A-V.R

PC

2022-02-09

###############################  
# Ucitavanje potrebnih paketa #  
###############################  
  
  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.3 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(caret)

## Warning: package 'caret' was built under R version 4.1.2

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(ggplot2)  
library(broom)  
library(dplyr)  
library(rpart)  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.1.2

library(rattle)

## Warning: package 'rattle' was built under R version 4.1.2

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.  
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(RColorBrewer)  
library(ISLR)

## Warning: package 'ISLR' was built under R version 4.1.2

library(readr)  
library(e1071)

## Warning: package 'e1071' was built under R version 4.1.2

library(ISOweek)

## Warning: package 'ISOweek' was built under R version 4.1.2

library(tibble)  
library(readxl)  
library(writexl)  
library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library(corrplot)

## Warning: package 'corrplot' was built under R version 4.1.2

## corrplot 0.92 loaded

###############################  
# Ucitavanje sirovih podataka #  
###############################  
  
# Podaci se odnose na vremenske prilike u Beogradu izmedju 01.01.2016. i 31.12.2020.  
# Skrejpovani su pomocu Pajton skripte u prilogu (Jupyter Notebook) i besplatne   
# verzije API-ja: https://www.worldweatheronline.com/developer/   
  
raw\_data <- read\_excel("Belgrade\_raw\_2016\_2020.xlsx")  
view(raw\_data)  
str(raw\_data)

## tibble [1,827 x 25] (S3: tbl\_df/tbl/data.frame)  
## $ date\_time : POSIXct[1:1827], format: "2016-01-01" "2016-01-02" ...  
## $ maxtempC : num [1:1827] 0 1 -6 3 0 3 1 2 6 9 ...  
## $ mintempC : num [1:1827] -6 -3 -11 -11 -7 0 -3 -5 2 5 ...  
## $ totalSnow\_cm : num [1:1827] 0 0.2 0.5 0.7 0 0 1.2 0 0 0 ...  
## $ sunHour : num [1:1827] 6.9 6.9 3.3 6.9 8.7 3.4 3.4 8.7 5.2 5.2 ...  
## $ uvIndex : num [1:1827] 1 1 1 1 1 1 1 2 2 2 ...  
## $ moon\_illumination: num [1:1827] 50 43 36 28 21 14 7 0 0 0 ...  
## $ moonrise : chr [1:1827] "No moonrise" "2.2916666666666669E-2" "6.3888888888888884E-2" "0.10347222222222223" ...  
## $ moonset : chr [1:1827] "0.50138888888888888" "0.51944444444444449" "0.53888888888888886" "0.55972222222222223" ...  
## $ sunrise : POSIXct[1:1827], format: "1899-12-31 08:16:00" "1899-12-31 08:16:00" ...  
## $ sunset : POSIXct[1:1827], format: "1899-12-31 17:07:00" "1899-12-31 17:08:00" ...  
## $ DewPointC : num [1:1827] -17 -9 -9 -4 -4 1 -1 -3 3 6 ...  
## $ FeelsLikeC : num [1:1827] -5 -6 -16 -6 -6 -2 -5 -2 1 5 ...  
## $ HeatIndexC : num [1:1827] -3 -2 -7 -2 -3 2 0 0 4 7 ...  
## $ WindChillC : num [1:1827] -5 -6 -16 -6 -6 -2 -5 -2 1 5 ...  
## $ WindGustKmph : num [1:1827] 10 20 44 22 19 22 34 17 33 28 ...  
## $ cloudcover : num [1:1827] 35 59 99 56 40 88 79 27 71 66 ...  
## $ humidity : num [1:1827] 35 58 85 85 92 95 93 85 92 91 ...  
## $ precipMM : num [1:1827] 0 0.2 5.8 1 0.2 16.1 11.9 0 1.7 4.7 ...  
## $ pressure : num [1:1827] 1029 1025 1018 1019 1008 ...  
## $ tempC : num [1:1827] 0 1 -6 3 0 3 1 2 6 9 ...  
## $ visibility : num [1:1827] 10 9 4 8 8 9 6 10 10 10 ...  
## $ winddirDegree : num [1:1827] 222 150 109 196 140 101 245 202 146 174 ...  
## $ windspeedKmph : num [1:1827] 6 14 31 12 10 12 22 8 16 13 ...  
## $ location : chr [1:1827] "Belgrade" "Belgrade" "Belgrade" "Belgrade" ...

