Machine Learning Methods for Site-specific Input Management

Shunkei Kakimto

library(knitr)  
library(here)

## here() starts at /Users/shunkeikakimoto/OneDrive - University of Nebraska-Lincoln/ML\_VRA\_shun

knitr::opts\_chunk$set(  
 cache = FALSE,  
 warning = FALSE,  
 message = FALSE  
 )  
  
opts\_knit$set(root.dir=here())

# === packages ===#  
library(sf)  
library(data.table)  
library(RColorBrewer)  
library(patchwork)  
library(grf)  
library(magrittr)  
library(ggplot2)  
library(ggthemes)  
library(viridis)  
library(tidyverse)  
library(ggpubr)  
library(flextable)  
library(officer)  
library(modelsummary)  
library(here)  
library(gridExtra)  
library(DiagrammeR)  
library(causalTree)  
library(rpart)  
library(rattle)  
library(webshot2) # remotes::install\_github("rstudio/webshot2")  
# library(sparkline)  
# library(htmltools)  
# library(htmlwidgets)  
# library(equatags)  
# library(equatiomatic)

# Load the data

# source(here("Codes", "functions.R"))  
source(here("Codes", "0\_0\_functions.R"))  
source(here("Codes", "functions\_stepwise\_vs\_base.R"))

# Results

## Source Results and Preparation

# ===================================  
# Forest resutls   
# ===================================  
  
res\_forest\_all\_honesty <- readRDS(here("CNN\_Results", "SimRes\_sp\_400.rds"))%>%  
 setnames("model", "Method")%>%  
 .[, var\_case:=factor(var\_case, levels = c("aby", "abytt", "aabbyy", "aabbyytt"))]%>%  
 .[, Method:=factor(Method, levels = c("RF", "BRF", "CF\_stepwise", "CF\_base"))]%>%  
 .[, Honesty := TRUE]  
  
  
res\_forest\_all\_nohonesty <- readRDS(here("CNN\_Results", "SimRes\_sp\_400\_nohonesty.rds"))%>%  
 setnames("model", "Method")%>%  
 .[, var\_case:=factor(var\_case, levels = c("aby", "abytt", "aabbyy", "aabbyytt"))]%>%  
 .[, Method:=factor(Method, levels = c("RF", "BRF", "CF\_stepwise", "CF\_base"))]%>%  
 .[, Honesty := FALSE]  
  
  
  
CNN\_y\_eval\_v1 <- readRDS(here("CNN\_Results", "CNN\_y\_eval\_v1.rds"))%>%  
 .[,r2\_y\_cell:=NA]%>%  
 .[,Honesty:=NA]%>%  
 .[,.(Method, var\_case, r2\_y\_cell, r2\_y\_agg, Honesty)]  
 # .[,r2\_y\_agg:=round(r2\_y\_agg,digits=3)]  
  
  
res\_CNN\_optN <- readRDS(here("CNN\_Results", "res\_CNN\_r2\_optN.rds"))%>%  
 .[,Method:="CNN"]%>%  
 setnames("r2\_N", "r2\_agg")%>%  
 .[,.(Method, var\_case, r2\_agg)]  
  
# /\*-------------------------------------------------------\*/  
#' ## Organize the data  
# /\*-------------------------------------------------------\*/  
res\_forest\_all <- rbind(res\_forest\_all\_honesty, res\_forest\_all\_nohonesty)  
 # .[,`:=`(  
 # r2\_cell=round(r2\_cell,digits=3),  
 # r2\_agg=round(r2\_agg,digits=3),  
 # r2\_y\_cell=round(r2\_y\_cell,digits=3),  
 # r2\_y\_agg=round(r2\_y\_agg,digits=3)  
 # )]   
  
res\_forest\_yield <- res\_forest\_all[Method%in%c("RF","BRF"), .(Method, var\_case, r2\_y\_cell, r2\_y\_agg, Honesty)]  
  
res\_forest\_optN <- res\_forest\_all[, .(Method, var\_case, r2\_cell, r2\_agg, Honesty)]  
  
res\_yield\_all <- rbind(res\_forest\_yield, CNN\_y\_eval\_v1)

