ML interpolation

Amber Lee

6/22/2021

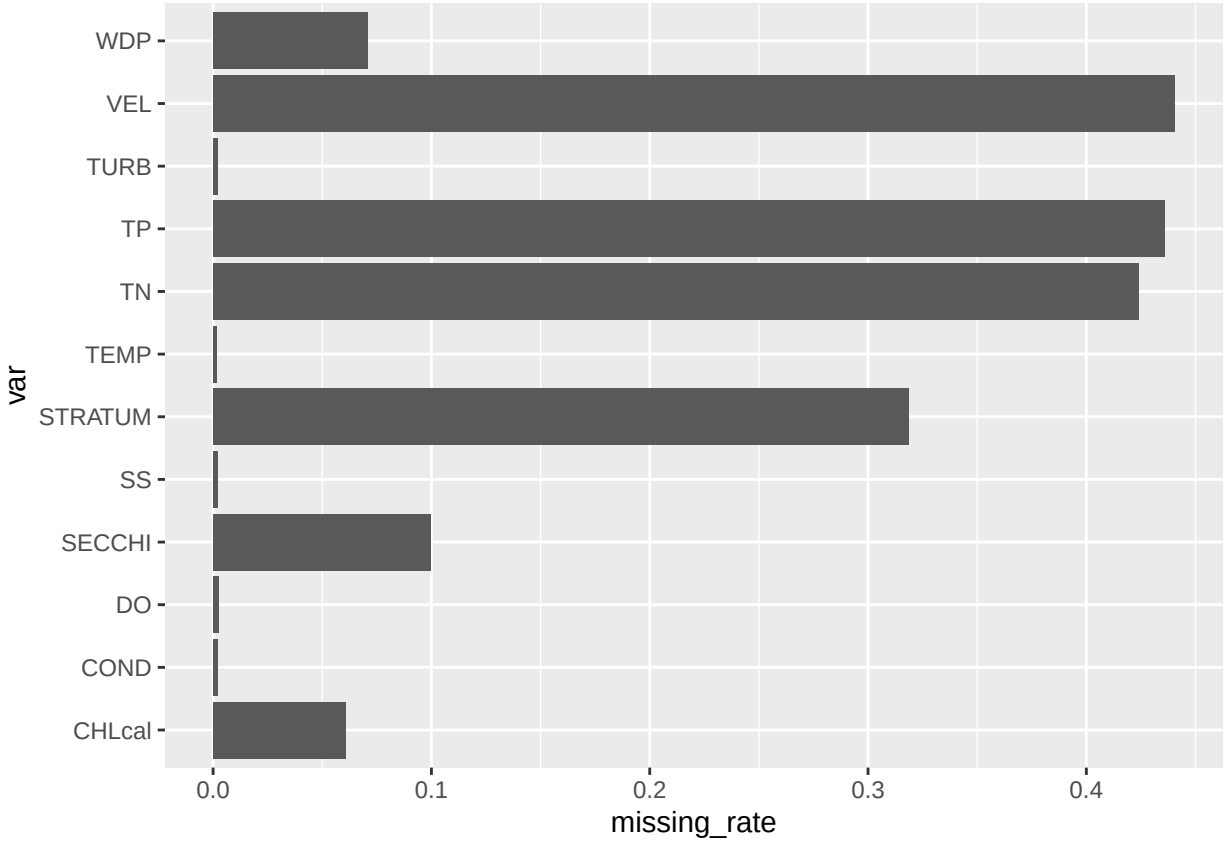
```
library(tidyverse)
library(stringr)
library(lubridate)

library(caret)
library(rattle)
```

```
water20 <- read.csv("cleaned_data.csv", header = TRUE)
```

```
# remove the QF code var
water_var <- names(water20)[str_detect(names(water20), "QF", negate = T)]

water20 <- water20 %>% select(all_of(water_var))
```

```
var_missing_rate <- sapply(water20, function(x) sum(is.na(x))/length(x))

var_missing_rate <- data.frame(var = names(var_missing_rate),
                               missing_rate = unname(var_missing_rate))
```

```
var_missing_rate %>% filter(missing_rate > 0) %>%
  ggplot(aes(x = var, y = missing_rate)) +
  geom_bar(stat = "identity") +
  coord_flip()
```



*

regression (what are the steps for it?)

- classification and regression trees (CART)
- neural networks
- support vector machines

CART

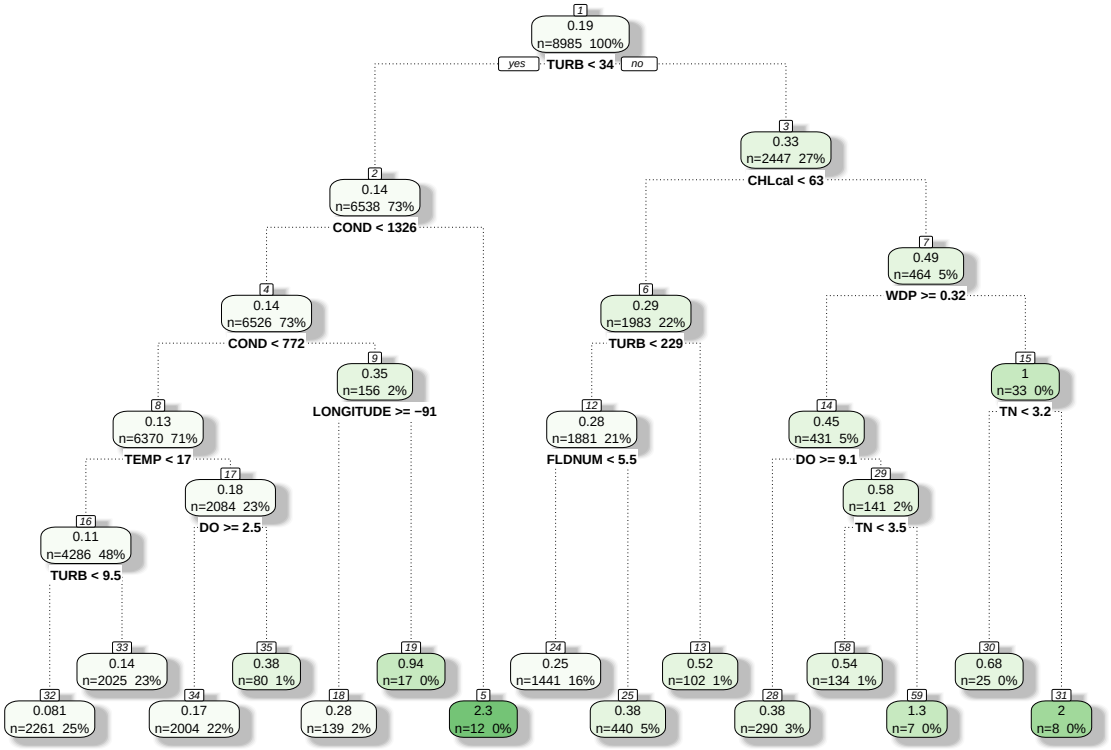
http://st47s.com/Math154/Notes/class.html#r-cart-example

TP

```
options(expressions = 5e5)

narm_water20 <- water20 %>%
  filter_at(vars(water_var), all_vars(!is.na(.))) %>%
  mutate(nice_date = mdy(DATE),
         year = year(nice_date),
         quarter = quarter(nice_date, fiscal_start = 3)) %>%
  select(-SHEETBAR, -nice_date, -DATE, -LOCATCD)
```

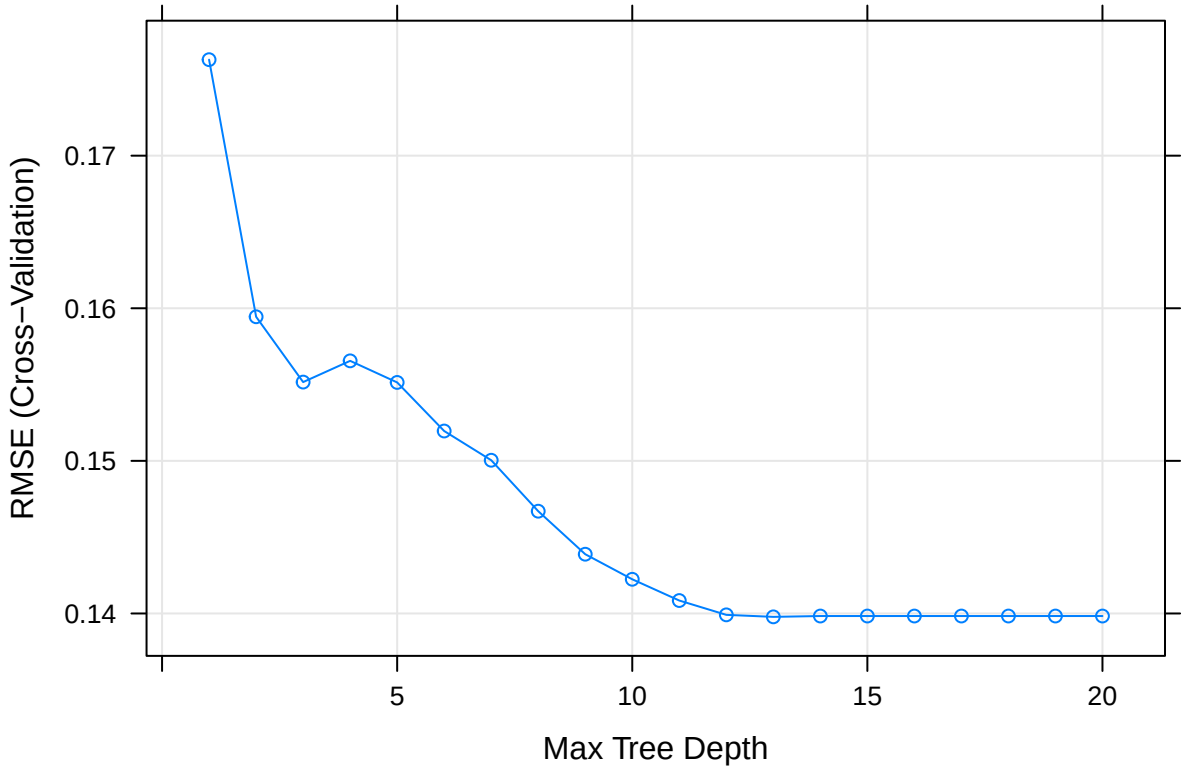
predicting TP with entire dataset

```
set.seed(4747)

fitControl <- caret::trainControl(method="cv")

tr.TP <- caret::train(TP ~ .,
                      data = (narm_water20 %>% sample_frac(0.5)),
                      method = "rpart2",
                      trControl = fitControl,
                      tuneGrid = data.frame(maxdepth=1:20))

fancyRpartPlot(tr.TP$finalModel)
```



Rattle 2021−Jun−22 13:47:21 amba

```
plot(tr.TP)
```

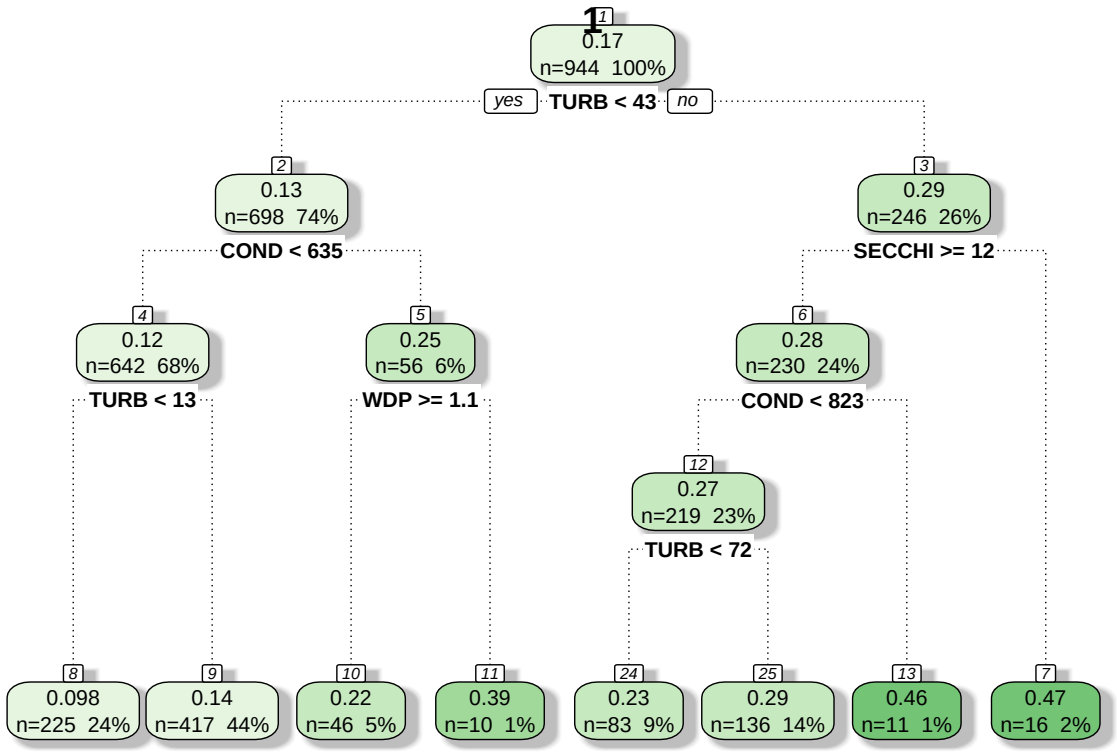


TP by year and season

```
make_year_touples <- function(index, year_partition){
  # index goes from 1 to length(year_partition) - 1

  return(c(year_partition[index], year_partition[index+1]))

}

tree_by_years <- function(year_touple, water_data){

  # filter for specific group of years
  water_data <- water_data %>% filter(year >= year_touple[1] &
                                        year <= year_touple[2])

  fitControl <- caret::trainControl(method="cv")

  tr.TP <- caret::train(TP ~ .,
                        data = water_data,
                        # what is method?
                        method = "rpart2",
                        trControl = fitControl,
                        # don't quite understand maxdepth
                        tuneGrid = data.frame(maxdepth=1:20))

  return(tr.TP)
```

