ML\_classifier

Tara Doma Lama

2023-04-13

#load the library

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.1.8  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(ggplot2)

# read the file

random <- read\_csv("~/CSB-BIOL425/r/sandbox/arezoo\_file.csv")#never set col\_name = FALSE

## New names:  
## Rows: 90 Columns: 11  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," dbl  
## (11): ...1, max depth, n\_estimators, training accuracy, training precisi...  
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ  
## Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
## • `` -> `...1`

colnames(random) <- c("id" ,"max\_depth", "n\_estimators", "train\_accurarcy", "train\_precise", "train\_recall", "train\_f1", "test\_accuracy", "test\_precise", "test\_recall", "test\_f1")  
str(random)

## spc\_tbl\_ [90 × 11] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ id : num [1:90] 0 1 2 3 4 5 6 7 8 9 ...  
## $ max\_depth : num [1:90] 5 5 5 5 5 5 5 5 5 5 ...  
## $ n\_estimators : num [1:90] 10 31 52 73 94 115 136 157 178 200 ...  
## $ train\_accurarcy: num [1:90] 0.952 0.986 0.993 1 1 ...  
## $ train\_precise : num [1:90] 1 1 1 1 1 1 1 1 1 1 ...  
## $ train\_recall : num [1:90] 0.93 0.982 0.987 1 1 ...  
## $ train\_f1 : num [1:90] 0.963 0.991 0.993 1 1 ...  
## $ test\_accuracy : num [1:90] 0.829 0.857 0.829 0.871 0.871 ...  
## $ test\_precise : num [1:90] 0.929 0.948 0.971 0.971 0.976 ...  
## $ test\_recall : num [1:90] 0.829 0.857 0.829 0.871 0.871 ...  
## $ test\_f1 : num [1:90] 0.866 0.893 0.886 0.909 0.915 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. ...1 = col\_double(),  
## .. `max depth` = col\_double(),  
## .. n\_estimators = col\_double(),  
## .. `training accuracy` = col\_double(),  
## .. `training precision` = col\_double(),  
## .. `training recall` = col\_double(),  
## .. `training f1 score` = col\_double(),  
## .. `test accuracy` = col\_double(),  
## .. `test precision` = col\_double(),  
## .. `test recall` = col\_double(),  
## .. `test f1 score` = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

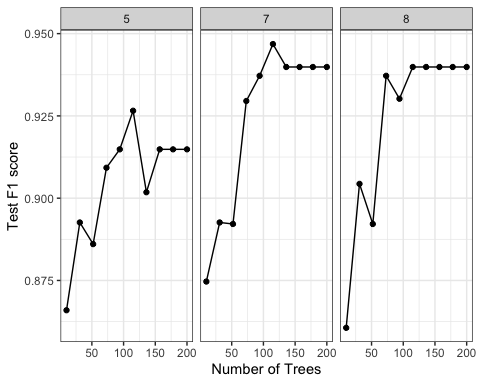
head(random)

## # A tibble: 6 × 11  
## id max\_de…¹ n\_est…² train…³ train…⁴ train…⁵ train…⁶ test\_…⁷ test\_…⁸ test\_…⁹  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0 5 10 0.952 1 0.930 0.963 0.829 0.929 0.829  
## 2 1 5 31 0.986 1 0.982 0.991 0.857 0.948 0.857  
## 3 2 5 52 0.993 1 0.987 0.993 0.829 0.971 0.829  
## 4 3 5 73 1 1 1 1 0.871 0.971 0.871  
## 5 4 5 94 1 1 1 1 0.871 0.976 0.871  
## 6 5 5 115 1 1 1 1 0.886 0.979 0.886  
## # … with 1 more variable: test\_f1 <dbl>, and abbreviated variable names  
## # ¹​max\_depth, ²​n\_estimators, ³​train\_accurarcy, ⁴​train\_precise, ⁵​train\_recall,  
## # ⁶​train\_f1, ⁷​test\_accuracy, ⁸​test\_precise, ⁹​test\_recall

random %>%   
 pull(max\_depth) %>% # for extract  
 table()

## .  
## 5 7 8 10 15 20 25 30 40   
## 10 10 10 10 10 10 10 10 10

random\_2 <- random %>%   
 filter(max\_depth %in% c(5, 7, 8))  
random\_2 %>%   
 ggplot(aes(x = n\_estimators , y = test\_f1, group = max\_depth))+  
 geom\_point()+  
 geom\_line()+  
 facet\_wrap(~max\_depth)+  
 theme\_bw()+  
 xlab("Number of Trees")+  
 ylab("Test F1 score")

 # max - depth is for the decision tree