midterm_project_code

Jiaqi Chen

3/26/2022

```r
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```r
library(base)
library(AppliedPredictiveModeling)
library(pdp)
```

```
##
## Attaching package: 'pdp'

## The following object is masked from 'package:purrr':
##
##     partial
```

```r
library(vip)
```

```
##
## Attaching package: 'vip'

## The following object is masked from 'package:utils':
##
##     vi
```

```r
library(klaR)
```

```
## Warning: package 'klaR' was built under R version 4.1.2

## Loading required package: MASS

##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(dplyr)
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-2
```

Data Entry and Cleanning

```
data = read.csv('Covid19_vacc_predict_handout.csv') %>%
  mutate(covid_vaccination = as.factor(covid_vaccination)) %>%
  dplyr::select(-id)

nonchr_data = data %>%
  dplyr::select(-hum_region & -sex_cd & -lang_spoken_cd)
```

I used the COVID19 vaccination data for illustration. The data contain 8308 observations and 19 variables. The outcome is binary variable `covid_vaccination`: vacc means vaccinated and no-vacc means not vaccinated.

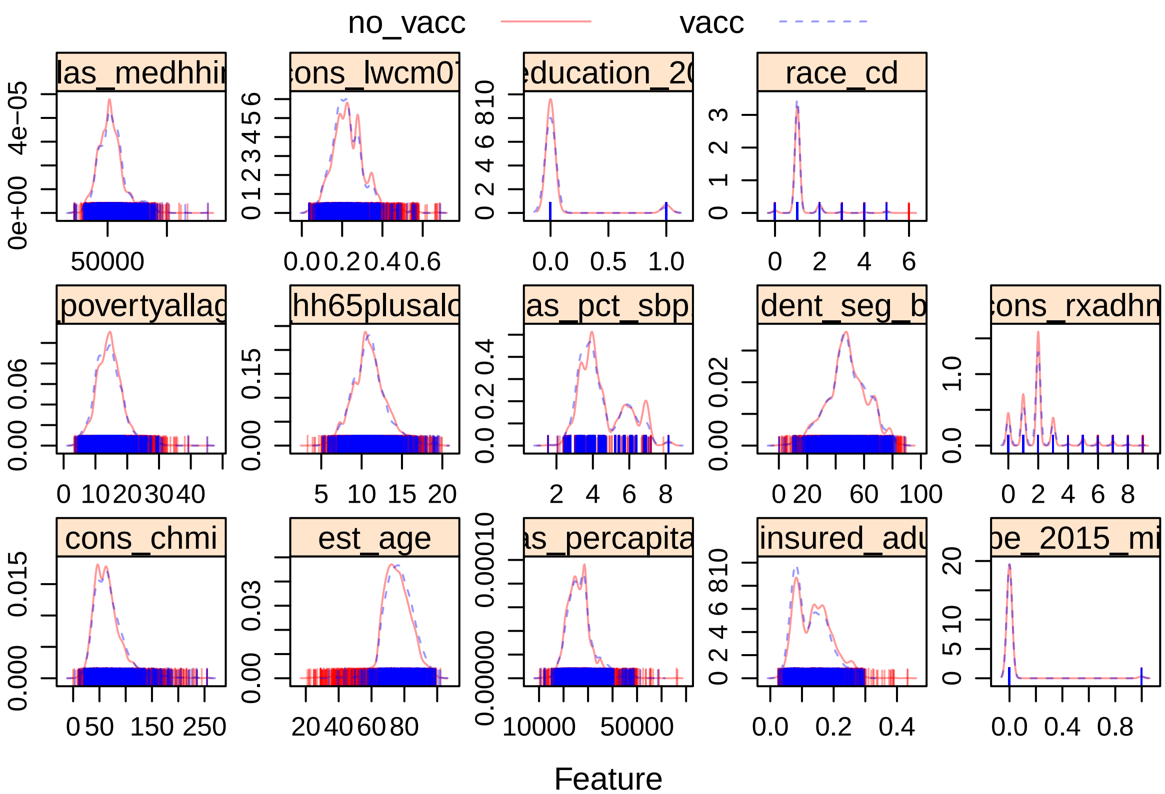
Split the dataset into two parts: training data (70%) and test data (30%)

```
set.seed(1)
train = createDataPartition(y = data$covid_vaccination, p = 0.7, list = FALSE)

x = data[, -7]
y = data$covid_vaccination

# Training Data
x1 = model.matrix(covid_vaccination ~., data)[train,-1]
y1 = data$covid_vaccination[train]

# Test Data
x2 = model.matrix(covid_vaccination ~., data)[-train,-1]
y2 = data$covid_vaccination[-train]
```

