DSII Midterm Project

Yiru Gong, yg2832

2022-05-08

Contents

```r
library(tidyverse)
library(summarytools)
library(corrplot)
library(caret)
library(MASS)
library(mlbench)
library(pROC) #ROCR
library(pdp)
library(vip)
library(AppliedPredictiveModeling) #for transparentTheme function
```

Data Input

```r
data = read.csv('Covid19_vacc_predict_handout.csv')
data = data %>%
  na.omit() %>%
  dplyr::select(-id) %>%
  mutate(
    atlas_type_2015_mining_no = factor(atlas_type_2015_mining_no),
    covid_vaccination = factor(covid_vaccination),
    hum_region = factor(hum_region),
    sex_cd = factor(sex_cd),
    race_cd = factor(race_cd),
    lang_spoken_cd = factor(lang_spoken_cd),
    atlas_low_education_2015_update = factor(atlas_low_education_2015_update)
    )
# summary(data)
# by(data[,c(5,7,8,10,11,17,18)], data$covid_vaccination, summary)
dfSummary(data[,c(5,7,8,10,11,17,18)])
```

Data Frame Summary
Dimensions: 8308 x 7
Duplicates: 7802

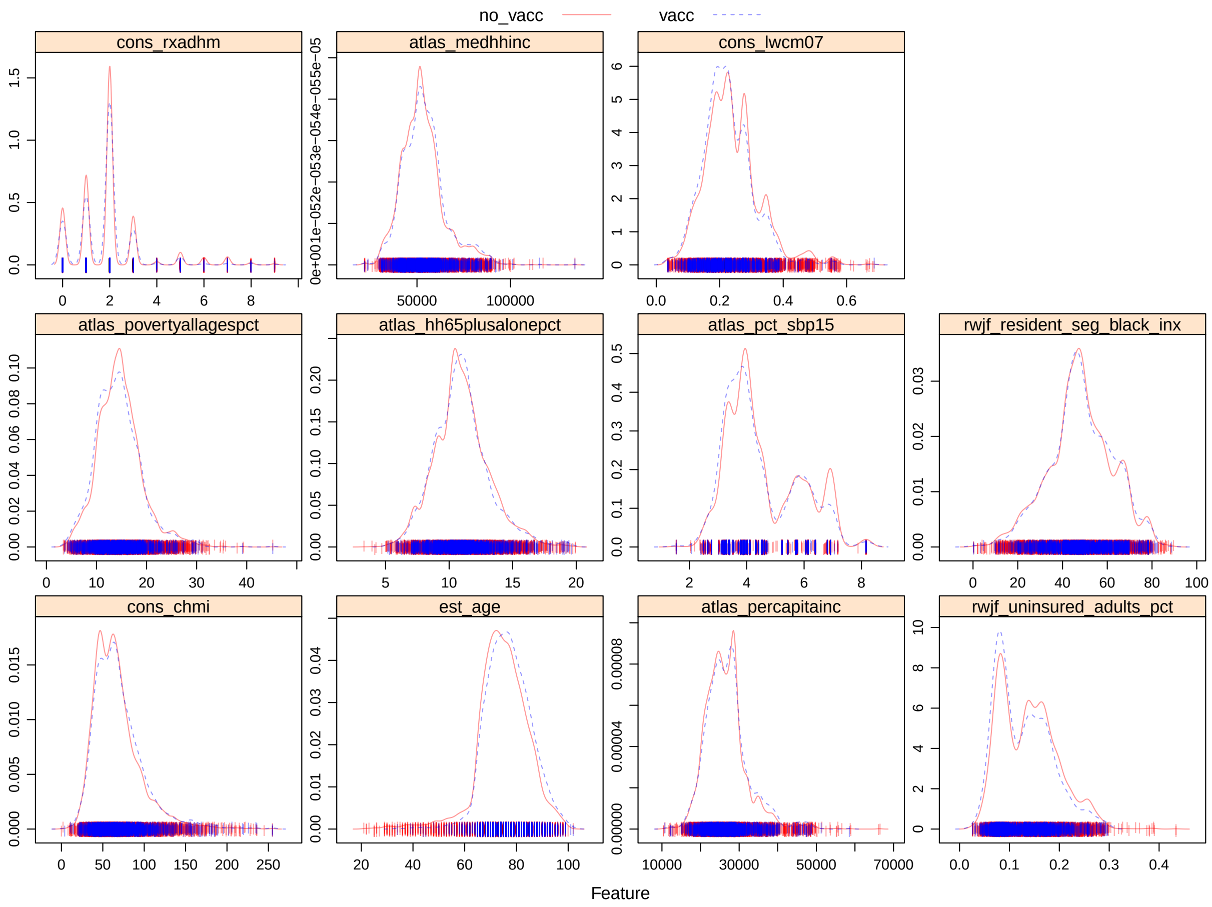
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	atlas_type_2015_mining_no [factor]	1. 0 2. 1	8177 (98.4%) 131 (1.6%)	IIIIIIIIIIIIIIIIIII	8308 (100.0%)	0 (0.0%)
