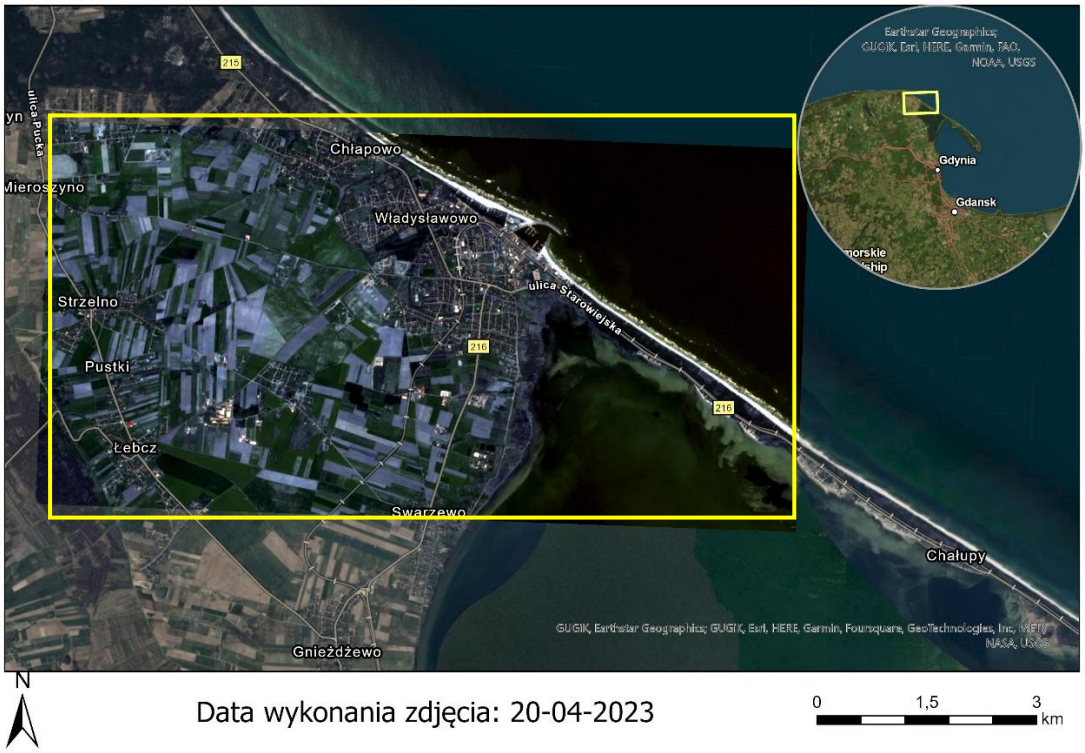
Wykonanie:
Adrian Szałkowski
Filip Lewczyk
https://github.com/Szauek/Land_cover_classification_with_ML

Praca zaliczeniowa pt: „Land Cover classification with ML in R”

Projekt został wykonany w oparciu o *Machine Learning*, z wykorzystaniem metody nadzorowanej poprzez tworzenie pól treningowych. Projekt ma za zadanie sklasyfikować teren pod względem pokrycia powierzchni.

Obszar badania to okolice Władysławowa nad Morzem Bałtyckim. Wyznaczona strefa pozwoli na sklasyfikowanie nie tylko terenów leśnych, uprawnych, bądź miejskich, ale również terenów piaszczystych oraz nadmorskich.

Badany obszar



Zdjęcia zostały pozyskane z portalu: https://www.sentinel-hub.com/explore/eobrowser/

W celu zbadania rastra pod względem pokrycia terenu, użyto następujących kanałów:

- Band 02
- Band 03
- Band 04
- Band 05
- Band 06
- Band 07
- Band 08
- Band 09
- Band 8A

```r
# Wczytanie plików obrazów jako RasterLayer
data = raster("raster_stack.grd")
class(data)
plot(data)

# Utworzenie RasterStack z dwóch RasterLayer
raster_stack <- stack(list.files(pattern = "\.tiff$"))
print(raster_stack)
plot(raster_stack)

writeRaster(raster_stack, "raster_stack.grd", format = "raster")
```

Złączenie .tiffów w raster stack o formacie .grd

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

Biblioteki użyte w projekcie

W projekcie została wykonana analiza porównawcza w oparciu o dwa zestawy pól treningowych:
z 20-ma i 40-ma obiektami.
Zostały one sklasyfikowane na poniższe typy (w warstwie z 40 obiektami):

- Woda 2x
- Płytka woda 4x
- Zabudowa 7x
- Zasiane pole 6x
- Pole 9x
- Las 5x
- Piasek 7x

```r
trainSites40 <- read_sf("data/trainFields2.gpkg")
print(trainSites40)

trainSites20 <- read_sf("data/trainFields.gpkg")
print(trainSites20)

viewRGB(main_data, r = 3, g = 2, b = 1, map.types = "Esri.WorldImagery")+
  mapview(trainSites40)
```

