

Final

2024-11-09

Set Up

```
libraries <- c("tidyverse", "splines", "rpart", "rpart.plot",  
  "randomForest", "gbm", "caret", "janitor", "survey", "irr")  
for (pkg in libraries) {  
  library(pkg, character.only = TRUE)  
}
```

Clean Up

The COVID-19 Vaccination Module is available on the Combined Land Line and Cell Phone survey and versions 1-3 for states that conducted more than one version of the questionnaires.

```
brfss <- read_csv("brfss.csv") %>%  
  clean_names()  
brfssv1 <- read_csv("v1_2023.csv") %>%  
  clean_names()  
brfssv2 <- read_csv("v2_2023.csv") %>%  
  clean_names()  
brfssv3 <- read_csv("v3_2023.csv") %>%  
  clean_names()
```

The following states had data on COVID-19 Vaccination status:

- Combined Land Line and Cell Phone data: Arkansas, Connecticut, Delaware, Georgia, Guam, Hawaii, Illinois, Michigan, Minnesota, Nebraska, New Hampshire, New Jersey, New Mexico, North Carolina, North Dakota, Puerto Rico, Vermont, Virginia, West Virginia, Wisconsin, Wyoming
- Combined Land Line and Cell Phone data, version 1: Oklahoma
- Combined Land Line and Cell Phone data, version 2: Maryland
- Combined Land Line and Cell Phone data, version 3: Maryland

```
# Subsetting and Combining States  
brfss_s <- brfss[brfss$state %in% c(5, 9, 10, 13, 66, 15, 17,  
  26, 27, 31, 33, 34, 35, 37, 38, 72, 50, 51, 54, 55, 56),  
  ]  
brfss_s <- brfss_s %>%  
  rename(finalwt = llcpwt) %>%  
  rename(finalcpwt = clcpwt)  
brfssv1_s <- brfssv1[brfssv1$state %in% c(40), ]  
brfssv1_s <- brfssv1_s %>%  
  rename(finalwt = lcpwtv1) %>%  
  rename(finalcpwt = clcwtv1)  
brfssv2_m <- brfssv2[brfssv2$state %in% c(24), ]  
brfssv2_m <- brfssv2_m %>%
```

```

mutate(finalwt = lcpwtv2 * 0.5) %>%
mutate(finalcpwt = clcwtv2 * 0.5) %>%
subset(select = -c(lcpwtv2, clcwtv2))
brfssv3_m <- brfssv3[brfssv3$state %in% c(24), ]
brfssv3_m <- brfssv3_m %>%
mutate(finalwt = lcpwtv3 * 0.5) %>%
mutate(finalcpwt = clcwtv3 * 0.5) %>%
subset(select = -c(lcpwtv3, clcwtv3))
brfss_t <- rbind(brfss_s, brfssv1_s, brfssv2_m, brfssv3_m)

```

Only people who had COVID-19 are able to develop long COVID.

```
# Subsetting Covid Positives
```

```
brfss_c <- subset(brfss_t, covidpo1 == 1)
```

```
# Long Covid - Defining missingness, factoring, creating
# level names
```

```
brfss_c$covidsm1[brfss_c$covidsm1 == 7 | brfss_c$covidsm1 ==
9] <- NA
```

```
brfss_c$covidsm1 <- factor(ifelse(brfss_c$covidsm1 == 1, "Pos",
"Neg"))
```

```
brfss_c$covidsm1 <- relevel(brfss_c$covidsm1, ref = "Pos")
```

```
# Defining predictor missingness Race
```

```
brfss_c$race[brfss_c$race == 9] <- NA
```

```
## Education
```

```
brfss_c$educag[brfss_c$educag == 9] <- NA
```

```
## Income
```

```
brfss_c$incomg1[brfss_c$incomg1 == 9] <- NA
```

```
## Insurance Status
```

```
brfss_c$hlthpl1[brfss_c$hlthpl1 == 9] <- NA
```

```
## Smoking
```

```
brfss_c$smoker3[brfss_c$smoker3 == 9] <- NA
```

```
# Factoring Predictors
```

```
variable <- c("age_g", "sexvar", "race", "educag", "incomg1",
"hlthpl1", "smoker3")
```

```
for (var in variable) {
  brfss_c[[var]] <- factor(brfss_c[[var]])
}
```

```
# BMI - Creating a continuous BMI variable, removing
```

```
# outliers
```

```
brfss_c$wt <- brfss_c$wtkg3/100
```

```
brfss_c$ht <- brfss_c$htm4/100
```

```
brfss_c$bmi <- brfss_c$wt/brfss_c$ht^2
```

```
iqr <- IQR(brfss_c$bmi, na.rm = TRUE)
```

```
qnt <- quantile(brfss_c$bmi, probs = c(0.25, 0.75), na.rm = TRUE)
```

```
lower <- qnt[1] - 1.5 * iqr
```

```
upper <- qnt[2] + 1.5 * iqr
```

```
brfss_c <- brfss_c %>%
```

```

subset(brfss_c$bmi > lower & brfss_c$bmi < upper)

# Mental Health - Defining missingness and 0
brfss_c$menthlth[brfss_c$menthlth == 77 | brfss_c$menthlth ==
99] <- NA
brfss_c$menthlth[brfss_c$menthlth == 88] <- 0

# COVID-19 Vaccine Status - Creating a factor vaccine
# variable starting with 0
brfss_c$cvax <- factor(ifelse(brfss_c$covidva1 == 2, 0, ifelse(brfss_c$covidnu2 ==
1, 1, ifelse(brfss_c$covidnu2 == 2, 2, ifelse(brfss_c$covidnu2 ==
3, 3, ifelse(brfss_c$covidnu2 == 4, 4, ifelse(brfss_c$covidnu2 ==
5, 5, NA))))))

# Complete Case Analysis
brfss_c <- na.omit(brfss_c[, c("age_g", "sexvar", "race", "educag",
"incomg1", "hlthpl1", "smoker3", "bmi", "menthlth", "cvax",
"covidsm1")])