2	covid_vaccination [factor]	1. no_vacc 2. vacc	6682 (80.4%) 1626 (19.6%)	IIIIIIIIIIIIIIII III	8308 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
3	hum_region [factor]	1. CALIFORNIA/NEVADA 2. CENTRAL 3. CENTRAL WEST 4. EAST 5. EAST CENTRAL 6. FLORIDA 7. GREAT LAKES/CENTRAL NORTH 8. GULF STATES 9. INTERMOUNTAIN 10. MID-ATLANTIC/NORTH CAROLI [5 others]	299 (3.6%) 551 (6.6%) 238 (2.9%) 491 (5.9%) 1370 (16.5%) 607 (7.3%) 1111 (13.4%) 454 (5.5%) 220 (2.6%) 845 (10.2%) 2122 (25.5%)	I	8308 (100.0%)	0 (0.0%)
4	sex_cd [factor]	1. F 2. M	4527 (54.5%) 3781 (45.5%)	IIIIIIIIII IIIIIIIII	8308 (100.0%)	0 (0.0%)
5	lang_spoken_cd [factor]	1. * 2. CHI 3. CRE 4. ENG 5. KOR 6. OTH 7. SPA 8. VIE	10 (0.1%) 13 (0.2%) 4 (0.0%) 7957 (95.8%) 7 (0.1%) 34 (0.4%) 276 (3.3%) 7 (0.1%)		8308 (100.0%)	0 (0.0%)
6	atlas_low_education_2015_update [factor]	1. 0 2. 1	7769 (93.5%) 539 (6.5%)	IIIIIIIIIIIIIIIIII I	8308 (100.0%)	0 (0.0%)
7	race_cd [factor]	1. 0 2. 1 3. 2 4. 3 5. 4 6. 5 7. 6	160 (1.9%) 7317 (88.1%) 558 (6.7%) 80 (1.0%) 56 (0.7%) 129 (1.6%) 8 (0.1%)	IIIIIIIIIIIIIIIII I	8308 (100.0%)	0 (0.0%)