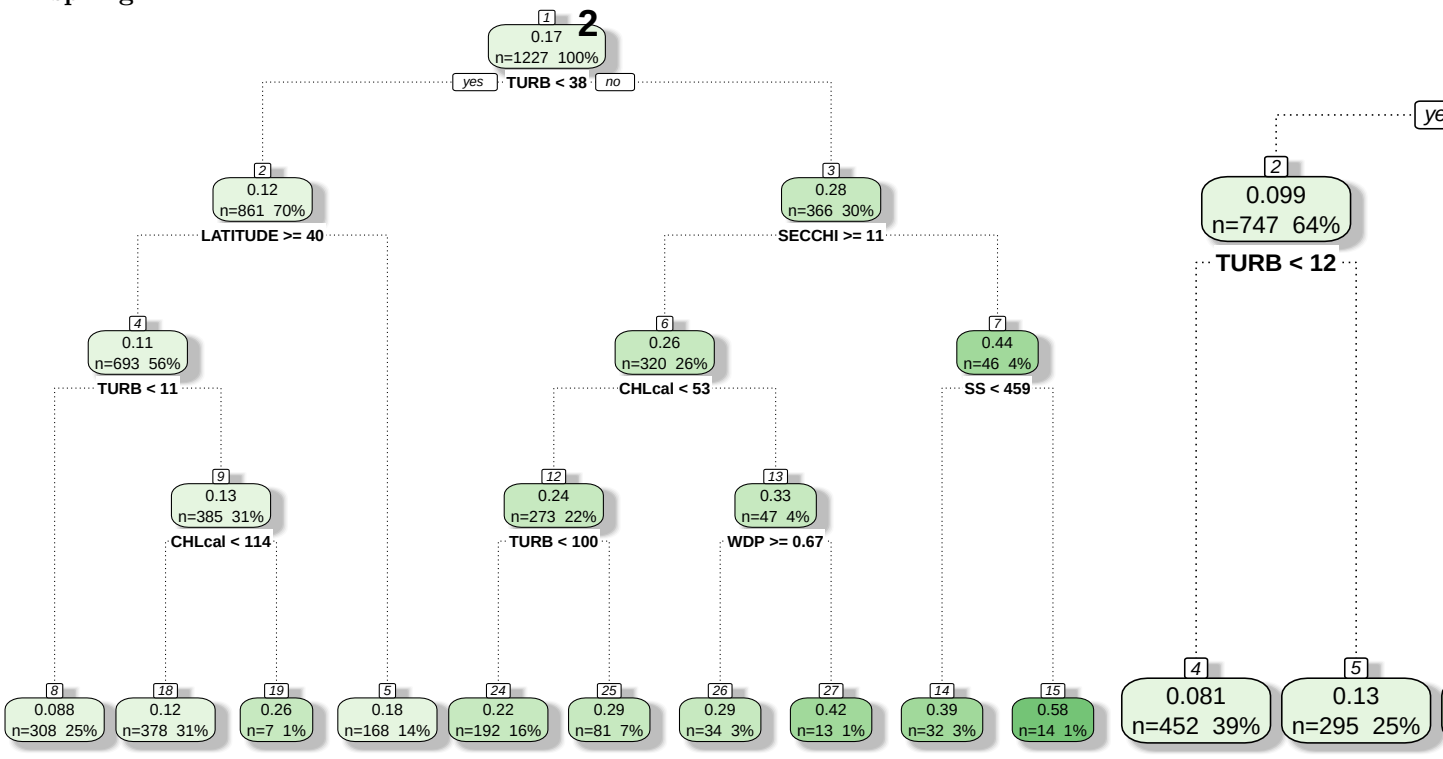
```
}

tree_by_season <- function(season, min_year, max_year, year_interval, water_data){
  # season can be 1, 2, 3, 4 with 1 being spring
  ## this is already processed in line 70

  year_partition <- seq(min_year, max_year, year_interval)
  # make a list of each year interval
  year_touples <- lapply(1:(length(year_partition)-1), make_year_touples, year_partition)

  water_data <- water_data %>% filter(quarter == season)

  tree_models <- lapply(year_touples, tree_by_years, water_data)

  return(tree_models)

}
```

```r
# model for spring, 2000-2005, ..., 2015-2020
trees.sp.2000.2020 <- tree_by_season(1, # 1 stands for spring
                                     2000, # minimum year
                                     2020, # maximum year
                                     5, # 5 year intervals
                                     narm_water20) # dataset

# the 4 comes from (2020 - 2000) / 5
lapply(1:4, function(x) return(fancyRpartPlot(trees.sp.2000.2020[[x]]$finalModel,
                                              main = paste(as.character(x)) )))
```

