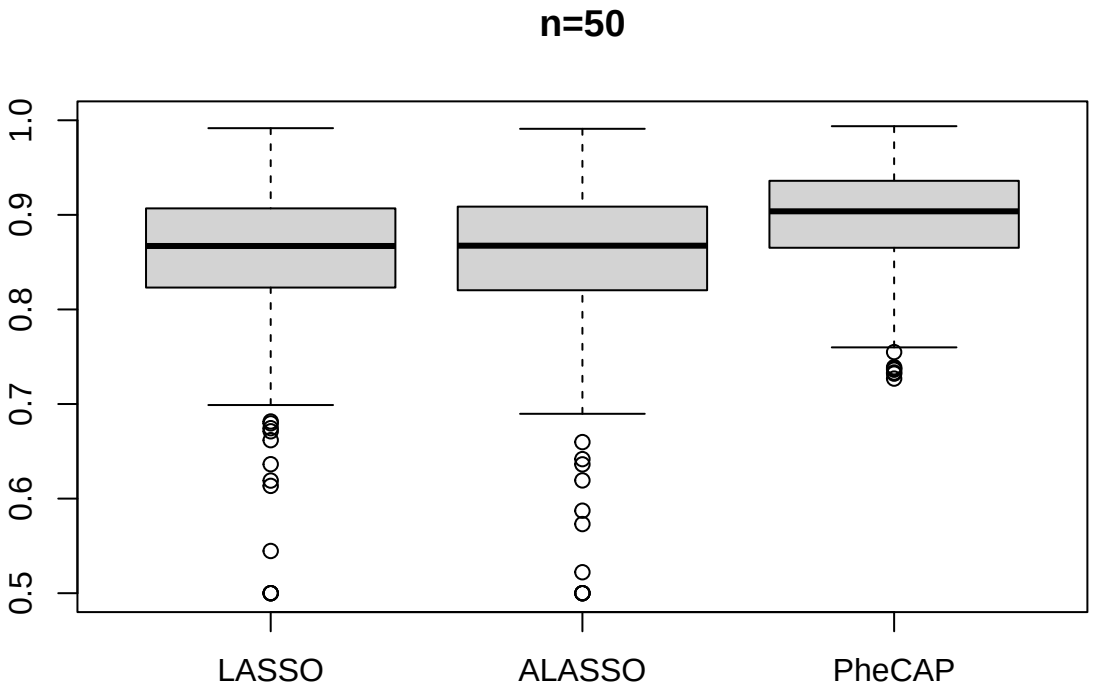
```
# median AUC
apply(auc_supervised, 2, median)
```

```
##  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
```