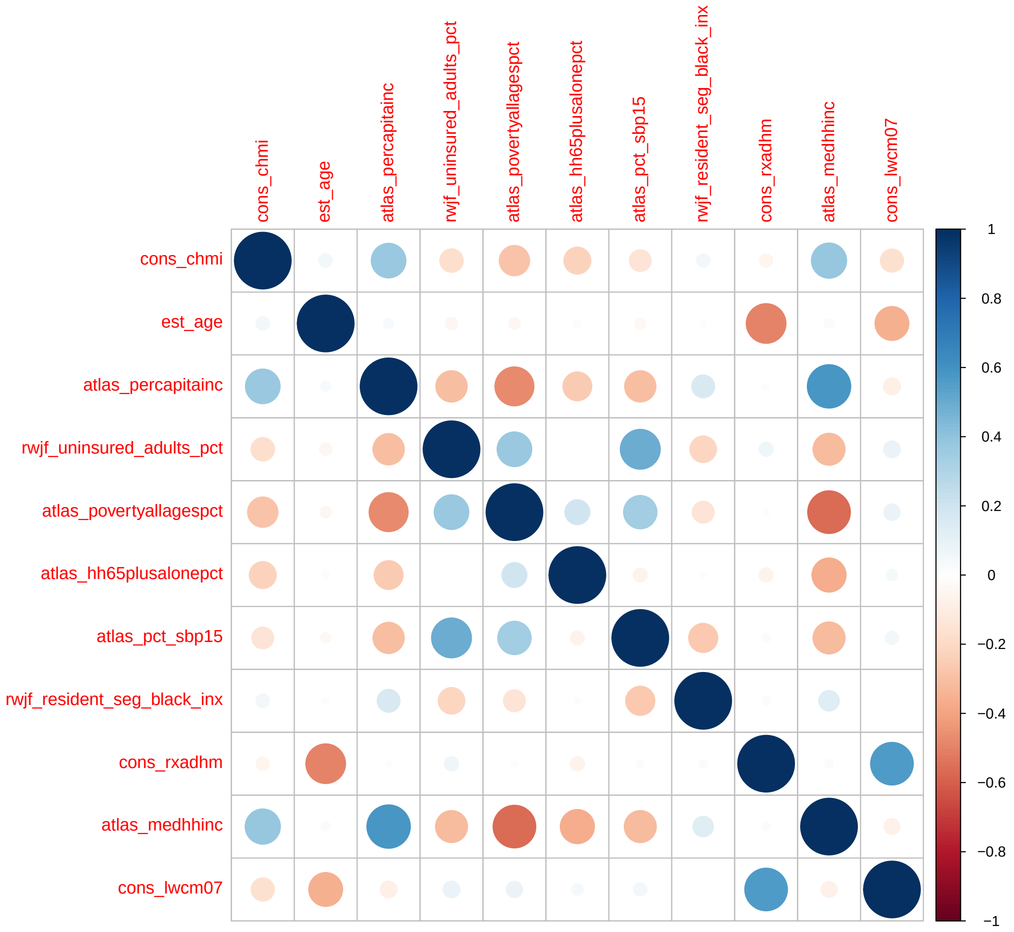
```
# cat_sum = NULL
# for (n in c(5,8,10,11,17,18)){
#   cat = data[,c(n,7)]
#   name = colnames(cat)[1]
#   cat2 = cat %>%
#     group_by(covid_vaccination,cat[,1]) %>%
#     count() %>%
#     rename(cat=`cat[, 1]`) %>%
#     pivot_wider(
#       names_from = covid_vaccination,
#       values_from = n
#     ) %>%
#     mutate(variable = name) %>%
#     relocate(variable,everything())
#   cat_sum = rbind(cat_sum,cat2)
# }
# knitr::kable(cat_sum)

# cat_sum %>%
#   pivot_longer(
#     c("no_vacc","vacc"),
#     names_to = 'covid_vaccination',
#     values_to = 'count'
#   ) %>%
#   ggplot(aes(variable,count,group=covid_vaccination,fill=cat))+geom_bar(stat = 'identity')

data2 = model.matrix(covid_vaccination ~ ., data)[ ,-1]
```