Data Visualization

Produce some graphical or numerical summaries of the data

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

nonchr_x = nonchr_data[, -7]
nonchr_y = nonchr_data$covid_vaccination

plot1 = featurePlot(nonchr_x, nonchr_y,
                    scales = list(x = list(relation = "free"),
                                  y = list(relation = "free")),
                    plot = "density", pch = "|",
                    auto.key = list(columns = 2))

plot1
```



Logistic Regression (GLM)

```
contrasts(data$covid_vaccination) #no_vacc-0, vacc-1
```

```
##          vacc
## no_vacc     0
## vacc        1
```

Fit a logistic regression model

```r
set.seed(1)
ctrl = trainControl(method = "repeatedcv",
                    summaryFunction = twoClassSummary,
                    classProbs = TRUE)

model.glm = train(x1, y1,
                  method = "glm",
                  metric = "ROC",
                  trControl = ctrl)

summary(model.glm)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0292  -0.7056  -0.6122  -0.4722   2.3262
##
## Coefficients:
##                                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)                            -3.067e+00  1.280e+00  -2.396   0.0166
## cons_chmi                               1.754e-03  1.248e-03   1.406   0.1597
## est_age                                 2.264e-02  4.136e-03   5.475 4.38e-08
## atlas_percapitainc                     -4.159e-06  7.889e-06  -0.527   0.5981
## rwjf_uninsured_adults_pct              -1.903e+00  9.171e-01  -2.075   0.0380
## atlas_type_2015_mining_no              -1.442e-01  3.296e-01  -0.438   0.6617
## atlas_povertyallagespct                 4.722e-03  9.999e-03   0.472   0.6367
## hum_regionCENTRAL                      -3.334e-01  2.232e-01  -1.494   0.1352
## `hum_regionCENTRAL WEST`               -5.627e-01  2.736e-01  -2.057   0.0397
## hum_regionEAST                         -2.708e-01  2.303e-01  -1.176   0.2397
## `hum_regionEAST CENTRAL`               -3.814e-01  1.964e-01  -1.942   0.0521
## hum_regionFLORIDA                      -5.587e-01  2.295e-01  -2.435   0.0149
## `hum_regionGREAT LAKES/CENTRAL NORTH`  -1.066e-01  1.981e-01  -0.538   0.5906
## `hum_regionGULF STATES`                 1.940e-01  2.355e-01   0.824   0.4099
## hum_regionINTERMOUNTAIN                -3.464e-01  2.732e-01  -1.268   0.2047
## `hum_regionMID-ATLANTIC/NORTH CAROLINA` -2.502e-01  2.034e-01  -1.230   0.2185
## `hum_regionMID-SOUTH`                  -6.091e-01  2.456e-01  -2.480   0.0132
## hum_regionNORTHEAST                    -3.844e-01  2.340e-01  -1.642   0.1005
## hum_regionPACIFIC                      -7.721e-01  1.096e+00  -0.704   0.4812
## hum_regionSOUTHEAST                    -2.765e-02  2.340e-01  -0.118   0.9059
## hum_regionTEXAS                        -5.225e-01  2.651e-01  -1.971   0.0487
## atlas_hh65plusalonepct                  1.449e-02  1.793e-02   0.808   0.4192
## sex_cdM                                -4.209e-02  7.155e-02  -0.588   0.5564
## lang_spoken_cdCHI                      -4.759e-01  1.524e+00  -0.312   0.7548
## lang_spoken_cdCRE                      -1.289e+01  6.221e+02  -0.021   0.9835
## lang_spoken_cdENG                       2.542e-01  1.094e+00   0.232   0.8162
## lang_spoken_cdKOR                      -1.272e+01  3.291e+02  -0.039   0.9692
## lang_spoken_cdOTH                       6.087e-01  1.208e+00   0.504   0.6142
## lang_spoken_cdSPA                       6.947e-03  1.114e+00   0.006   0.9950
## lang_spoken_cdVIE                      -1.254e+01  3.541e+02  -0.035   0.9718
```