```

There are 6,942 participants with long COVID and 46,886 without. I am creating class weights to address the imbalance.

```

class_counts <- table(brfss_c$covidsm1)
total_samples <- sum(class_counts)
num_classes <- length(class_counts)
class_weight <- total_samples/(num_classes * class_counts)
class_weight

##
##          Pos          Neg
## 3.8769807 0.5740306

brfss_c$classwt <- ifelse(brfss_c$covidsm1 == "Pos", class_weight["Pos"],
class_weight["Neg"])

```

Machine Learning Models

```

# Logistic Regression
set.seed(1234)
lc_log <- train(covidsm1 ~ age_g + sexvar + race + educag + incomg1 +
hlthpl1 + smoker3 + bmi + menthlth + cvax, data = brfss_c,
method = "glm", family = binomial(link = "logit"), trControl = trainControl(method = "cv",
number = 10, classProbs = TRUE, savePredictions = TRUE),
weights = brfss_c$classwt)
# Deriving Sensitivity, Specificity, Accuracy, Kappa
confusionMatrix(lc_log$pred$pred, lc_log$pred$obs)

## Confusion Matrix and Statistics
##
##          Reference
## Prediction  Pos  Neg
##          Pos 3936 16337
##          Neg 3006 30549
##

```

```
##               Accuracy : 0.6407
##               95% CI : (0.6366, 0.6447)
##      No Information Rate : 0.871
##      P-Value [Acc > NIR] : 1
##
##               Kappa : 0.1202
##
##      McNemar's Test P-Value : <2e-16
##
##               Sensitivity : 0.56698
##               Specificity : 0.65156
##      Pos Pred Value : 0.19415
##      Neg Pred Value : 0.91042
##               Prevalence : 0.12897
##      Detection Rate : 0.07312
##      Detection Prevalence : 0.37663
##      Balanced Accuracy : 0.60927
##
##      'Positive' Class : Pos
##
```

```
# Deriving Variable Importance
lc_log_vi <- varImp(lc_log$finalModel)
lc_log_vi[order(-lc_log_vi$Overall), , drop = FALSE]
```

```
##               Overall
## menthlth 30.7896302
## sexvar2  21.2593027
## bmi      11.5372208
## incomg17 10.9771059
## cvax4    10.7315813
## age_g5   10.1041195
## cvax5    9.8308270
## age_g4   9.5288422
## smoker34 9.0572932
## incomg16 8.7554823
## age_g6   6.7468310
## age_g3   6.7390838
## cvax3    6.2871783
## incomg15 6.2364498
## race3    5.3238129
## race4    5.2449761
## incomg14 5.1500566
## race8    3.9970241
## smoker33 2.9481852
## hlthpl12 2.5959671
## race7    2.2974561
## smoker32 2.0909296
## educag3  1.9355299
## educag4  1.5115062
## incomg13 1.3657638
## race5    1.0481121
## cvax1    0.9065789
## race2    0.7143208
## cvax2    0.5771904
```

```
## incomg12 0.4428572
## race6    0.4364542
## age_g2   0.3398988
## educag2  0.1156161
```

```
# Seeing Model Coefficients and Odds Ratios
exp(coef(lc_log$finalModel))
```

```
## (Intercept)    age_g2    age_g3    age_g4    age_g5    age_g6
## 1.8559852 0.9827138 0.7161553 0.6250138 0.6116441 0.7225679
## sexvar2      race2      race3      race4      race5      race6
## 0.6730545 1.0273378 0.6873606 1.3490776 1.1043024 0.9436239
## race7      race8    educag2    educag3    educag4    incomg12
## 0.8834348 0.8826536 0.9940004 0.9046815 1.0822418 1.0243546
## incomg13    incomg14    incomg15    incomg16    incomg17    hlthpl12
## 1.0734759 1.2953563 1.3425750 1.5363026 1.8579909 0.8830135
## smoker32    smoker33    smoker34      bmi    menthlth    cvax1
## 0.8793995 1.1219119 1.4071223 0.9811657 0.9666615 1.0417338
## cvax2      cvax3      cvax4      cvax5
## 1.0172691 1.2015593 1.4280568 1.4829118
```