Exploratory analysis

```
theme1 <- transparentTheme(trans = .4)
trellis.par.set(theme1)

#figure 1
featurePlot(x = data[,-c(5,7,8,10,11,17,18)],
            y = data$covid_vaccination,
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            plot = "density", pch = "|",
            auto.key = list(columns = 2))
```



```
#correlation
corrplot(cor(data[,-c(5,7,8,10,11,17,18)]), method = "circle", type = "full")
```



Data split

```
ctrl <- trainControl(method = "repeatedcv",
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)

set.seed(1)
rowTrain <- createDataPartition(y = data$covid_vaccination,
                                p = 0.7,
                                list = FALSE)
x = data2[rowTrain,]
y = data$covid_vaccination[rowTrain]
x2 = data2[-rowTrain,]
y2 = data$covid_vaccination[-rowTrain]
```

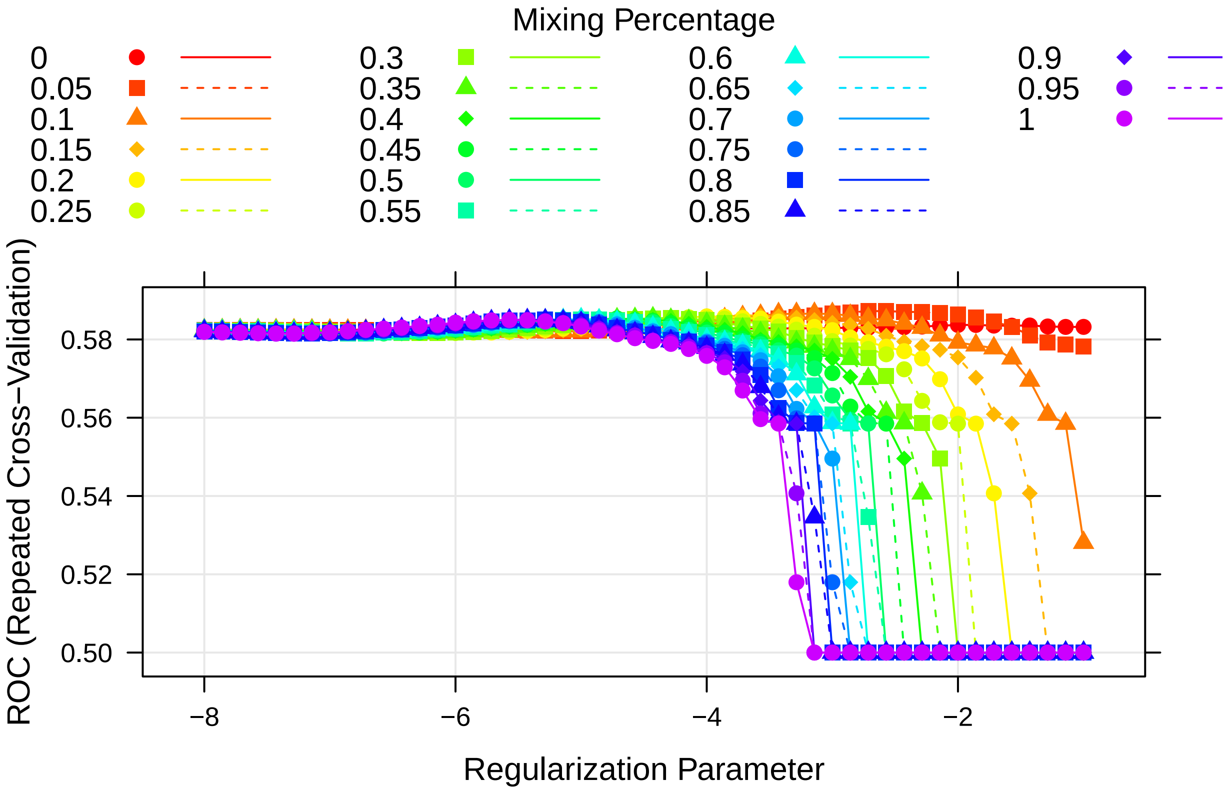
Model fitting

Penalized logistic regression

```
glmnGrid <- expand.grid(.alpha = seq(0, 1, length = 21),
                        .lambda = exp(seq(-8, -1, length = 50)))
set.seed(1)
model.glmn <- train(x, y,
                    method = "glmnet",
                    tuneGrid = glmnGrid,
                    metric = "ROC",
                    trControl = ctrl)

model.glmn$bestTune
```

```
##    alpha     lambda
## 89  0.05 0.07642629
```

```
myCol<- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))

plot(model.glmn, par.settings = myPar, xTrans = function(x) log(x))
```