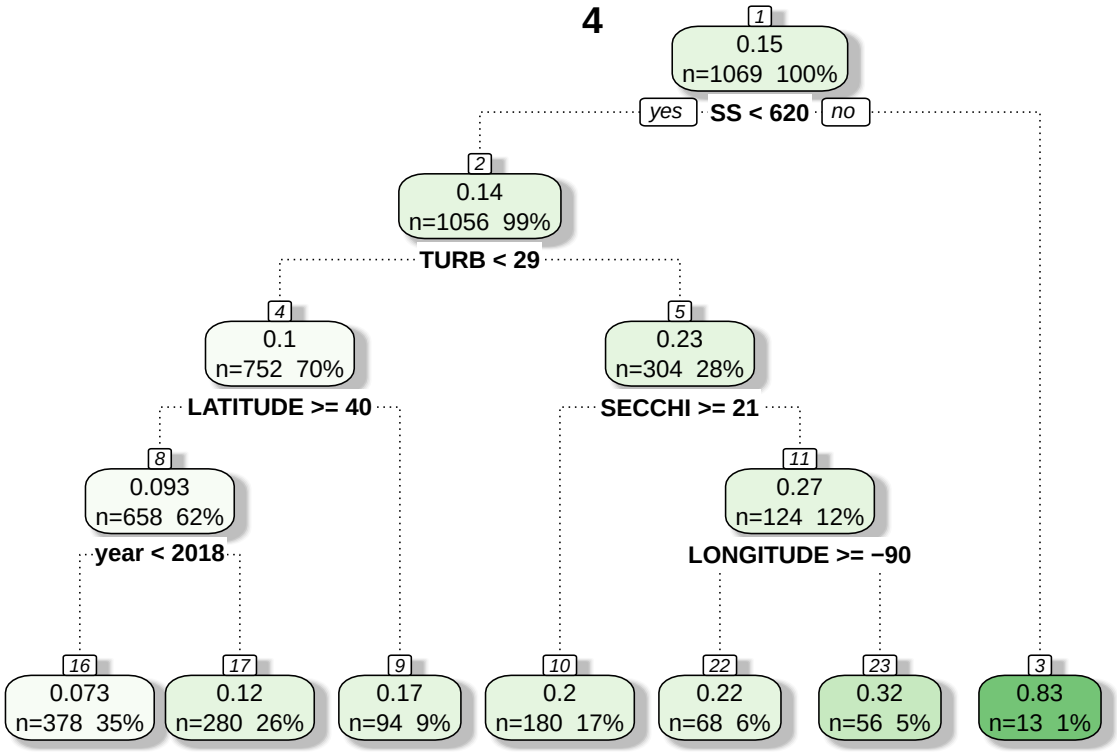


Rattle 2021−Jun−22 13:47:27 amba

TP spring



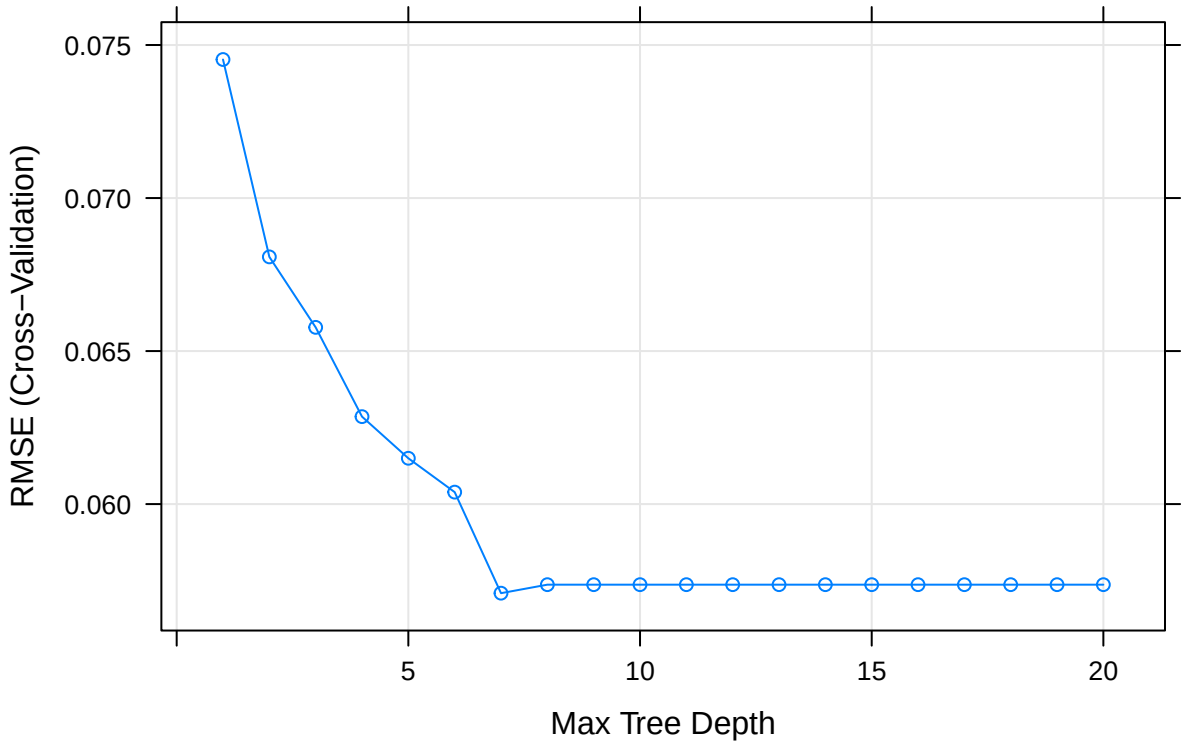
Rattle 2021−Jun−22 13:47:27 amba



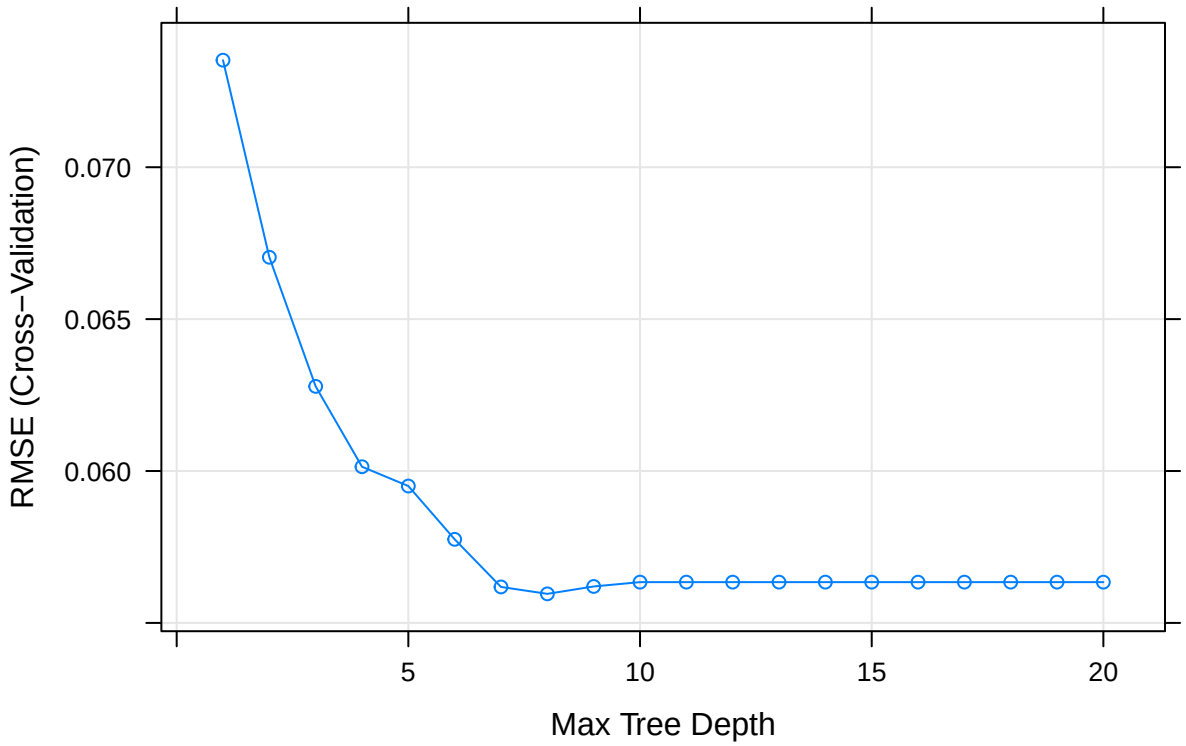

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
```

```
lapply(1:4, function(x) return(plot(trees.sp.2000.2020[[x]],
                                    main = paste(as.character(x)) )))
```

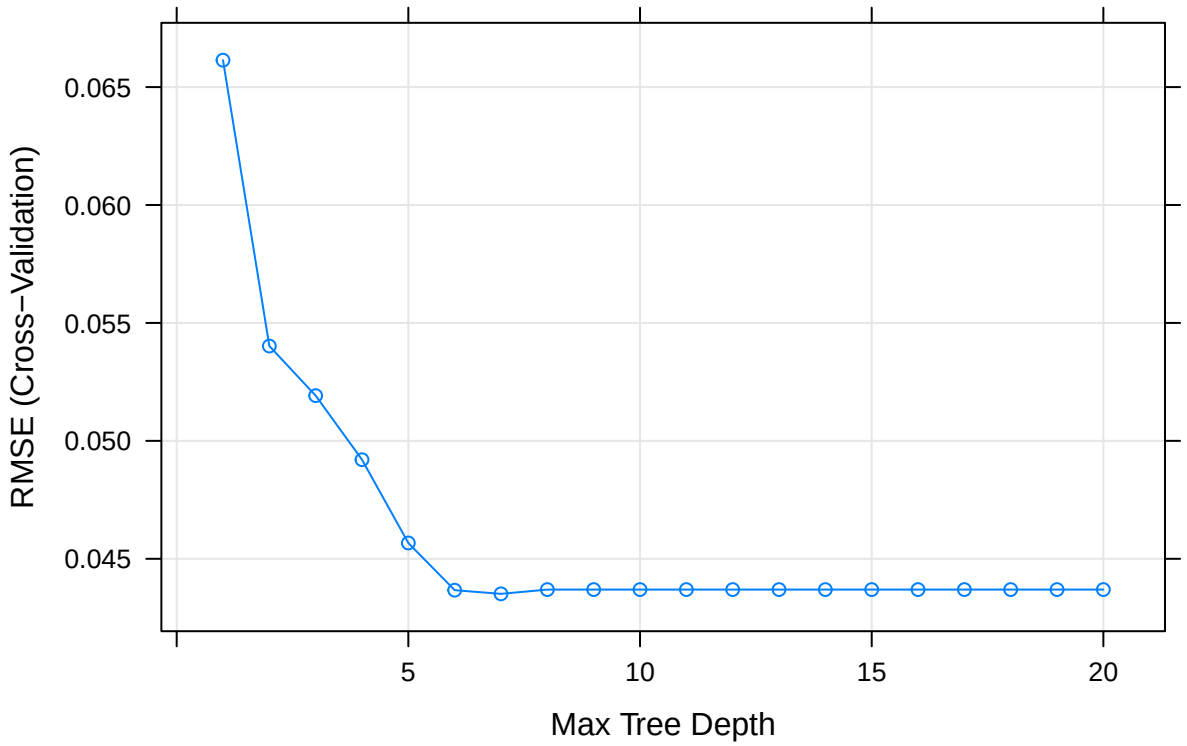
```
## [[1]]
```


1
0.075
0.070
0.065
0.060
RMSE (Cross−Validation)
5
10
15
20
Max Tree Depth

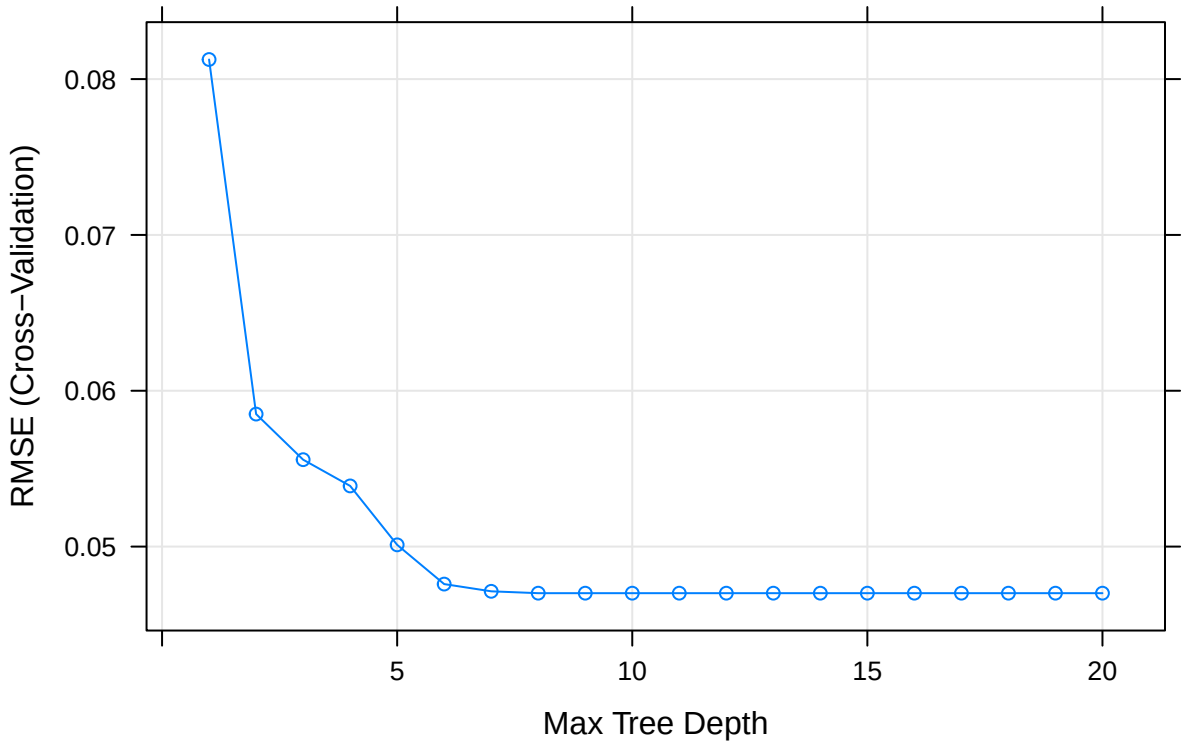
```
##
## [[2]]
```


2
RMSE (Cross−Validation)
0.070
0.065
0.060
5
10
15
20
Max Tree Depth

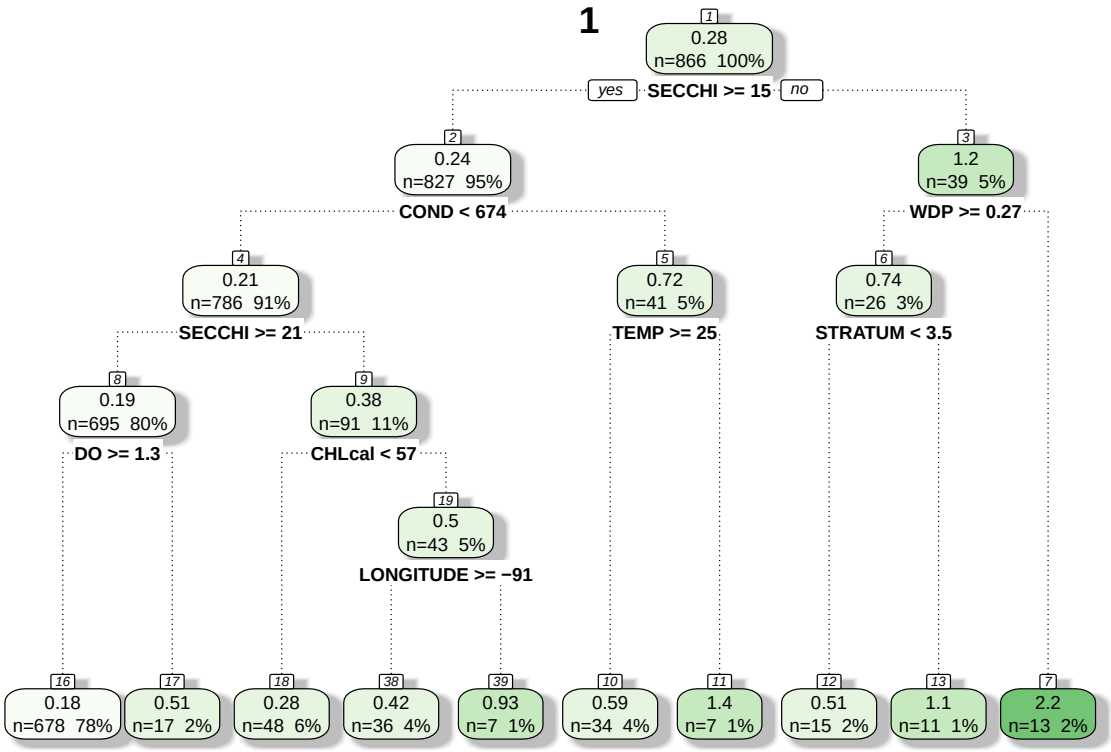
```
##
## [[3]]
```


3
RMSE (Cross−Validation)
0.065
0.060
0.055
0.050
0.045
5
10
15
20
Max Tree Depth

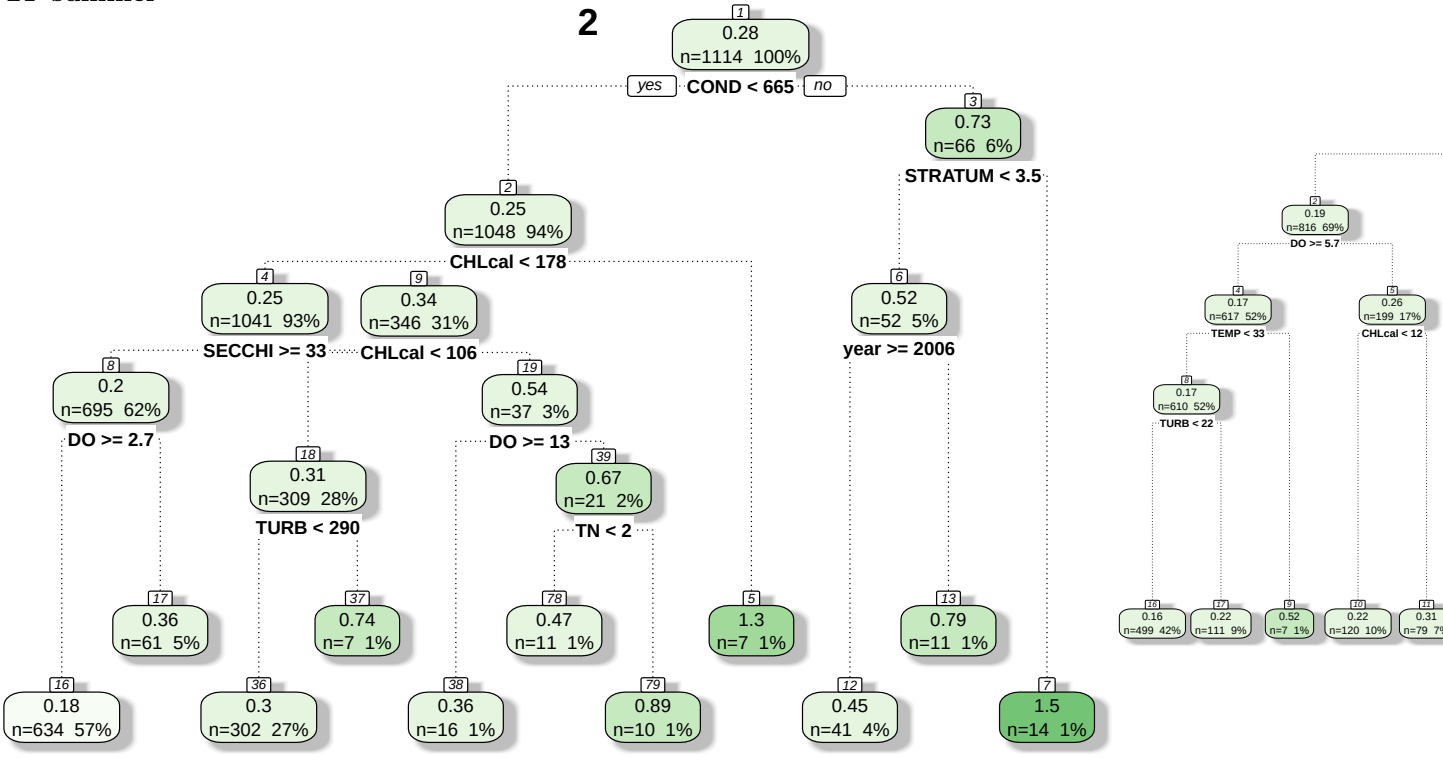
```
##
## [[4]]
```

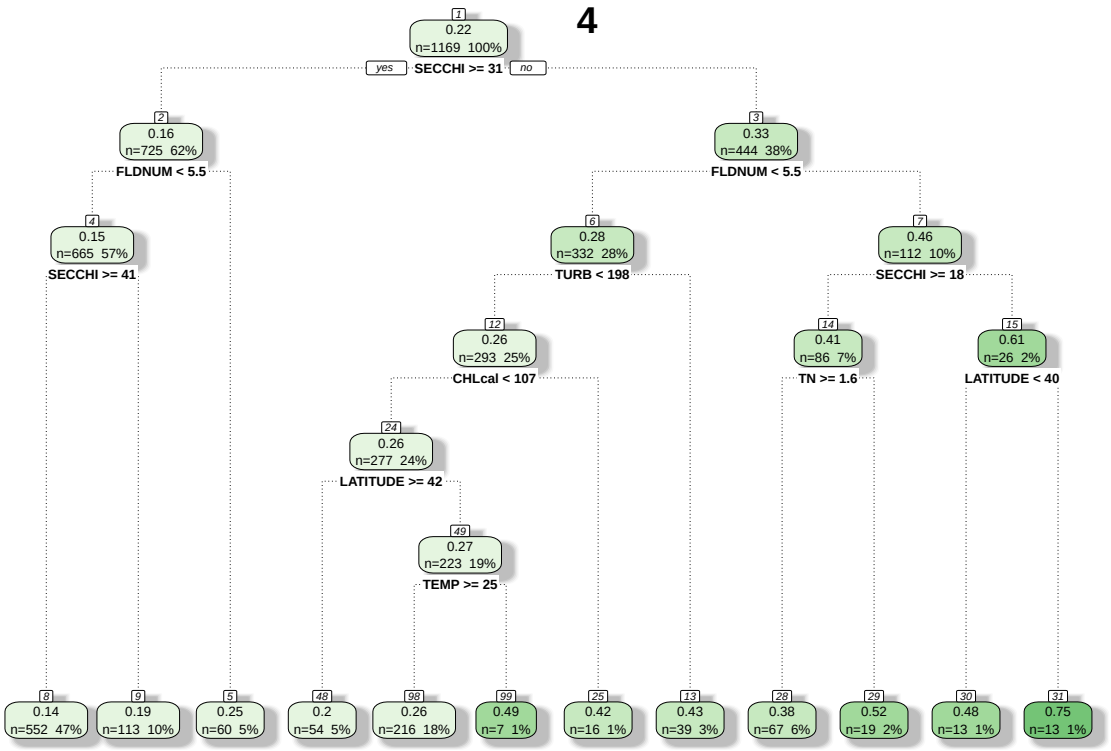

4
RMSE (Cross−Validation)
0.08
0.07
0.06
0.05
5
10
15
20
Max Tree Depth