```
## atlas_pct_sbp15                         -9.756e-03  4.993e-02  -0.195   0.8451
## rwjf_resident_seg_black_inx              8.350e-04  2.749e-03   0.304   0.7613
## cons_rxadhm                              2.794e-02  3.231e-02   0.865   0.3872
## atlas_medhhinc                           5.692e-06  4.159e-06   1.369   0.1711
## cons_lwcm07                             -1.126e+00  5.278e-01  -2.133   0.0329
## atlas_low_education_2015_update         -3.436e-02  1.614e-01  -0.213   0.8314
## race_cd                                 -6.681e-02  6.276e-02  -1.065   0.2871
##
## (Intercept)                             *
## cons_chmi
## est_age                                 ***
## atlas_percapitainc
## rwjf_uninsured_adults_pct               *
## atlas_type_2015_mining_no
## atlas_povertyallagespct
## hum_regionCENTRAL
## `hum_regionCENTRAL WEST`                *
## hum_regionEAST
## `hum_regionEAST CENTRAL`                .
## hum_regionFLORIDA                       *
## `hum_regionGREAT LAKES/CENTRAL NORTH`
## `hum_regionGULF STATES`
## hum_regionINTERMOUNTAIN
## `hum_regionMID-ATLANTIC/NORTH CAROLINA`
## `hum_regionMID-SOUTH`                   *
## hum_regionNORTHEAST
## hum_regionPACIFIC
## hum_regionSOUTHEAST
## hum_regionTEXAS                         *
## atlas_hh65plusalonepct
## sex_cdM
## lang_spoken_cdCHI
## lang_spoken_cdCRE
## lang_spoken_cdENG
## lang_spoken_cdKOR
## lang_spoken_cdOTH
## lang_spoken_cdSPA
## lang_spoken_cdVIE
## atlas_pct_sbp15
## rwjf_resident_seg_black_inx
## cons_rxadhm
## atlas_medhhinc
## cons_lwcm07                             *
## atlas_low_education_2015_update
## race_cd
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5753.4  on 5816  degrees of freedom
## Residual deviance: 5612.6  on 5780  degrees of freedom
## AIC: 5686.6
##
```

```
## Number of Fisher Scoring iterations: 13
```

From the summary of GLM model, we can see that residual deviance is 5611.8, close to 5779 degrees of freedom. Therefore, GLM model is a good fit.

In this model, `est_age`, `rwjf_uninsured_adults_pct`, `hum_regionCENTRAL WEST`, `hum_regionFLORIDA`, `hum_regionFLORIDA`, `hum_regionMID-SOUTH`, `hum_regionTEXAS`, `cons_lwcm07` are statistically significant predictors, since the p-value of these three predictors are less than 0.05.

Confusion Matrix

```r
test.pred.prob = predict(model.glm,
                         newdata = x2,
                         type = "prob")[,2]

test.pred = rep("no_vacc", length(test.pred.prob))
test.pred[test.pred.prob > 0.5] = "vacc"

confusionMatrix(data = as.factor(test.pred),
                reference = y2,
                positive = "vacc")
```

```
## Warning in confusionMatrix.default(data = as.factor(test.pred), reference =
## y2, : Levels are not in the same order for reference and data. Refactoring data
## to match.
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction no_vacc vacc
##    no_vacc    2004  487
##    vacc          0    0
##
##                Accuracy : 0.8045
##                  95% CI : (0.7884, 0.8199)
##     No Information Rate : 0.8045
##     P-Value [Acc > NIR] : 0.5121
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.8045
##              Prevalence : 0.1955
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : vacc
##
```

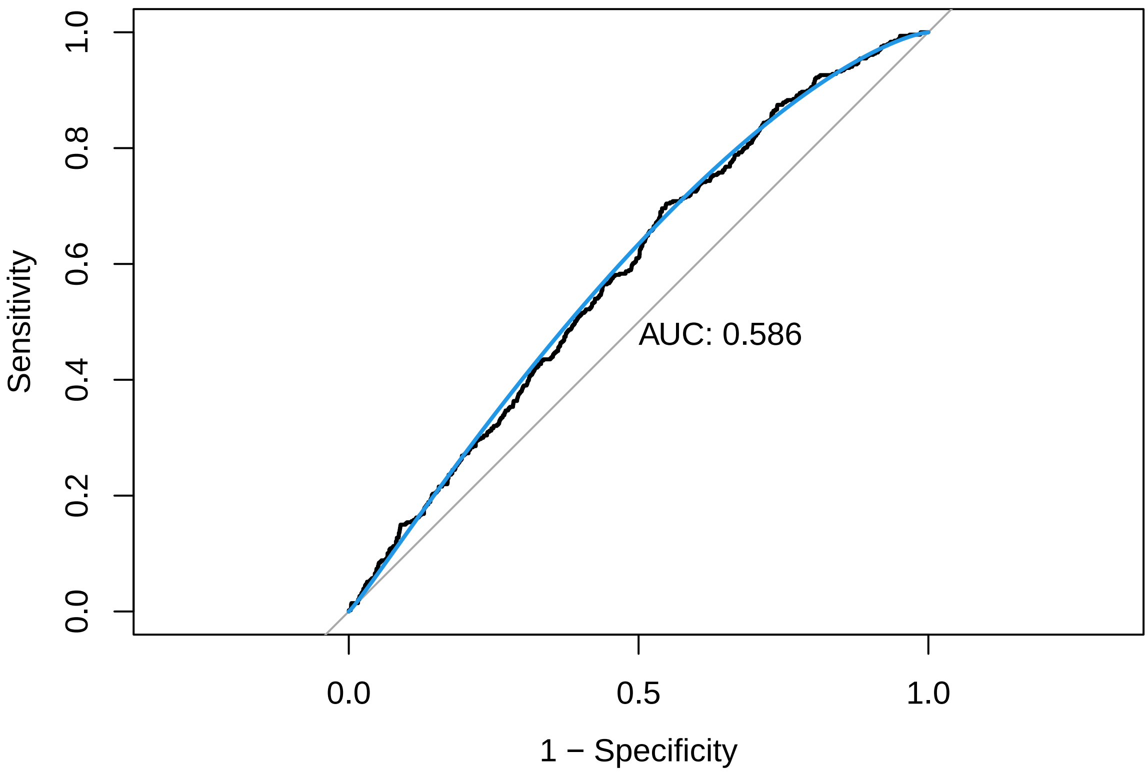
Plot the test ROC curve

