COVID STEROID 2: HTE Analysis

Bryan Blette

2022-06-07

Analysis

First the required R packages are loaded, as well as the data set.

```
rm(list = ls())
library(BART)
library(caret)
library(rpart)
library(rpart.plot)

# Load data from appropriate directory (edit as needed)
dat <- read.csv2("~/Downloads/synth_covid.csv")
```

Next we do a small amount of data cleaning/preparation.

```
# Clean up data variables types and remove the small amount of missing data
dat$resp_sup <- as.factor(dat$resp_sup)
dat$dead90 <- ifelse(dat$dead90 == TRUE, 1, 0)
dat <- dat[complete.cases(dat), ]

# Standardize continuous covariates
dat$age <- (dat$age - mean(dat$age)) / sd(dat$age)
dat$BL9_Weight <- (dat$BL9_Weight - mean(dat$BL9_Weight)) / sd(dat$BL9_Weight)

# Make datasets under each counterfactual
dat1 <- dat0 <- dat
dat1$allocation <- TRUE
dat0$allocation <- FALSE
```

Then we run a BART analysis focused on the binary mortality outcome.

```
# Create 10 folds of the data set for cross-validation
set.seed(60622)
folds <- createFolds(dat$dead90, k = 10, list = TRUE, returnTrain = FALSE)

# Initialize output matrices for prediction error from each model
cvoutput <- expand.grid(1:3, c(0.25, 0.5, 0.95), c(50, 200, 400), NA)
colnames(cvoutput) <- c("Power", "Base", "Ntrees", "CVMSE")
mse <- array(NA, dim = c(27, 10))

# Perform cross validation (may take >2 hours)
```

```
for (hp in 1:27) {

  for (i in 1:10) {

    # BART model
    bartmod <- lbart(x.train = dat[-folds[[i]], c(1, 4:13)],
                     y.train = dat$dead90[-folds[[i]]],
                     x.test = dat[folds[[i]], c(1, 4:13)],
                     power = cvoutput$Power[hp], base = cvoutput$Base[hp],
                     ntree = cvoutput$Ntrees[hp])
    pred <- exp(colMeans(bartmod$yhat.test)) /
            (1 + exp(colMeans(bartmod$yhat.test)))
    mse[hp, i] <- mean((dat$dead90[folds[[i]]] - pred)^2)

  }

}

# Calculate 10-fold CV error for each hyperparameter combination
cvoutput$CVMSE <- rowMeans(mse)

# Fit final model under hyperparameters with minimum CV error
set.seed(60622)
bartmod1 <- lbart(x.train = dat[, c(1, 4:13)], y.train = dat$dead90,
                  x.test = dat1[, c(1, 4:13)],
                  power = cvoutput$Power[which.min(cvoutput$CVMSE)],
                  base = cvoutput$Base[which.min(cvoutput$CVMSE)],
                  ntree = cvoutput$Ntrees[which.min(cvoutput$CVMSE)])
bartmod0 <- lbart(x.train = dat[, c(1, 4:13)], y.train = dat$dead90,
                  x.test = dat0[, c(1, 4:13)],
                  power = cvoutput$Power[which.min(cvoutput$CVMSE)],
                  base = cvoutput$Base[which.min(cvoutput$CVMSE)],
                  ntree = cvoutput$Ntrees[which.min(cvoutput$CVMSE)])
```

Then conditional average treatment effects are estimated using add math.

```
dat$cate <-
  exp(colMeans(bartmod1$yhat.test)) / (1 + exp(colMeans(bartmod1$yhat.test))) -
  exp(colMeans(bartmod0$yhat.test)) / (1 + exp(colMeans(bartmod0$yhat.test)))
```