```r
# model for summer, 2000-2020
trees.su.2000.2020 <- tree_by_season(2, 2000, 2020, 5, narm_water20)

# the 4 comes from (2020 - 2000) / 5
lapply(1:4, function(x) return(fancyRpartPlot(trees.su.2000.2020[[x]]$finalModel,
                                              main = paste(as.character(x)) )))
```


1
1
0.28
n=866 100%
yes
SECCHI >= 15
no
2
0.24
n=827 95%
3
1.2
n=39 5%
COND < 674
WDP >= 0.27
4
0.21
n=786 91%
5
0.72
n=41 5%
6
0.74
n=26 3%
SECCHI >= 21
TEMP >= 25
STRATUM < 3.5
8
0.19
n=695 80%
9
0.38
n=91 11%
DO >= 1.3
CHLcal < 57
19
0.5
n=43 5%
LONGITUDE >= −91
16
0.18
n=678 78%
17
0.51
n=17 2%
18
0.28
n=48 6%
38
0.42
n=36 4%
39
0.93
n=7 1%
10
0.59
n=34 4%
11
1.4
n=7 1%
12
0.51
n=15 2%
13
1.1
n=11 1%
7
2.2
n=13 2%
Rattle 2021−Jun−22 13:47:35 amba

TP summer


2
1
0.28
n=1114 100%
yes
COND < 665
no
3
0.73
n=66 6%
STRATUM < 3.5
2
0.25
n=1048 94%
CHLcal < 178
4
0.25
n=1041 93%
9
0.34
n=346 31%
6
0.52
n=52 5%
SECCHI >= 33
CHLcal < 106
year >= 2006
8
0.2
n=695 62%
19
0.54
n=37 3%
DO >= 2.7
DO >= 13
18
0.31
n=309 28%
39
0.67
n=21 2%
TURB < 290
TN < 2
17
0.36
n=61 5%
37
0.74
n=7 1%
78
0.47
n=11 1%
5
1.3
n=7 1%
13
0.79
n=11 1%
16
0.18
n=634 57%
36
0.3
n=302 27%
38
0.36
n=16 1%
79
0.89
n=10 1%
12
0.45
n=41 4%
7
1.5
n=14 1%
Rattle 2021−Jun−22 13:47:35 amba

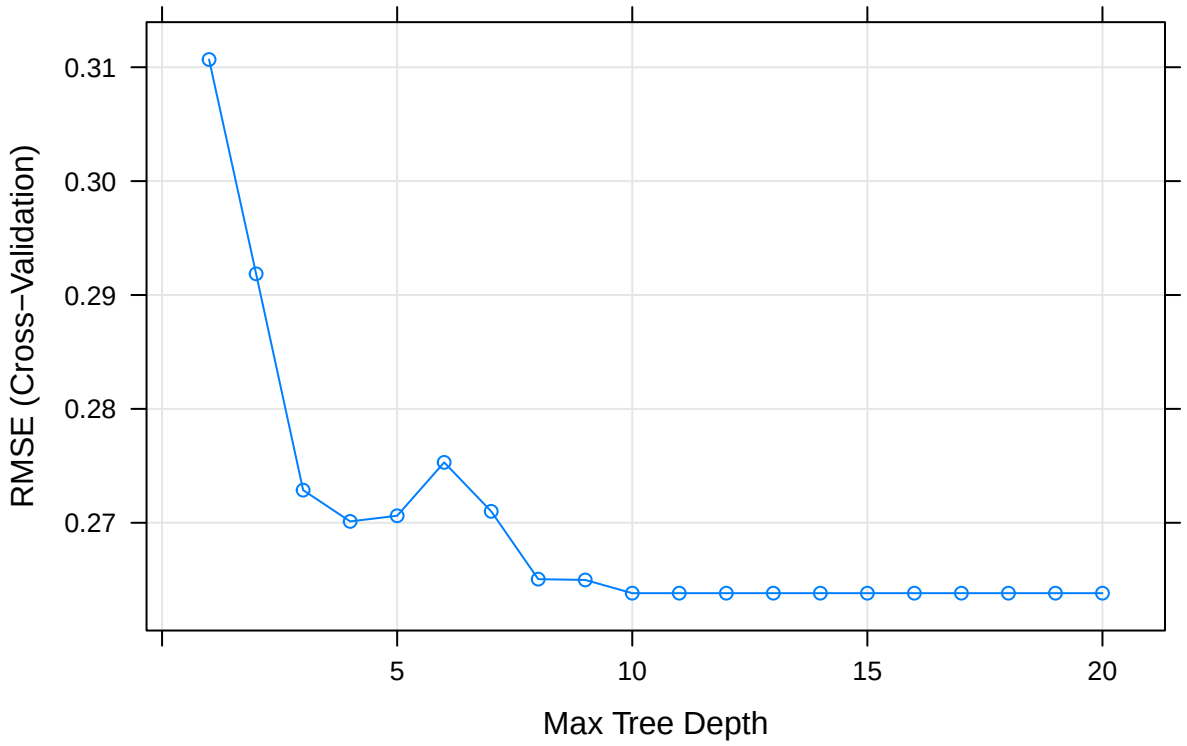


Rattle 2021−Jun−22 13:47:36 amba

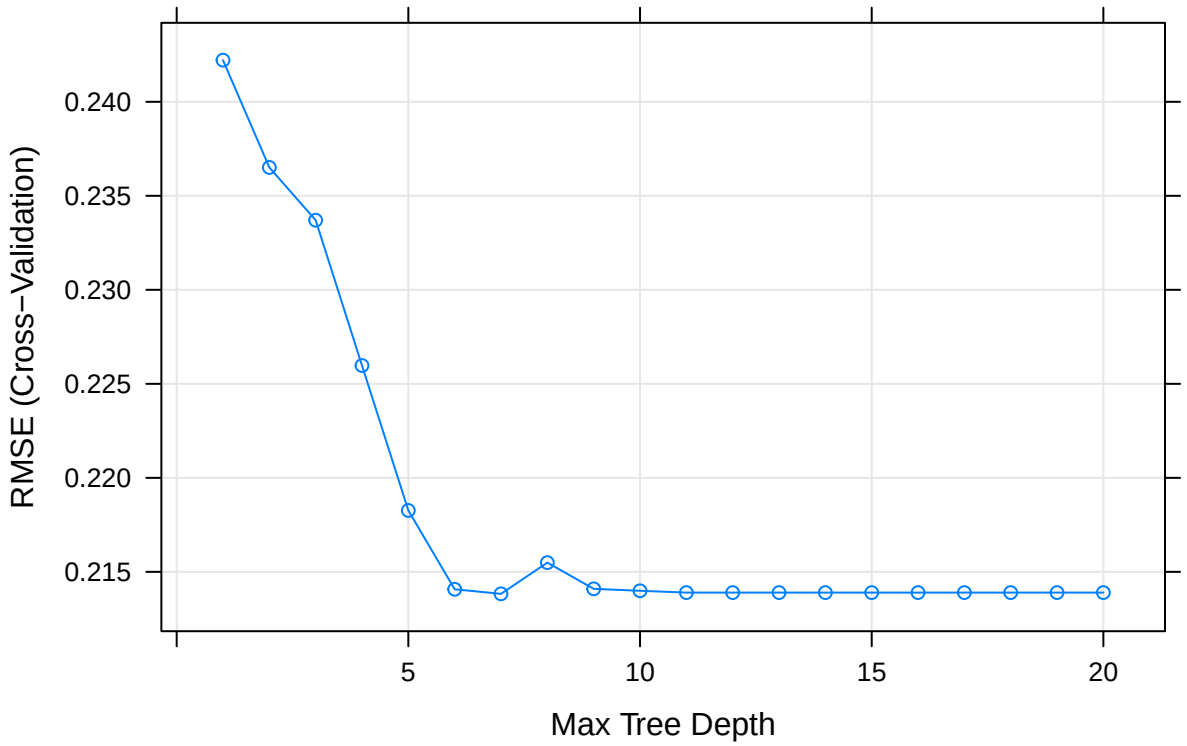
```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
```