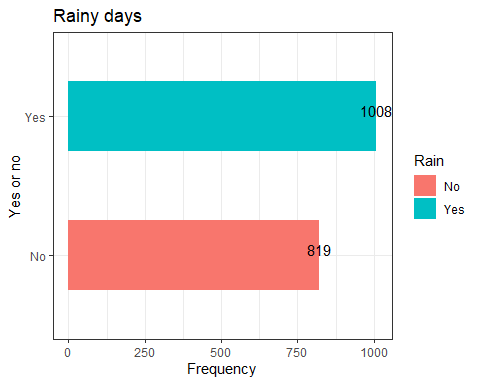
################################  
# Priprema podataka za analizu #  
################################  
  
# Cilj projekta je naci najbolji decision tree klasifikacioni algoritam za predvidjanje  
# da li ce odredenog dana padati kisa spram ostalih vremenskih varijabli.  
# Za te potrebe cemo prvo srediti podatke i konstruisati  
# zavisnu varijablu "rain" kao binarnu varijablu sa dva nivoa 1/0), kao i par dodatnih nezavisnih.  
  
# Selekcija i reimenovanje kolona (varijabli)  
daily\_weather\_df <- as\_tibble(raw\_data)  
  
daily\_weather\_df <- daily\_weather\_df %>%  
 select(  
 date\_time,  
 maxtempC,  
 mintempC,  
 humidity,  
 windspeedKmph,  
 WindGustKmph,  
 DewPointC,  
 cloudcover,  
 precipMM,  
 totalSnow\_cm,   
 pressure  
 ) %>%  
 rename(  
 date = date\_time,  
 max\_temp\_c = maxtempC,  
 min\_temp\_c = mintempC,  
 wind\_speed\_kmph = windspeedKmph,  
 wind\_gust\_kmph = WindGustKmph,  
 dew\_point\_c = DewPointC,  
 cloud\_cover = cloudcover,  
 precip\_mm = precipMM,  
 total\_snow\_cm = totalSnow\_cm,   
 )  
  
# Konstrukcija zavisne varijable  
daily\_weather\_df$rain <- ifelse(daily\_weather\_df$precip\_mm > 0, 1, 0)  
table(daily\_weather\_df$rain) #0:819; 1:1008

##   
## 0 1   
## 819 1008

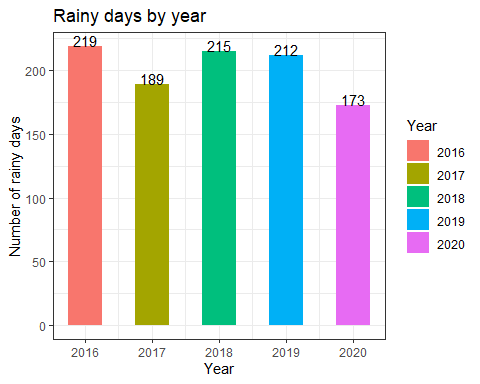
# Konstrukcija novih nezavisnih varijabli  
daily\_weather\_df$mean\_temp\_c <- (daily\_weather\_df$max\_temp\_c + daily\_weather\_df$min\_temp\_c) / 2  
daily\_weather\_df$temp\_difference <- daily\_weather\_df$max\_temp\_c - daily\_weather\_df$min\_temp\_c  
daily\_weather\_df$temp\_difference\_extreme <- ifelse(daily\_weather\_df$temp\_difference > 10, 1, 0)  
daily\_weather\_df$year <- year(daily\_weather\_df$date)  
daily\_weather\_df$month <- month(daily\_weather\_df$date)  
daily\_weather\_df$iso\_week <- isoweek(daily\_weather\_df$date)  
  
# Eksportovanje pripremljenih podataka u eksel fajl  
clean\_data <- daily\_weather\_df[, c(12, 1, 16, 17, 18, 13, 2, 3, 14, 15, 4, 5, 6, 7, 8, 10, 11)]  
write\_xlsx(clean\_data, "clean\_data.xlsx")   
read\_excel('clean\_data.xlsx')