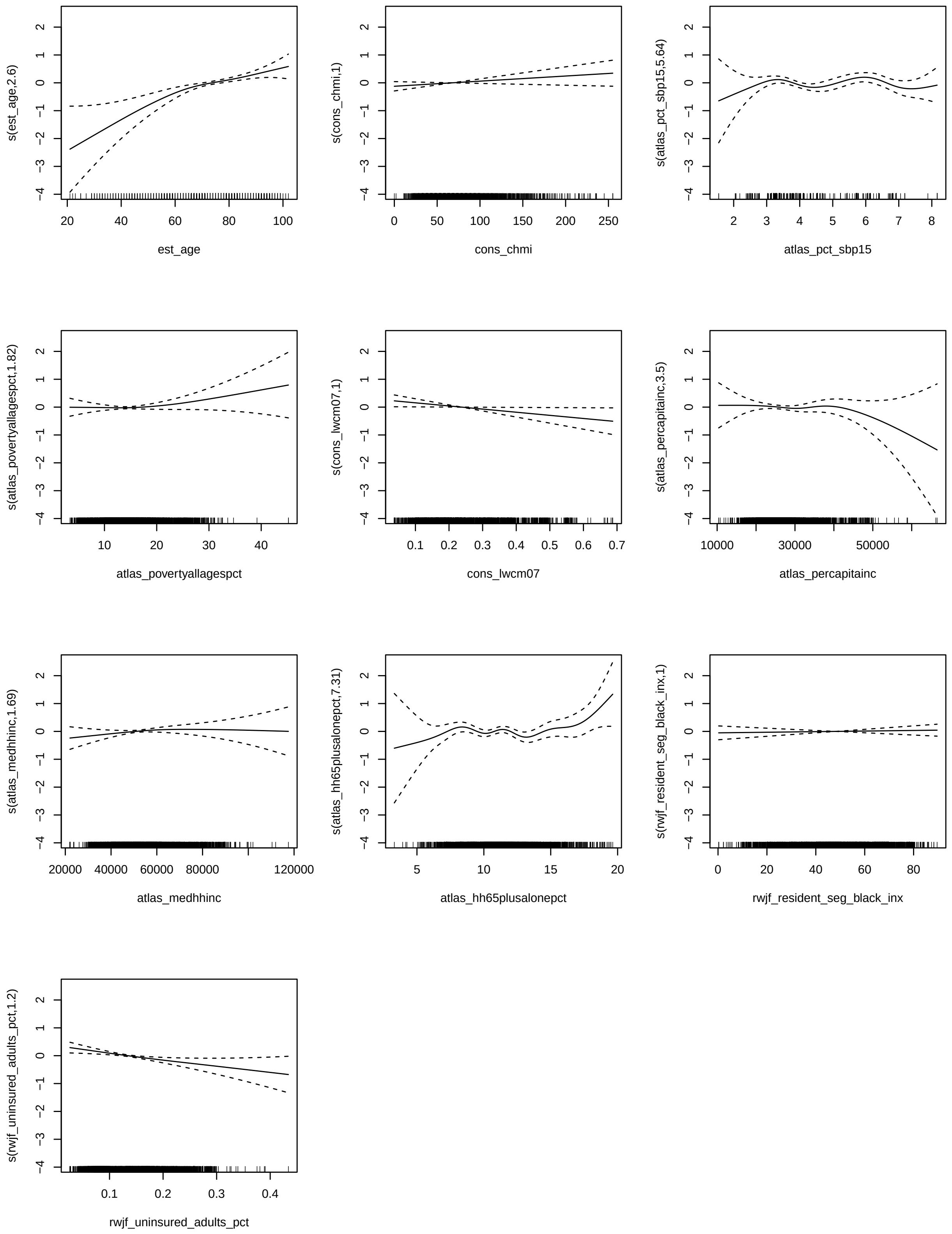


GAM

```
set.seed(1)
model.gam <- train(data[rowTrain,-c(7:8)], y,
                   method = "gam",
                   metric = "ROC",
                   trControl = ctrl)
### row 8: hum_region report error

model.gam$finalModel
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## .outcome ~ sex_cd + atlas_low_education_2015_update + race_cd +
##     cons_rxadhm + s(est_age) + s(cons_chmi) + s(atlas_pct_sbp15) +
##     s(atlas_povertyallagespct) + s(cons_lwcm07) + s(atlas_percapitainc) +
##     s(atlas_medhhinc) + s(atlas_hh65plusalonepct) + s(rwjf_resident_seg_black_inx) +
##     s(rwjf_uninsured_adults_pct)
##
## Estimated degrees of freedom:
## 2.60 1.00 5.64 1.82 1.00 3.50 1.69
## 7.31 1.00 1.20  total = 36.76
##
## UBRE score: -0.02449249
```