```
lapply(1:4, function(x) return(plot(trees.su.2000.2020[[x]],
                                    main = paste(as.character(x)) )))
```

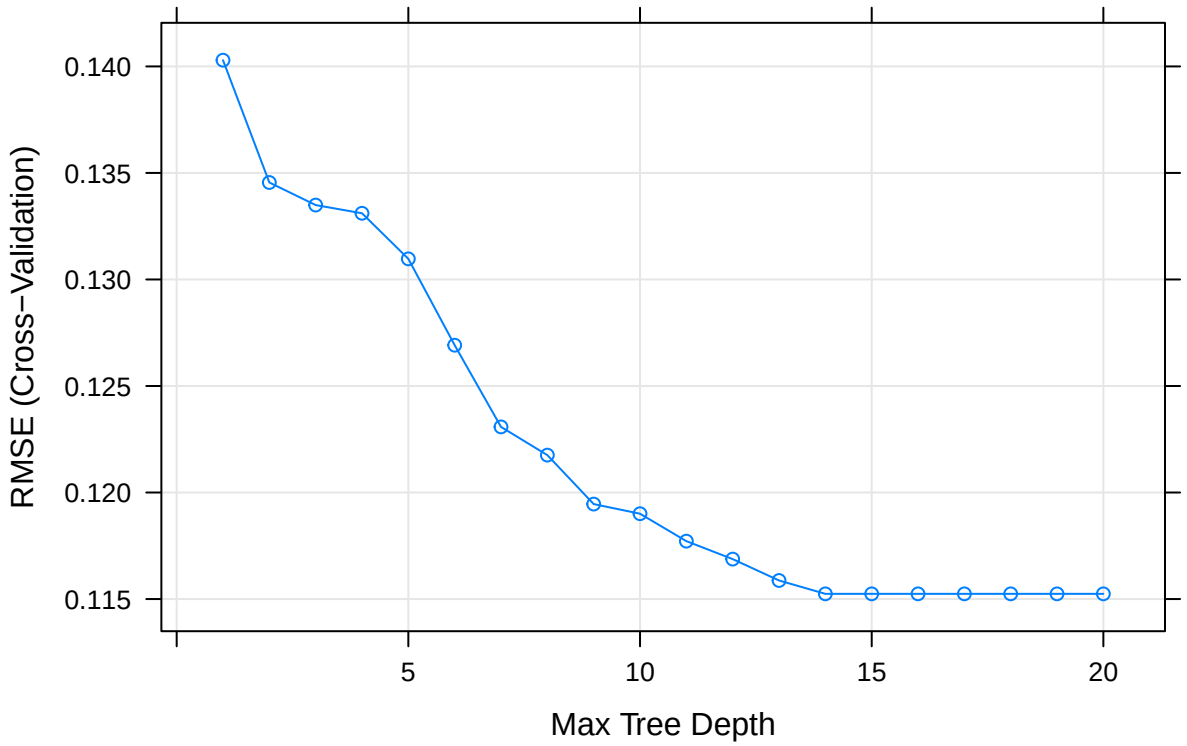
```
## [[1]]
```


1
RMSE (Cross−Validation)
0.31
0.30
0.29
0.28
0.27
5
10
15
20
Max Tree Depth

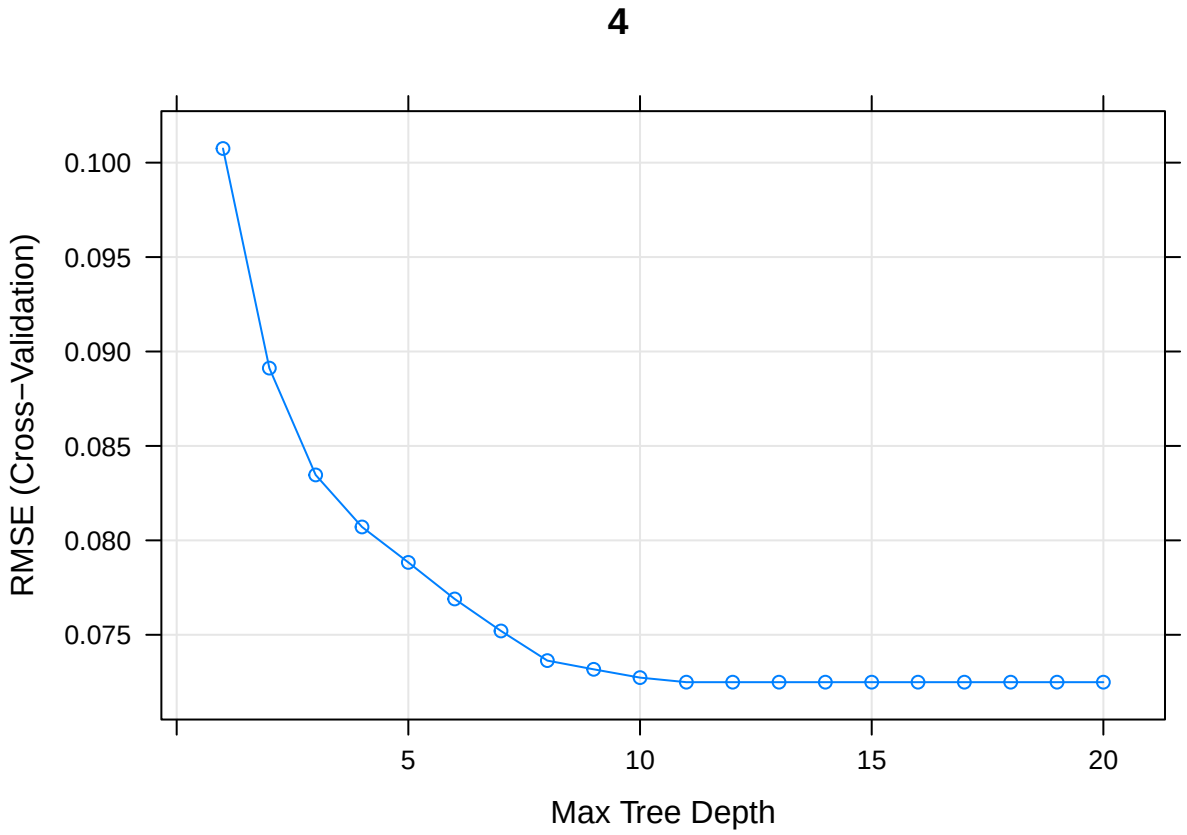
```
##
## [[2]]
```


2
RMSE (Cross−Validation)
0.240
0.235
0.230
0.225
0.220
0.215
5
10
15
20
Max Tree Depth

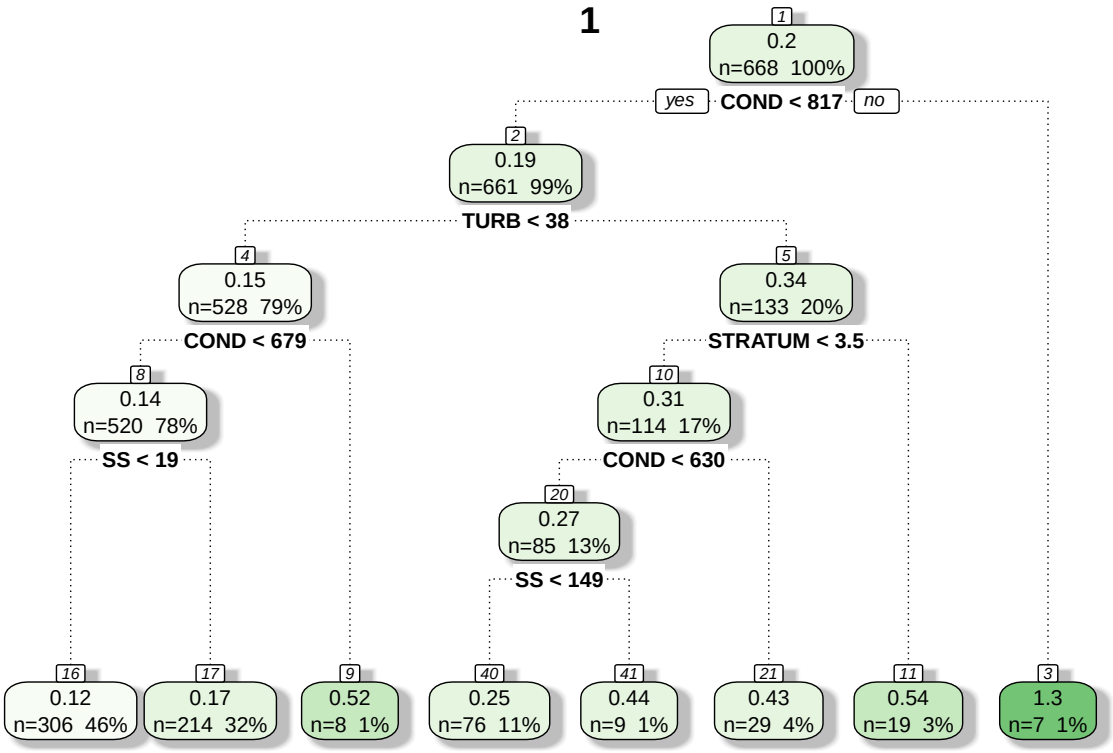
```
##
## [[3]]
```


3
RMSE (Cross−Validation)
0.140
0.135
0.130
0.125
0.120
0.115
5
10
15
20
Max Tree Depth

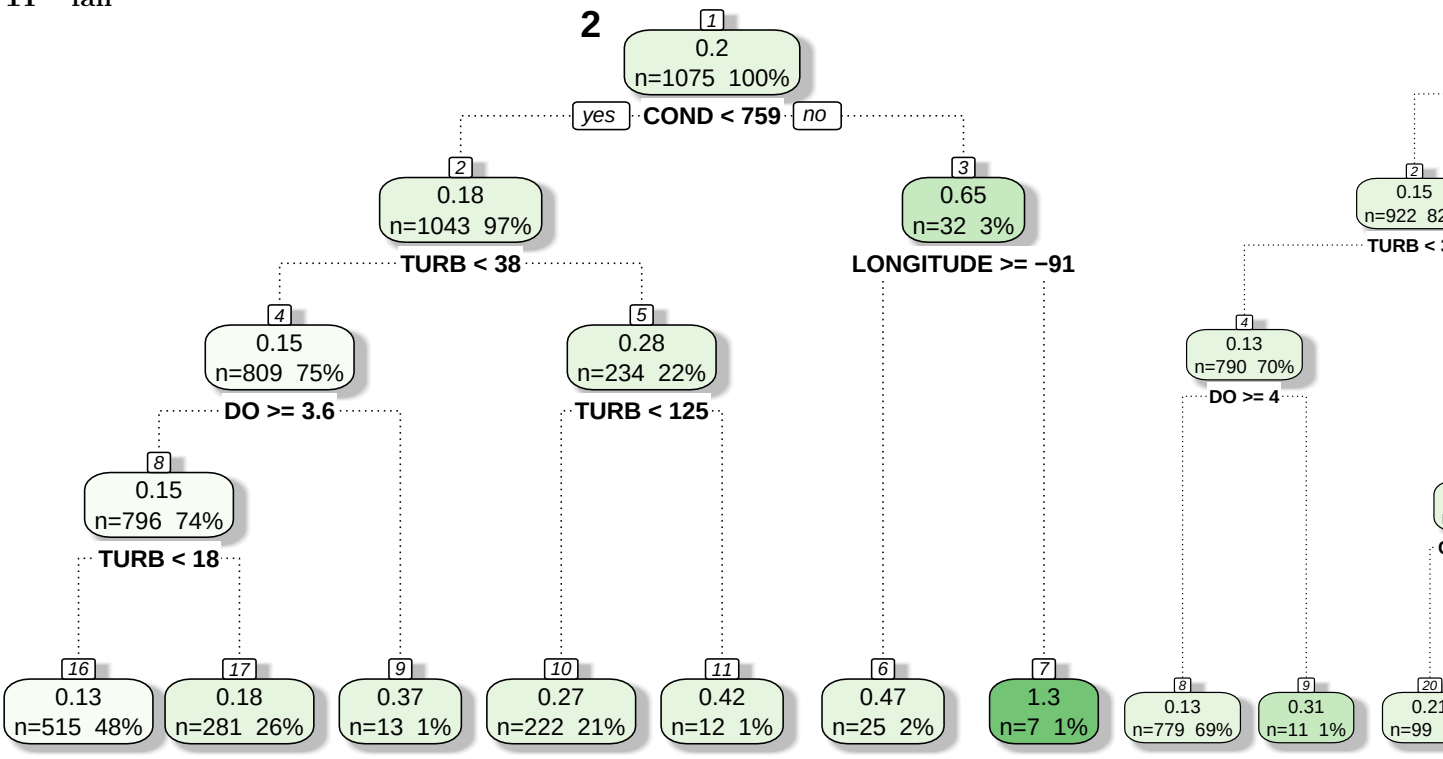
```
##
## [[4]]
```

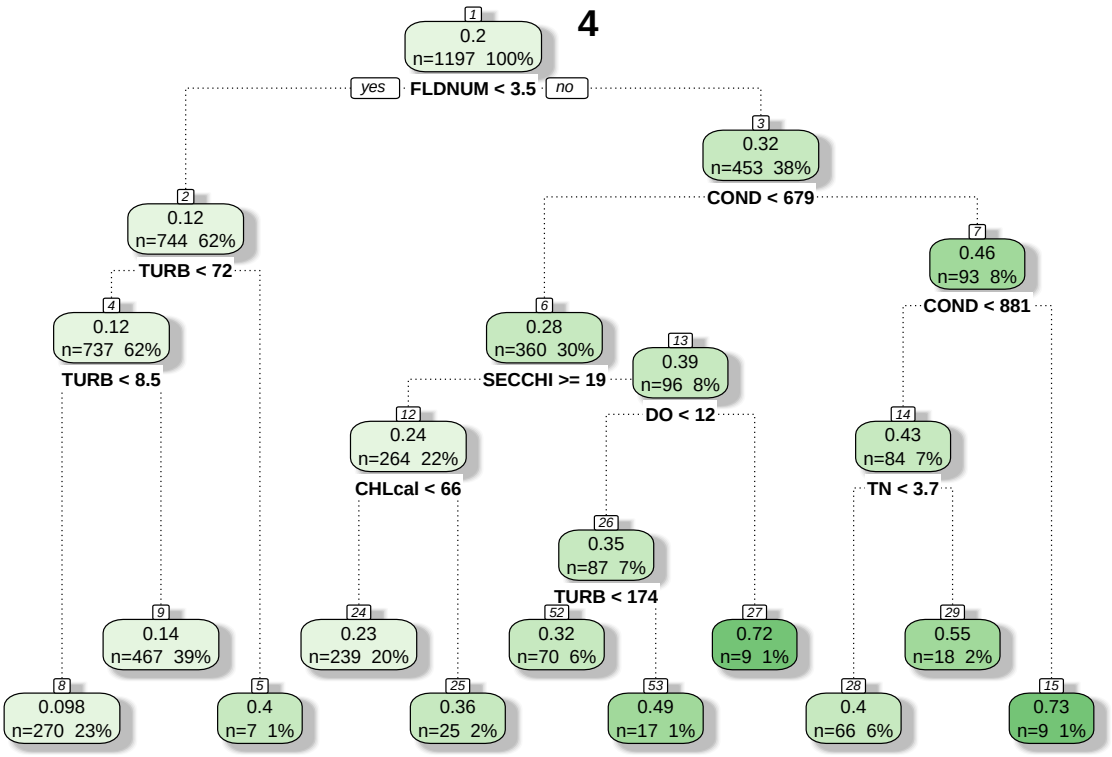

4
RMSE (Cross−Validation)
0.100
0.095
0.090
0.085
0.080
0.075
5
10
15
20
Max Tree Depth

