

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)

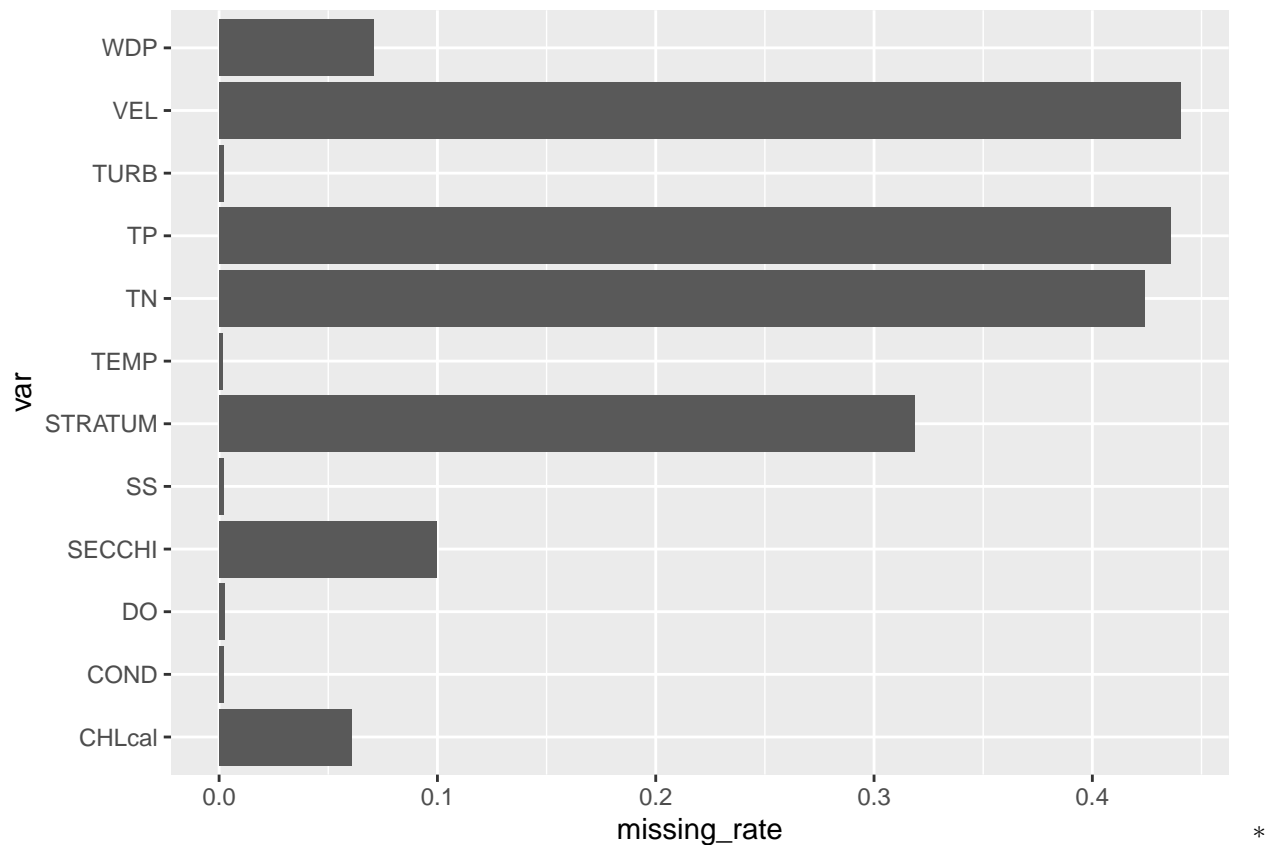
# remove the QF code var
water_var <- names(water20)[str_detect(names(water20), "QF", negate = T)]

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)
```

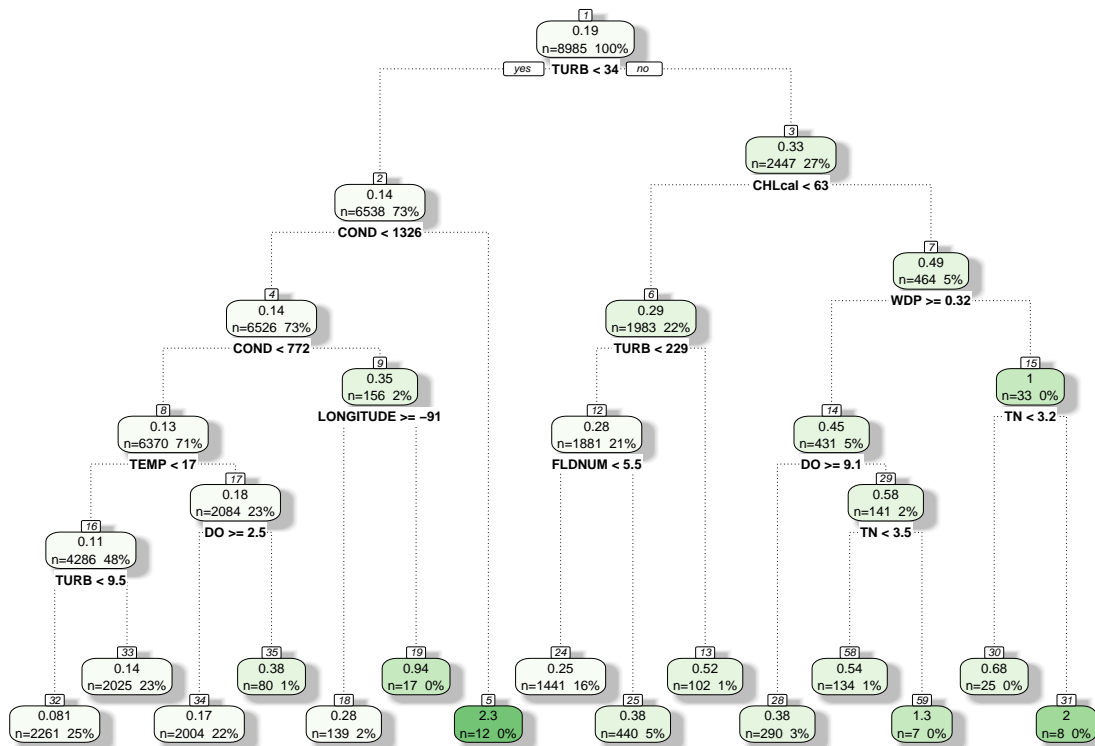
predicting TP with entire dataset

```
set.seed(4747)

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

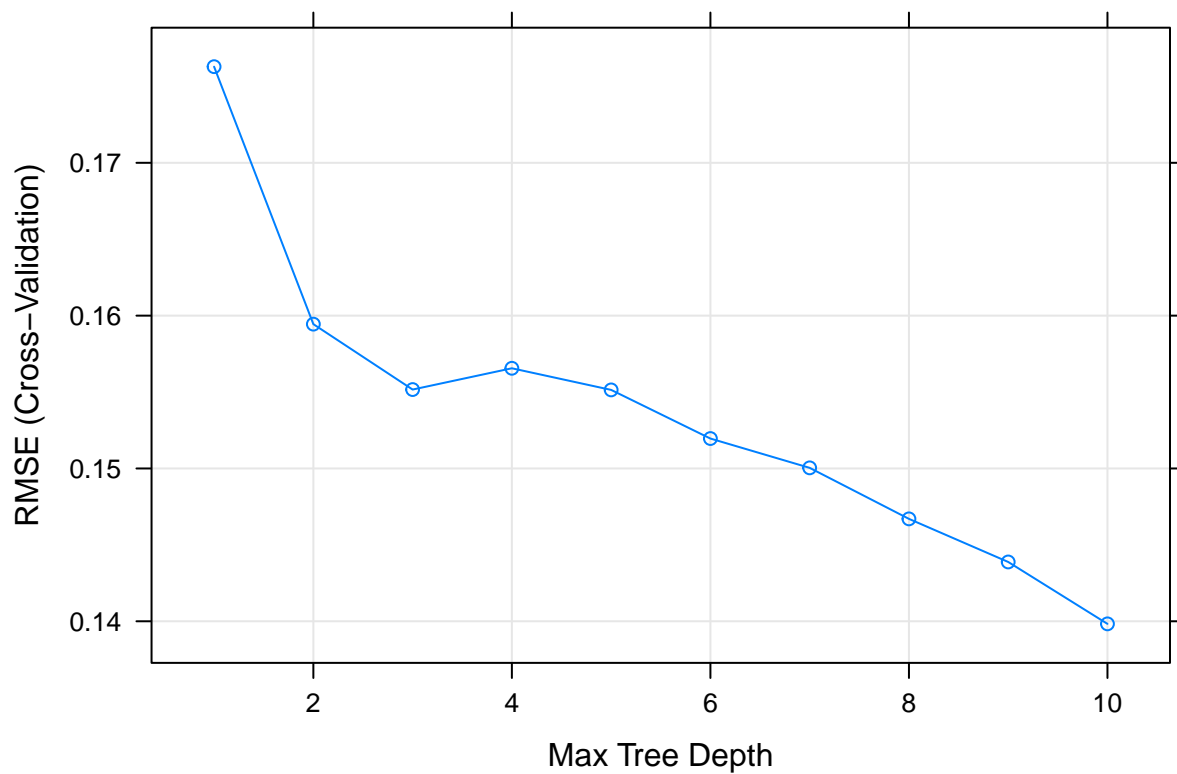
tr.TP <- caret::train(TP ~ .,
  data = (narm_water20 %>% sample_frac(0.5)),
  method = "rpart2",
  trControl = fitControl,
  tuneGrid = data.frame(maxdepth=1:10))