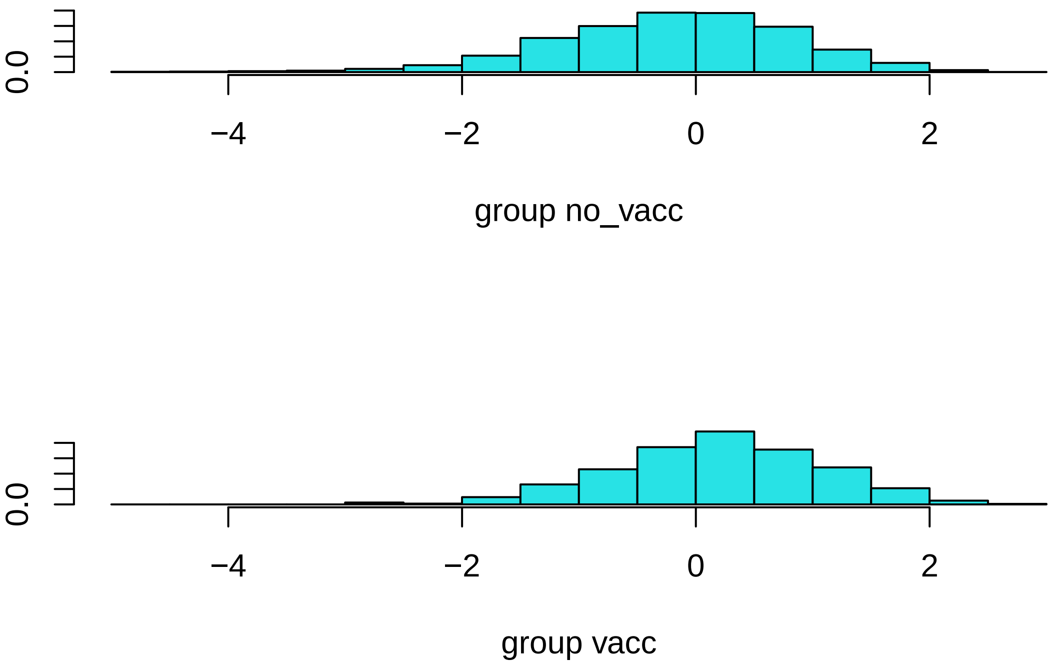
```
# fig 2
par(mfrow=c(4,3))
plot(model.gam$finalModel)
```


s(est_age,2.6)
est_age
20 40 60 80 100
−4 −3 −2 −1 0 1 2
s(cons_chmi,1)
cons_chmi
0 50 100 150 200 250
s(atlas_pct_sbp15,5.64)
atlas_pct_sbp15
2 3 4 5 6 7 8
s(atlas_povertyallagespct,1.82)
atlas_povertyallagespct
10 20 30 40
s(cons_lwcm07,1)
cons_lwcm07
0.1 0.2 0.3 0.4 0.5 0.6 0.7
s(atlas_percapitainc,3.5)
atlas_percapitainc
10000 30000 50000
s(atlas_medhhinc,1.69)
atlas_medhhinc
20000 40000 60000 80000 120000
s(atlas_hh65plusalonepct,7.31)
atlas_hh65plusalonepct
5 10 15 20
s(rwjf_resident_seg_black_inx,1)
rwjf_resident_seg_black_inx
0 20 40 60 80
s(rwjf_uninsured_adults_pct,1.2)
rwjf_uninsured_adults_pct
0.1 0.2 0.3 0.4

```
# ## add data preprocessing
# model.gam.proc <- train(data[rowTrain,-c(7:8)], y,
#                          preProcess = c("zv"),
#                     method = "gam",
#                     metric = "ROC",
#                     trControl = ctrl)
# plot(model.gam.proc$finalModel, select = 3)
```

LDA

```
lda.fit <- lda(y~x)
plot(lda.fit)
```



```
set.seed(1)
model.lda <- train(x, y,
                   method = "lda",
                   metric = "ROC",
                   trControl = ctrl)
```