```
roc.glm = roc(data$covid_vaccination[-train], test.pred.prob)
```

```
## Setting levels: control = no_vacc, case = vacc
```

```
## Setting direction: controls < cases
```

```
plot(roc.glm, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm), col = 4, add = TRUE)
```



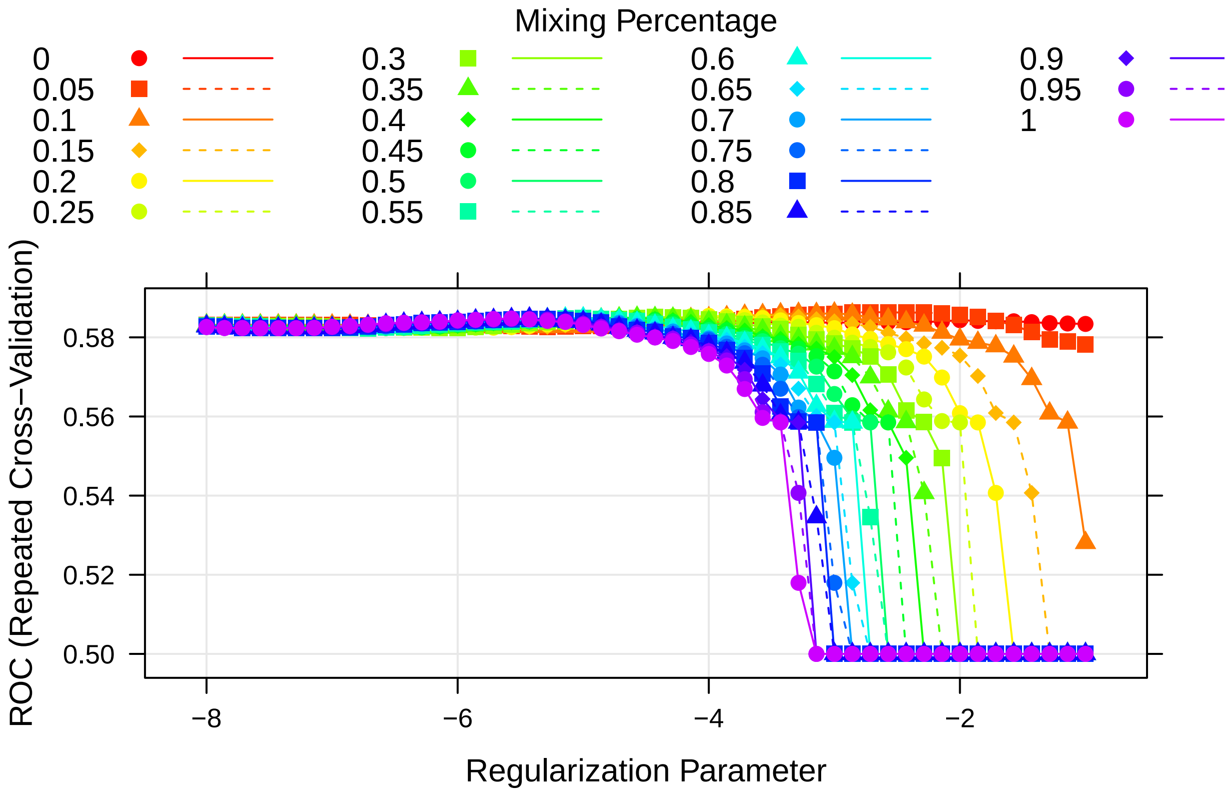
Penalized logistic regression (GLMN)

Fit a GLMN model

