Raport\_ML

Adrian Szalkowski Filip Lewczyk

2023-05-09

library(raster)  
library(caret)  
library(mapview)  
library(sf)  
library(CAST)  
library(tmap)  
library(terra)  
library(gridExtra)

# Dane pobrane z przegladarki EO Browser, Sentinel 2 w formacie .tiff.

# Utworzony zostal raster stack i wczytany do programu

main\_data <- stack(“data/raster\_stack.grd”)

# Sprawdźmy jak przedstawiaja sie dane

plot(main\_data) plotRGB(main\_data, stretch =“lin”, r = 3, g = 2, b = 1)

# Należy wczytać warstwe z poligonami, wykonane ręcznie w QGIS,

# które posłużą jako warstwy treningowe

trainSites40 <- read\_sf(“data/trainFields2.gpkg”) print(trainSites40) trainSites20 <- read\_sf(“data/trainFields.gpkg”)

# Wyświetlenie rastra wraz z polami treningowymi 40

viewRGB(main\_data, r = 3, g = 2, b = 1, map.types = “Esri.WorldImagery”)+ mapview(trainSites40)

# Wydobycie wspólnych punktów

extr40 <- extract(main\_data, trainSites40, df=TRUE) extr20 <- extract(main\_data, trainSites20, df=TRUE)

# Połączenie danych

extr40 <- merge(extr40, trainSites40, by.x=“ID”, by.y=“ID”) extr20 <- merge(extr20, trainSites20, by.x=“ID”, by.y=“ID”) head(extr40)

# Ustawienie parametrów modelu z biblioteki CARET

set.seed(100) trainids40 <- createDataPartition(extr40$ID,list=FALSE,p=0.1) trainDat40 <- extr40[trainids40,]

trainids20 <- createDataPartition(extr20$ID,list=FALSE,p=0.1) trainDat20 <- extr20[trainids20,]

# Metoda cv czyli cross walidacja, słuzy do lepszego przewidywania zmian,

# zamiast na wszystkich zmiennych, model uczy się jedynie na polu treningowym

# Zapewnia większą pewność, by model lepiej pracował.

# Przewidywanymi polami, na których będą występować wyniki, zostaną zapisane do

# konkretnego kanału

predictors <- names(main\_data) response <- “typ” indices40 <- CreateSpacetimeFolds(trainDat40,spacevar = “ID”,k=3,class = “typ”) #CAST ctrl40 <- trainControl(method=“cv”, index = indices40$index, savePredictions = TRUE)

indices20 <- CreateSpacetimeFolds(trainDat20,spacevar = “ID”,k=3,class=“typ”) #CAST ctrl20 <- trainControl(method=“cv”, index = indices20$index, savePredictions = TRUE)

# Trenowanie modelu w oparciu o ffs z dodanym kontrolerem

set.seed(100) model40 <- ffs(trainDat40[,predictors], trainDat40[,response], method=“rf”, metric=“Kappa”, trControl=ctrl40, importance=TRUE, ntree=60)

print(model40) plot(model40) plot(varImp(model40))

set.seed(100) model20 <- ffs(trainDat20[,predictors], trainDat20[,response], method=“rf”, metric=“Kappa”, trControl=ctrl20, importance=TRUE, ntree=60)

print(model20) plot(model20) plot(varImp(model20))

# Próba przewidzenia

cvPredictions40 <- model40predbestTune$mtry,]

# Tabela z ilością obiektów

table(cvPredictions40obs)

cvPredictions20 <- model20predbestTunepred,cvPredictions20$obs)

prediction40 <- predict(main\_data,model40) prediction20 <- predict(main\_data,model20) cols <- rev(c(“palegreen”, “grey”, “blue”, “forestgreen”, “brown”,“beige”,“yellowgreen”))

# Kompozycja mapowa

tm\_shape(prediction40) + tm\_raster(palette = cols,title = “Legenda\_40”)+ tm\_scale\_bar(bg.color=“white”,bg.alpha=0.75)+ tm\_layout(legend.bg.color = “white”, legend.bg.alpha = 0.75)

tm\_shape(prediction20) + tm\_raster(palette = cols,title = “Legenda\_20”)+ tm\_scale\_bar(bg.color=“white”,bg.alpha=0.75)+ tm\_layout(legend.bg.color = “white”, legend.bg.alpha = 0.75)

# Model AOA

#model20 <- train(trainDat20[,predictors],trainDat20[,response]) #AOA <- aoa(main\_data,model20)

# Sprawdzenie wyników

#plot(AOA) #plot(AOA$AOA) #plot(AOA$DI) #plot(AOA$parameters)

# Przygotowanie wizualizacji mapowej aoa

#predplotaoa <- spplot((AOA$AOA),col.regions=c(“yellowgreen”, “beige”), main = list(label=“Prediction AOA\_20”))

# Przygotowanie wizualizacji mapowej

predplot40 <- spplot(deratify(prediction40),col.regions=cols, main = list(label=“Prediction 40”)) predplot20 <- spplot(deratify(prediction20),col.regions=cols, main = list(label=“Prediction 20”))

# Wyswietlenie mapy, w zakładce plot można będzie się szybciej przemieszczać

plotRGB(main\_data, stretch =“lin”, r = 3, g = 2, b = 1)

# Wyświetlenie wyników

grid.arrange(predplot20 ,predplot40, ncol = 2)