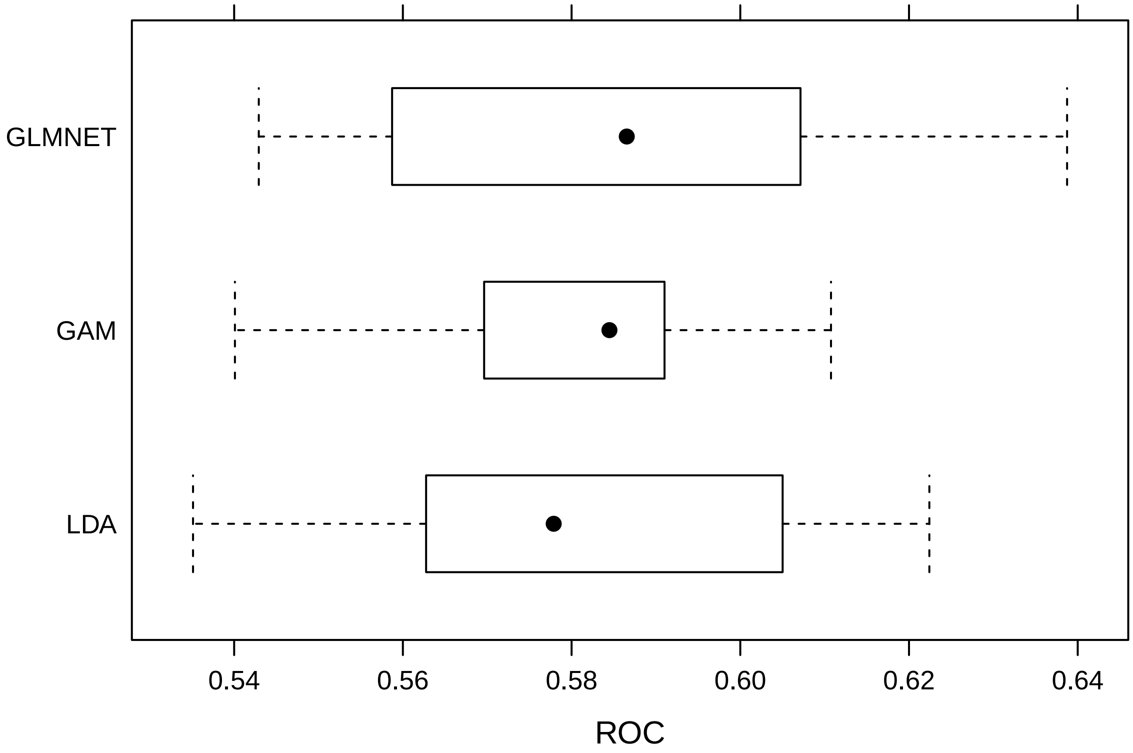
Model Comparison

CV Compare

```
res <- resamples(list(GLMNET = model.glmn,
                      GAM = model.gam,
                      LDA = model.lda))

#KNN
summary(res)
```