```
summary(lc_log$finalModel)
```

```
##
## Call:
## NULL
##
## Coefficients:
##      Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.618416  0.094151   6.568 5.09e-11 ***
## age_g2      -0.017437  0.051302  -0.340  0.73393
## age_g3      -0.333858  0.049541  -6.739 1.59e-11 ***
## age_g4      -0.469982  0.049322  -9.529 < 2e-16 ***
## age_g5      -0.491605  0.048654 -10.104 < 2e-16 ***
## age_g6      -0.324944  0.048162  -6.747 1.51e-11 ***
## sexvar2     -0.395929  0.018624 -21.259 < 2e-16 ***
## race2        0.026971  0.037757   0.714  0.47503
## race3       -0.374896  0.070419  -5.324 1.02e-07 ***
## race4        0.299421  0.057087   5.245 1.56e-07 ***
## race5        0.099214  0.094660   1.048  0.29459
## race6       -0.058028  0.132952  -0.436  0.66251
## race7       -0.123938  0.053946  -2.297  0.02159 *
## race8       -0.124822  0.031229  -3.997 6.41e-05 ***
## educag2     -0.006018  0.052049  -0.116  0.90796
## educag3     -0.100172  0.051754  -1.936  0.05293 .
## educag4      0.079035  0.052289   1.512  0.13066
## incomg12     0.024063  0.054335   0.443  0.65787
## incomg13     0.070902  0.051914   1.366  0.17201
## incomg14     0.258786  0.050249   5.150 2.60e-07 ***
## incomg15     0.294589  0.047237   6.236 4.48e-10 ***
## incomg16     0.429379  0.049041   8.755 < 2e-16 ***
## incomg17     0.619496  0.056435  10.977 < 2e-16 ***
## hlthpl12    -0.124415  0.047926  -2.596  0.00943 **
## smoker32    -0.128516  0.061464  -2.091  0.03653 *
## smoker33     0.115034  0.039019   2.948  0.00320 **
```

```

## smoker34      0.341547    0.037710    9.057 < 2e-16 ***
## bmi          -0.019014    0.001648   -11.537 < 2e-16 ***
## menthlth     -0.033907    0.001101   -30.790 < 2e-16 ***
## cvax1         0.040886    0.045100    0.907  0.36463
## cvax2         0.017122    0.029664    0.577  0.56381
## cvax3         0.183620    0.029205    6.287 3.23e-10 ***
## cvax4         0.356315    0.033202   10.732 < 2e-16 ***
## cvax5         0.394008    0.040079    9.831 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 74621 on 53827 degrees of freedom
## Residual deviance: 70475 on 53794 degrees of freedom
## AIC: 97810
##
## Number of Fisher Scoring iterations: 5
# Naive Bayes
set.seed(1234)
lc_naive <- train(covidsml ~ age_g + sexvar + race + educag +
  incomg1 + hlthpl1 + smoker3 + bmi + menthlth + cvax, data = brfss_c,
  method = "naive_bayes", tuneGrid = data.frame(laplace = 0,
    usekernel = FALSE, adjust = FALSE), trControl = trainControl(method = "cv",
    number = 10, classProbs = TRUE, savePredictions = TRUE),
  weights = brfss_c$classwt)
# Deriving Sensitivity, Specificity, Accuracy, Kappa
confusionMatrix(lc_naive$pred$pred, lc_naive$pred$obs)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Pos  Neg
##           Pos 2097 8077
##           Neg 4845 38809
##
##           Accuracy : 0.7599
##           95% CI : (0.7563, 0.7635)
##           No Information Rate : 0.871
##           P-Value [Acc > NIR] : 1
##
##           Kappa : 0.1083
##
##           Mcnemar's Test P-Value : <2e-16
##
##           Sensitivity : 0.30207
##           Specificity : 0.82773
##           Pos Pred Value : 0.20611
##           Neg Pred Value : 0.88901
##           Prevalence : 0.12897
##           Detection Rate : 0.03896
##           Detection Prevalence : 0.18901
##           Balanced Accuracy : 0.56490
##

