Module 4: Alternative approaches

2-step Semi-supervised Approach

i) Regress the surrogate on the features with penalized least square to get the direction of beta.

```
x <- log(ehr_data %>% select(starts_with("health") | starts_with("COD") |
  starts_with("NLP")) + 1) # COD + NLP + HU
S <- log(ehr_data$main_ICD + ehr_data$main_NLP + 1)

# Step 1
beta_step1 <- adaptive_lasso_fit(
  y = S[], # surrogate
  x = x[], # all X
  family = "gaussian",
  tuning = "ic"
)

# Features selected
names(beta_step1[abs(beta_step1) > 0])[-1]
```

```
##  [1] "COD10"  "NLP6"   "NLP14"  "NLP24"  "NLP31"  "NLP44"  "NLP56"  "NLP59"
##  [9] "NLP61"  "NLP68"  "NLP73"  "NLP74"  "NLP93"  "NLP127" "NLP130" "NLP160"
## [17] "NLP161" "NLP172" "NLP176" "NLP193" "NLP199" "NLP202" "NLP215" "NLP225"
## [25] "NLP231" "NLP243" "NLP294" "NLP295" "NLP302" "NLP304" "NLP306" "NLP309"
## [33] "NLP321" "NLP349" "NLP350" "NLP361" "NLP403" "NLP434" "NLP446" "NLP451"
## [41] "NLP456" "NLP463" "NLP465" "NLP482" "NLP495" "NLP507" "NLP536" "NLP539"
## [49] "NLP541" "NLP544" "NLP560" "NLP564"
```

2-step Semi-supervised Approach

i) Regress the surrogate on the features with penalized least square to get the direction of beta.

ii) Regress the outcome on the linear predictor to get the intercept and multiplier for the beta.

```
# linear predictor without intercept
bhatx <- linear_model_predict(beta = beta_step1, x = as.matrix(x))

# Step 2
step2 <- glm(train_y ~ bhatx[train_data$patient_id] + S[train_data$patient_id],
  family = "binomial"
)
beta_step2 <- coef(step2)
beta_step2
```

```
##                   (Intercept) bhatx[train_data$patient_id]
##                    -1.9395295                    0.6361248
##     S[train_data$patient_id]
##                      0.6534730
```

```
# recover beta
beta <- beta_step2[2] * beta_step1
```

Compare betas

```
# LASSO
beta_lasso[!beta_lasso == 0][-1]
```

```
##        COD2       COD10        NLP1       NLP17       NLP56       NLP82
## -0.07891435 -0.07964064 -0.15656996 -0.10698323  0.43476973 -0.14774013
##       NLP93      NLP104      NLP118      NLP130      NLP144      NLP164
## -0.95721897 -1.14198338 -0.83985826 -0.02971022 -0.39607669 -0.13824534
##      NLP172      NLP193      NLP199      NLP222      NLP231      NLP265
##  0.11876041  0.11493486 -0.16297872 -2.01541309  0.40654328 -0.84088955
##      NLP274      NLP280      NLP297      NLP299      NLP346      NLP362
## -0.17839805  0.62463549 -0.54371389  0.86087307 -0.40862069  0.17883546
##      NLP375      NLP382      NLP396      NLP401      NLP409      NLP435
##  0.79214450 -0.47973944 -0.08726960 -0.17450935  0.53175298  0.20241840
##      NLP451      NLP462      NLP488      NLP533      NLP536      NLP552
##  0.61949264 -0.24987822  0.46166193 -0.37801422  0.53979607  0.04623370
##      NLP568    main_NLP
##  0.40970337  1.28008994
```

```
# ALASSO
beta_alasso[!beta_alasso == 0][-1]
```

```
##      NLP56      NLP93     NLP104     NLP118     NLP222     NLP231     NLP265
##  0.1966447 -1.0538342 -1.7011315 -1.5489010 -2.0758094  0.3598780 -0.9584738
##     NLP280     NLP297     NLP299     NLP409     NLP536    main_NLP
##  0.6256635 -0.2093127  1.0106695  0.4019735  0.1038460  1.4248803
```