Wczytanie oraz wyświetlenie danych rastrowych, pól treningowych oraz podkładu mapowego.

```
table(cvPredictions40$pred,cvPredictions40$obs)

               las piasek płytka woda pole woda zabudowa zasiane pole
  las          551      0           0    0    0       192           34
  piasek         0    452           0   30    0        14            0
  płytka woda    2      0        1302    0    0        32            0
  pole           0      0           0 3810    0        41            0
  woda           0      0           7    0 2890         0           16
  zabudowa      22     40           0   70    0       686            1
  zasiane pole   0      0           0    0    0        13         1382
```

Tabela z 40 polami treningowymi

```
table(cvPredictions20$pred,cvPredictions20$obs)

               las piasek płytka woda pole woda zabudowa zasiane pole
  las           46      0           0    0    0         0            0
  piasek         0    124           0    6    0         0            0
  płytka woda    0      0         160    0    0         0            0
  pole           0      0           0  105    0         0            0
  woda           0      0           0    0 2847         0            0
  zabudowa       0      1           0   56    0         0            0
  zasiane pole   0      0           0    0    0         0          287
```

Tabela z 20 polami treningowymi

Funkcje *extract* oraz *merge* umożliwiają wydobycie wspólnych mianowników oraz połączenie ze sobą danych z rastra oraz pól treningowych.

Następnym krokiem było przygotowanie zmiennych odpowiedzialnych za działanie naszego nauczania maszynowego.

```
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)
```

Zmienna *predictors* otrzymuje nazwy kolumn z tabeli rastra (tj.: Band 02, Band 03…), w celu utworzenia zmiennych objaśniających w procesie uczenia modelu. Następną zmienną jest *response* będąca zmienną celu w trakcie uczenia modelu. Zmienna *indices40* opiera się na funkcji *CreateSpacetimeFolds*. Służy ona do podziału danych przestrzennych i czasowych na zbiory uczące i testowe. Zmienna *ctrl40* kontroluje proces uczenia modelu w oparciu o metode *cross validation* (*cv*).

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

Powyższy kod trenuje z wykorzystaniem modelu *Random Forest* na danych *trainDat40* z użyciem metody *ffs*, która służy do selekcji zmiennych. Metoda ta wykorzystuje zbiór zmiennych niezależnych (cech) *trainDat40[,predictors]* oraz zmienną zależną *trainDat40[,response]*. Ocenę jakości modelu przeprowadzono za pomocą metryki *Kappa*. Ustawiono również parametry kontroli procesu uczenia modelu oraz określoną ilość powtórzeń w modelu RF. Funkcja *set.seed(100)* zapewnia deterministyczne wyniki modelu.

```
> print(model20)
Random Forest

4321 samples
   3 predictor
   7 classes: 'las', 'piasek', 'płytka woda', 'pole', 'woda', 'zabudowa', 'zasiane pole'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1638, 3372, 3632
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.9668119  0.9145658
  3     0.9877647  0.9496911

Kappa was used to select the optimal model using the largest value.
The final value used for the model was mtry = 3.
```

```
> print(model40)
Random Forest

6743 samples
   4 predictor
   7 classes: 'las', 'piasek', 'płytka woda', 'pole', 'woda', 'zabudowa', 'zasiane pole'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 3090, 4589, 5807
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.9800866  0.9709890
  3     0.9779768  0.9676957
  4     0.9733528  0.9599354

Kappa was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

Powyżej zostały przedstawione wyniki działania dwóch modeli dla 20 oraz 40 pól treningowych. Główną różnicą jest liczba próbek wykorzystanych w obydwu badaniach. Skuteczność modeli jest bardzo wysoka, zbliżona do wartości 1. Współczynnik *Kappa* dla *model20* ma średnią ~0,93 , natomiast dla *model40* ~0,96. Dla modelu z 20 polami najlepsza próba została uzyskana za trzecim razem, a dla modelu z 40 polami za drugą próbą.

Jedna z pierwszych prób działania modelu na 20 polach treningowych.

Mapy porównawcze dwóch modeli, prezentują się następująco:

Prediction 20

Prediction 40

Jak można zauważyć, przy użyciu 40 pól treningowych wyniki są lepsze. Obraz zawiera mniej błędów, model działa zaskakująco dobrze. Na obrazie z 20 pól zostały zaznaczone najbardziej rzucające się w oczy błędy. Warto również porównać wynik mapy po jednej z pierwszych prób, z próbą ostateczną. Widać efekty uczenia się modelu, który lepiej klasyfikuje pokrycie terenu.