```

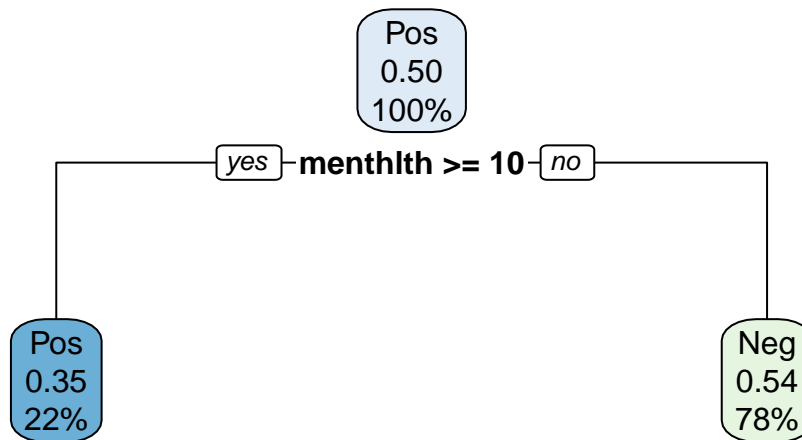
```

##          'Positive' Class : Pos
##
# Naive Bayes is incompatible with varImp()

# Setting up cp tuning grid
set.seed(1234)
lc_tree_cp <- rpart(covidsm1 ~ age_g + sexvar + race + educag +
  incomg1 + hlthpl1 + smoker3 + bmi + menthlth + cvax, data = brfss_c,
  method = "class", parms = list(split = "gini"), control = rpart.control(minsplit = 30,
    minbucket = 10), cp = 0, weights = brfss_c$classwt)
tg_lc <- data.frame(cp = lc_tree_cp$cptable[, 1])
# Classification Tree
set.seed(1234)
lc_tree <- train(covidsm1 ~ age_g + sexvar + race + educag +
  incomg1 + hlthpl1 + smoker3 + bmi + menthlth + cvax, data = brfss_c,
  method = "rpart", parms = list(split = "gini"), control = rpart.control(minsplit = 30,
    minbucket = 10), tuneGrid = tg_lc, trControl = trainControl(method = "cv",
    number = 10, selectionFunction = "oneSE", classProbs = TRUE,
    savePredictions = TRUE), weights = brfss_c$classwt)
lc_tree

## CART
##
## 53828 samples
##    10 predictor
##    2 classes: 'Pos', 'Neg'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 48445, 48446, 48444, 48445, 48445, 48446, ...
## Resampling results across tuning parameters:
##
##    cp          Accuracy    Kappa
## 0.01000000  0.6336287  0.09808499
## 0.02430715  0.7362354  0.10815231
## 0.13361051  0.6439378  0.07544708
##
## Accuracy was used to select the optimal model using the one SE rule.
## The final value used for the model was cp = 0.02430715.
rpart.plot(lc_tree$finalModel)

```



```
# Deriving Sensitivity, Specificity, Accuracy, Kappa
confusionMatrix(lc_tree$pred$pred, lc_tree$pred$obs)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Pos   Neg
##          Pos  8941 41199
##          Neg 11885 99459
##
##           Accuracy : 0.6713
##           95% CI   : (0.669, 0.6736)
##    No Information Rate : 0.871
##    P-Value [Acc > NIR] : 1
##
##           Kappa : 0.0853
##
##  McNemar's Test P-Value : <2e-16
##
##           Sensitivity : 0.42932
##           Specificity : 0.70710
##           Pos Pred Value : 0.17832
##           Neg Pred Value : 0.89326
##           Prevalence : 0.12897
##           Detection Rate : 0.05537
##           Detection Prevalence : 0.31050
##           Balanced Accuracy : 0.56821
##
##           'Positive' Class : Pos
##
```

```
# Deriving Variable Importance
```

```
lc_tree_vi <- varImp(lc_tree$finalModel)
lc_tree_vi[order(-lc_tree_vi$Overall), , drop = FALSE]
```

```
##           Overall
## menthlth 693.8085
## sexvar2  343.9497
## educag4  329.7100
## smoker34 233.2008
## bmi      160.4325
```



```
## age_g2      0.0000
## age_g3      0.0000
## age_g4      0.0000
## age_g5      0.0000
## age_g6      0.0000
## race2       0.0000
## race3       0.0000
## race4       0.0000
## race5       0.0000
## race6       0.0000
## race7       0.0000
## race8       0.0000
## educag2     0.0000
## educag3     0.0000
## incomg12    0.0000
## incomg13    0.0000
## incomg14    0.0000
## incomg15    0.0000
## incomg16    0.0000
## incomg17    0.0000
## hlthpl12    0.0000
## smoker32    0.0000
## smoker33    0.0000
## cvax1       0.0000
## cvax2       0.0000
## cvax3       0.0000
## cvax4       0.0000
## cvax5       0.0000
```

```
# Random Forest
```

```
set.seed(1234)
```

```
lc_rf <- randomForest(covidsm1 ~ age_g + sexvar + race + educag +
  incomg1 + hlthpl1 + smoker3 + bmi + menthlth + cvax, data = brfss_c,
  classwt = class_weight, importance = TRUE)
lc_rf
```

```
##
```

```
## Call:
```

```
## randomForest(formula = covidsm1 ~ age_g + sexvar + race + educag +      incomg1 + hlthpl1 + smoker3
```

```
##           Type of random forest: classification
```

```
##           Number of trees: 500
```

```
## No. of variables tried at each split: 3
```

```
##
```

```
##           OOB estimate of  error rate: 33.97%
```

```
## Confusion matrix:
```

```
##           Pos    Neg class.error
```