## # A tibble: 1,827 x 17  
## rain date year month iso\_week mean\_temp\_c max\_temp\_c  
## <dbl> <dttm> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0 2016-01-01 00:00:00 2016 1 53 -3 0  
## 2 1 2016-01-02 00:00:00 2016 1 53 -1 1  
## 3 1 2016-01-03 00:00:00 2016 1 53 -8.5 -6  
## 4 1 2016-01-04 00:00:00 2016 1 1 -4 3  
## 5 1 2016-01-05 00:00:00 2016 1 1 -3.5 0  
## 6 1 2016-01-06 00:00:00 2016 1 1 1.5 3  
## 7 1 2016-01-07 00:00:00 2016 1 1 -1 1  
## 8 0 2016-01-08 00:00:00 2016 1 1 -1.5 2  
## 9 1 2016-01-09 00:00:00 2016 1 1 4 6  
## 10 1 2016-01-10 00:00:00 2016 1 1 7 9  
## # ... with 1,817 more rows, and 10 more variables: min\_temp\_c <dbl>,  
## # temp\_difference <dbl>, temp\_difference\_extreme <dbl>, humidity <dbl>,  
## # wind\_speed\_kmph <dbl>, wind\_gust\_kmph <dbl>, dew\_point\_c <dbl>,  
## # cloud\_cover <dbl>, total\_snow\_cm <dbl>, pressure <dbl>

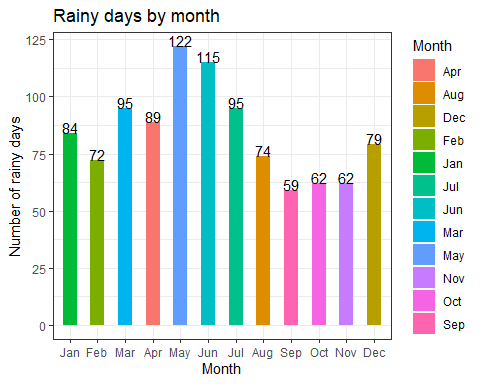
#########################  
# Eksploratorna analiza #  
#########################  
  
# Bar chart kisnih dana  
rain\_tab <- data.frame(table(clean\_data$rain))  
colnames(rain\_tab) <- c('Rain', 'Frequency')  
rain\_tab$Rain <- c('No', 'Yes')  
  
ggplot(rain\_tab, aes(x = Frequency, y = reorder(Rain, Frequency), fill = Rain)) +  
 geom\_bar(stat = "identity", width = 0.5) +  
 geom\_text(aes(label= round(Frequency)), vjust=0) +  
 labs(y = 'Yes or no', x = "Frequency", title= "Rainy days") +  
 theme(legend.position="none")+  
 theme\_bw()



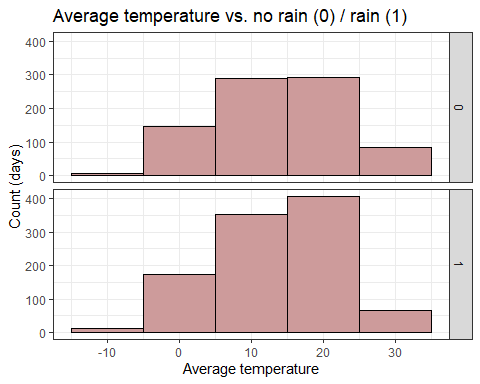
# Bar chart kisnih dana po godinama  
rain\_year <- clean\_data %>% group\_by(year) %>%  
 summarise(rainy\_days\_sum = sum(rain))  
  
ggplot(rain\_year, aes(x = year, y = rainy\_days\_sum, fill = factor(year))) +  
 geom\_bar(stat = "identity", width = 0.5) +  
 geom\_text(aes(label= rainy\_days\_sum), vjust=0) +  
 labs(y= "Number of rainy days", x = "Year", title= "Rainy days by year", fill = "Year") +  
 theme(legend.position="none") +  
 theme\_bw()