```
glmnGrid = expand.grid(.alpha = seq(0, 1, length = 21),
                       .lambda = exp(seq(-8, -1, length = 50)))

set.seed(1)
model.glmn = train(x1, y1,
                   method = "glmnet",
                   tuneGrid = glmnGrid,
                   metric = "ROC",
                   trControl = ctrl)

model.glmn$bestTune
```

```
##    alpha     lambda
## 88  0.05 0.06625226
```

```r
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

Fit a GAM Model

```r
set.seed(1)

gam_x = model.matrix(covid_vaccination ~., nonchr_data)[train,-1]
test_gam_x = model.matrix(covid_vaccination ~., nonchr_data)[-train, -1]

model.gam = train(gam_x, y1,
                  method = "gam",
                  metric = "ROC",
                  trControl = ctrl)
```

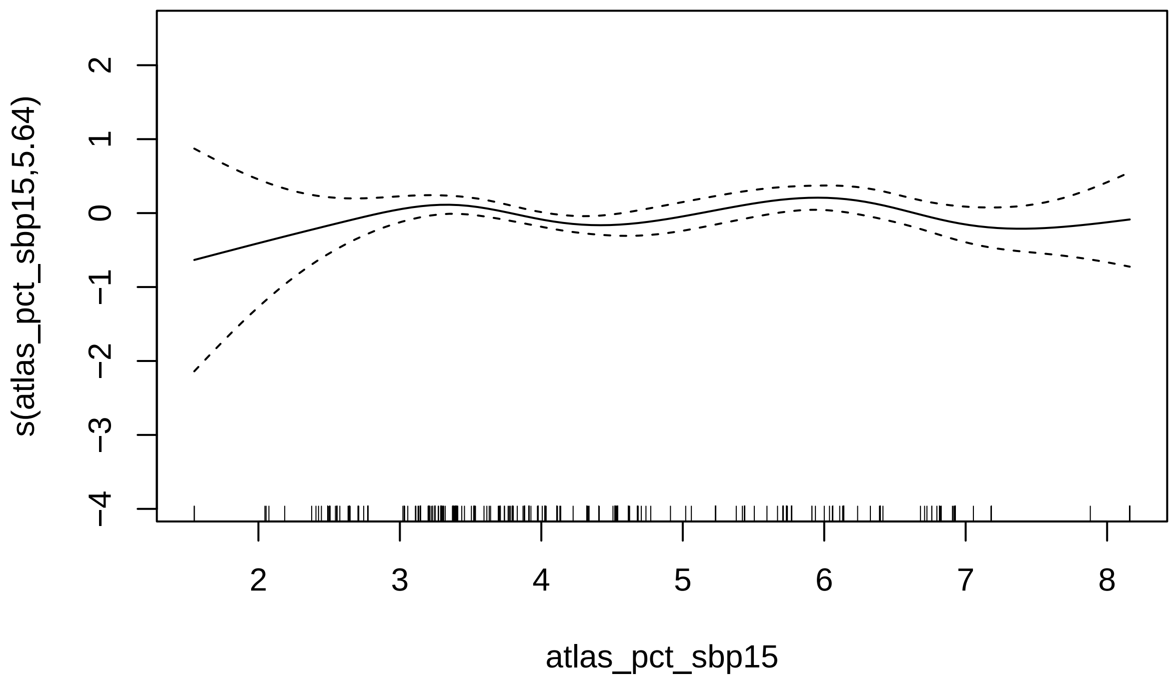
```
## Loading required package: mgcv

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
```

```
##     collapse
## This is mgcv 1.8-36. For overview type 'help("mgcv-package")'.
```

```
model.gam$finalModel
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## .outcome ~ 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(rwjf_resident_seg_black_inx) + s(atlas_hh65plusalonepct) +
##     s(rwjf_uninsured_adults_pct)
##
## Estimated degrees of freedom:
## 2.59 1.00 5.64 1.86 1.00 3.51 1.70
## 1.00 7.34 1.15  total = 30.79
##
## UBRE score: -0.02549095
```

```
plot(model.gam$finalModel, select = 3)
```