```r
# model for fall, 2000-2020
trees.fa.2000.2020 <- tree_by_season(3, 2000, 2020, 5, narm_water20)

# the *2 is to insert line breaks
lapply(1:4*2, function(x)
  if (x %% 2 == 1) { # if x is even
    asis_output("\\\\[10cm]")
    } else {
      return(fancyRpartPlot(trees.fa.2000.2020[[x/2]]$finalModel,
                            main = paste(as.character(x/2)) ))
    } )
```


1
1
0.2
n=668 100%
yes
COND < 817
no
2
0.19
n=661 99%
TURB < 38
4
0.15
n=528 79%
COND < 679
5
0.34
n=133 20%
STRATUM < 3.5
8
0.14
n=520 78%
SS < 19
10
0.31
n=114 17%
COND < 630
20
0.27
n=85 13%
SS < 149
16
0.12
n=306 46%
17
0.17
n=214 32%
9
0.52
n=8 1%
40
0.25
n=76 11%
41
0.44
n=9 1%
21
0.43
n=29 4%
11
0.54
n=19 3%
3
1.3
n=7 1%
Rattle 2021−Jun−22 13:47:44 amba

TP fall

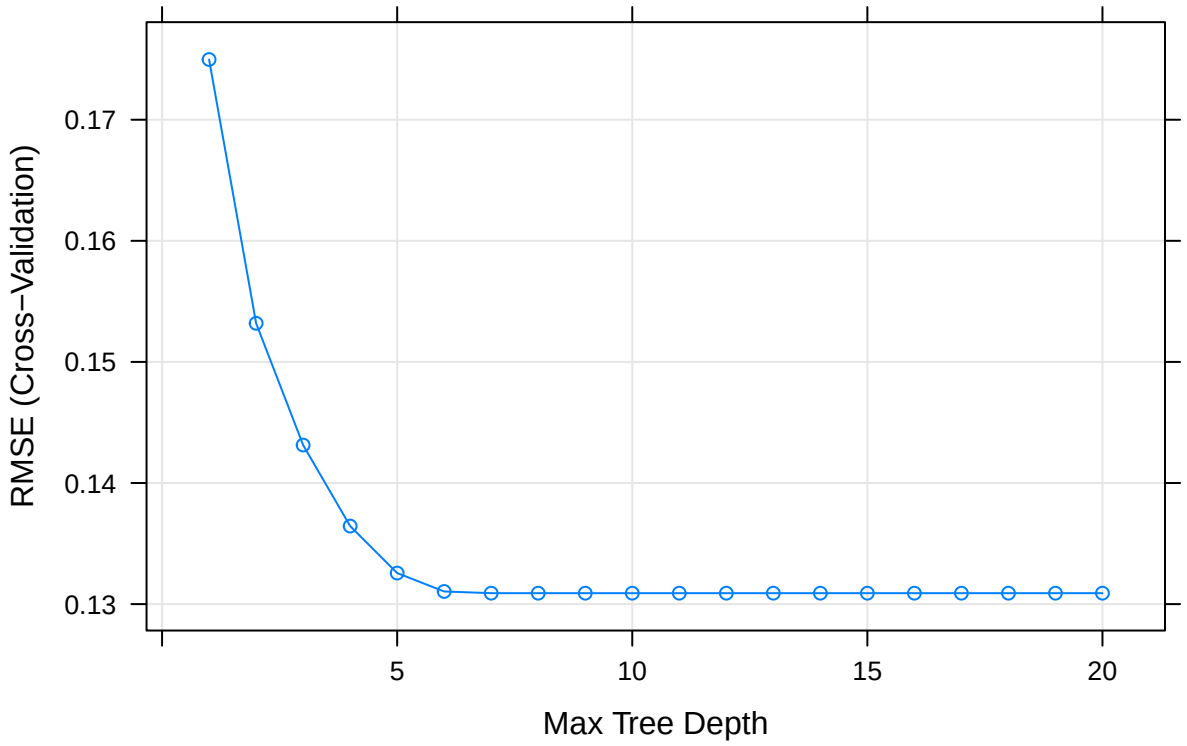

2
1
0.2
n=1075 100%
yes
COND < 759
no
2
0.18
n=1043 97%
TURB < 38
3
0.65
n=32 3%
LONGITUDE >= −91
4
0.15
n=809 75%
DO >= 3.6
5
0.28
n=234 22%
TURB < 125
8
0.15
n=796 74%
TURB < 18
16
0.13
n=515 48%
17
0.18
n=281 26%
9
0.37
n=13 1%
10
0.27
n=222 21%
11
0.42
n=12 1%
6
0.47
n=25 2%
7
1.3
n=7 1%
4
0.13
n=790 70%
DO >= 4
8
0.13
n=779 69%
9
0.31
n=11 1%
Rattle 2021−Jun−22 13:47:44 amba



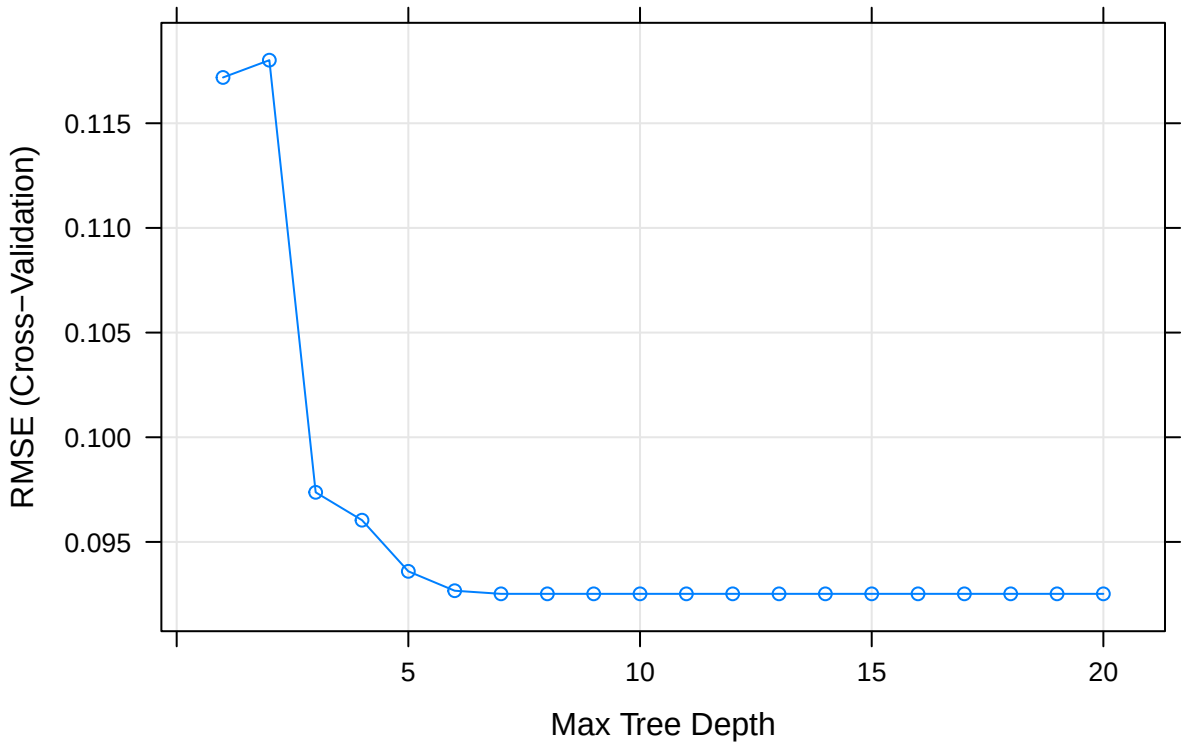

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
```

```
lapply(1:4, function(x) return(plot(trees.fa.2000.2020[[x]],
                                    main = paste(as.character(x)) )))
```

