lab_7

Lindsey Greenhill

4/5/2022

Question 1

```
## [1] "In generation 1, parent_rank = 57.9, child_rank = 53.6329"
## [1] "In generation 2, parent_rank = 53.6329, child_rank = 52.1351479"
## [1] "In generation 3, parent_rank = 52.1351479, child_rank = 51.6094369129"
## [1] "In generation 4, parent_rank = 51.6094369129, child_rank = 51.4249123564279"
## [1] "In generation 5, parent_rank = 51.4249123564279, child_rank = 51.3601442371062"
## [1] "In generation 6, parent_rank = 51.3601442371062, child_rank = 51.3374106272243"
## [1] "In generation 7, parent_rank = 51.3374106272243, child_rank = 51.3294311301557"
```

```
## [1] "In generation 1, parent_rank = 32.7, child_rank = 44.7877"
## [1] "In generation 2, parent_rank = 44.7877, child_rank = 49.0304827"
## [1] "In generation 3, parent_rank = 49.0304827, child_rank = 50.5196994277"
## [1] "In generation 4, parent_rank = 50.5196994277, child_rank = 51.0424144991227"
## [1] "In generation 5, parent_rank = 51.0424144991227, child_rank = 51.2258874891921"
## [1] "In generation 6, parent_rank = 51.2258874891921, child_rank = 51.2902865087064"
## [1] "In generation 7, parent_rank = 51.2902865087064, child_rank = 51.312890564556"
```

```
## [1] "In generation 1, parent_rank = 32.7, child_rank = 34.556"
## [1] "In generation 2, parent_rank = 34.556, child_rank = 35.07568"
## [1] "In generation 3, parent_rank = 35.07568, child_rank = 35.2211904"
## [1] "In generation 4, parent_rank = 35.2211904, child_rank = 35.261933312"
## [1] "In generation 5, parent_rank = 35.261933312, child_rank = 35.27334132736"
## [1] "In generation 6, parent_rank = 35.27334132736, child_rank = 35.2765355716608"
## [1] "In generation 7, parent_rank = 35.2765355716608, child_rank = 35.277429960065"
```

```
## [1] "In generation 1, parent_rank = 36.17, child_rank = 45.5442"
## [1] "In generation 2, parent_rank = 45.5442, child_rank = 47.981492"
## [1] "In generation 3, parent_rank = 47.981492, child_rank = 48.61518792"
## [1] "In generation 4, parent_rank = 48.61518792, child_rank = 48.7799488592"
## [1] "In generation 5, parent_rank = 48.7799488592, child_rank = 48.822786703392"
## [1] "In generation 6, parent_rank = 48.822786703392, child_rank = 48.8339245428819"
## [1] "In generation 7, parent_rank = 48.8339245428819, child_rank = 48.8368203811493"
```