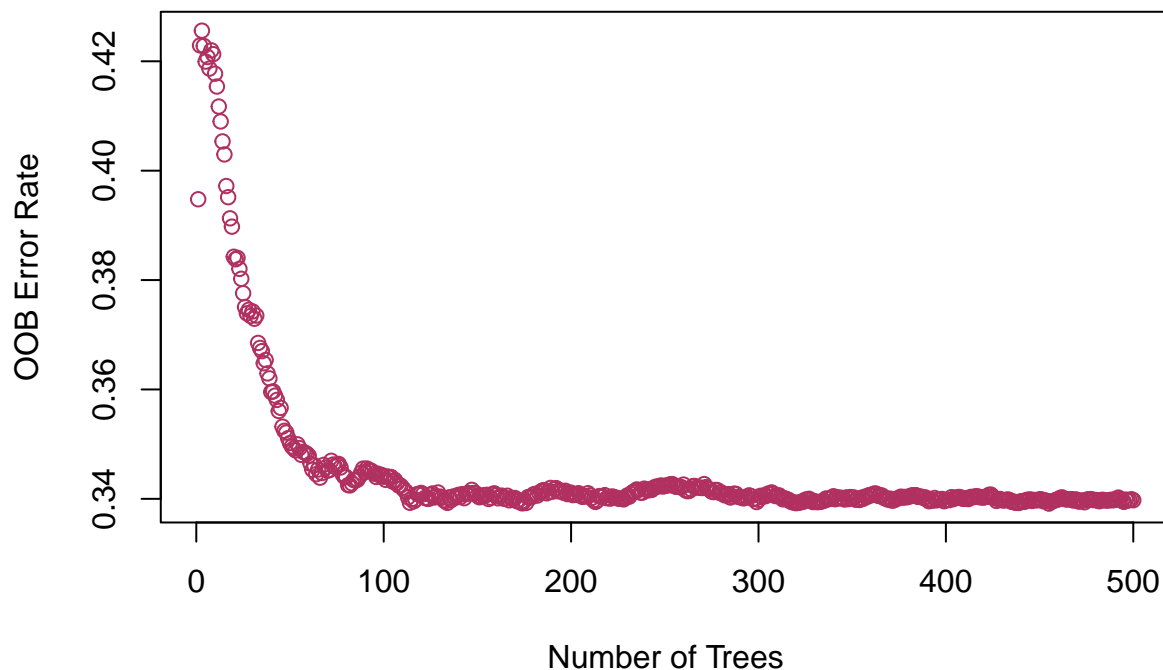
```
## Pos  1515  5427   0.7817632
```

```
## Neg 12861 34025   0.2743036
```

```
# Since the OOB Error Rate does appear to flatten, 500
```

```
# trees is sufficient
```

```
plot(lc_rf$err.rate[, 1], col = "maroon", xlab = "Number of Trees",
  ylab = "OOB Error Rate")
```



```
# Calculating Kappa
```

```
oob_predictions <- lc_rf$votes
oob_predicted_classes <- factor(ifelse(oob_predictions[, 1] >
  0.5, "Pos", "Neg"))
oob_predicted_classes <- relevel(oob_predicted_classes, ref = "Pos")
actual_classes <- brfss_c$covidsm1
conf_matrix <- table(Predicted = oob_predicted_classes, Actual = actual_classes)
kappa(conf_matrix)
```

```
##      Estimate Std.Err    2.5%   97.5% P-value
## kappa -0.03876 0.003698 -0.04601 -0.03151 1.059e-25
```