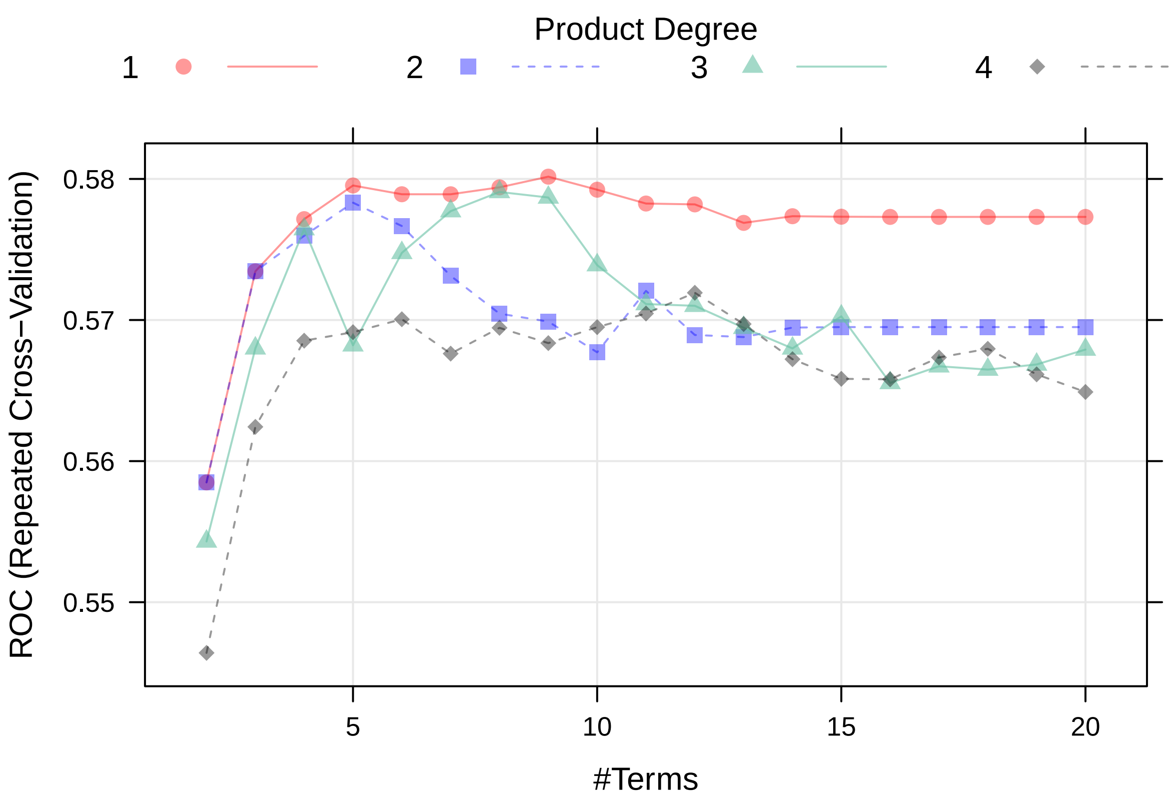
MARS

```
set.seed(1)
model.mars = train(x1, y1,
                   method = "earth",
                   tuneGrid = expand.grid(degree = 1:4,
                                          nprune = 2:20),
                   metric = "ROC",
```

```
                    trControl = ctrl)
```

```
## Loading required package: earth
## Loading required package: Formula
## Loading required package: plotmo
## Loading required package: plotrix
## Loading required package: TeachingDemos
##
## Attaching package: 'TeachingDemos'
## The following object is masked from 'package:klaR':
##
##     triplot
```

```
plot(model.mars)
```




```
model.mars$bestTune
```