fancyRpartPlot(tr.TP$finalModel)
```



Rattle 2021-Jun-22 13:02:18 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=1:10))

  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)
}

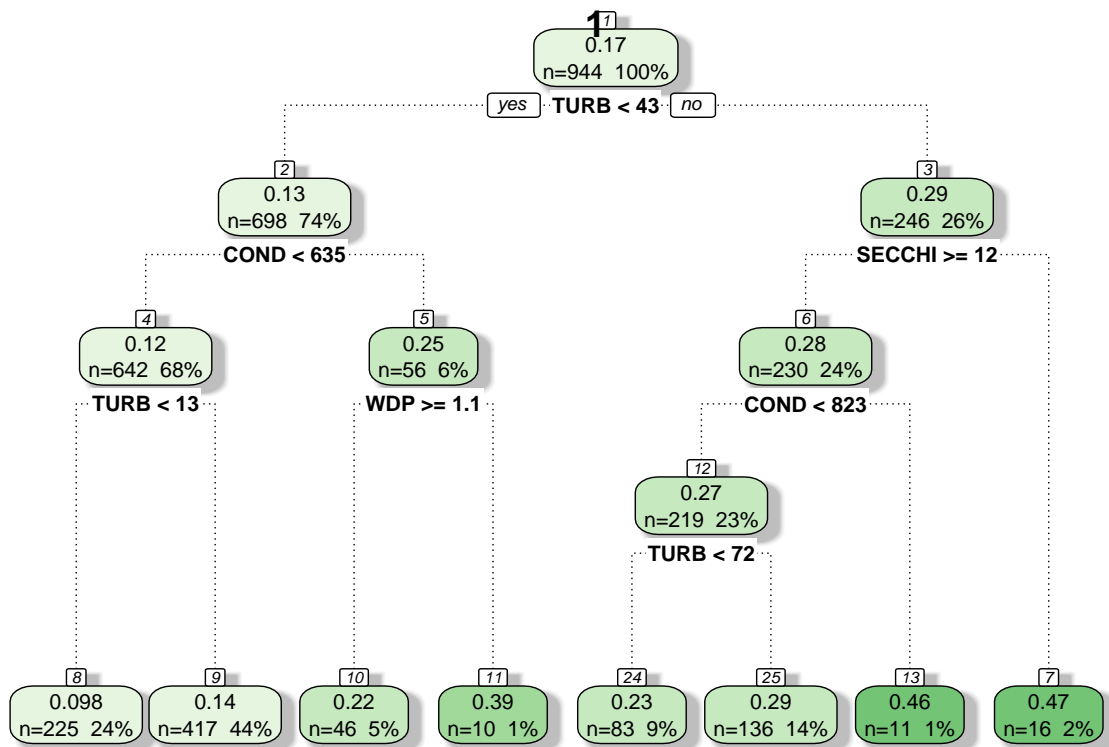
```

```

# 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

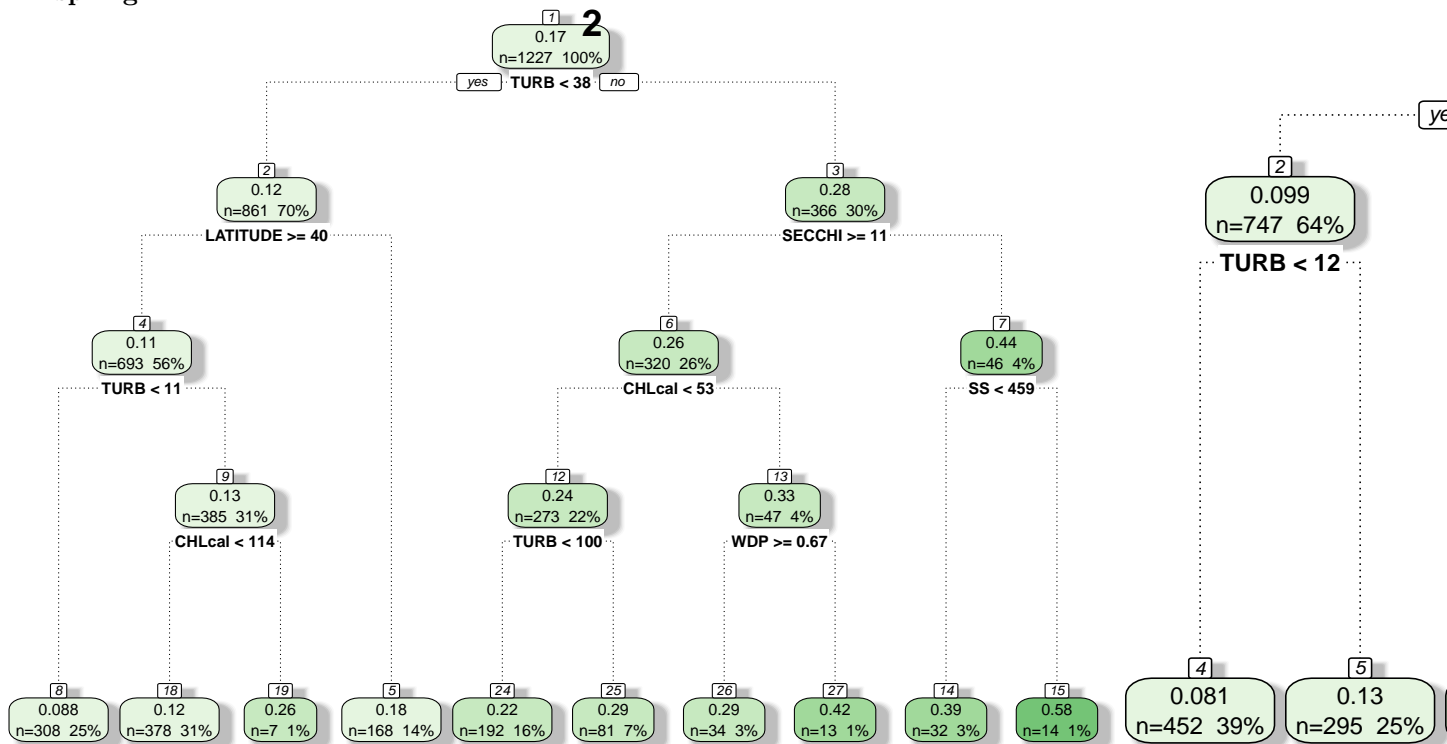
# the 4 comes from (2020 - 2000) / 5
lapply(1:4, function(x) return(fancyRpartPlot(trees.sp.2000.2020[[x]]$finalModel,
                                              main = paste(as.character(x)) )))

```

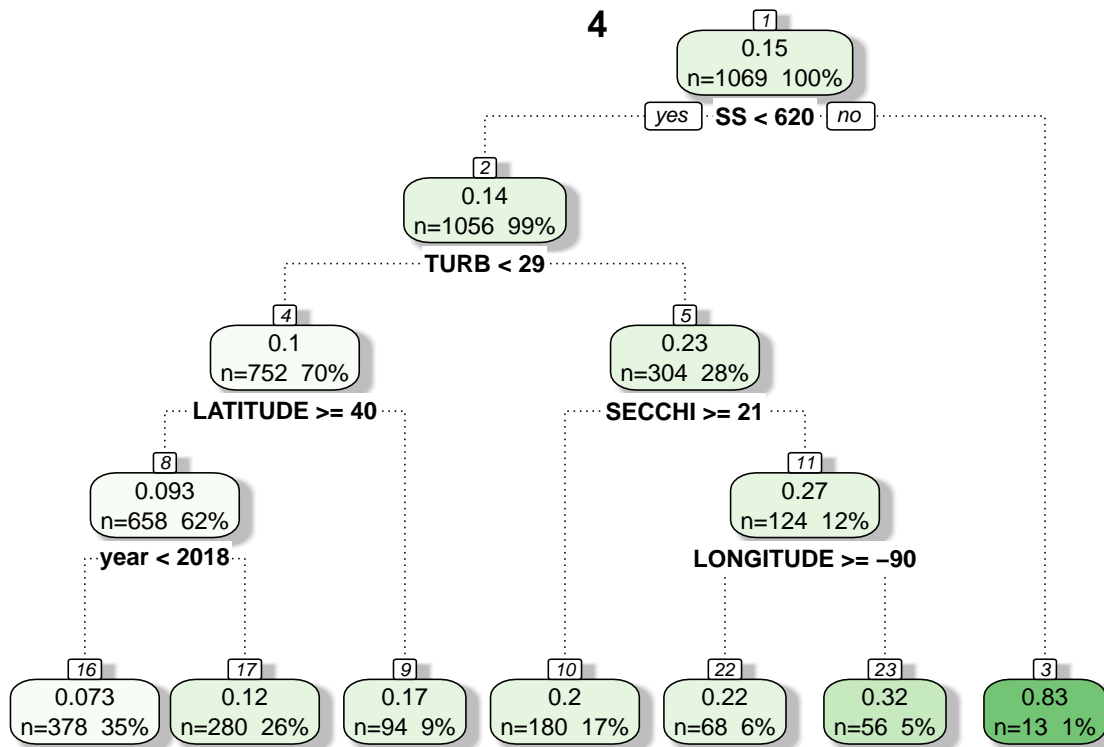


TP spring

Rattle 2021-Jun-22 13:02:23 amba



Rattle 2021-Jun-22 13:02:23 amba



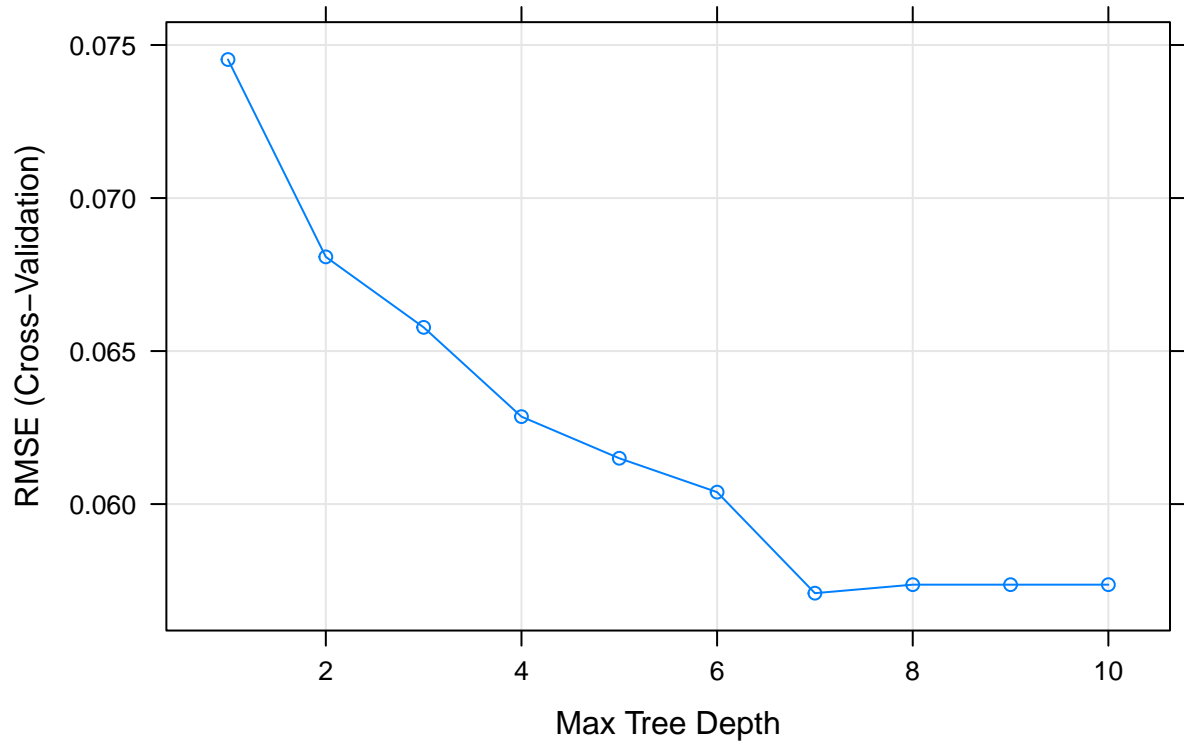
Rattle 2021-Jun-22 13:02:23 amba

```
## [[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)) )))
```