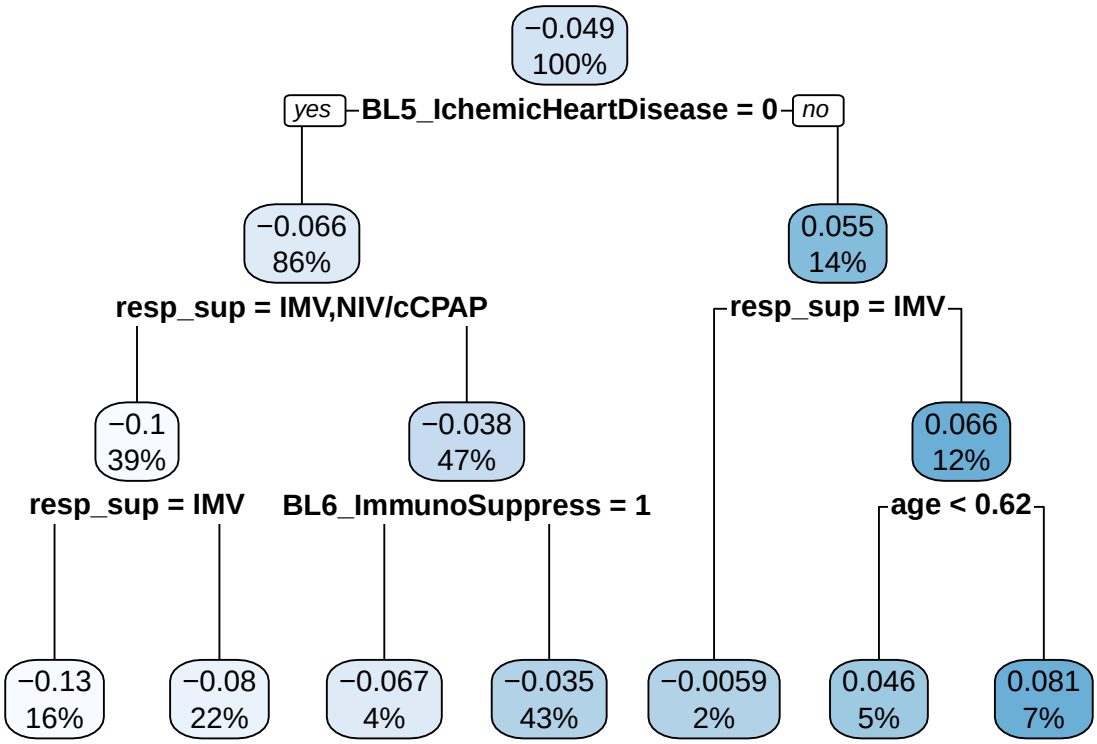
This full process is then repeated for the continuous outcome (days alive without life support by day 90).

```
# Initialize output for prediction error from each model
cvoutput$CVMSE_c <- NA
mse_c <- array(NA, dim = c(27, 10))

# Perform cross validation (should take much less time than the binary outcome)
set.seed(60622)
for (hp in 1:27) {

  for (i in 1:10) {

    # BART model
```

```
    bartmod_c <- wbart(x.train = dat[-folds[[i]], c(1, 4:13)],
                       y.train = dat$dawols90[-folds[[i]]],
                       x.test = dat[folds[[i]], c(1, 4:13)],
                       power = cvoutput$Power[hp], base = cvoutput$Base[hp],
                       ntree = cvoutput$Ntrees[hp])
    pred_c <- colMeans(bartmod_c$yhat.test)
    mse_c[hp, i] <- mean((dat$dawols90[folds[[i]]] - pred_c)^2)

  }

}

# Calculate 10-fold CV error for each hyperparameter combination
cvoutput$CVMSE_c <- rowMeans(mse_c)

# Fit final models under hyperparameters with minimum CV error
set.seed(60622)
bartmod1_c <- wbart(x.train = dat[, c(1, 4:13)], y.train = dat$dawols90,
                    x.test = dat1[, c(1, 4:13)],
                    power = cvoutput$Power[which.min(cvoutput$CVMSE_c)],
                    base = cvoutput$Base[which.min(cvoutput$CVMSE_c)],
                    ntree = cvoutput$Ntrees[which.min(cvoutput$CVMSE_c)])
bartmod0_c <- wbart(x.train = dat[, c(1, 4:13)], y.train = dat$dawols90,
                    x.test = dat0[, c(1, 4:13)],
                    power = cvoutput$Power[which.min(cvoutput$CVMSE_c)],
                    base = cvoutput$Base[which.min(cvoutput$CVMSE_c)],
                    ntree = cvoutput$Ntrees[which.min(cvoutput$CVMSE_c)])

# Estimate CATEs
dat$cate_c <- colMeans(bartmod1_c$yhat.test) - colMeans(bartmod0_c$yhat.test)
```