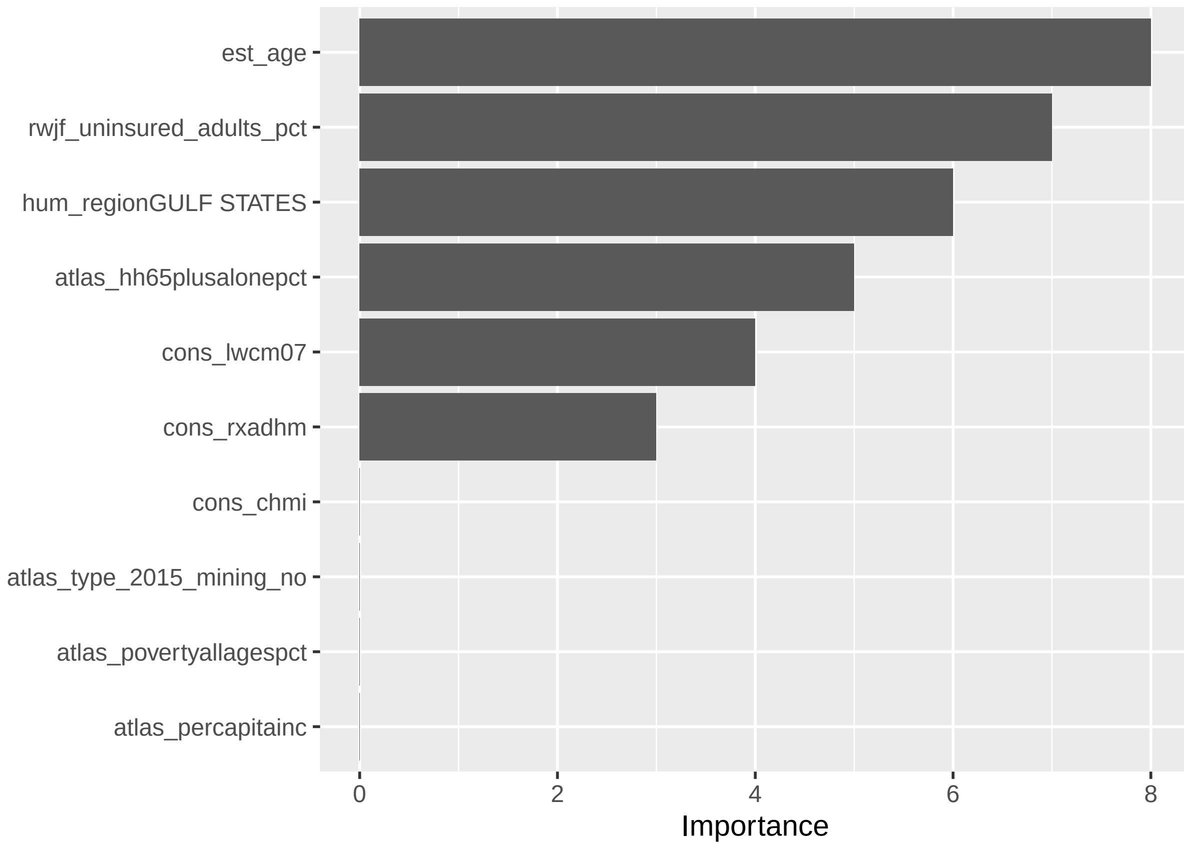
```
##   nprune degree
## 8      9      1
```

```
coef(model.mars$finalModel)
```

```
##                            (Intercept)                       h(est_age-99)
##                            -0.71284405                          1.17730183
##                          h(99-est_age) h(rwjf_uninsured_adults_pct-0.121157)
##                            -0.02194124                         -2.34116156
## h(0.121157-rwjf_uninsured_adults_pct)                 hum_regionGULF STATES
##                             4.97107373                          0.51029417
```

```
##     h(atlas_hh65plusalonepct-18.2505)               h(cons_lwcm07-0.13151)
##                            1.61017435                          -1.78115352
##                      h(2-cons_rxadhm)
##                           -0.12973872
```

```
vip(model.mars$finalModel)
```