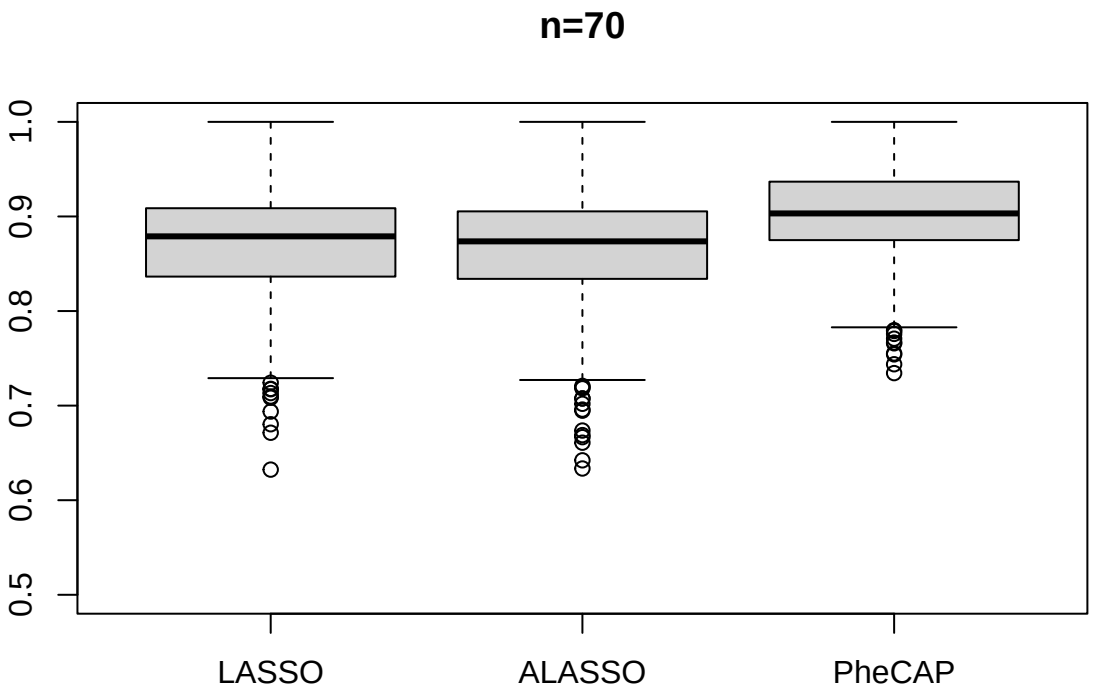
```
# se
apply(auc_supervised, 2, sd)
```

```
##  n=50,LASSO  n=70,LASSO  n=90,LASSO n=50,ALASSO n=70,ALASSO n=90,ALASSO
##  0.07197573  0.05587566  0.05184181  0.07300341  0.05871336  0.05415953
```

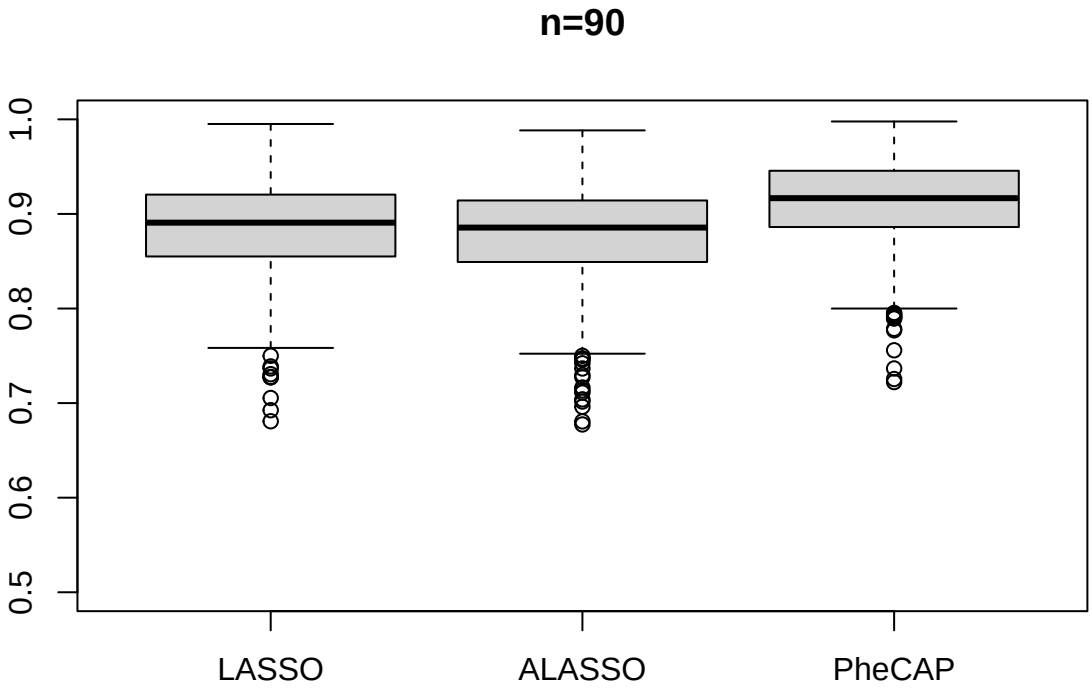
```
boxplot(cbind(auc_supervised, auc_phecap) %>% select(starts_with("n=50")),
        ylim = c(0.5, 1), names = c("LASSO", "ALASSO", "PheCAP"), main = "n=50")
```