The steady state prediction for Hispanic children is 48.8

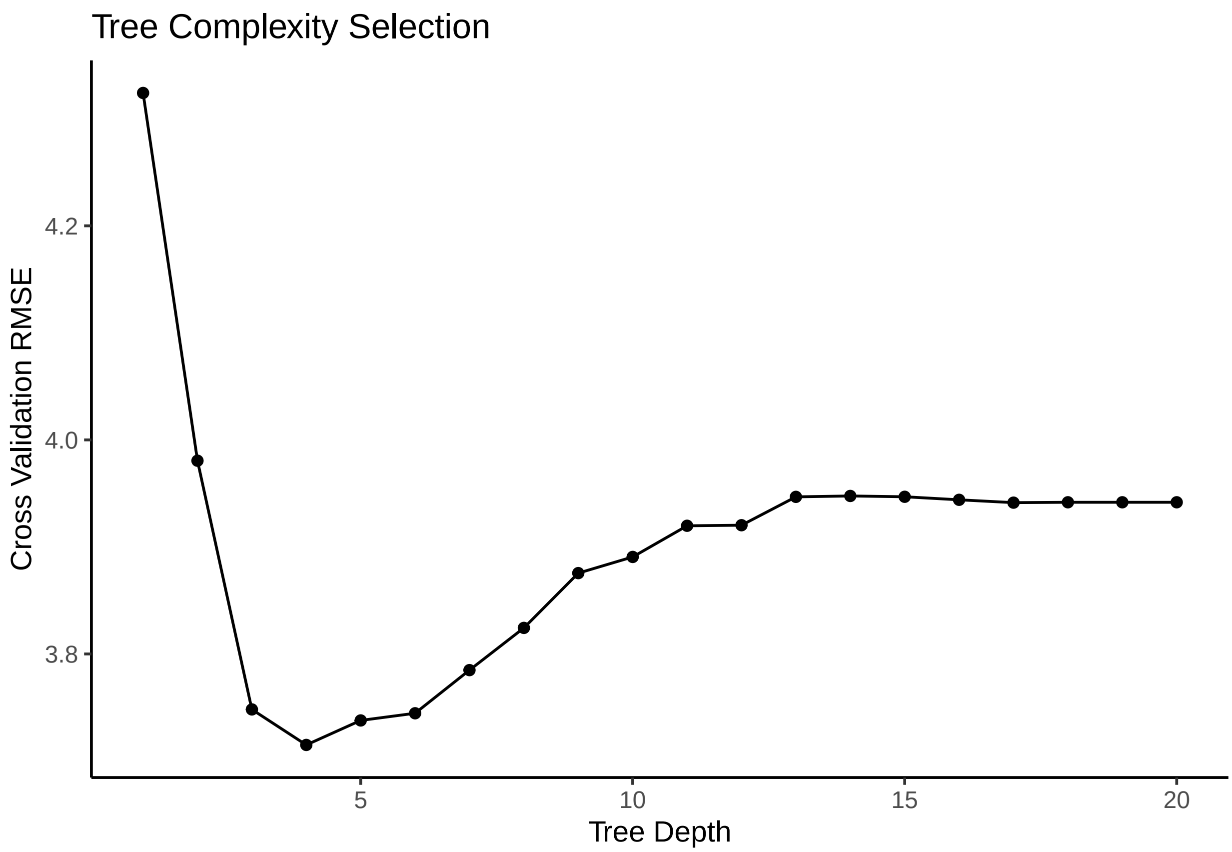
Question 2

Cross Validation helps avoid the overfit problem by allowing you to test different levels of tree complexities on data you didn't use to create the model and allows you to systematically select the level of complexity that minimizes out of sample prediction error.

Question 3

Part a

I am using P_26 – fraction of residents with a college degree or more in 2000 and P_55 (physically unhealthy days per month)

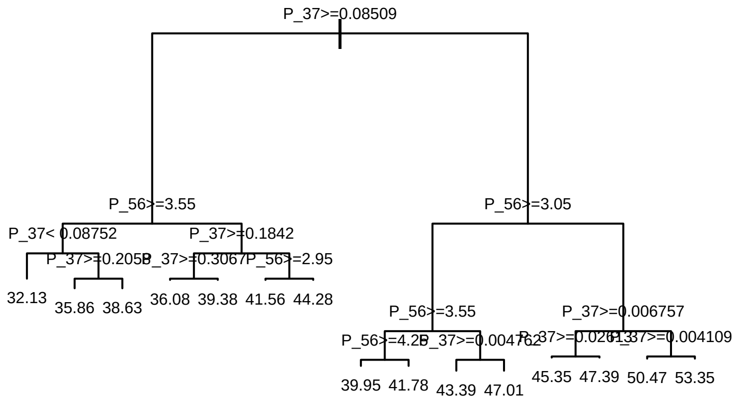


Part b

Using the graph above, a tree depth of 4 is optimal

Part c

P_37 (share black in 2000) and P_56 (mentally unhealthy days per month) are being used in the first few splits in the tree



Part d

Question 4

Random forests improve upon decision using bagging because it allows you to take the average of many different decision trees to find the optimal one and eliminates the "noise" of the individual models. Random forests use input randomization to decrease the correlation between the predictions of the trees.

Question 5

```
##
## Call:
##  randomForest(formula = kfr_pooled_pooled_p25 ~ P_26 + P_55, data = training,      ntree = 1000, mtry
##                Type of random forest: regression
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 15.58605
##                     % Var explained: 41.66
```

Question 6

```
##
```

```
## Call:
##  randomForest(formula = kfr_pooled_pooled_p25 ~ ., data = training,      ntree = 1000, mtry = 121, i
##                Type of random forest: regression
##                      Number of trees: 1000
## No. of variables tried at each split: 121
##
##           Mean of squared residuals: 4.823905
##                     % Var explained: 81.94
```