## EONR estimation and Yield Prediction

###=== subplot-level ===###  
res\_y\_subplot <- copy(res\_yield\_all)%>%  
 .[Honesty%in%c(TRUE, NA),.(Method, var\_case, r2\_y\_agg)]  
  
res\_optN\_subplot <- copy(res\_forest\_optN)%>%  
 .[Honesty==TRUE,.(Method, var\_case, r2\_agg)]%>%  
 rbind(.,res\_CNN\_optN)%>%  
 .[, Method:=factor(Method, levels = c("RF", "BRF", "CNN", "CF\_stepwise", "CF\_base"))]%>%  
 .[,var\_case:= case\_when(  
 var\_case == "aby" ~ "Model 1",  
 var\_case == "abytt" ~ "Model 2",  
 var\_case == "aabbyy" ~ "Model 3",  
 var\_case == "aabbyytt" ~ "Model 4"  
 )]  
  
  
##== Summary Table ==##  
table\_y\_subplot\_wide\_prep <- copy(res\_y\_subplot)%>%  
 .[,var\_case:= case\_when(  
 var\_case == "aby" ~ "Model 1",  
 var\_case == "abytt" ~ "Model 2",  
 var\_case == "aabbyy" ~ "Model 3",  
 var\_case == "aabbyytt" ~ "Model 4"  
 )]%>%  
 .[, .(r2\_y\_agg = mean(r2\_y\_agg)), by=.(Method, var\_case)]%>%  
 .[, r2\_y\_agg:= format(round(r2\_y\_agg,3), nsmall=3)]%>%  
 dcast(var\_case~Method, value.var = "r2\_y\_agg")  
  
  
avg\_y <- copy(res\_y\_subplot)%>%  
 .[, .(Avg= mean(r2\_y\_agg)), by=Method]%>%  
 .[, Avg:= format(round(Avg, 3), nsmall = 3)]  
  
  
table\_optN\_subplot\_wide\_prep <- copy(res\_optN\_subplot)%>%  
 .[, .(r2\_agg = mean(r2\_agg)), by=.(Method, var\_case)]%>%  
 .[, r2\_agg:= format(round(r2\_agg, digits=3), nsmall = 3)]  
  
avg\_optN <- copy(res\_optN\_subplot)%>%  
 .[, .(Avg= mean(r2\_agg)), by=Method]%>%  
 .[, Avg:= format(round(Avg, 3), nsmall = 3)]%>%  
 dcast(...~Method, value.var = "Avg")  
  
  
  
table\_optN\_subplot\_wide <- copy(table\_optN\_subplot\_wide\_prep)%>%  
 dcast(var\_case~Method, value.var = "r2\_agg")%>%  
 setnames("var\_case", "Model")%>%  
 mutate(  
 across(  
 everything(),  
 as.character  
 )  
 ) %>%   
 .[,RF := paste0(RF, " (",table\_y\_subplot\_wide\_prep[,RF],")")]%>%  
 .[,BRF := paste0(BRF, " (",table\_y\_subplot\_wide\_prep[,BRF],")")]%>%  
 .[,CNN := paste0(CNN, " (",table\_y\_subplot\_wide\_prep[,CNN],")")]%>%  
 .[,CF\_stepwise := paste0(CF\_stepwise, " ( - )")]%>%  
 .[,CF\_base := paste0(CF\_base, " ( - )")]%>%  
 add\_row(Model = NA, RF = NA, BRF = NA, CF\_stepwise = NA, CF\_base = NA, .after = 4)%>%  
 flextable(.)%>%  
 add\_body(  
 Model="Avg.",  
 RF = paste0(avg\_optN[,RF]," (", avg\_y[Method=="RF",Avg], ")"),  
 BRF = paste0(avg\_optN[,BRF]," (", avg\_y[Method=="BRF",Avg], ")"),  
 CNN = paste0(avg\_optN[,CNN]," (", avg\_y[Method=="CNN",Avg], ")"),  
 CF\_stepwise = paste0(avg\_optN[,CF\_stepwise]," ( - )"),  
 CF\_base = paste0(avg\_optN[,CF\_base]," ( - )"),  
 top = FALSE  
 )%>%  
 theme\_booktabs()%>%  
 set\_header\_labels(values = list(  
 Model = "Model",  
 RF = "RF",  
 BRF = "BRF",  
 CNN = "CNN",  
 CF\_stepwise = "CF-stepwise",  
 CF\_base = "CF-base"  
 ))%>%  
 align(align = "center", part = "all")%>%  
 align(j=1, align = "left", part = "all")%>%  
 # width(width = 2)%>%  
 autofit()%>%  
 add\_footer\_row(  
 values = "The value in ( ) indicates yield prediction performances",  
 colwidths = 6  
 )%>%  
 set\_caption("Table1: Mean R-squared of EONR Estimates and Predicted Yield by ML Methods and Modeling Scenarios")%>%  
 font(fontname="Helvetica", part = "all",  
 # cs.family = fontname,  
 # hansi.family = fontname,  
 # eastasia.family = fontname  
 )