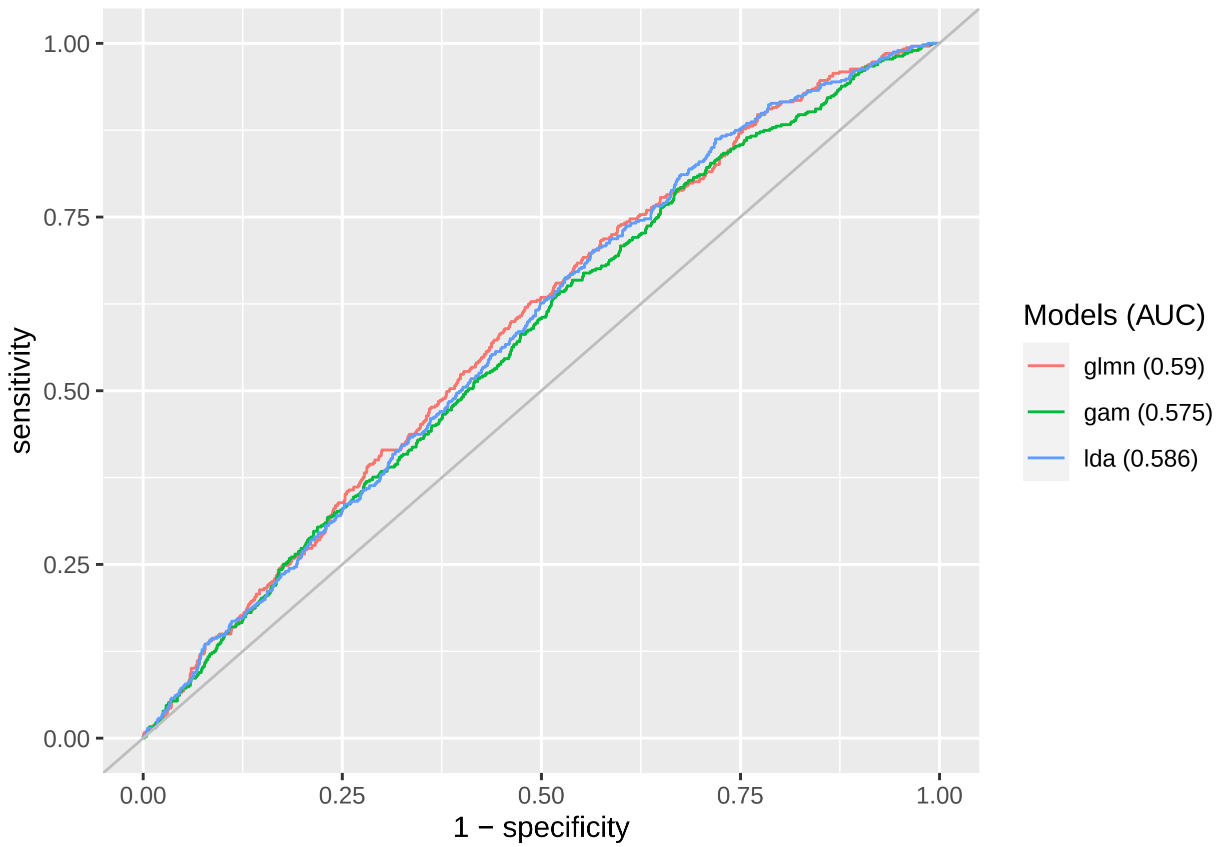
```
##
## Call:
## summary.resamples(object = res)
##
## Models: GLMNET, GAM, LDA
## Number of resamples: 10
##
## ROC
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLMNET 0.5429205 0.5624063 0.5865405 0.5872412 0.6059043 0.6387389    0
## GAM    0.5400922 0.5726839 0.5844861 0.5816581 0.5903860 0.6107550    0
## LDA    0.5351253 0.5645383 0.5778804 0.5822242 0.6033940 0.6224134    0
##
## Sens
##             Min. 1st Qu. Median      Mean 3rd Qu. Max. NA's
## GLMNET 1.0000000       1      1 1.0000000       1    1    0
## GAM    0.9978632       1      1 0.9995726       1    1    0
## LDA    1.0000000       1      1 1.0000000       1    1    0
##
## Spec
##        Min. 1st Qu. Median        Mean 3rd Qu.       Max. NA's
## GLMNET    0       0      0 0.000000000       0 0.00000000    0
## GAM       0       0      0 0.000877193       0 0.00877193    0
## LDA       0       0      0 0.000000000       0 0.00000000    0
```

```
# figure 4
bwplot(res, metric = "ROC")
```



Test data performance

```
glmn.pred <- predict(model.glmn, newdata = x2, type = "prob")[,2]
gam.pred <- predict(model.gam, newdata = data[-rowTrain,-c(7:8)], type = "prob")[,2]
lda.pred <- predict(model.lda, newdata = x2, type = "prob")[,2]

roc.glmn <- roc(y2, glmn.pred)
roc.gam <- roc(y2, gam.pred)
roc.lda <- roc(y2, lda.pred)

auc <- c(roc.glmn$auc[1], roc.gam$auc[1], roc.lda$auc[1])

modelNames <- c("glmn","gam","lda")

# fig 5
ggroc(list(roc.glmn, roc.gam, roc.lda), legacy.axes = TRUE) +
  scale_color_discrete(labels = paste0(modelNames, " (", round(auc,3),")"),
                       name = "Models (AUC)") +
  geom_abline(intercept = 0, slope = 1, color = "grey")
```


1.00
0.75
0.50
0.25
0.00
sensitivity
0.00
0.25
0.50
0.75
1.00
1 − specificity
Models (AUC)
glmn (0.59)
gam (0.575)
lda (0.586)