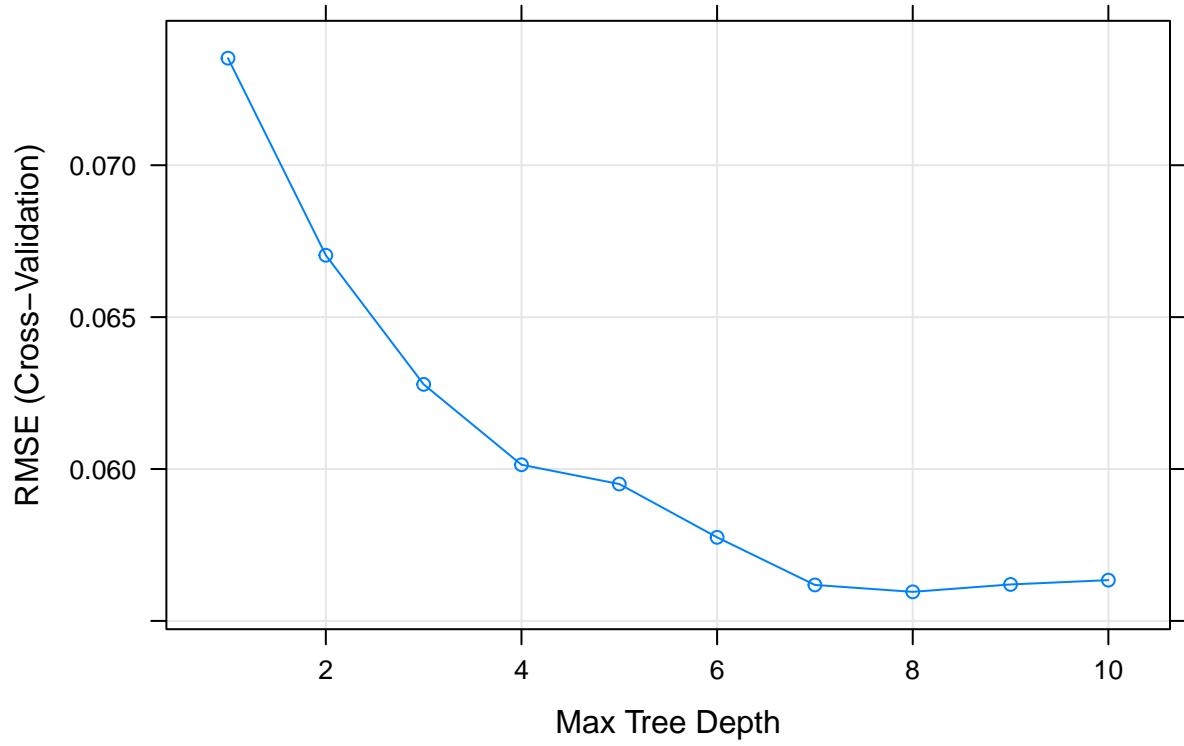
```
## [[1]]
```


1



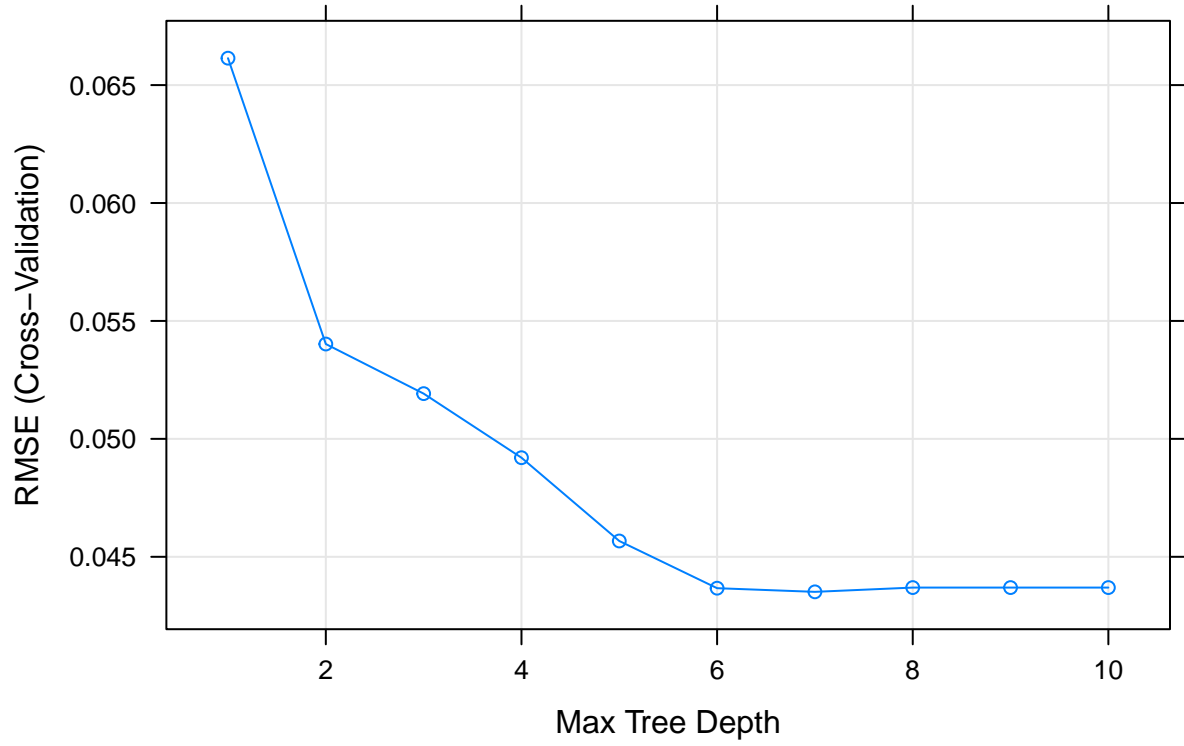
```
##  
## [[2]]
```

2



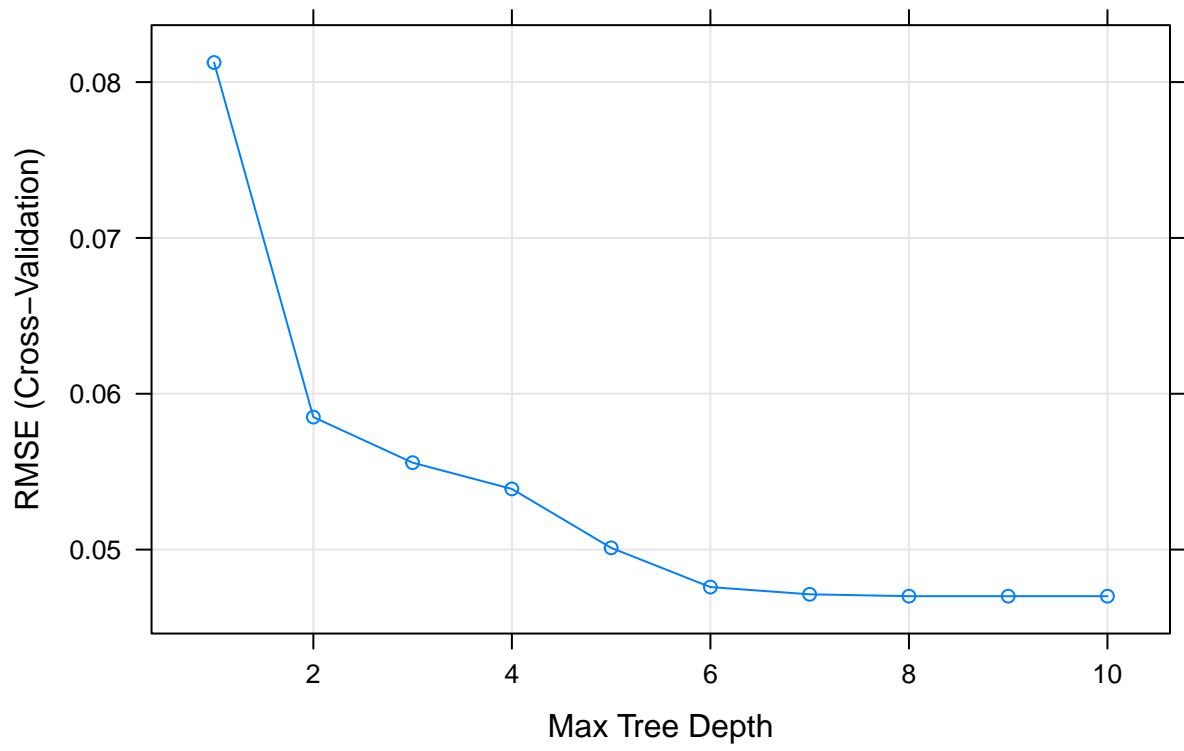
```
##  
## [[3]]
```

3



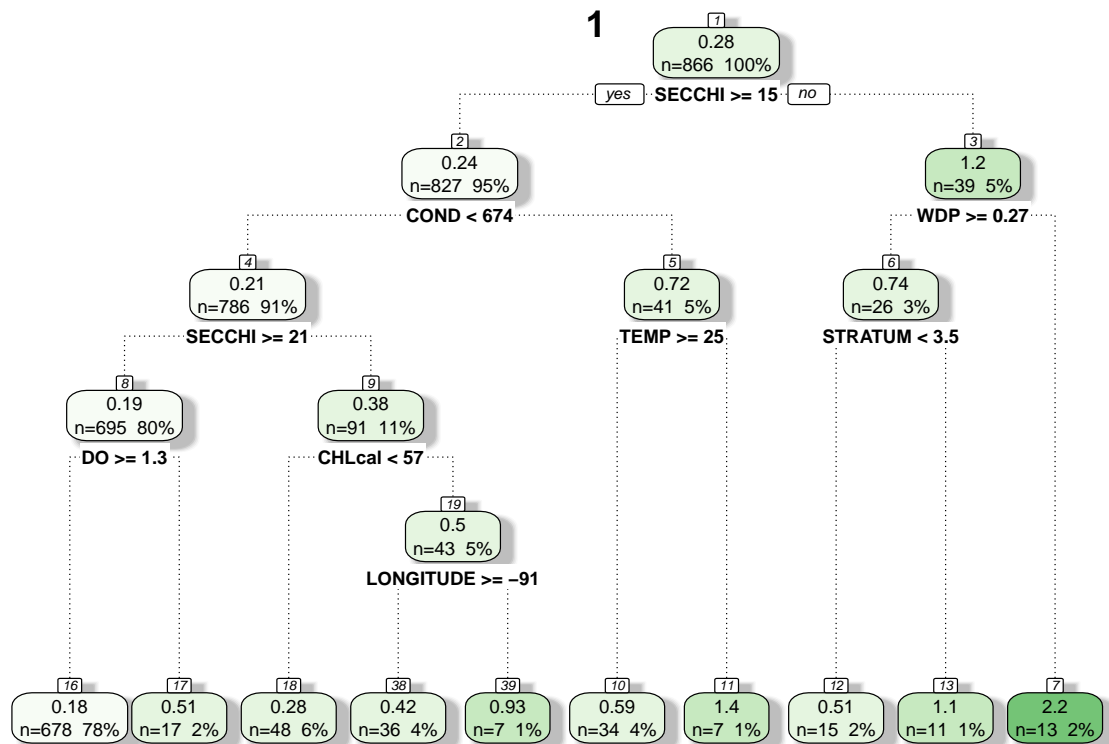
```
##  
## [[4]]
```

4

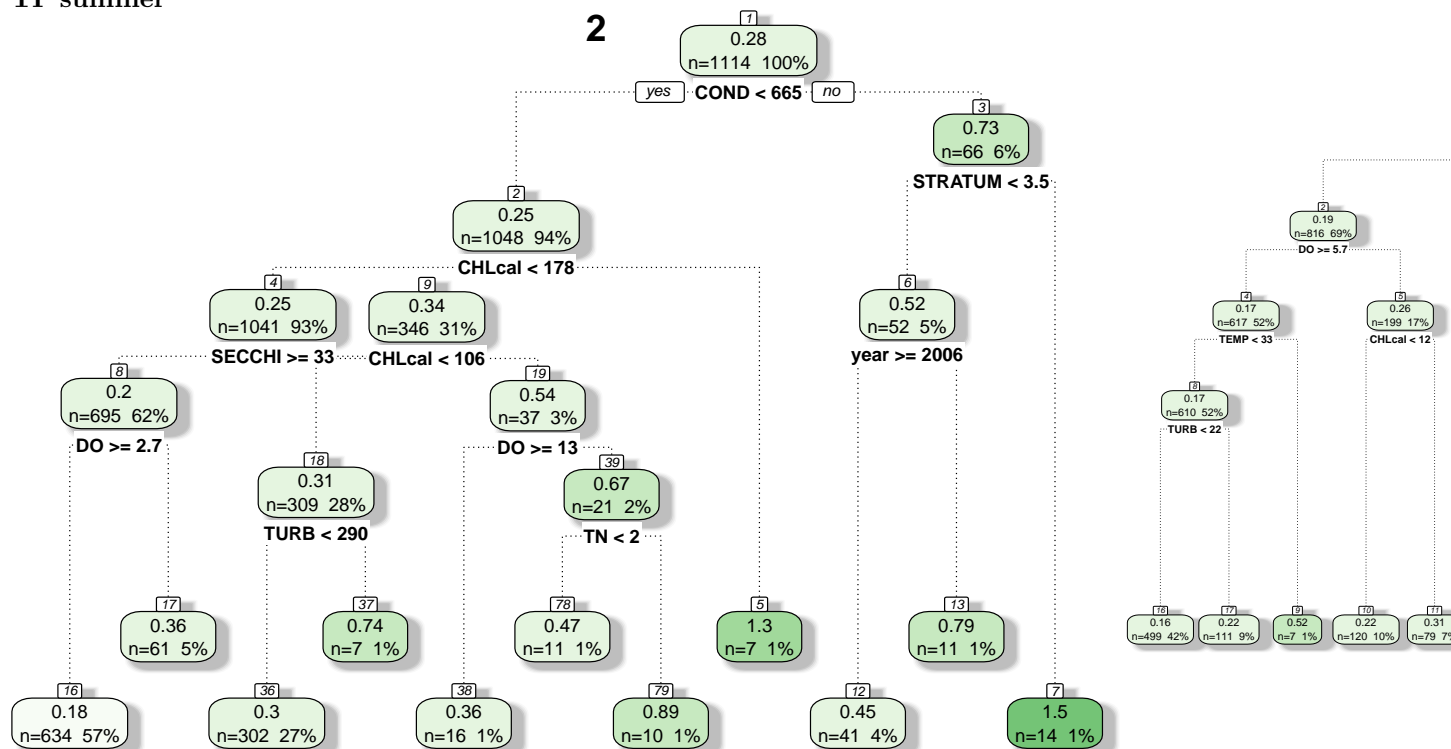


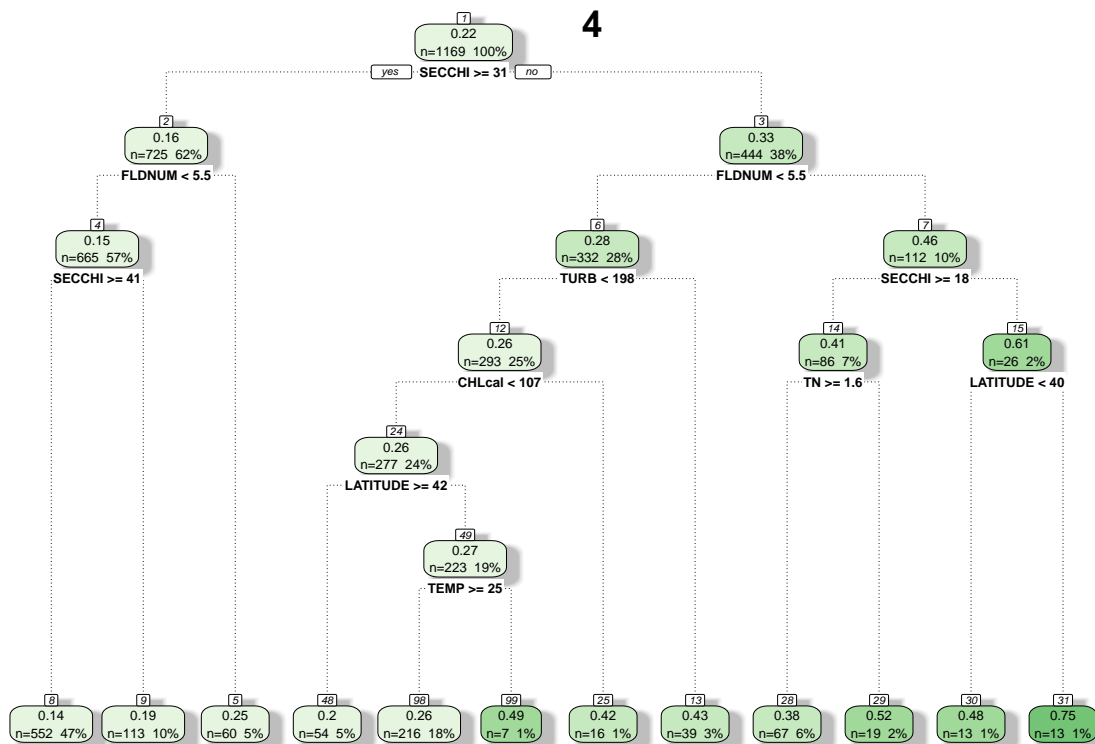
```