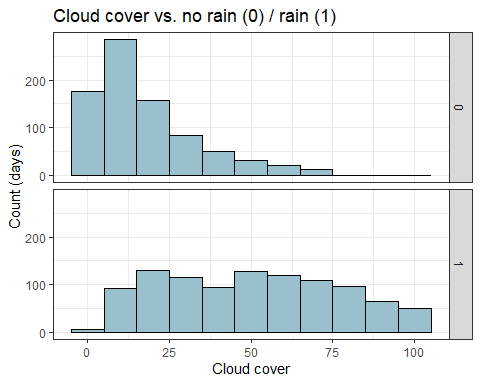
# Bar chart kisnih dana po mesecima  
rain\_month <- clean\_data %>% group\_by(month) %>%  
 summarise(rainy\_days\_sum = sum(rain))  
rain\_month$month\_name <- c('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',  
 'Sep', 'Oct', 'Nov', 'Dec')  
  
  
ggplot(rain\_month, aes(x = reorder(month\_name, month), y = rainy\_days\_sum, fill = month\_name)) +  
 geom\_bar(stat = "identity", width = 0.5) +  
 geom\_text(aes(label= rainy\_days\_sum), vjust=0) +  
 labs(y= "Number of rainy days", x = "Month", title= "Rainy days by month", fill = 'Month') +  
 theme(legend.position="none") +  
 theme\_bw()



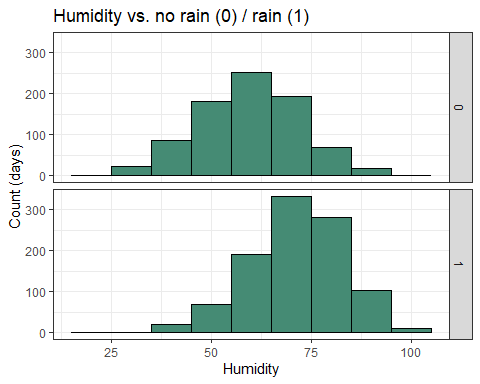
# Histogram prosecne temperature tokom kisnih i suvih dana  
  
ggplot(clean\_data, aes(x = mean\_temp\_c)) +  
 geom\_histogram(binwidth = 10, fill = "rosybrown3", colour = "black") +   
 facet\_grid(clean\_data$rain) +  
 labs(y= "Count (days)", x = "Average temperature", title= "Average temperature vs. no rain (0) / rain (1)") +  
 theme(legend.position="none") +  
 theme\_bw()



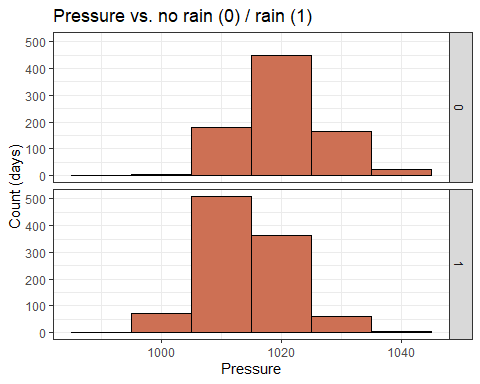
# Histogram oblacnosti tokom kisnih i suvih dana  
  
ggplot(clean\_data, aes(x = cloud\_cover)) +  
 geom\_histogram(binwidth = 10, fill = "lightblue3", colour = "black") +   
 facet\_grid(clean\_data$rain) +  
 labs(y= "Count (days)", x = "Cloud cover", title= "Cloud cover vs. no rain (0) / rain (1)") +  
 theme(legend.position="none") +  
 theme\_bw()



# Histogram vlaznosti tokom kisnih i suvih dana  
  
ggplot(clean\_data, aes(x = humidity)) +  
 geom\_histogram(binwidth = 10, fill = "aquamarine4", colour = "black") +   
 facet\_grid(clean\_data$rain) +  
 labs(y= "Count (days)", x = "Humidity", title= "Humidity vs. no rain (0) / rain (1)") +  
 theme(legend.position="none") +  
 theme\_bw()



# Historgam vazdusnog pritiska tokom kisnih i suvih dana  
  
ggplot(daily\_weather\_df, aes(x = pressure)) +  
 geom\_histogram(binwidth = 10, fill = "salmon3", colour = "black") +   
 facet\_grid(clean\_data$rain) +  
 labs(y= "Count (days)", x = "Pressure", title= "Pressure vs. no rain (0) / rain (1)") +  
 theme(legend.position="none") +  
 theme\_bw()



########################  
# Decision Tree Models #  
########################  
  
#Izbacivanje datuma kao varijable koje nece biti u modelu i transformacija binarne varijable "raine" u kategorijsku varijablu sa dva nivoa (Yes/No)   
ready\_data <- clean\_data[,-2]  
for (i in 1:1827) {  
 if (ready\_data$rain[i] == 1) {ready\_data$rain[i] <- 'Yes'}  
 if (ready\_data$rain[i] == 0) {ready\_data$rain[i] <- 'No'}  
}  
  