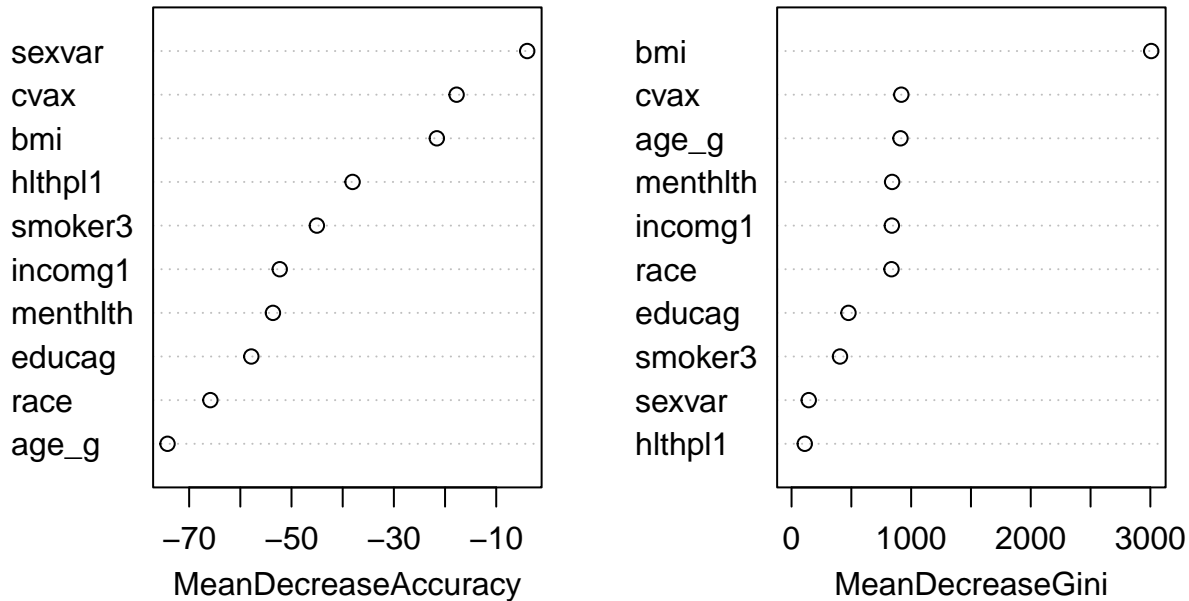
```
# Deriving Variable Importance
```

```
lc_rf_vi <- varImp(lc_rf)
lc_rf_vi[order(-lc_rf_vi$Neg), , drop = FALSE]
```

```
##           Pos           Neg
## sexvar  10.8155799  10.8155799
## cvax     0.1227626   0.1227626
## bmi     -7.8785674  -7.8785674
## incomg1 -11.7889363 -11.7889363
## hlthpl1 -16.6880161 -16.6880161
## age_g   -17.3285175 -17.3285175
## smoker3 -18.2429150 -18.2429150
## educag  -19.4314192 -19.4314192
## race    -21.7561637 -21.7561637
## menthlth -25.6022399 -25.6022399
```

```
varImpPlot(lc_rf, main = "Variable Importance")
```

Variable Importance



```
# Setting up tuning grid
tg_boostclass <- expand.grid(n.trees = c(5, 10, 15, 20, 25, 30,
    35, 40, 45, 50), interaction.depth = 1:6, shrinkage = c(0.2,
    0.25, 0.3, 0.35, 0.4, 0.45), n.minobsinnode = 10)
# Boosted Tree
set.seed(1234)
lc_boost <- train(covidsm1 ~ age_g + sexvar + race + educag +
    incomg1 + hlthpl1 + smoker3 + bmi + menthlth + cvax, data = brfss_c,
    method = "gbm", bag.fraction = 0.5, tuneGrid = tg_boostclass,
    trControl = trainControl(method = "cv", number = 10, classProbs = TRUE,
    savePredictions = TRUE), weights = brfss_c$classwt)

# Deriving Sensitivity, Specificity, Accuracy, Kappa
confusionMatrix(lc_boost$pred$pred, lc_boost$pred$obs)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      Pos      Neg
##      Pos 1432919 6108000
##      Neg 1066201 10770960
##
##           Accuracy : 0.6298
##           95% CI : (0.6296, 0.63)
##      No Information Rate : 0.871
##      P-Value [Acc > NIR] : 1
##
##           Kappa : 0.1137
##
```

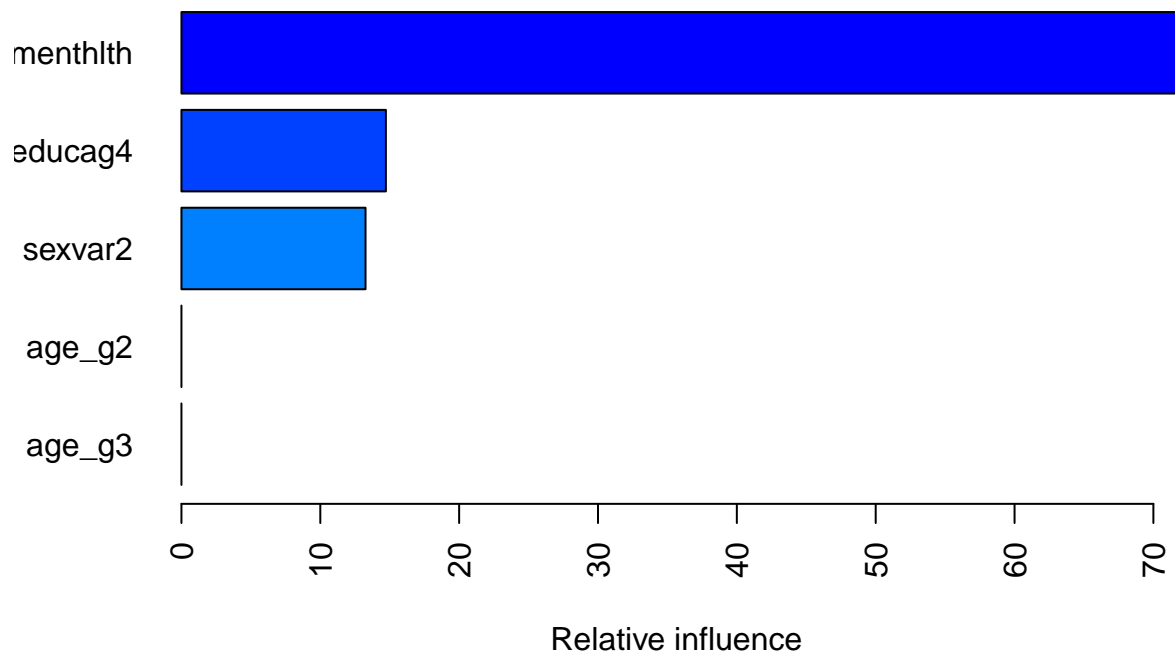
```
## McNemar's Test P-Value : <2e-16
##
##          Sensitivity : 0.57337
##          Specificity : 0.63813
##          Pos Pred Value : 0.19002
##          Neg Pred Value : 0.90993
##          Prevalence : 0.12897
##          Detection Rate : 0.07395
##          Detection Prevalence : 0.38915
##          Balanced Accuracy : 0.60575
##
##          'Positive' Class : Pos
##
```