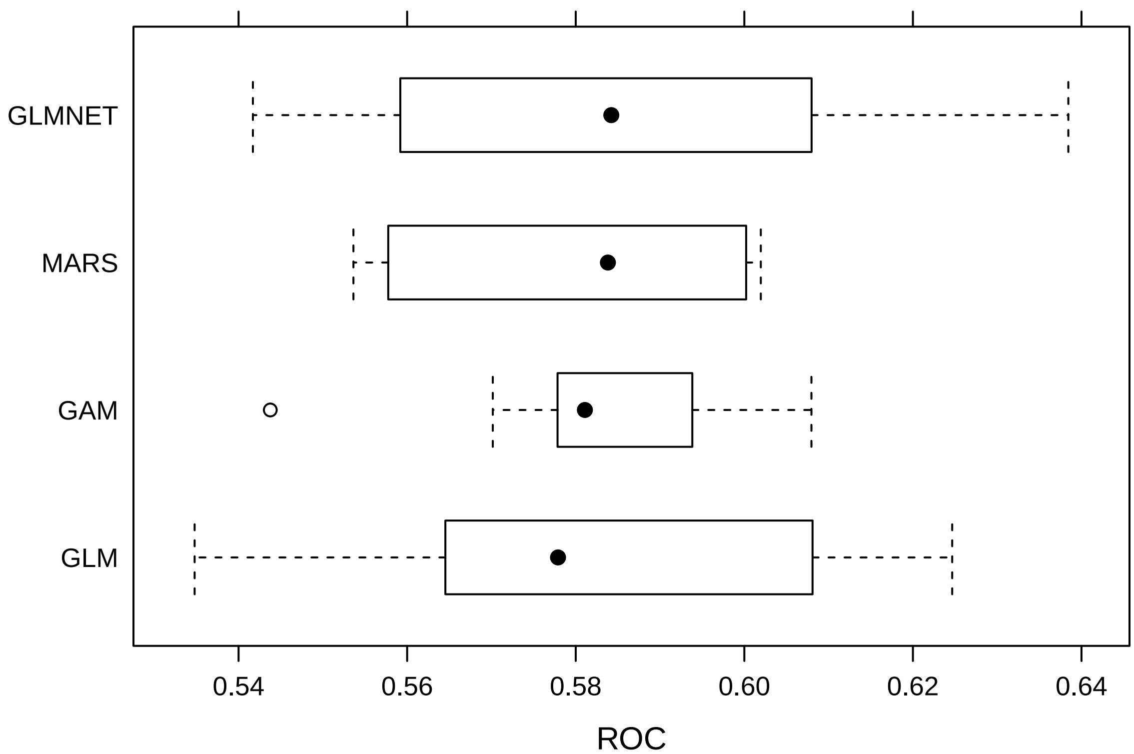
Compare Model

```
res = resamples(list(GLM = model.glm,
                     GLMNET = model.glmn,
                     GAM = model.gam,
                     MARS = model.mars))

summary(res)
```

```
##
## Call:
## summary.resamples(object = res)
##
## Models: GLM, GLMNET, GAM, MARS
## Number of resamples: 10
##
## ROC
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM    0.5347872 0.5652538 0.5778977 0.5828427 0.6070231 0.6246626    0
```

```
## GLMNET 0.5416995 0.5625843 0.5842221 0.5863508 0.6067827 0.6384390    0
## GAM    0.5437659 0.5778503 0.5810874 0.5828090 0.5922693 0.6079622    0
## MARS   0.5536271 0.5612606 0.5838160 0.5801600 0.5980946 0.6019549    0
##
## Sens
##             Min.   1st Qu. Median      Mean 3rd Qu. Max. NA's
## GLM    1.0000000 1.0000000      1 1.0000000       1    1    0
## GLMNET 1.0000000 1.0000000      1 1.0000000       1    1    0
## GAM    0.9978632 1.0000000      1 0.9995726       1    1    0
## MARS   0.9978587 0.9983974      1 0.9993585       1    1    0
##
## Spec
##        Min. 1st Qu. Median        Mean 3rd Qu.       Max. NA's
## GLM       0       0      0 0.000000000       0 0.00000000    0
## GLMNET    0       0      0 0.000000000       0 0.00000000    0
## GAM       0       0      0 0.000877193       0 0.00877193    0
## MARS      0       0      0 0.001754386       0 0.00877193    0
```

```
bwplot(res, metric = "ROC")
```



Test Data Performance

```
glm.pred = predict(model.glm, newdata = x2, type = "prob")[,2]
glmn.pred = predict(model.glmn, newdata = x2, type = "prob")[,2]
gam.pred = predict(model.gam, newdata = test_gam_x, type = "prob")[,2]
mars.pred = predict(model.mars, newdata = x2, type = "prob")[,2]

roc.glm = roc(data$covid_vaccination[-train], glm.pred)
```

```
## Setting levels: control = no_vacc, case = vacc
## Setting direction: controls < cases
```

```
roc.glmn = roc(data$covid_vaccination[-train], glmn.pred)
```

```
## Setting levels: control = no_vacc, case = vacc
## Setting direction: controls < cases
```

```
roc.gam = roc(data$covid_vaccination[-train], gam.pred)
```

```
## Setting levels: control = no_vacc, case = vacc
## Setting direction: controls < cases
```

```
roc.mars = roc(data$covid_vaccination[-train], mars.pred)
```

```
## Setting levels: control = no_vacc, case = vacc
## Setting direction: controls < cases
```

```
auc = c(roc.glm$auc[1], roc.glmn$auc[1],
roc.gam$auc[1], roc.mars$auc[1])

modelNames = c("glm","glmn","gam","mars")

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