Compare betas

```
# PheCAP

# 2 Step
beta[!beta == 0][-1]
```

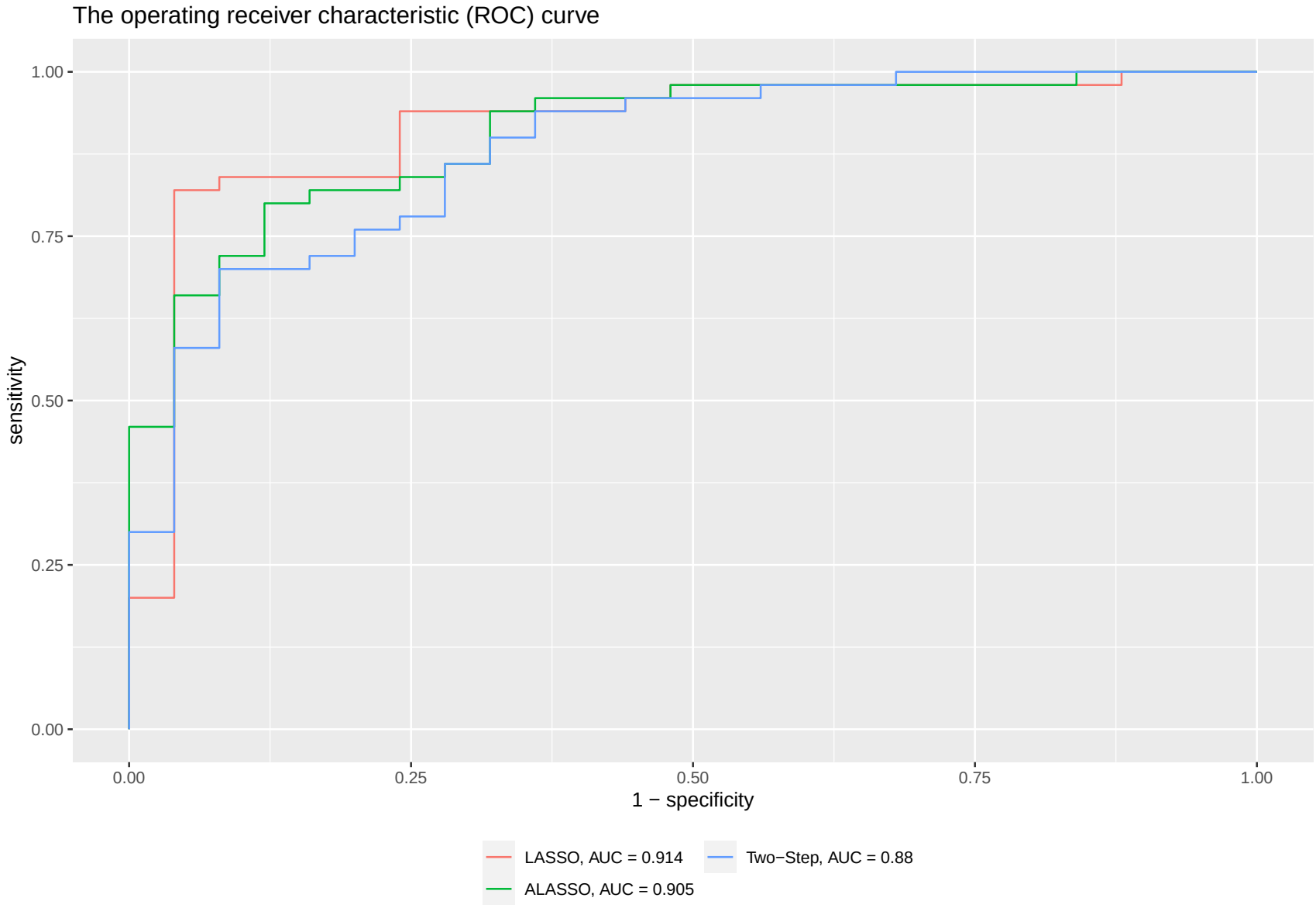
```
##        COD10         NLP6        NLP14        NLP24        NLP31        NLP44
## -0.052684289  0.016671720 -0.006957855 -0.023926364  0.025759944  0.019471833
##        NLP56        NLP59        NLP61        NLP68        NLP73        NLP74
##  0.198290388 -0.039200882  0.026074774  0.032826245 -0.028517111 -0.020899707
##        NLP93       NLP127       NLP130       NLP160       NLP161       NLP172
## -0.244610912 -0.017342083  0.019679635  0.269205071  0.113374310 -0.056810746
##       NLP176       NLP193       NLP199       NLP202       NLP215       NLP225
##  0.040699524 -0.005262443  0.013997988 -0.029110663  0.027935000 -0.101808993
##       NLP231       NLP243       NLP294       NLP295       NLP302       NLP304
##  0.083571172 -0.028323332 -0.066200601 -0.033656008 -0.037062329 -0.162835333
##       NLP306       NLP309       NLP321       NLP349       NLP350       NLP361
##  0.154919464  0.103247281  0.147671004  0.097088218  0.040739431 -0.009832788
##       NLP403       NLP434       NLP446       NLP451       NLP456       NLP463
##  0.138386240  0.054695149  0.018172852 -0.009528431 -0.061582265 -0.042908283
##       NLP465       NLP482       NLP495       NLP507       NLP536       NLP539
## -0.020442361 -0.037878047  0.028522562 -0.012282138  0.034586854 -0.053668635
##       NLP541       NLP544       NLP560       NLP564
##  0.047804629 -0.040011100  0.053525402  0.074898869
```

ROC

```
# mu
mu <- beta_step2[1] +
  as.numeric(as.matrix(x[test_data$patient_id, ])
  %*% beta[-1]) +
  as.numeric(beta_step2[3] %*% S[test_data$patient_id])
# expit
y_hat_twostep <- plogis(mu)

roc_twostep <- roc(test_y, y_hat_twostep)
```

ROC


The operating receiver characteristic (ROC) curve
sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 − specificity
LASSO, AUC = 0.914
Two−Step, AUC = 0.88
ALASSO, AUC = 0.905

Model Evaluation

```
auc_twostep <- validate_ss(
  dat = labeled_data, nsim = 600,
  n.train = c(50, 70, 90),
  beta = beta_step1,
  S = S,
  x = x
)
```