####==== CNN vs RF vs BRF ====####  
res\_CNN\_y <- readRDS(here("CNN\_Results", "CNN\_y\_eval\_v1.rds"))%>%  
 .[,.(sim, Method, var\_case, r2\_y\_agg)]%>%  
 setnames("r2\_y\_agg", "r2\_y")  
 # .[,r2\_y\_agg:=round(r2\_y\_agg,digits=3)]  
  
res\_CNN\_optN <- readRDS(here("CNN\_Results", "res\_CNN\_r2\_optN.rds"))%>%  
 setnames("r2\_N", "r2")  
  
res\_CNN <- res\_CNN\_y[res\_CNN\_optN, on=c("sim", "var\_case")]%>%  
 dcast(sim + var\_case ~ Method, value.var = c("r2", "r2\_y"))%>%  
 .[,var\_case:= case\_when(  
 var\_case == "aby" ~ "Model 1",  
 var\_case == "abytt" ~ "Model 2",  
 var\_case == "aabbyy" ~ "Model 3",  
 var\_case == "aabbyytt" ~ "Model 4"  
 )]  
  
  
  
res\_CNN\_RF\_BRF <- readRDS(here("CNN\_Results", "SimRes\_sp\_400.rds"))%>%  
 .[,!c("r2\_cell", "r2\_y\_cell")]%>%  
 .[model%in%c("RF","BRF"),] %>%  
 .[, var\_case:=factor(var\_case, levels = c("aby", "abytt", "aabbyy", "aabbyytt"))]%>%  
 setnames("model", "Method")%>%  
 setnames(c("r2\_agg", "r2\_y\_agg"), c("r2", "r2\_y"))%>%  
 .[,var\_case:= case\_when(  
 var\_case == "aby" ~ "Model 1",  
 var\_case == "abytt" ~ "Model 2",  
 var\_case == "aabbyy" ~ "Model 3",  
 var\_case == "aabbyytt" ~ "Model 4"  
 )]%>%  
 dcast(sim + var\_case ~ Method, value.var = c("r2", "r2\_y"))%>%  
 .[res\_CNN, on=c("sim", "var\_case")]%>%  
 setnames("var\_case", "Model")%>%  
 .[,.(sim, Model, r2\_RF, r2\_BRF, r2\_CNN, r2\_y\_RF, r2\_y\_BRF, r2\_y\_CNN)]%>%  
 .[,which\_optN :=   
 ifelse(r2\_BRF>r2\_RF & r2\_BRF>r2\_CNN, "BRF",  
 ifelse(r2\_RF>r2\_BRF & r2\_RF>r2\_CNN, "RF",  
 ifelse(r2\_CNN>r2\_BRF & r2\_CNN>r2\_RF, "CNN", NA)  
 )  
 )  
 ]%>%  
 .[,which\_y :=   
 ifelse(r2\_y\_BRF>r2\_y\_RF & r2\_y\_BRF>r2\_y\_CNN, "BRF",  
 ifelse(r2\_y\_RF>r2\_y\_BRF & r2\_y\_RF>r2\_y\_CNN, "RF",  
 ifelse(r2\_y\_CNN>r2\_y\_BRF & r2\_y\_CNN>r2\_y\_RF, "CNN", NA)  
 )  
 )  
 ]%>%  
 .[,index\_consist := ifelse(which\_optN==which\_y, 1, 0)]%>%  
 .[,cons\_y\_optN := ifelse(index\_consist==1, which\_optN, NA)]%>%  
 # RF vs BRF  
 .[,r2\_y\_ratio\_RF\_BRF := r2\_y\_RF/r2\_y\_BRF]%>%  
 .[,r2\_optN\_ratio\_RF\_BRF := r2\_RF/r2\_BRF]%>%  
 # CNN vs RF  
 .[,r2\_y\_ratio\_CNN\_RF := r2\_y\_CNN/r2\_y\_RF]%>%  
 .[,r2\_optN\_ratio\_CNN\_RF := r2\_CNN/r2\_RF]%>%  
 # CNN vs BRF   
 .[,r2\_y\_ratio\_CNN\_BRF := r2\_y\_CNN/r2\_y\_BRF]%>%  
 .[,r2\_optN\_ratio\_CNN\_BRF := r2\_CNN/r2\_BRF]  
  