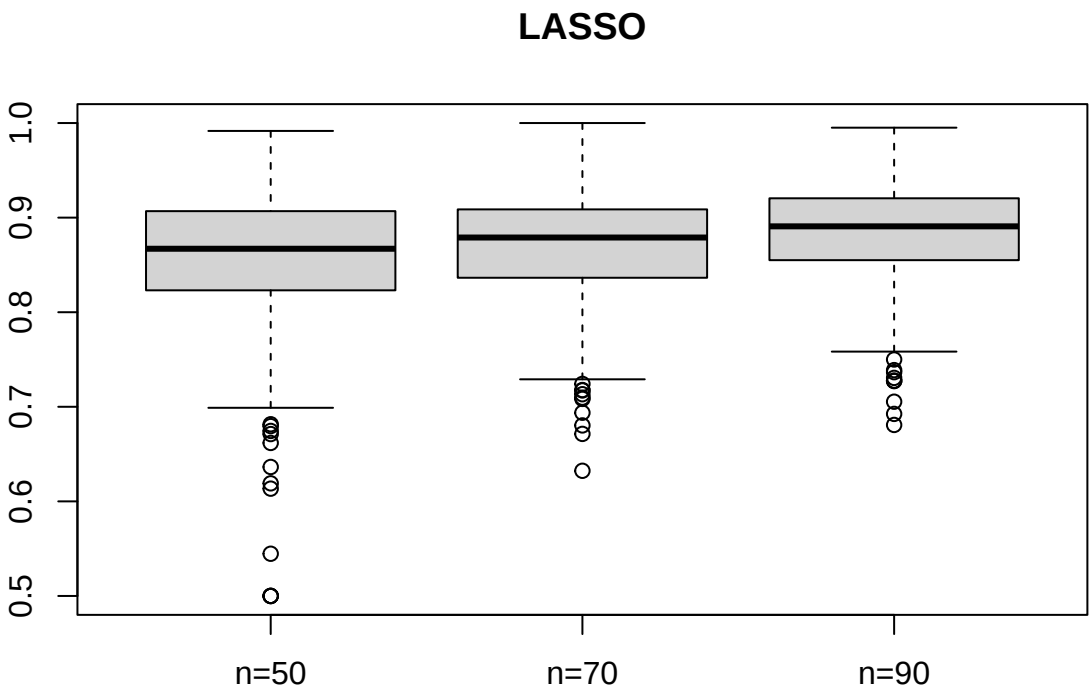

```
boxplot(cbind(auc_supervised, auc_phecap) %>% select(starts_with("n=70")),
        ylim = c(0.5, 1), names = c("LASSO", "ALASSO", "PheCAP"), main = "n=70")
```