Deriving Variable Importance

```
lc_boost_vi <- varImp(lc_boost$finalModel)
lc_boost_vi[order(-lc_boost_vi$Overall), , drop = FALSE]
```

```
##          Overall
## menthlth 401.60558
## educag4   82.10914
## sexvar2   73.93005
## age_g2     0.00000
## age_g3     0.00000
## age_g4     0.00000
## age_g5     0.00000
## age_g6     0.00000
## race2      0.00000
## race3      0.00000
## race4      0.00000
## race5      0.00000
## race6      0.00000
## race7      0.00000
## race8      0.00000
## educag2    0.00000
## educag3    0.00000
## incomg12   0.00000
## incomg13   0.00000
## incomg14   0.00000
## incomg15   0.00000
## incomg16   0.00000
## incomg17   0.00000
## hlthpl12   0.00000
## smoker32   0.00000
## smoker33   0.00000
## smoker34   0.00000
## bmi        0.00000
## cvax1      0.00000
## cvax2      0.00000
## cvax3      0.00000
## cvax4      0.00000
## cvax5      0.00000
```

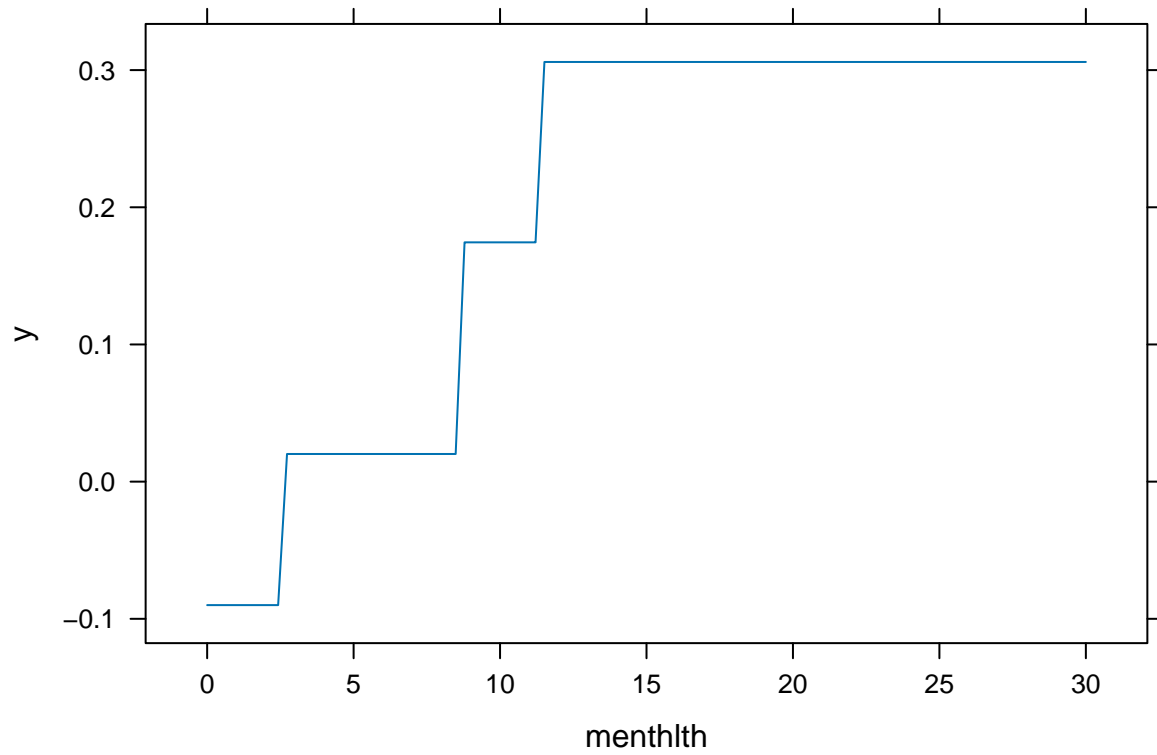
```
summary(lc_boost$finalModel, method = relative.influence, normalize = TRUE,
        las = 2, order = TRUE, cBars = 5)
```