  
  
# Summary Table   
sumry\_res\_CNN\_RF\_BRF <- res\_CNN\_RF\_BRF%>%  
 .[,.(  
 count\_BRF = nrow(.SD[cons\_y\_optN=="BRF"]),  
 count\_RF = nrow(.SD[cons\_y\_optN=="RF"]),  
 count\_CNN = nrow(.SD[cons\_y\_optN=="CNN"]),  
 count\_y\_BRF = nrow(.SD[which\_y=="BRF"]),  
 count\_y\_RF = nrow(.SD[which\_y=="RF"]),  
 count\_y\_CNN = nrow(.SD[which\_y=="CNN"]),  
 Total= sum(index\_consist)  
 ), by=Model]%>%  
 .[,BRF := paste0(count\_BRF, " (",count\_y\_BRF,")")]%>%  
 .[,RF := paste0(count\_RF, " (",count\_y\_RF,")")]%>%  
 .[,CNN := paste0(count\_CNN, " (",count\_y\_CNN,")")]%>%  
 .[,.(Model, RF, BRF, CNN, Total)]%>%  
 flextable()%>%  
 align(align = "center", part = "all")%>%  
 align(j=1, align = "left", part = "all")%>%  
 autofit()  
  
# save\_as\_image(x = table\_optN\_subplot\_wide, path = here("Writing/OFE Presentation/Res\_y\_cound.pdf"))

# Examination of Treatment Effects

# ======================================================================  
# Examination of te esimation (CF-stepwise vs CF-base)  
# ======================================================================  
  
te\_res <- readRDS(here("CNN\_Results", "res\_eval\_tre\_effects.rds"))%>%  
 .[,.(method, N\_index, R\_squared)]%>%  
 .[, N\_index := case\_when(  
 N\_index == "1-2" ~ "(N1,N2)",  
 N\_index == "2-3" ~ "(N2,N3)",  
 N\_index == "3-4" ~ "(N3,N4)",  
 N\_index == "4-5" ~ "(N4,N5)",  
 N\_index == "1-3" ~ "(N1,N3)",  
 N\_index == "1-4" ~ "(N1,N4)",  
 N\_index == "1-5" ~ "(N1,N5)"  
 )]%>%  
 .[,sim:=rep(1:1000, each=8)]  
  
  
  