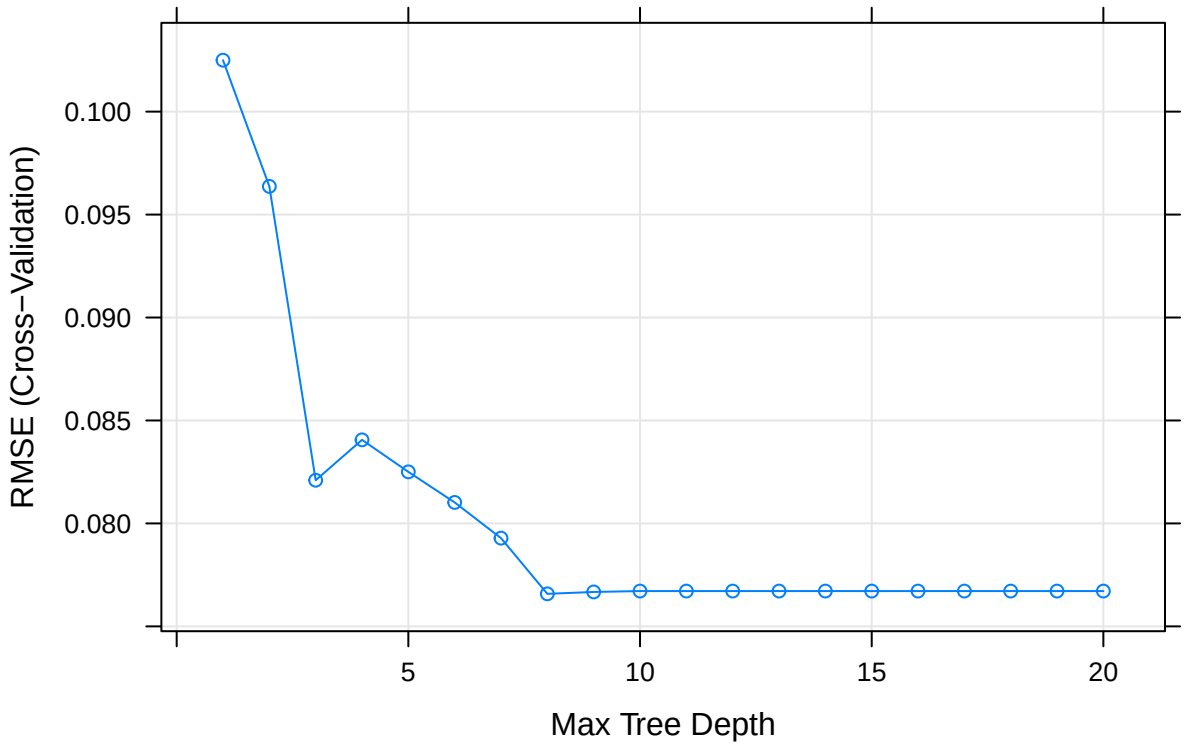
```
## [[1]]
```


1
RMSE (Cross−Validation)
0.17
0.16
0.15
0.14
0.13
5
10
15
20
Max Tree Depth

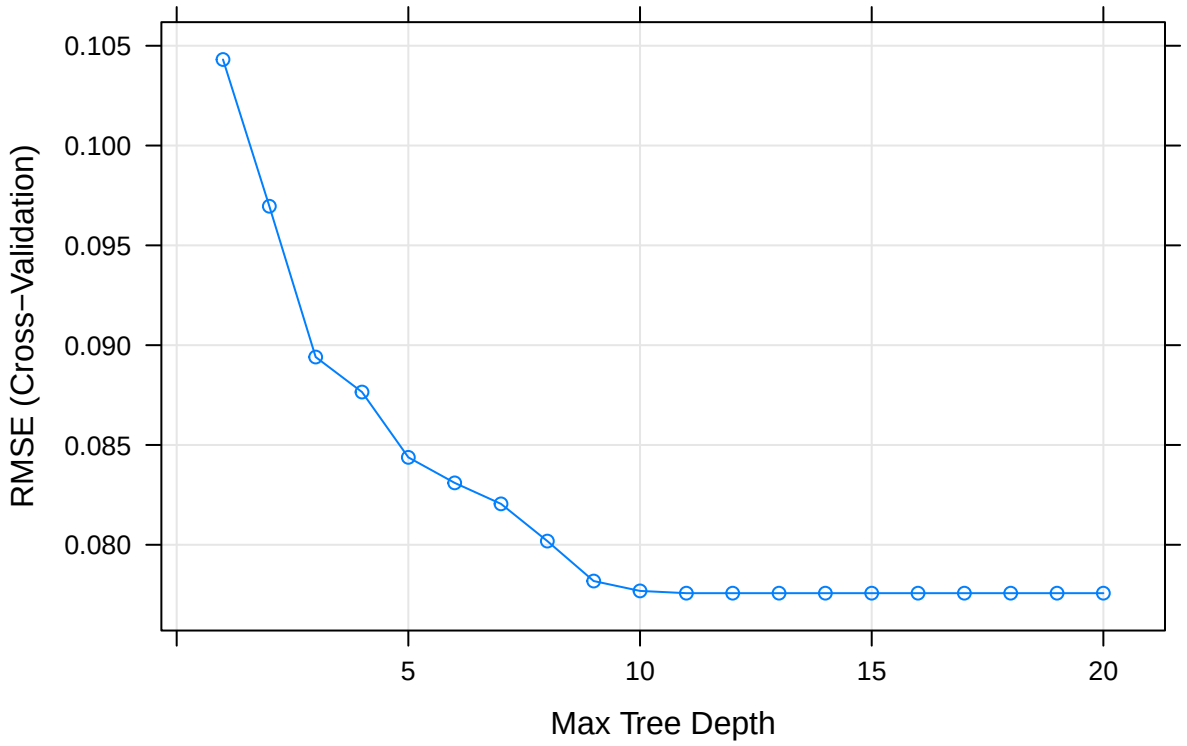
```
##
## [[2]]
```


2
RMSE (Cross−Validation)
0.115
0.110
0.105
0.100
0.095
5
10
15
20
Max Tree Depth

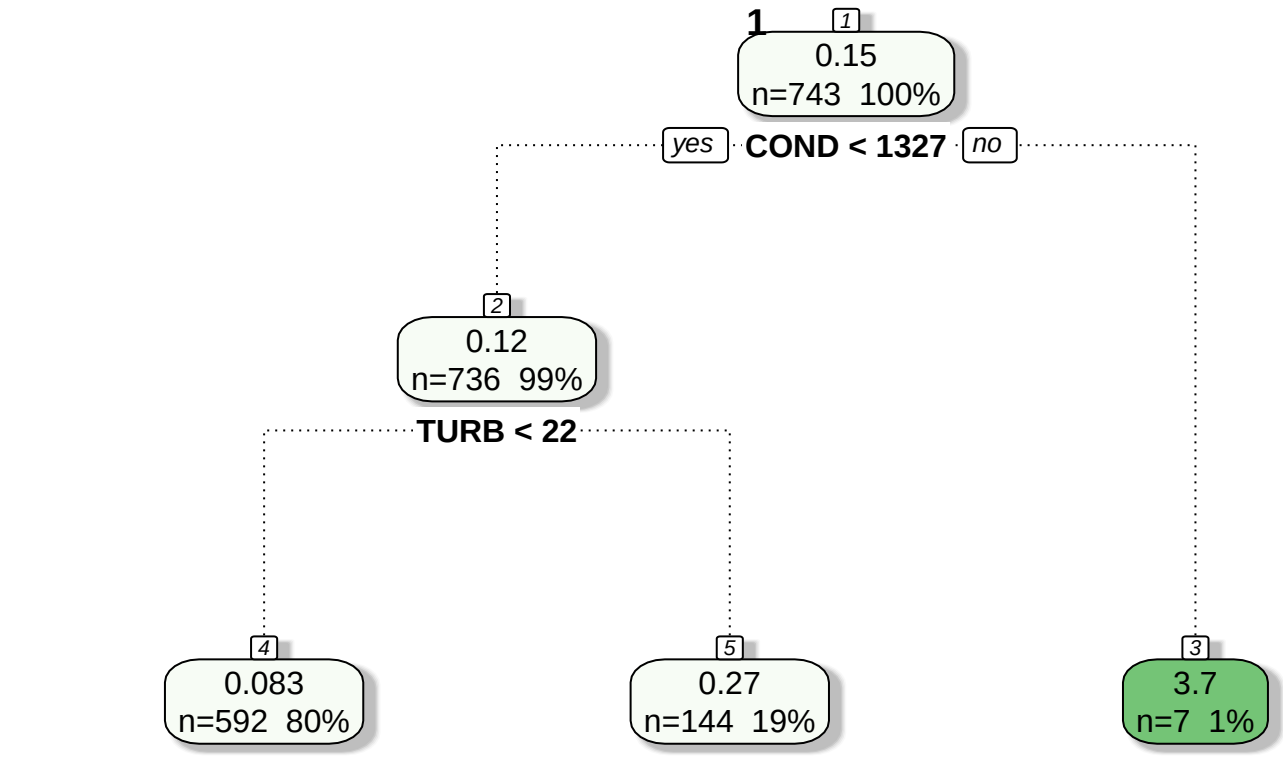
```
##
## [[3]]
```


3
RMSE (Cross−Validation)
0.100
0.095
0.090
0.085
0.080
5
10
15
20
Max Tree Depth

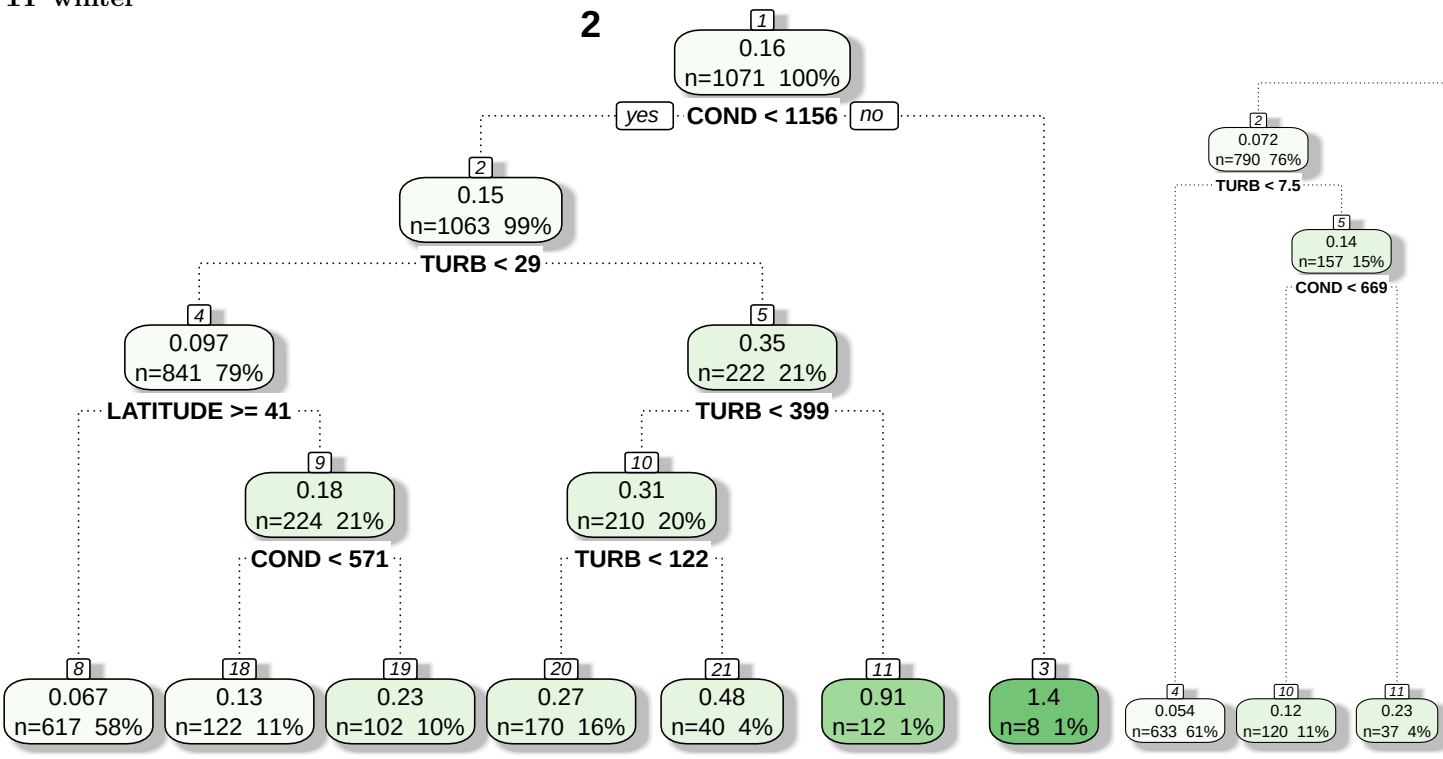
```
##
## [[4]]
```

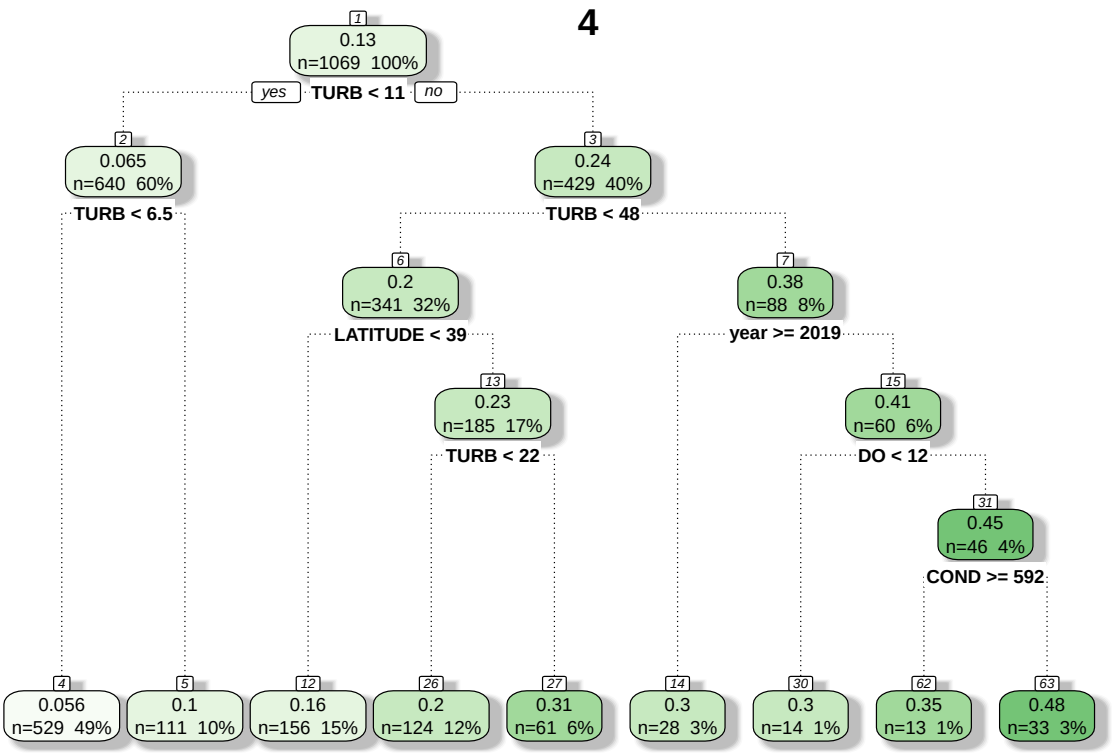

4
RMSE (Cross−Validation)
0.105
0.100
0.095
0.090
0.085
0.080
5
10
15
20
Max Tree Depth

```r
# model for winter, 2000-2020
trees.wi.2000.2020 <- tree_by_season(4, 2000, 2020, 5, narm_water20)

lapply(1:4, function(x) return(fancyRpartPlot(trees.wi.2000.2020[[x]]$finalModel,
                                              main = paste(as.character(x)) )))
```


1
1
0.15
n=743 100%
yes
COND < 1327
no
2
0.12
n=736 99%
TURB < 22
4
0.083
n=592 80%
5
0.27
n=144 19%
3
3.7
n=7 1%
Rattle 2021−Jun−22 13:47:52 amba