Finally, the "fit-the-fit" approach is used to find subgroups exhibiting heterogeneity of treatment effect, starting with the binary outcome. In particular, a CART model is fit with the CATE for 90-day mortality as the outcome and the covariates as possible predictors. The model is first fit under default CART hyperparameter settings.

```
# CART model for 90 day mortality with default CART hyperparameter and
# all covariates considered
cartmod <- rpart(cate ~ ., data = dat[, c(4:14)], method = "anova")
rpart.plot(cartmod)
```



Now we prune the tree for interpretability using the stepwise approach of Hu et al. In particular, covariates are added to the CART model sequentially according to greatest increase in model R^2 until an increase of less than 1% occurs.

```
# Pruned tree using stepwise approach of Hu et al.
r2 <- c()
covar_include <- c()
covar_cols <- 4:13
currentr2 <- 0
maxcovars <- 4

for (numcovars in 1:maxcovars) {

  for (covar in covar_cols) {
    cartmod <- rpart(cate ~ ., data = dat[, c(covar, covar_include, 14)],
                     method = "anova")
    r2[which(covar_cols == covar)] <-
      1 - printcp(cartmod)[dim(printcp(cartmod))[1], 3]
  }

  if ((max(r2) - currentr2) / currentr2 > 0.01) {
    covar_include <- c(covar_include, covar_cols[which.max(r2)])
    covar_cols <- covar_cols[covar_cols != covar_cols[which.max(r2)]]
    currentr2 <- max(r2)
    r2 <- c()
  } else {
```

```
    break
  }

}
```

Print out the number of covariates selected and the resulting CART model output.

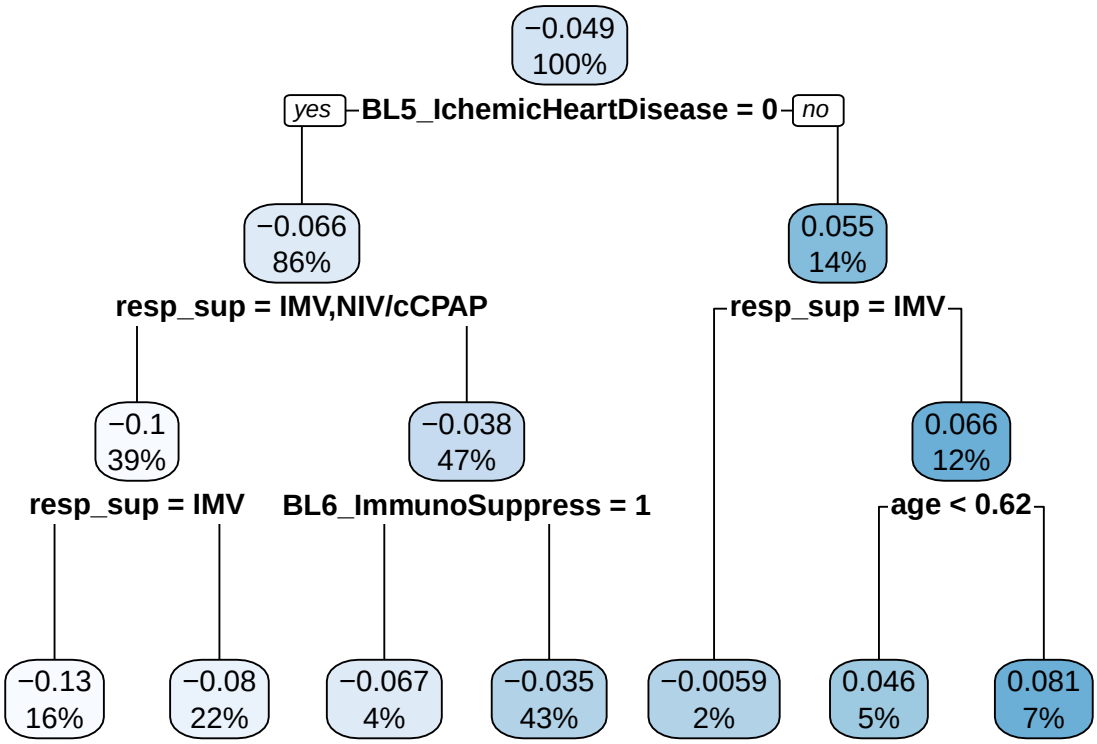
```
print("Number of covariates:")
```

```
## [1] "Number of covariates:"
```

```
print(numcovars - 1)
```

```
## [1] 3
```

```
cartmod <- rpart(cate ~ ., data = dat[, c(covar_include, 14)],
                 method = "anova")
rpart.plot(cartmod)
```