# model for summer, 2000-2020
trees.su.2000.2020 <- tree_by_season(2, 2000, 2020, 5, narm_water20)

# 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)) )))
```



Rattle 2021-Jun-22 13:02:31 amba





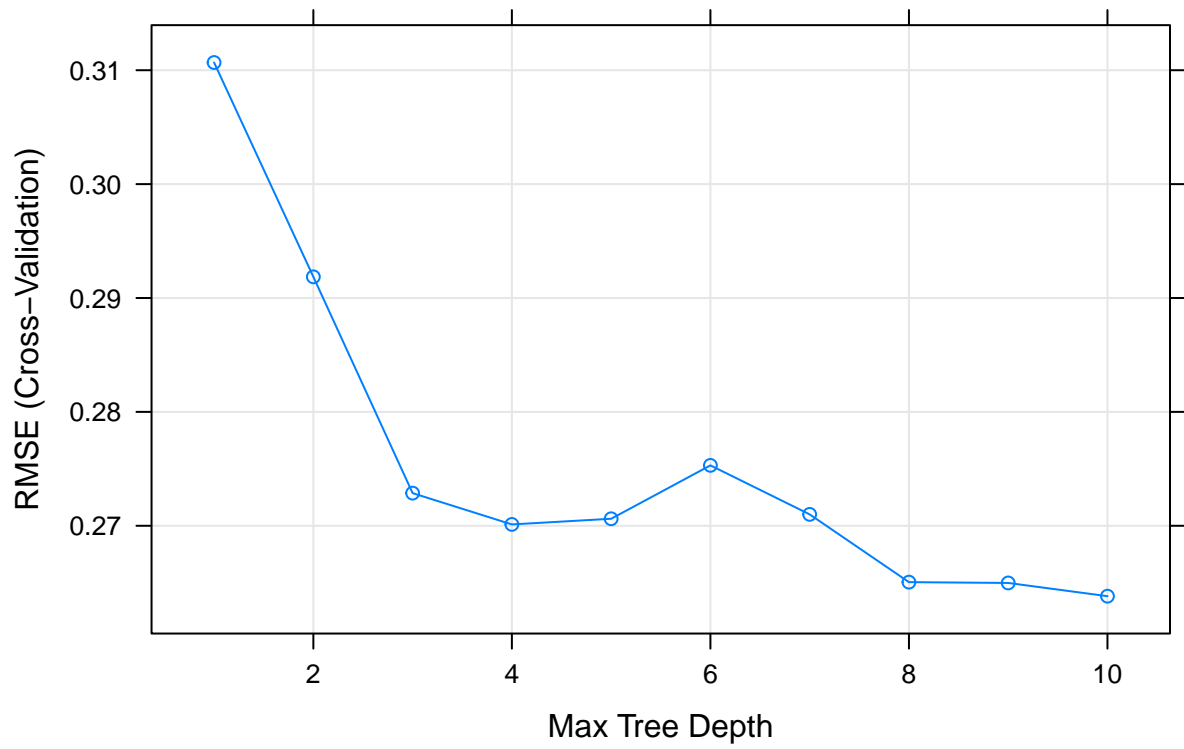
Rattle 2021-Jun-22 13:02:31 amba

```
## [[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)) )))
```