# te\_res <- readRDS(here("CNN\_Results", "res\_eval\_tre\_effects\_no\_honesty.rds"))%>%  
# .[,.(method, N\_index, R\_squared)]%>%  
# .[, N\_index := case\_when(  
# N\_index == "1-2" ~ "(N1,N2)",  
# N\_index == "2-3" ~ "(N2,N3)",  
# N\_index == "3-4" ~ "(N3,N4)",  
# N\_index == "4-5" ~ "(N4,N5)",  
# N\_index == "1-3" ~ "(N1,N3)",  
# N\_index == "1-4" ~ "(N1,N4)",  
# N\_index == "1-5" ~ "(N1,N5)"  
# )]%>%  
# .[,sim:=rep(1:1000, each=8)]  
# diff\_2 <- function(x) c(0,diff(x))  
# test <- te\_res[method=="CF-stepwise", ]%>%  
# .[, diff := diff\_2(R\_squared), by="sim"]  
  
# test[, .SD[diff<0,], by=sim]%>%.[,sim]%>%table()  
  
# te\_res[method=="CF-stepwise"&sim==8,]  
  
  
##== Treatment Effects Summary Table ==##  
  
te\_res\_summary <- te\_res[,.(R\_squared=mean(R\_squared)), by=c("method", "N\_index")]%>%  
 .[, R\_squared:= round(R\_squared,3)]%>%  
 flextable(.)%>%  
 set\_header\_labels(  
 values = list(  
 method="Method",  
 N\_index ="N Rates Combinations",  
 R\_squared = "R-squared"  
 ))%>%  
 align(align = "center", part = "all")%>%  
 align(j=1, align = "left", part = "all")%>%  
 # align(align = "center", part = "header")%>%  
 # align\_text\_col(align = "center", header = TRUE, footer = TRUE)%>%  
 autofit()  
  
  
##== Graph ==##  
#== tre calculation ==#  
  
# reg\_data\_all <- readRDS(here("Data/CNN\_Simulations/reg\_data.rds"))  
# test\_agg\_data\_all <- readRDS(here("Data/CNN\_Simulations/test\_agg\_data.rds"))  
  
# # estimate\_tre() is defined in "functions\_stepwise\_vs\_base.R"  
# sim\_ex <- estimate\_tre(x=32, reg\_data\_all=reg\_data\_all, test\_data\_all=test\_agg\_data\_all)  
  
# sim\_ex$r2  
  
# saveRDS(sim\_ex, here("Data","CNN\_writing","tre\_demo.rds"))  
  
sim\_output <- readRDS(here("Data","CNN\_writing","tre\_demo.rds"))  
  
# r2 <-sim\_output$r2  
eval <- sim\_output$output  
  
cf\_step <- eval[Method=="CF-stepwise"]%>%  
 ggplot()+  
 geom\_density(aes(x=value, fill=factor(variable)),alpha=0.8)+  
 facet\_wrap(~N\_index, ncol = 2)+  
 theme\_few()+  
 theme\_few()+  
 theme(  
 legend.position="bottom",  
 plot.title = element\_text(hjust = 0.5),  
 panel.background = element\_rect(fill = "white"),  
 axis.ticks = element\_blank(),  
 legend.title = element\_blank(),  
 strip.text.x = element\_text(size=12,face="bold"),  
 legend.text = element\_text(size=12, face="bold"),  
 )+  
 labs(  
 x = "Treatment Effects(kg/ha)",  
 colour = "Cylinders",  
 shape = "Transmission"  
 )  
  
  
cf\_base <- eval[Method=="CF-base"]%>%  
 ggplot()+  
 geom\_density(aes(x=value, fill=factor(variable)),alpha=0.8)+  
 facet\_wrap(~N\_index, ncol = 2)+  
 theme\_few()+  
 theme\_few()+  
 theme(  
 legend.position="bottom",  
 plot.title = element\_text(hjust = 0.5),  
 panel.background = element\_rect(fill = "white"),  
 axis.ticks = element\_blank(),  
 legend.title = element\_blank(),  
 strip.text.x = element\_text(size=12,face="bold"),  
 legend.text = element\_text(size=12, face="bold"),  
 )+  
 labs(  
 x = "Treatment effects(kg/ha)",  
 colour = "Cylinders",  
 shape = "Transmission"  
 )