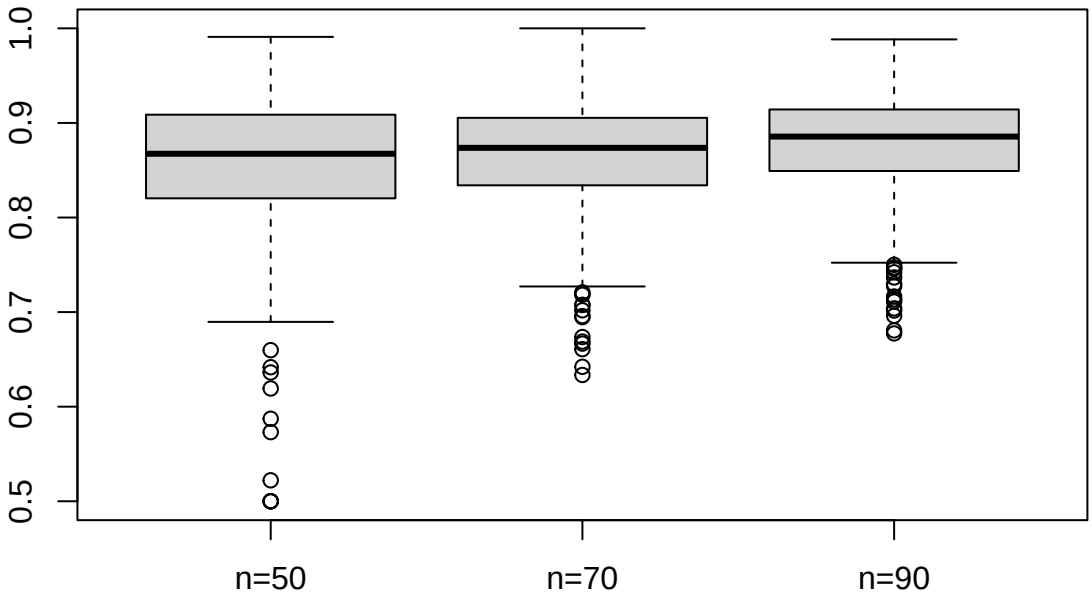

```
boxplot(cbind(auc_supervised, auc_phecap) %>% select(starts_with("n=90")),
        ylim = c(0.5, 1), names = c("LASSO", "ALASSO", "PheCAP"), main = "n=90")
```



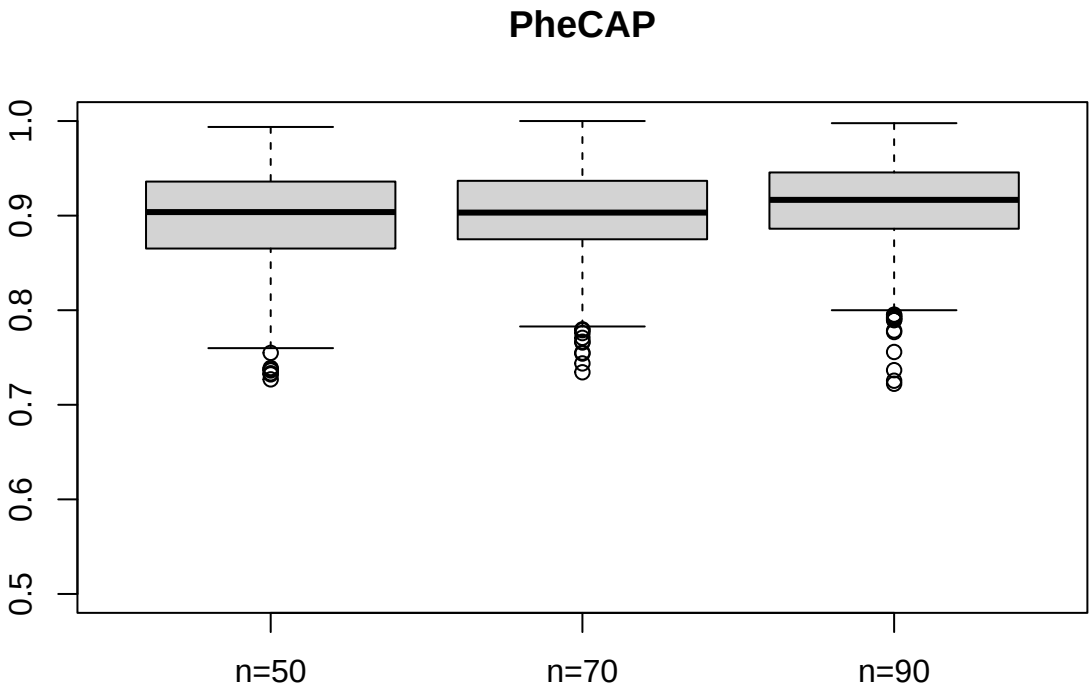

```
boxplot(auc_supervised[,1:3], ylim = c(0.5, 1),
        names = c("n=50", "n=70", "n=90"), main = "LASSO")
```




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




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



Save the data and feature selected for module 4 and model fitting.

```
save(list = ls(), file = "../module4/environment.RData")
```