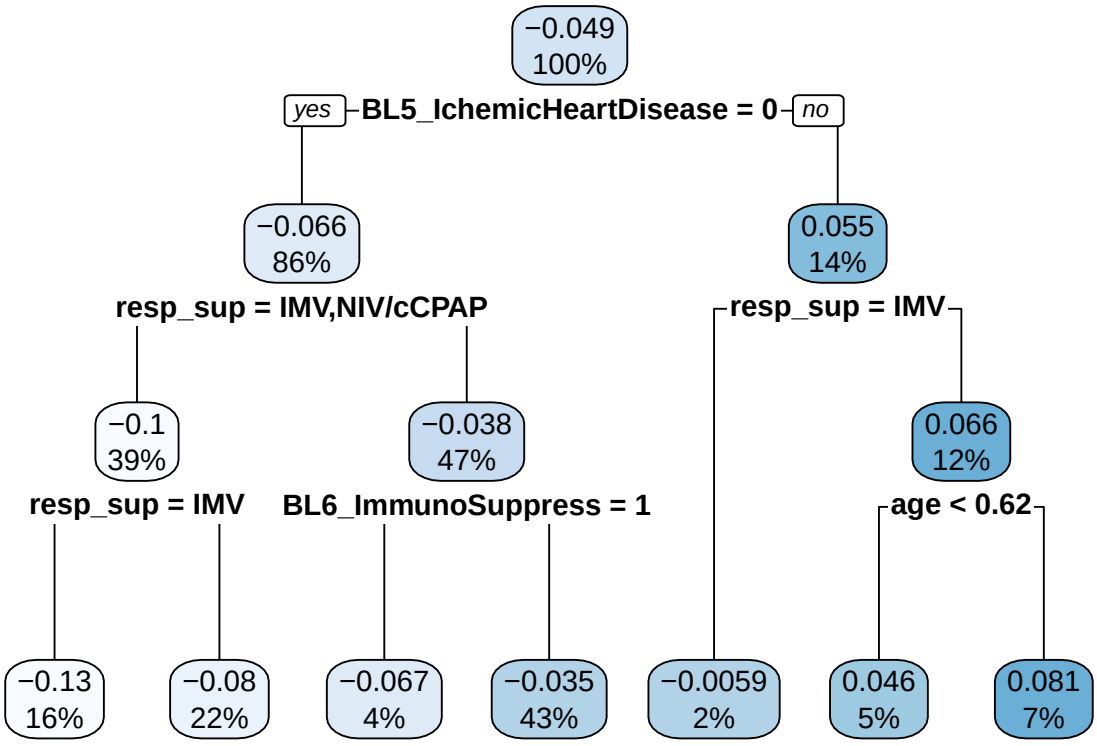


Finally we explore further pruning:

```
printcp(cartmod)
```

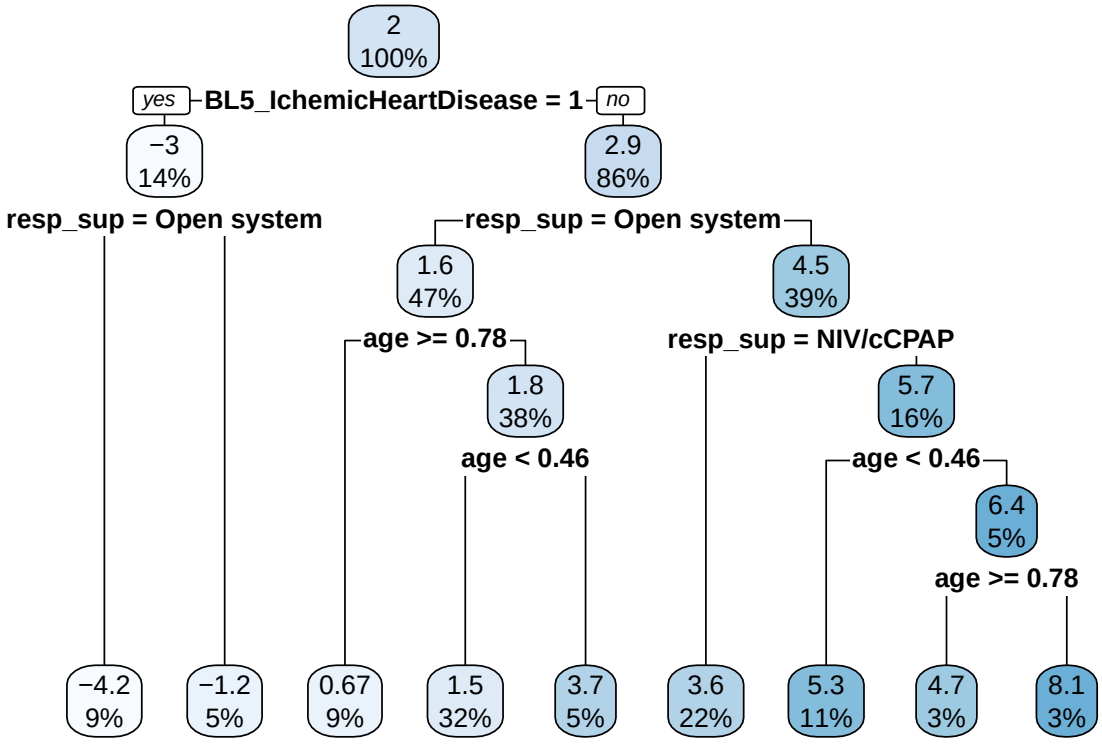
```
##
## Regression tree:
## rpart(formula = cate ~ ., data = dat[, c(covar_include, 14)],
##     method = "anova")
##
## Variables actually used in tree construction:
## [1] age                     BL5_IchemicHeartDisease BL6_ImmunoSuppress
## [4] resp_sup
##
## Root node error: 3.4469/968 = 0.0035609
##
## n= 968
##
##         CP nsplit rel error  xerror      xstd
## 1 0.508314      0   1.00000 1.00255 0.0495106
## 2 0.237191      1   0.49169 0.49290 0.0196796
## 3 0.065258      2   0.25449 0.25552 0.0116221
## 4 0.028046      3   0.18924 0.19119 0.0096375
## 5 0.010370      4   0.16119 0.16336 0.0077576
## 6 0.010018      5   0.15082 0.16196 0.0077717
## 7 0.010000      6   0.14080 0.15830 0.0076955
```

```
prunedtree <-
  prune(cartmod,
        cp = cartmod$cptable[which.min(cartmod$cptable[, "xerror"]), "CP"])
rpart.plot(prunedtree)
```



Next the same fit-the-fit approach is used to summarize the results for the continuous outcome, starting with a CART model using all covariates and default hyperparameter.

```
# CART model for 90 day mortality with default CART hyperparameter and
# all covariates considered
cartmod <- rpart(cate_c ~ ., data = dat[, c(4:13, 15)], method = "anova")
rpart.plot(cartmod)
```