  
  
  
#Podela seta na deo za treniranje (75%) i deo za testiranje (25%) pomocu stratified partitioning  
set.seed(123)  
  
train.indices <- createDataPartition(ready\_data$rain,  
 p = .75,   
 list = FALSE)  
train.data <- ready\_data[train.indices,]  
test.data <- ready\_data[-train.indices,]  
  
#Provera da li su klase jednako zastupljene u train i test setu  
prop.table(table(train.data$rain))

##   
## No Yes   
## 0.4485777 0.5514223

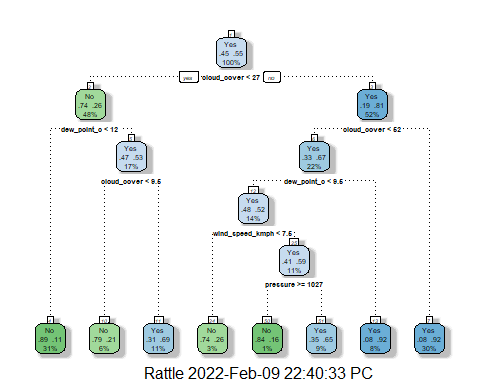
prop.table(table(test.data$rain))

##   
## No Yes   
## 0.4473684 0.5526316

#Pravljenje funkcije evaluaciju performansa drveta  
compute.eval.metrics <- function(cmatrix) {  
 TP <- cmatrix[1,1] # true positive  
 TN <- cmatrix[2,2] # true negative  
 FP <- cmatrix[2,1] # false positive  
 FN <- cmatrix[1,2] # false negative  
 acc <- sum(diag(cmatrix)) / sum(cmatrix)  
 precision <- TP / (TP + FP)  
 recall <- TP / (TP + FN)  
 F1 <- 2\*precision\*recall / (precision + recall)  
 c(accuracy = acc, precision = precision, recall = recall, F1 = F1)  
}   
  
#Pravljenje i testiranje modela tree1 sa niskim / standardnim parametrima kontrole (default: minsplit = 20, cp = 0.01)  
  
tree1 <- rpart(rain ~ ., data = train.data, method = "class")  
print(tree1)

## n= 1371   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 1371 615 Yes (0.44857768 0.55142232)   
## 2) cloud\_cover< 26.5 655 173 No (0.73587786 0.26412214)   
## 4) dew\_point\_c< 11.5 420 48 No (0.88571429 0.11428571) \*  
## 5) dew\_point\_c>=11.5 235 110 Yes (0.46808511 0.53191489)   
## 10) cloud\_cover< 9.5 78 16 No (0.79487179 0.20512821) \*  
## 11) cloud\_cover>=9.5 157 48 Yes (0.30573248 0.69426752) \*  
## 3) cloud\_cover>=26.5 716 133 Yes (0.18575419 0.81424581)   
## 6) cloud\_cover< 51.5 304 101 Yes (0.33223684 0.66776316)   
## 12) dew\_point\_c< 9.5 191 92 Yes (0.48167539 0.51832461)   
## 24) wind\_speed\_kmph< 7.5 42 11 No (0.73809524 0.26190476) \*  
## 25) wind\_speed\_kmph>=7.5 149 61 Yes (0.40939597 0.59060403)   
## 50) pressure>=1026.5 19 3 No (0.84210526 0.15789474) \*  
## 51) pressure< 1026.5 130 45 Yes (0.34615385 0.65384615) \*  
## 13) dew\_point\_c>=9.5 113 9 Yes (0.07964602 0.92035398) \*  
## 7) cloud\_cover>=51.5 412 32 Yes (0.07766990 0.92233010) \*

fancyRpartPlot(tree1)



tree1.pred <- predict(object = tree1, newdata = test.data, type = "class")  
  
tree1.cm <- table(test.data$rain, tree1.pred)  
tree1.cm

## tree1.pred  
## No Yes  
## No 157 47  
## Yes 35 217

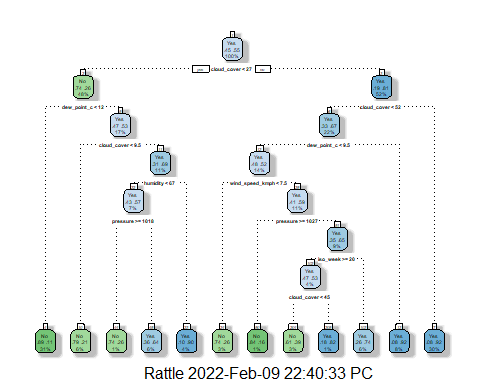
tree1.eval <- compute.eval.metrics(tree1.cm)   
tree1.eval