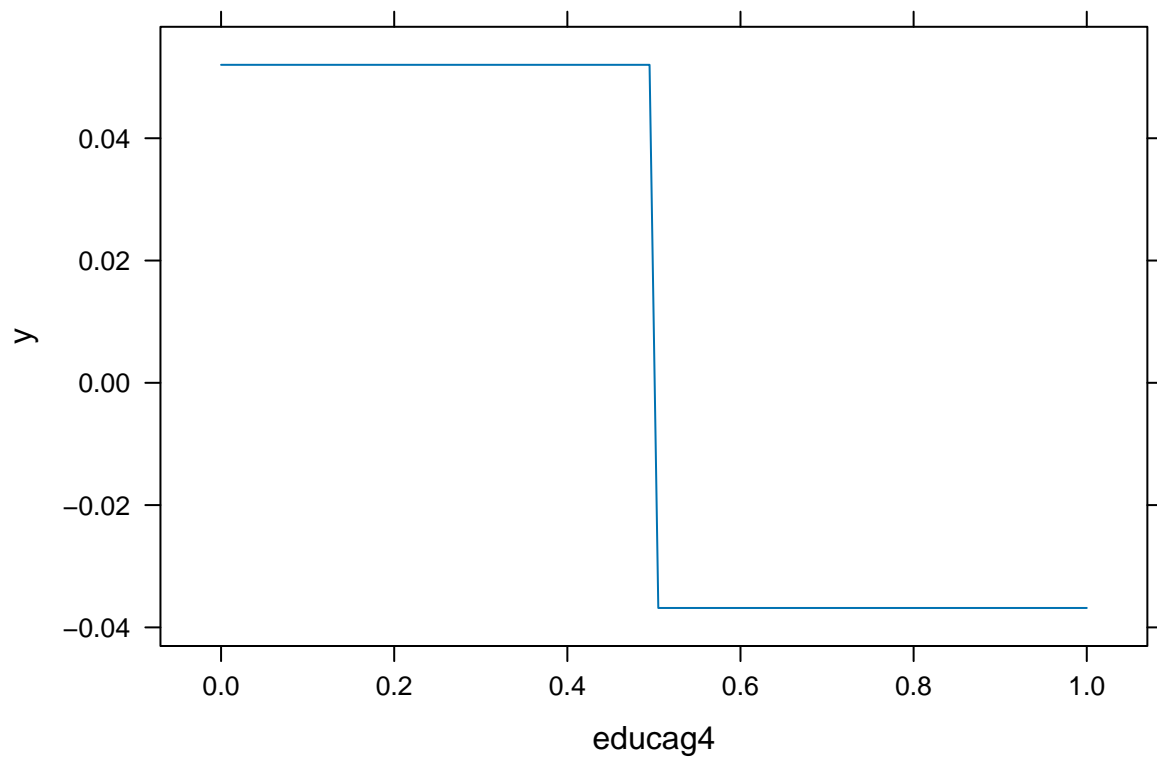
```
##          var  rel.inf
## menthlth menthlth 72.01817
## educag4   educag4 14.72427
## sexvar2   sexvar2 13.25755
## age_g2    age_g2  0.00000
## age_g3    age_g3  0.00000
## age_g4    age_g4  0.00000
## age_g5    age_g5  0.00000
## age_g6    age_g6  0.00000
## race2     race2  0.00000
## race3     race3  0.00000
## race4     race4  0.00000
## race5     race5  0.00000
## race6     race6  0.00000
## race7     race7  0.00000
## race8     race8  0.00000
## educag2   educag2  0.00000
## educag3   educag3  0.00000
## incomg12  incomg12 0.00000
## incomg13  incomg13 0.00000
## incomg14  incomg14 0.00000
## incomg15  incomg15 0.00000
## incomg16  incomg16 0.00000
## incomg17  incomg17 0.00000
## hlthpl12  hlthpl12 0.00000
## smoker32  smoker32 0.00000
## smoker33  smoker33 0.00000
## smoker34  smoker34 0.00000
## bmi       bmi     0.00000
## cvax1     cvax1   0.00000
## cvax2     cvax2   0.00000
## cvax3     cvax3   0.00000
## cvax4     cvax4   0.00000
```

```
## cvax5      cvax5  0.00000
```

```
plot(lc_boost$finalModel, i = "menthlth")
```



```
plot(lc_boost$finalModel, i = "educag4")
```



```
plot(lc_boost$finalModel, i = "sexvar2")
```