Now we prune the tree for interpretability using the stepwise approach of Hu et al.

```
# Pruned tree using stepwise approach of Hu et al.
r2 <- c()
covar_include <- c()
covar_cols <- 4:13
currentr2 <- 0
maxcovars <- 4

for (numcovars in 1:maxcovars) {

  for (covar in covar_cols) {
    cartmod <- rpart(cate_c ~ ., data = dat[, c(covar, covar_include, 15)],
                     method = "anova")
    r2[which(covar_cols == covar)] <-
      1 - printcp(cartmod)[dim(printcp(cartmod))[1], 3]
  }
```

```
  if ((max(r2) - currentr2) / currentr2 > 0.01) {
    covar_include <- c(covar_include, covar_cols[which.max(r2)])
    covar_cols <- covar_cols[covar_cols != covar_cols[which.max(r2)]]
    currentr2 <- max(r2)
    r2 <- c()
  } else {
    break
  }

}
```

Print out the number of covariates selected and the resulting CART model output.

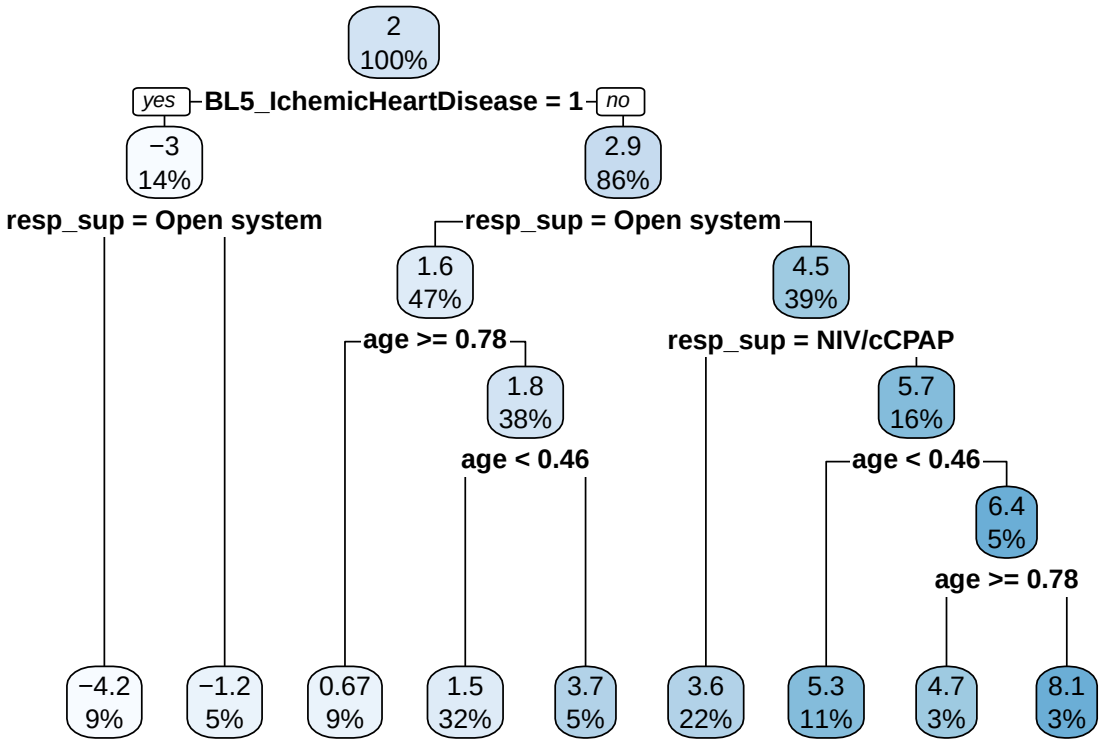
```
print("Number of covariates:")
```

```
## [1] "Number of covariates:"
```

```
print(numcovars - 1)
```

```
## [1] 3
```

```
cartmod <- rpart(cate_c ~ ., data = dat[, c(covar_include, 15)],
                 method = "anova")
rpart.plot(cartmod)
```