TP winter

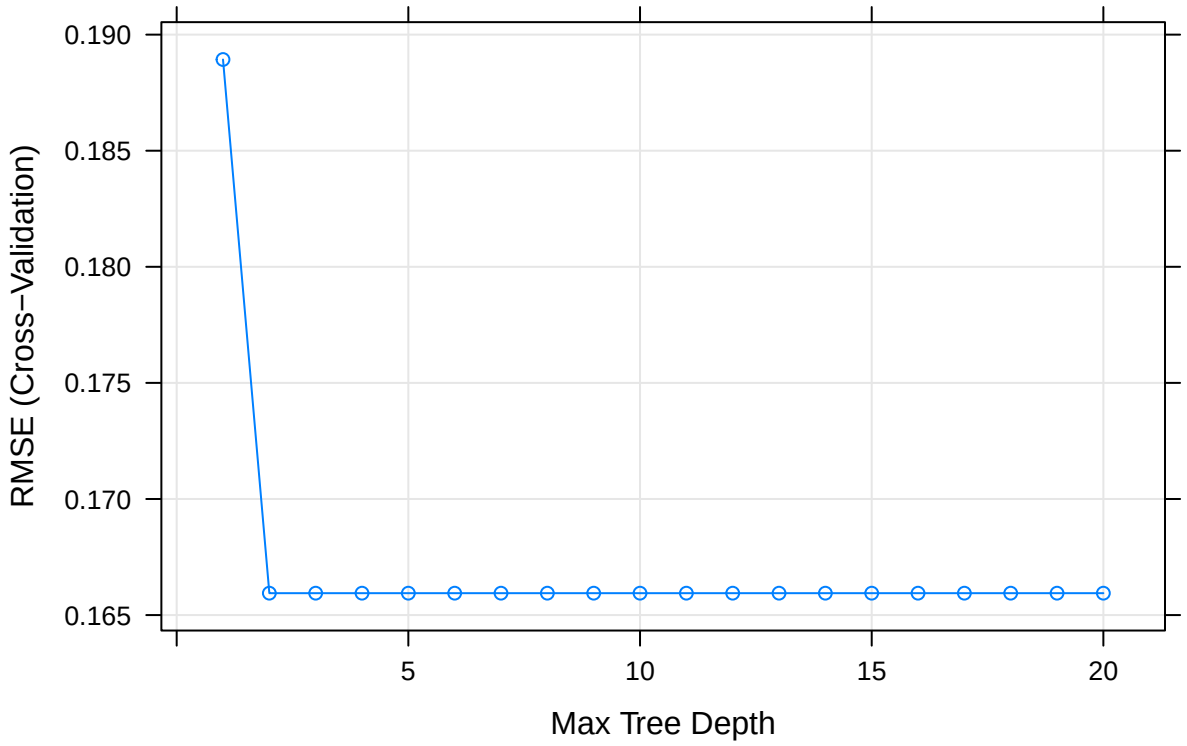

2
1
0.16
n=1071 100%
yes
COND < 1156
no
2
0.15
n=1063 99%
TURB < 29
4
0.097
n=841 79%
LATITUDE >= 41
9
0.18
n=224 21%
COND < 571
8
0.067
n=617 58%
18
0.13
n=122 11%
19
0.23
n=102 10%
5
0.35
n=222 21%
TURB < 399
10
0.31
n=210 20%
TURB < 122
20
0.27
n=170 16%
21
0.48
n=40 4%
11
0.91
n=12 1%
3
1.4
n=8 1%
2
0.072
n=790 76%
TURB < 7.5
5
0.14
n=157 15%
COND < 669
4
0.054
n=633 61%
10
0.12
n=120 11%
11
0.23
n=37 4%
Rattle 2021−Jun−22 13:47:52 amba



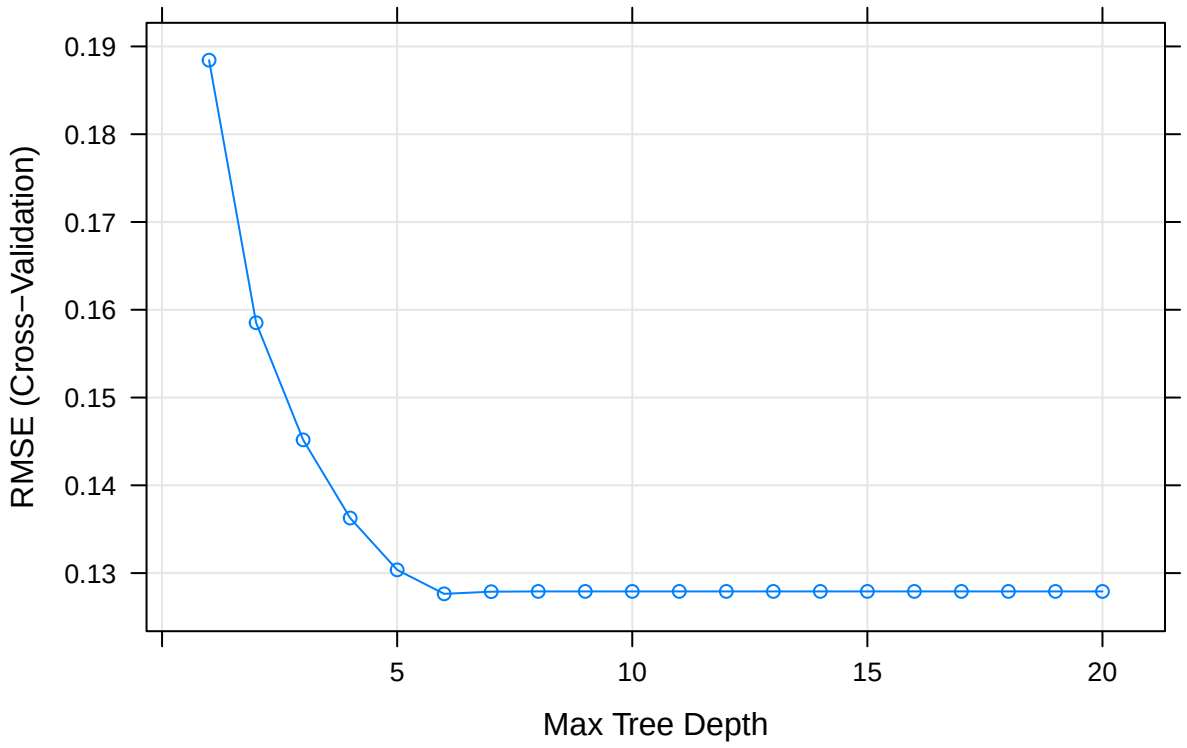

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
```

```
lapply(1:4, function(x) return(plot(trees.wi.2000.2020[[x]],
                                    main = paste(as.character(x)) )))
```

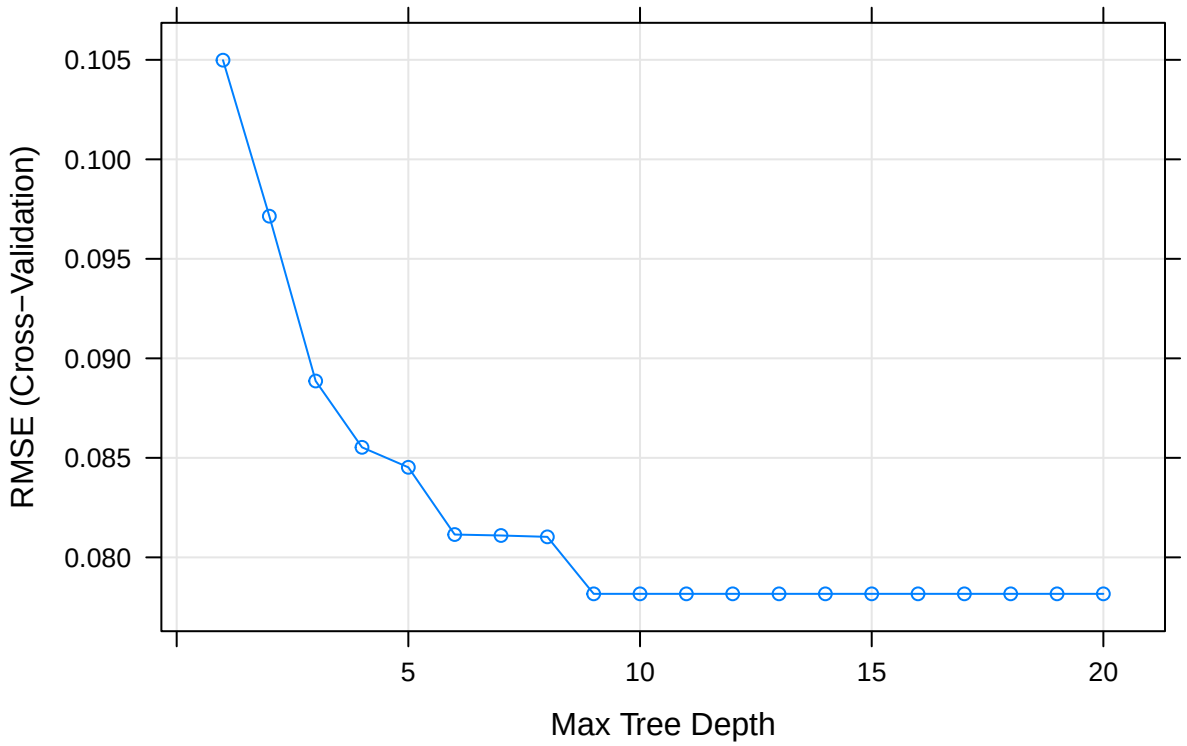
```
## [[1]]
```


1
0.190
0.185
0.180
0.175
0.170
0.165
RMSE (Cross−Validation)
5
10
15
20
Max Tree Depth

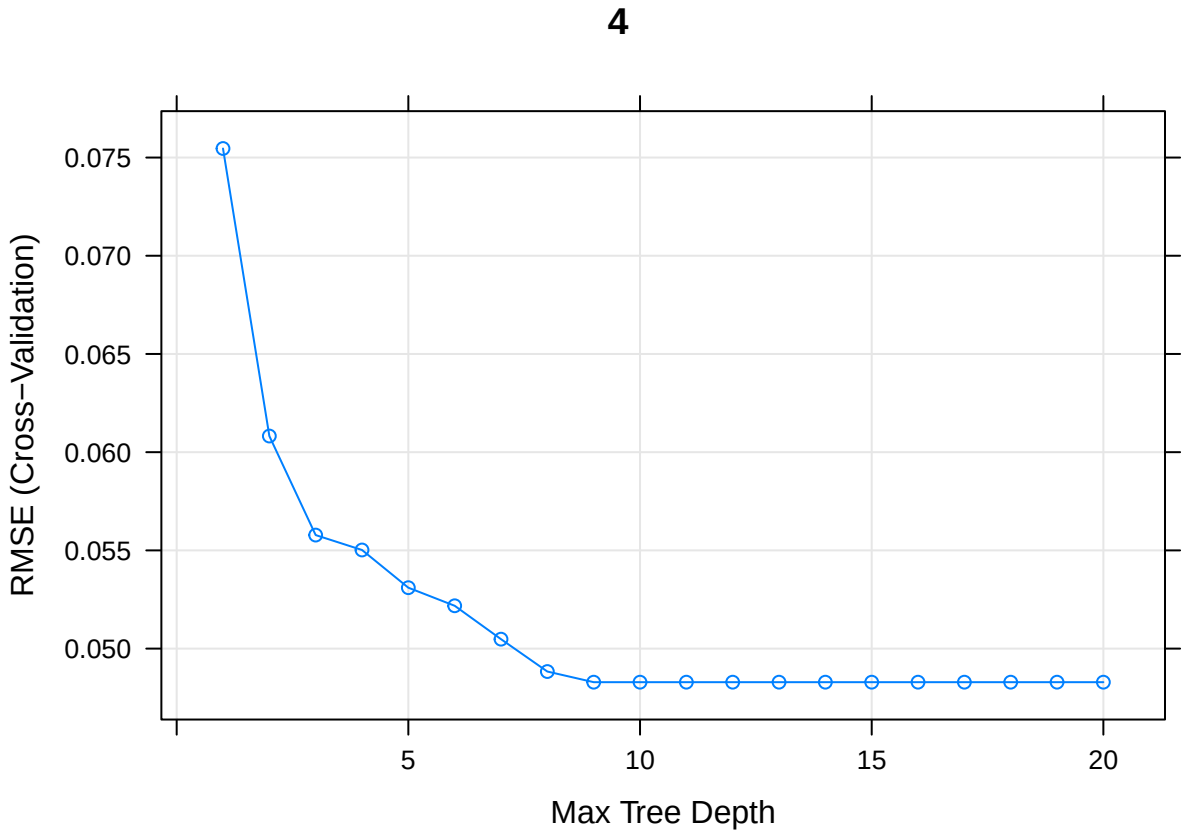
```
##
## [[2]]
```


2
RMSE (Cross−Validation)
0.19
0.18
0.17
0.16
0.15
0.14
0.13
5
10
15
20
Max Tree Depth

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


3
RMSE (Cross−Validation)
0.105
0.100
0.095
0.090
0.085
0.080
5
10
15
20
Max Tree Depth

```
##
## [[4]]
```


4
RMSE (Cross−Validation)
0.075
0.070
0.065
0.060
0.055
0.050
5
10
15
20
Max Tree Depth