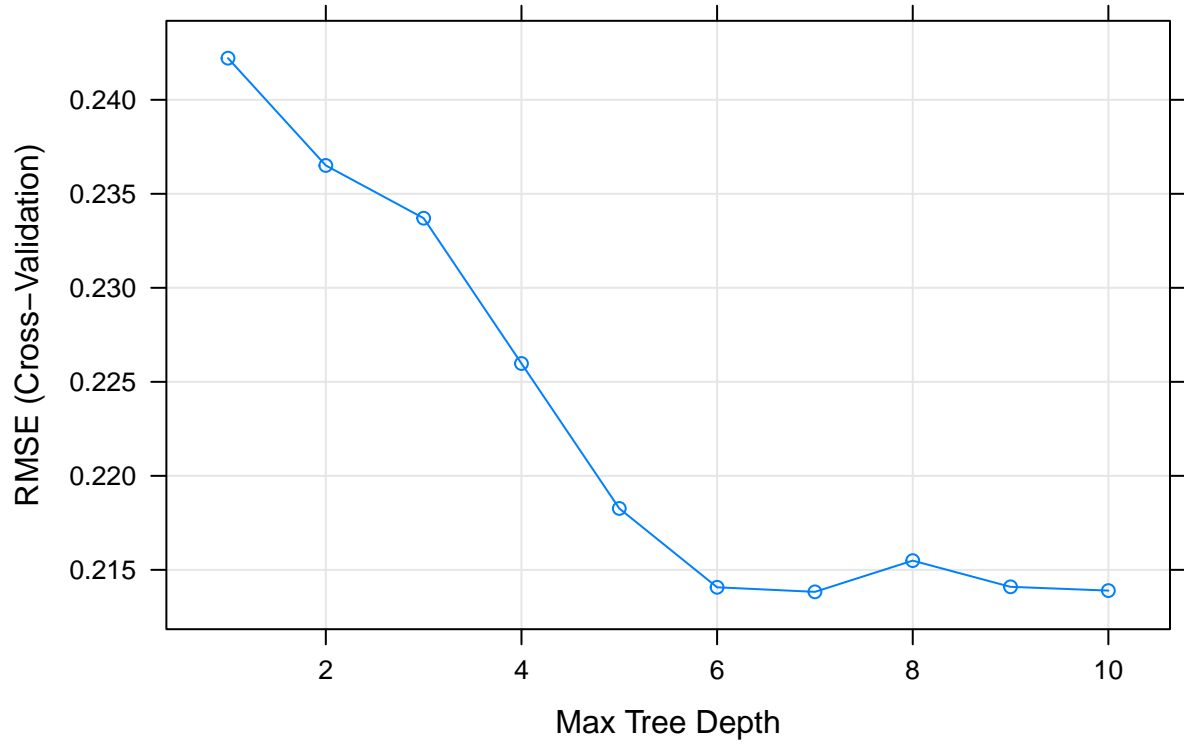
```
## [[1]]
```

1



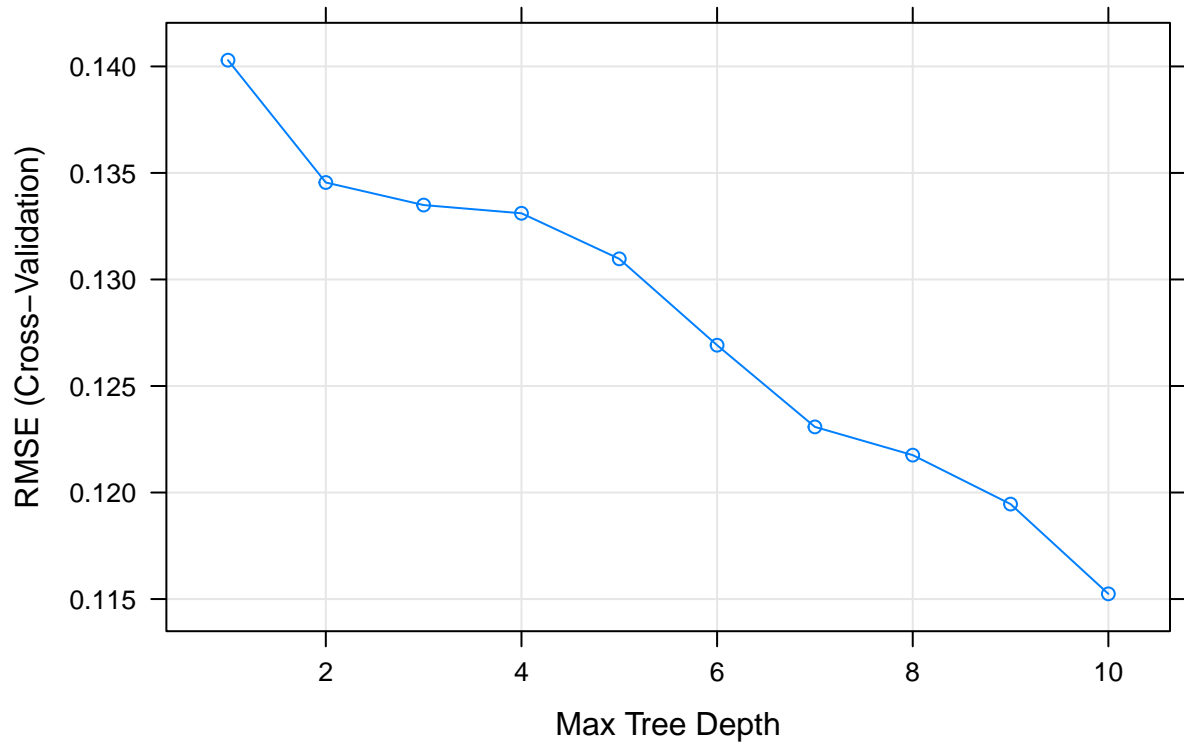
```
##  
## [[2]]
```


2



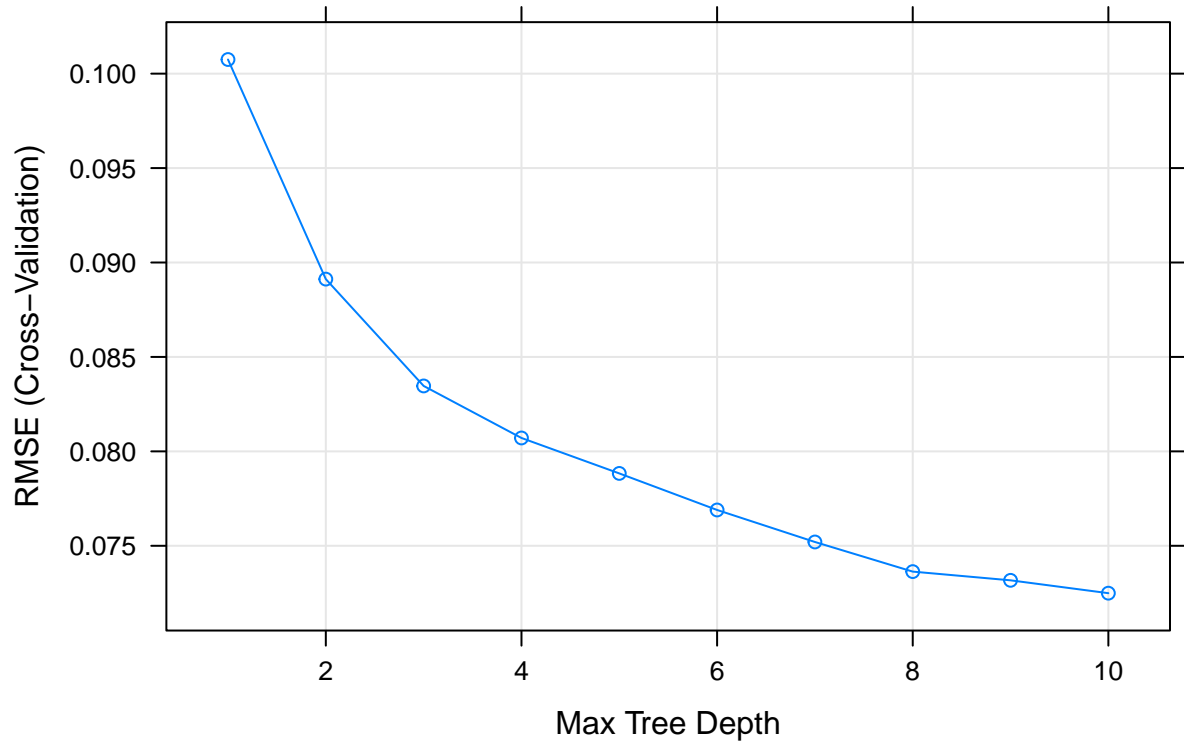
```
##  
## [[3]]
```

3



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

4

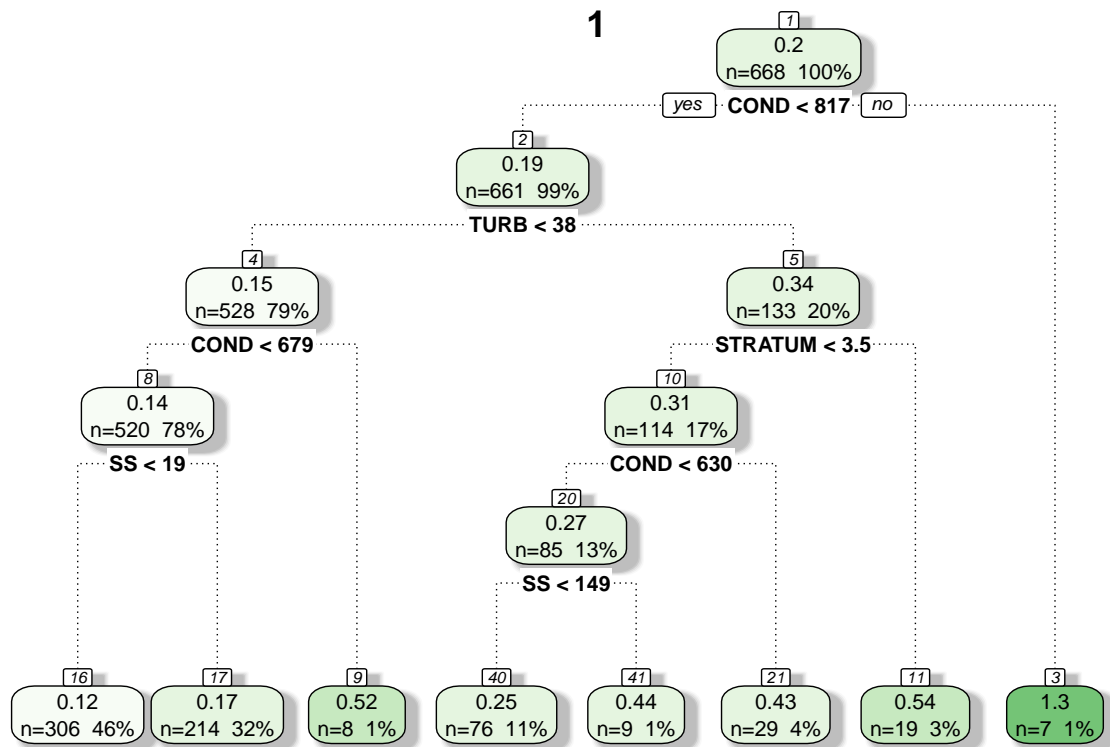


```

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

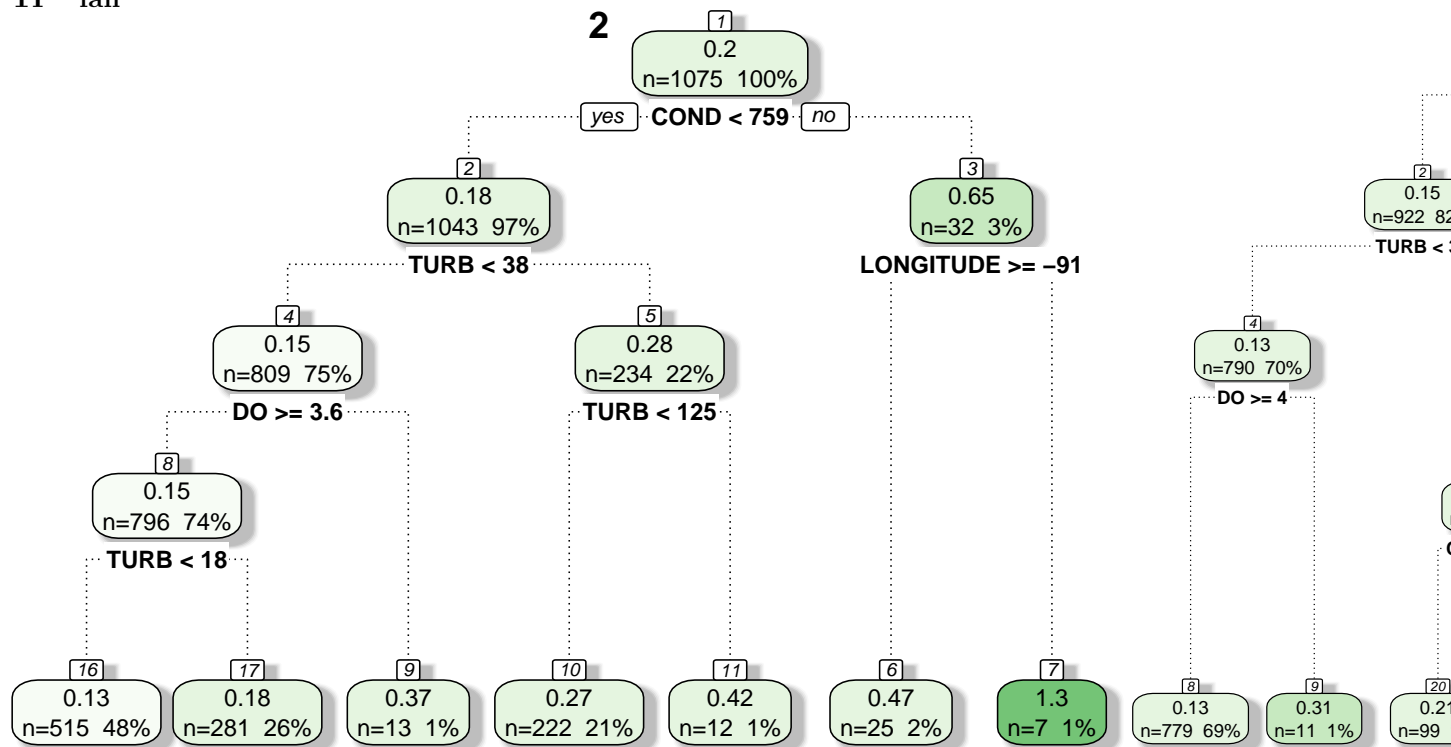
# the *2 is to insert line breaks
lapply(1:4*2, function(x)
  if (x %% 2 == 1) { # if x is even
    asis_output("\\\\[2cm]")
  } else {
    return(fancyRpartPlot(trees.fa.2000.2020[[x/2]]$finalModel,
                          main = paste(as.character(x/2)) ))
  } )