## accuracy precision recall F1   
## 0.8201754 0.8177083 0.7696078 0.7929293

#Pravljenje i testiranje modela tree2 sa smanjenim parametrima kontrole: minsplit = 10, cp = 0.005   
  
  
tree2 <- rpart(rain ~ ., data = train.data, method = "class", control = rpart.control(minsplit = 10, cp = 0.005))  
print(tree2)

## n= 1371   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 1371 615 Yes (0.44857768 0.55142232)   
## 2) cloud\_cover< 26.5 655 173 No (0.73587786 0.26412214)   
## 4) dew\_point\_c< 11.5 420 48 No (0.88571429 0.11428571) \*  
## 5) dew\_point\_c>=11.5 235 110 Yes (0.46808511 0.53191489)   
## 10) cloud\_cover< 9.5 78 16 No (0.79487179 0.20512821) \*  
## 11) cloud\_cover>=9.5 157 48 Yes (0.30573248 0.69426752)   
## 22) humidity< 66.5 97 42 Yes (0.43298969 0.56701031)   
## 44) pressure>=1017.5 19 5 No (0.73684211 0.26315789) \*  
## 45) pressure< 1017.5 78 28 Yes (0.35897436 0.64102564) \*  
## 23) humidity>=66.5 60 6 Yes (0.10000000 0.90000000) \*  
## 3) cloud\_cover>=26.5 716 133 Yes (0.18575419 0.81424581)   
## 6) cloud\_cover< 51.5 304 101 Yes (0.33223684 0.66776316)   
## 12) dew\_point\_c< 9.5 191 92 Yes (0.48167539 0.51832461)   
## 24) wind\_speed\_kmph< 7.5 42 11 No (0.73809524 0.26190476) \*  
## 25) wind\_speed\_kmph>=7.5 149 61 Yes (0.40939597 0.59060403)   
## 50) pressure>=1026.5 19 3 No (0.84210526 0.15789474) \*  
## 51) pressure< 1026.5 130 45 Yes (0.34615385 0.65384615)   
## 102) iso\_week>=19.5 53 25 Yes (0.47169811 0.52830189)   
## 204) cloud\_cover< 44.5 36 14 No (0.61111111 0.38888889) \*  
## 205) cloud\_cover>=44.5 17 3 Yes (0.17647059 0.82352941) \*  
## 103) iso\_week< 19.5 77 20 Yes (0.25974026 0.74025974) \*  
## 13) dew\_point\_c>=9.5 113 9 Yes (0.07964602 0.92035398) \*  
## 7) cloud\_cover>=51.5 412 32 Yes (0.07766990 0.92233010) \*

fancyRpartPlot(tree2)



tree2.pred <- predict(object = tree2, newdata = test.data, type = "class")  
  
tree2.cm <- table(test.data$rain, tree2.pred)  
tree2.cm

## tree2.pred  
## No Yes  
## No 168 36  
## Yes 45 207

tree2.eval <- compute.eval.metrics(tree2.cm)   
tree2.eval

## accuracy precision recall F1   
## 0.8223684 0.7887324 0.8235294 0.8057554

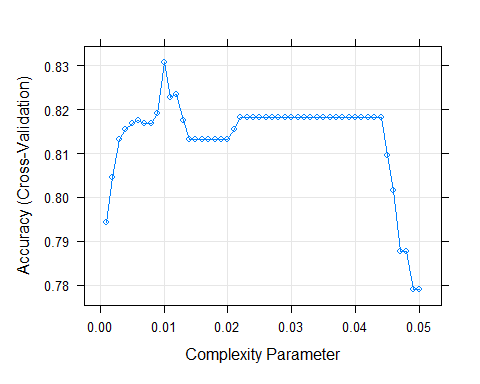
#Pravljenje dejtafrejma za uporedivanje performansa razlicitih modela  
  
model.performance.comparison <- data.frame(rbind(tree1.eval, tree2.eval),  
 row.names = c("tree 1", "tree 2"))  
model.performance.comparison