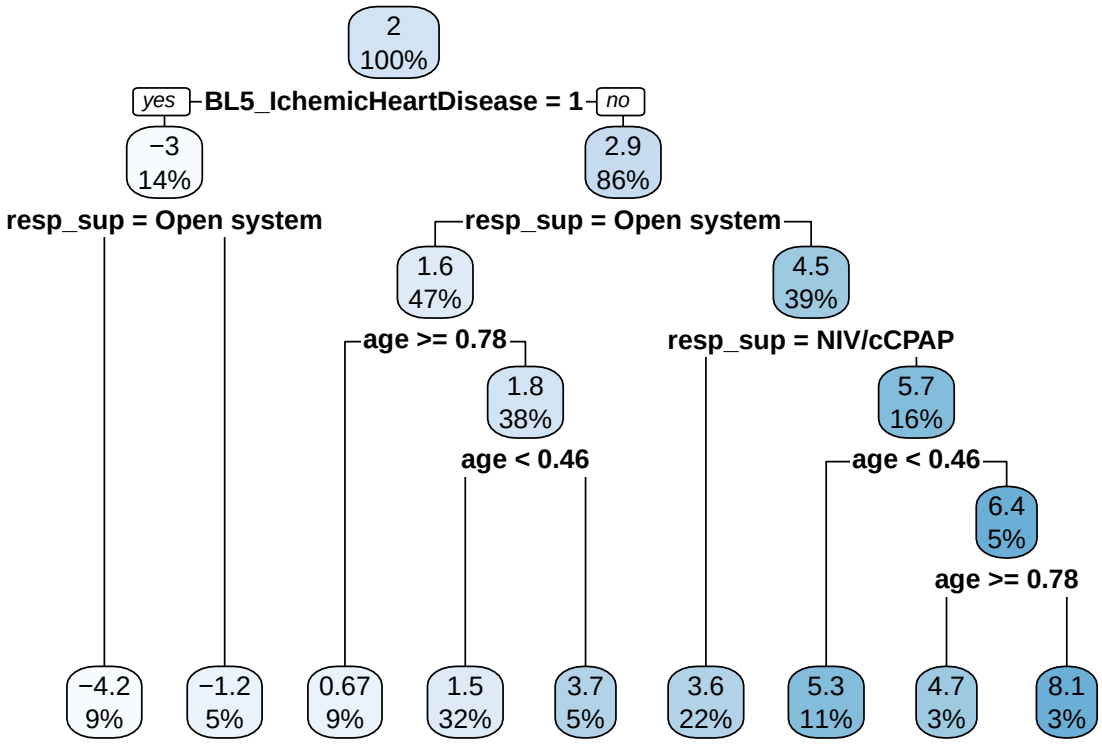


Finally we explore further pruning:

```
printcp(cartmod)
```

```
##
## Regression tree:
## rpart(formula = cate_c ~ ., data = dat[, c(covar_include, 15)],
##     method = "anova")
##
## Variables actually used in tree construction:
## [1] age                      BL5_IchemicHeartDisease resp_sup
##
## Root node error: 7838.7/968 = 8.0978
##
## n= 968
##
##         CP nsplit rel error  xerror      xstd
## 1 0.531003      0  1.000000 1.00255 0.0531382
## 2 0.221122      1  0.468997 0.47195 0.0245254
## 3 0.049294      2  0.247875 0.25025 0.0168189
## 4 0.036653      3  0.198582 0.20074 0.0145302
## 5 0.019989      4  0.161928 0.16466 0.0122864
## 6 0.011861      6  0.121951 0.12485 0.0102299
## 7 0.010000      8  0.098229 0.11171 0.0089859
```

```
prunedtree <-
  prune(cartmod,
        cp = cartmod$cptable[which.min(cartmod$cptable[, "xerror"]), "CP"])
rpart.plot(prunedtree)
```


2
100%
yes
BL5_IchemicHeartDisease = 1
no
−3
14%
2.9
86%
resp_sup = Open system
resp_sup = Open system
1.6
47%
4.5
39%
age >= 0.78
resp_sup = NIV/cCPAP
1.8
38%
5.7
16%
age < 0.46
age < 0.46
6.4
5%
age >= 0.78
−4.2
9%
−1.2
5%
0.67
9%
1.5
32%
3.7
5%
3.6
22%
5.3
11%
4.7
3%
8.1
3%