```

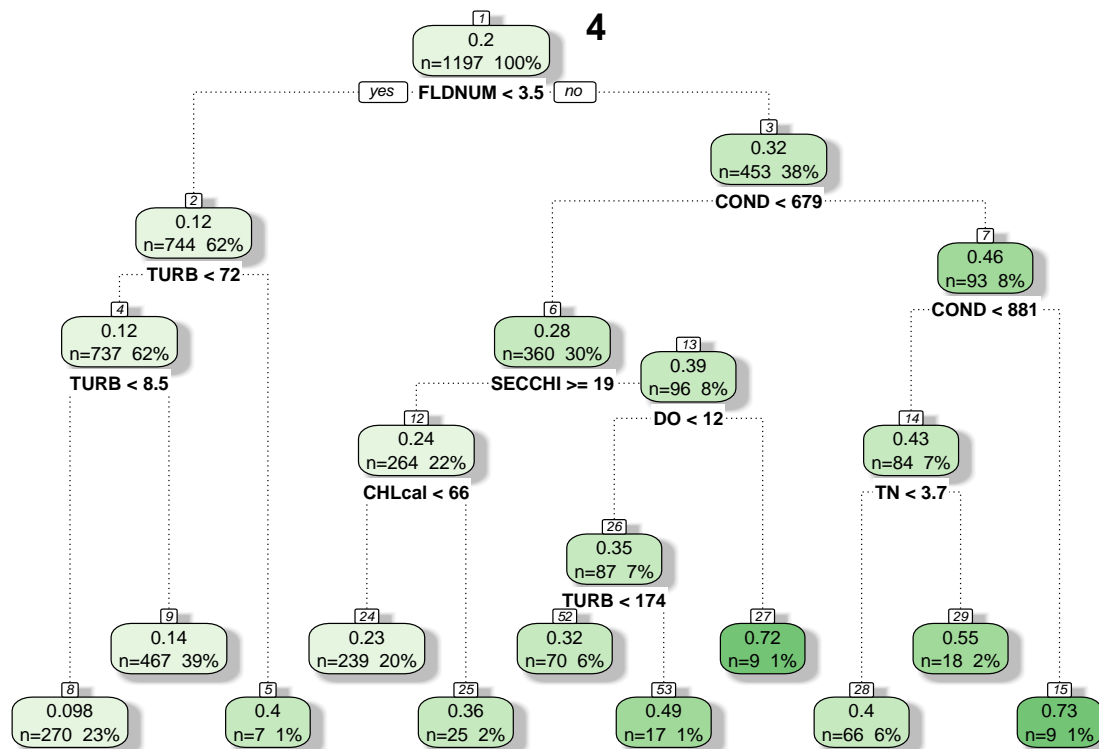


TP fall

Rattle 2021-Jun-22 13:02:41 amba



Rattle 2021-Jun-22 13:02:41 amba



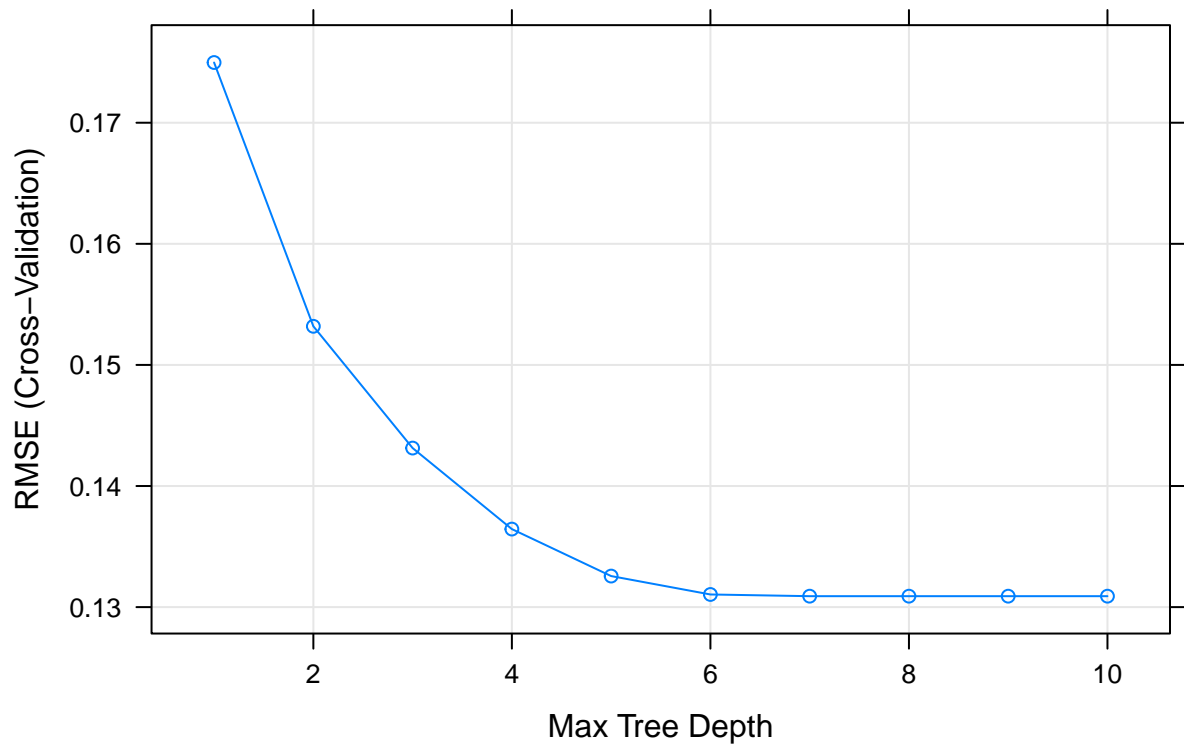
Rattle 2021-Jun-22 13:02:42 amba

```
## [[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)) )))
```