## accuracy precision recall F1  
## tree 1 0.8201754 0.8177083 0.7696078 0.7929293  
## tree 2 0.8223684 0.7887324 0.8235294 0.8057554

##################################################################################  
  
#Sistematsko trazenje parametra cp naoptimalnijeg za preciznost predvidanja modela (Cross-validation)  
set.seed(123)  
  
numFolds = trainControl(method = "cv", number = 10)  
cpGrid = expand.grid(.cp = seq(0.001, to = 0.05, by = 0.001))   
  
set.seed(1)  
dt.cv <- train(rain ~ .,  
 data = train.data,  
 method = "rpart",  
 control = rpart.control(minsplit = 10),  
 trControl = numFolds,  
 tuneGrid = cpGrid)  
dt.cv

## CART   
##   
## 1371 samples  
## 15 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 1234, 1234, 1234, 1234, 1233, 1234, ...   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.001 0.7943332 0.5842274  
## 0.002 0.8045630 0.6047666  
## 0.003 0.8133009 0.6217165  
## 0.004 0.8154959 0.6264349  
## 0.005 0.8169346 0.6295303  
## 0.006 0.8176327 0.6304046  
## 0.007 0.8168975 0.6288394  
## 0.008 0.8169027 0.6286904  
## 0.009 0.8190714 0.6327309  
## 0.010 0.8307504 0.6551189  
## 0.011 0.8227424 0.6382697  
## 0.012 0.8234670 0.6394158  
## 0.013 0.8176061 0.6268443  
## 0.014 0.8132266 0.6170136  
## 0.015 0.8132266 0.6170136  
## 0.016 0.8132266 0.6170136  
## 0.017 0.8132266 0.6170136  
## 0.018 0.8132266 0.6170136  
## 0.019 0.8132266 0.6170136  
## 0.020 0.8132266 0.6170136  
## 0.021 0.8154164 0.6208065  
## 0.022 0.8183361 0.6262146  
## 0.023 0.8183361 0.6262146  
## 0.024 0.8183361 0.6262146  
## 0.025 0.8183361 0.6262146  
## 0.026 0.8183361 0.6262146  
## 0.027 0.8183361 0.6262146  
## 0.028 0.8183361 0.6262146  
## 0.029 0.8183361 0.6262146  
## 0.030 0.8183361 0.6262146  
## 0.031 0.8183361 0.6262146  
## 0.032 0.8183361 0.6262146  
## 0.033 0.8183361 0.6262146  
## 0.034 0.8183361 0.6262146  
## 0.035 0.8183361 0.6262146  
## 0.036 0.8183361 0.6262146  
## 0.037 0.8183361 0.6262146  
## 0.038 0.8183361 0.6262146  
## 0.039 0.8183361 0.6262146  
## 0.040 0.8183361 0.6262146  
## 0.041 0.8183361 0.6262146  
## 0.042 0.8183361 0.6262146  
## 0.043 0.8183361 0.6262146  
## 0.044 0.8183361 0.6262146  
## 0.045 0.8095769 0.6094168  
## 0.046 0.8016059 0.5943757  
## 0.047 0.7877902 0.5683625  
## 0.048 0.7877902 0.5683625  
## 0.049 0.7790311 0.5522119  
## 0.050 0.7790311 0.5522119  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.01.

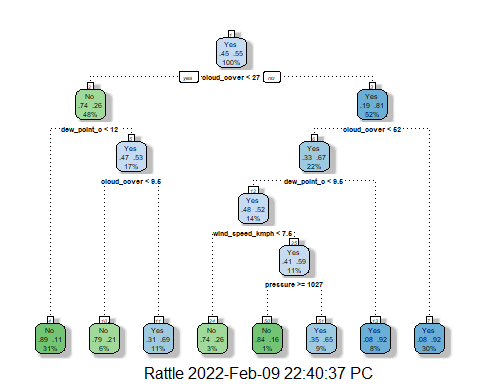
plot(dt.cv)



###################################################################################  
  