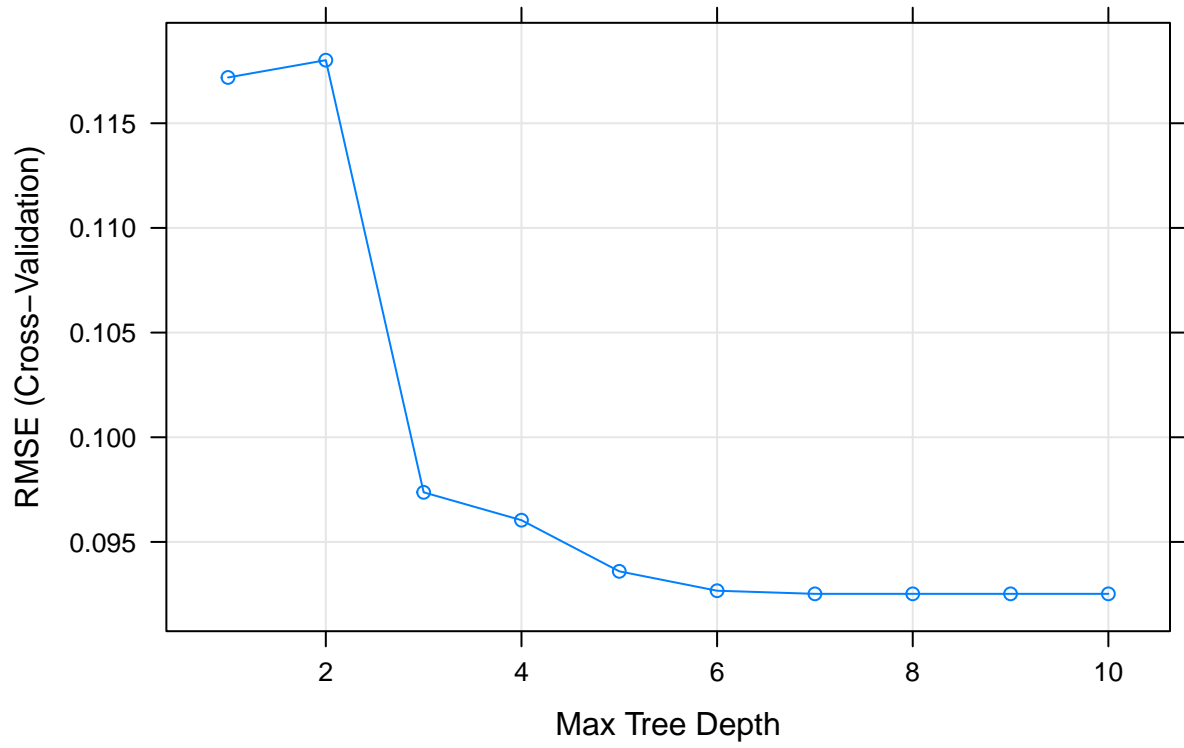
```
## [[1]]
```

1



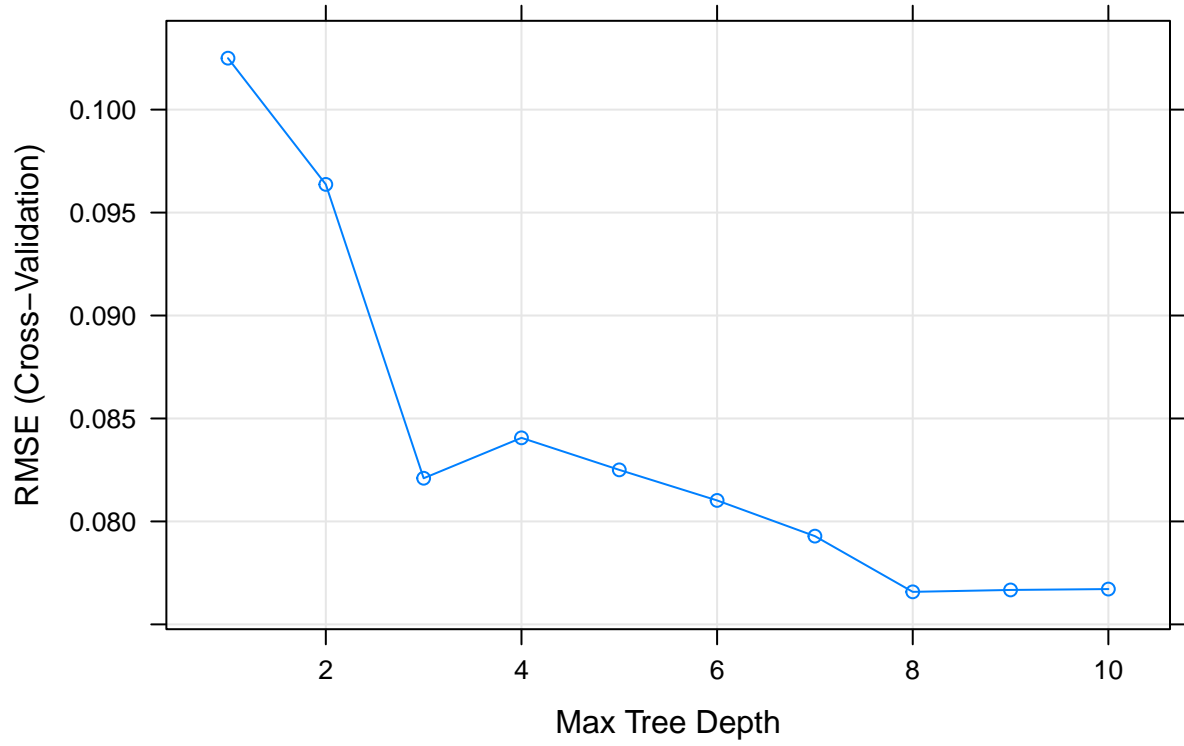
```
##  
## [[2]]
```

2



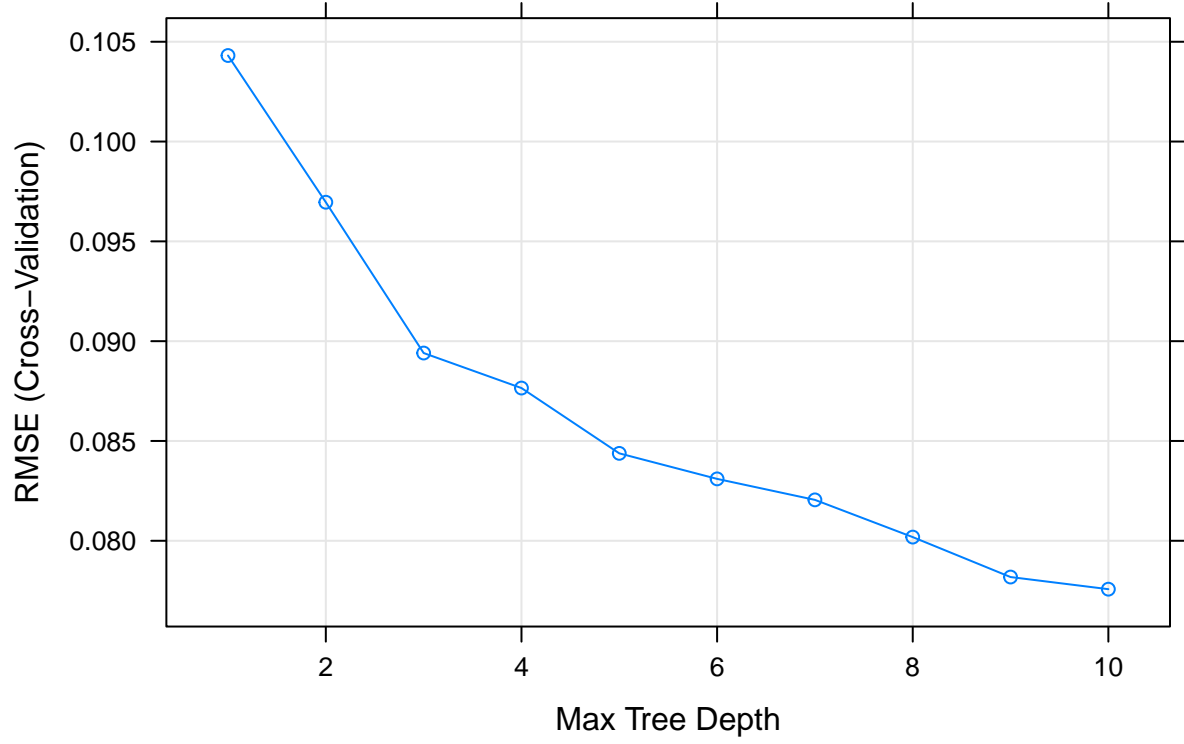
```
##  
## [[3]]
```


3



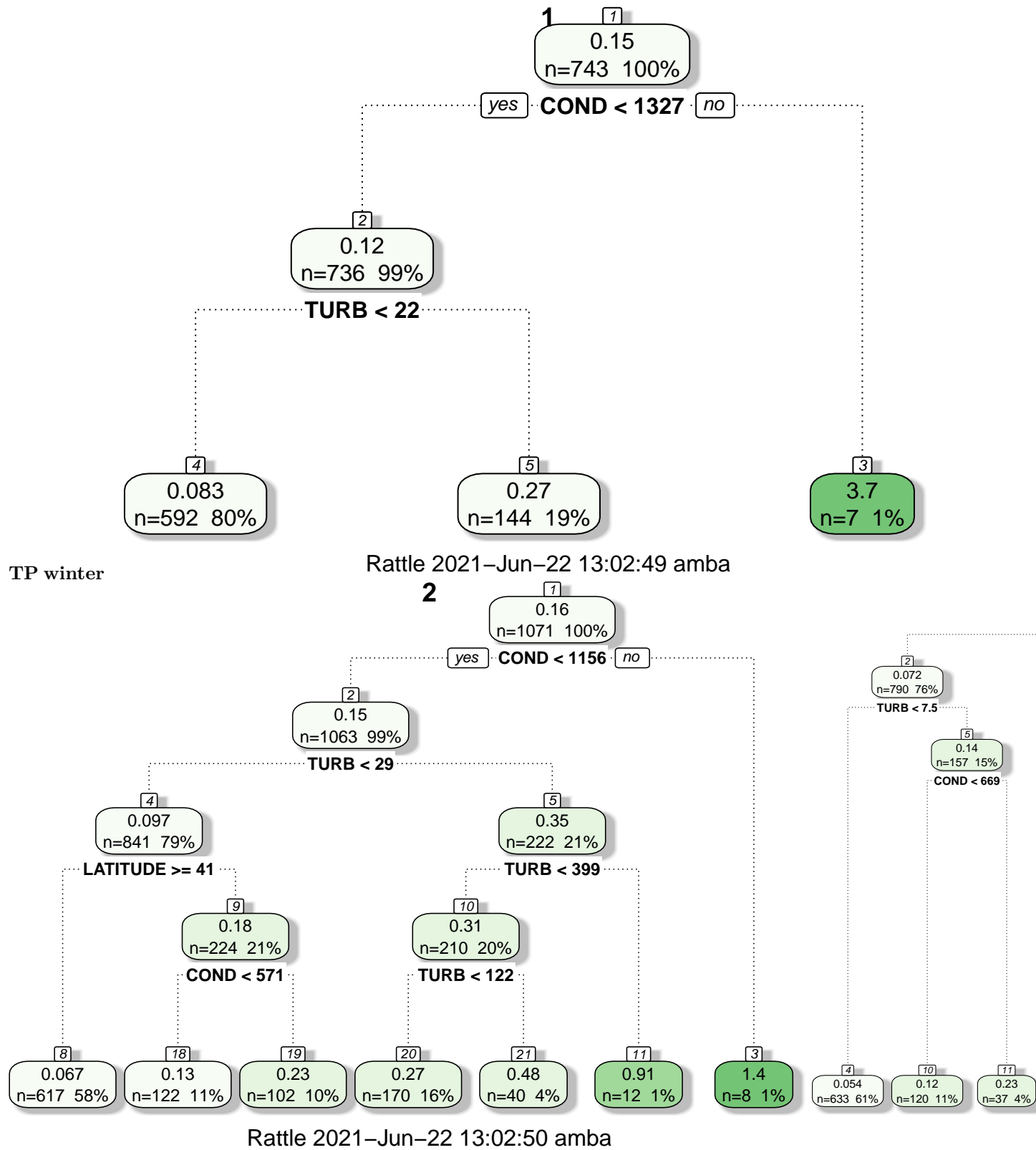
```
##  
## [[4]]
```

4

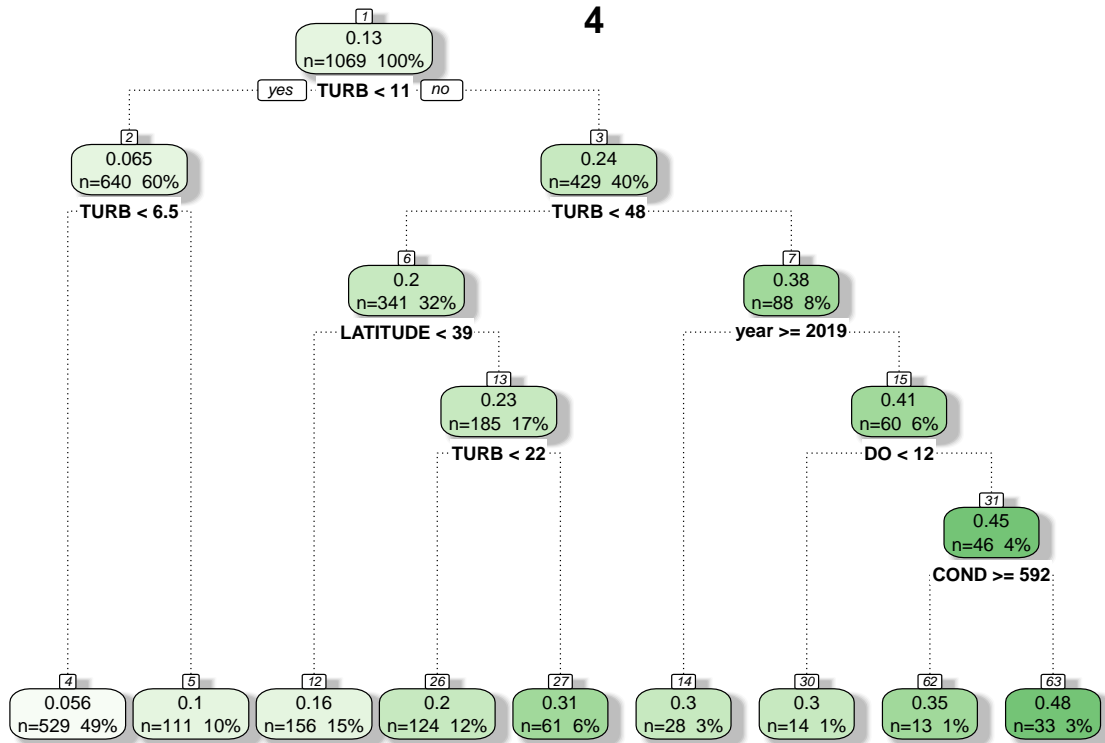


```
# model for winter, 2000-2020
trees.wi.2000.2020 <- tree_by_season(4, 2000, 2020, 5, norm_water20)

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



4



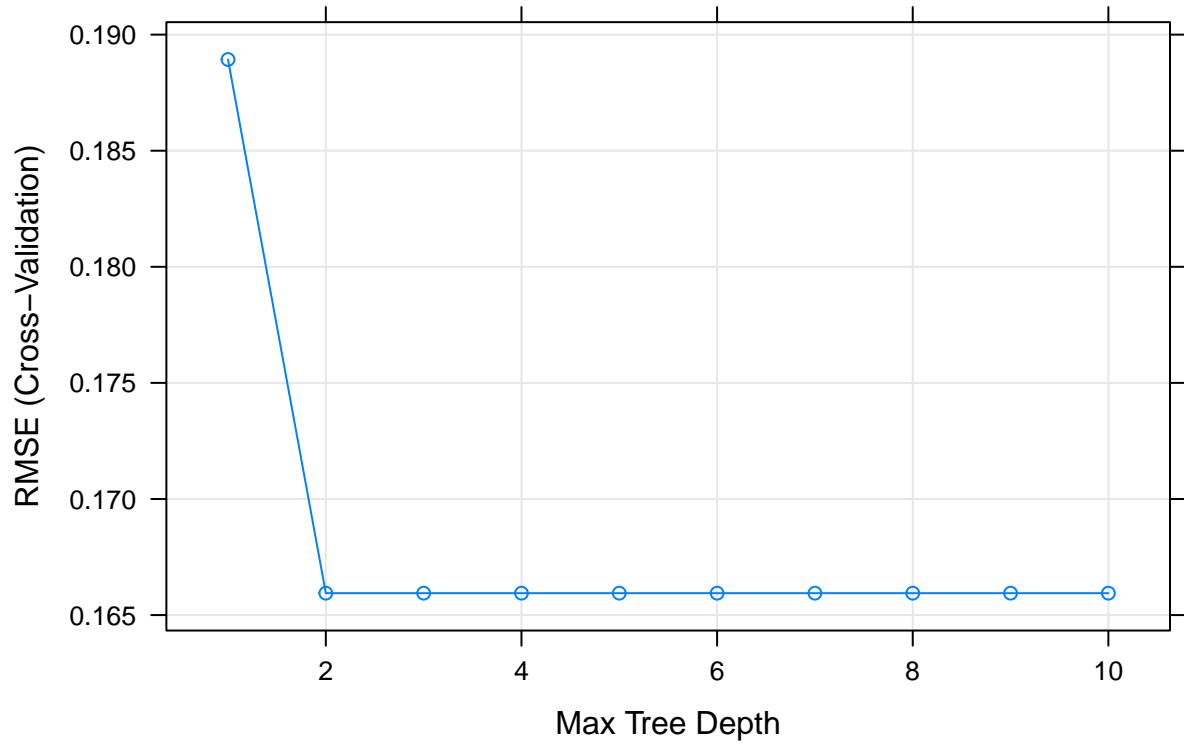
Rattle 2021-Jun-22 13:02:50 amba

```
## [[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)) )))
```

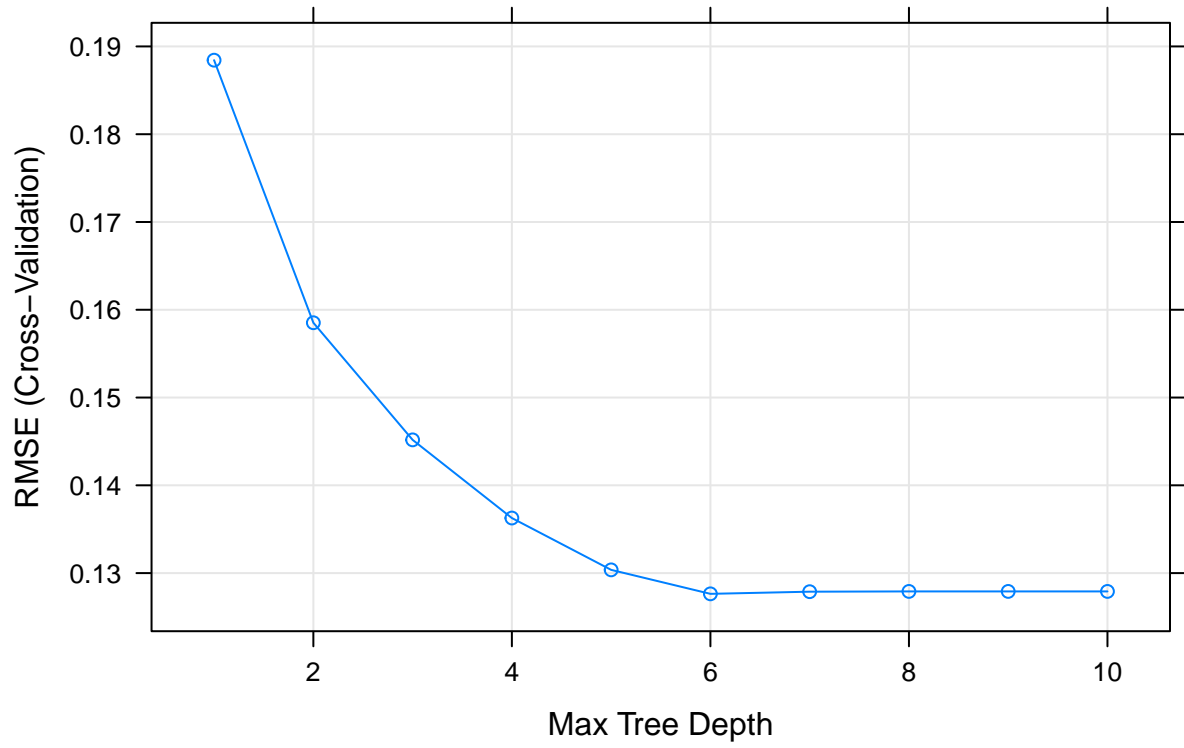
```
## [[1]]
```

1



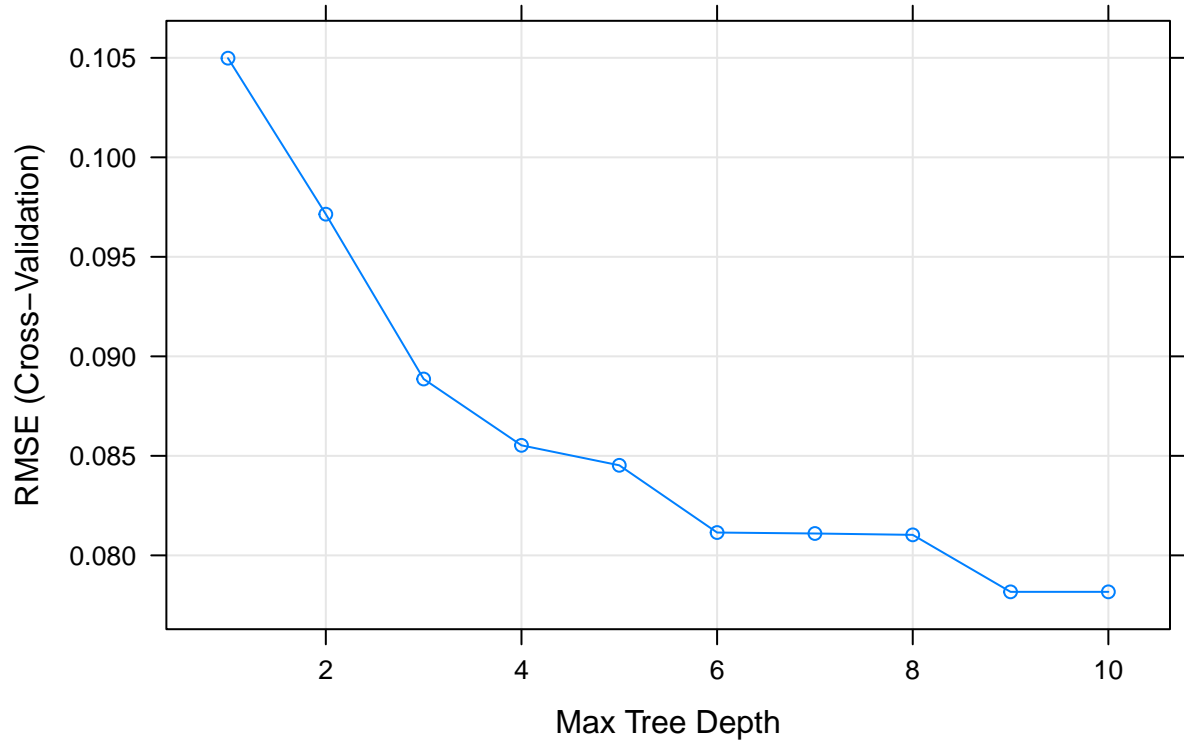
```
##  
## [[2]]
```

2



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

3



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


4