#Pravljenje i testiranje modela tree3 sa optimalnom vrednoscu cp = 0.01  
  
tree3 <- prune(tree2, cp = 0.01)  
print(tree3)

## n= 1371   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 1371 615 Yes (0.44857768 0.55142232)   
## 2) cloud\_cover< 26.5 655 173 No (0.73587786 0.26412214)   
## 4) dew\_point\_c< 11.5 420 48 No (0.88571429 0.11428571) \*  
## 5) dew\_point\_c>=11.5 235 110 Yes (0.46808511 0.53191489)   
## 10) cloud\_cover< 9.5 78 16 No (0.79487179 0.20512821) \*  
## 11) cloud\_cover>=9.5 157 48 Yes (0.30573248 0.69426752) \*  
## 3) cloud\_cover>=26.5 716 133 Yes (0.18575419 0.81424581)   
## 6) cloud\_cover< 51.5 304 101 Yes (0.33223684 0.66776316)   
## 12) dew\_point\_c< 9.5 191 92 Yes (0.48167539 0.51832461)   
## 24) wind\_speed\_kmph< 7.5 42 11 No (0.73809524 0.26190476) \*  
## 25) wind\_speed\_kmph>=7.5 149 61 Yes (0.40939597 0.59060403)   
## 50) pressure>=1026.5 19 3 No (0.84210526 0.15789474) \*  
## 51) pressure< 1026.5 130 45 Yes (0.34615385 0.65384615) \*  
## 13) dew\_point\_c>=9.5 113 9 Yes (0.07964602 0.92035398) \*  
## 7) cloud\_cover>=51.5 412 32 Yes (0.07766990 0.92233010) \*

fancyRpartPlot(tree3)



tree3.pred <- predict(object = tree3, newdata = test.data, type = "class")  
  
tree3.cm <- table(test.data$rain, tree3.pred)  
tree3.cm

## tree3.pred  
## No Yes  
## No 157 47  
## Yes 35 217

tree3.eval <- compute.eval.metrics(tree3.cm)   
tree3.eval

## accuracy precision recall F1   
## 0.8201754 0.8177083 0.7696078 0.7929293

###################################################################################  
  
#Pravljenje dejtafrejma za uporedivanje performansa sva 3 modela  
  
model.performance.comparison <- data.frame(rbind(tree1.eval, tree2.eval, tree3.eval),  
 row.names = c("tree 1", "tree 2", "tree 3"))  
model.performance.comparison

## accuracy precision recall F1  
## tree 1 0.8201754 0.8177083 0.7696078 0.7929293  
## tree 2 0.8223684 0.7887324 0.8235294 0.8057554  
## tree 3 0.8201754 0.8177083 0.7696078 0.7929293

#Vizualizacija performansa modela  
  
df\_all <- data.frame()  
  
  
eval\_res <- compute.eval.metrics(tree1.cm)  
  
df\_tree <- data.frame(eval\_res)  
df\_tree$tree <- "tree1"  
df\_tree$metric <- row.names(df\_tree)  
row.names(df\_tree) <- NULL  
names(df\_tree)[1] <- "value"  
  
df\_all <- rbind(df\_all, df\_tree)  
  
eval\_res\_1 <- compute.eval.metrics(tree2.cm)  
  
df\_tree1 <- data.frame(eval\_res\_1)  
df\_tree1$tree <- "tree2"  
df\_tree1$metric <- row.names(df\_tree1)  
row.names(df\_tree1) <- NULL  
names(df\_tree1)[1] <- "value"  
  
  
df\_all <- rbind(df\_all,df\_tree1)  
  
eval\_res\_2 <- compute.eval.metrics(tree3.cm)  
  
df\_tree2 <- data.frame(eval\_res\_2)  
df\_tree2$tree <- "tree3"  
df\_tree2$metric <- row.names(df\_tree2)  
row.names(df\_tree2) <- NULL  
names(df\_tree2)[1] <- "value"  
  
df\_all <- rbind(df\_all,df\_tree2)  
  
  
df\_all$metric <- factor(df\_all$metric, levels = c("accuracy","precision","recall","F1"))  
df\_all$tree <- as.factor(df\_all$tree)  
  
ggplot(df\_all,   
 aes(x = metric, y = value, fill = tree)) +   
 geom\_bar(position = "dodge", stat = "identity") + ylim(0, 1) +  
 theme\_bw()

