BSS

BIAŁOWIESKA SZKOŁA STATYSTYKI

Machine learning

ostateczne rozwiązania

Uczenie maszynowe

Duża liczba obserwacji (>50)

Duża liczba zmiennych (ale p<n zwykle)

Zwykle interesuje nas wynik a nie mechanizm*

ale zwykle można poznać oba

rozpoznawanie twarzy

wyszukiwanie obrazów

ocena ryzyka kredytowego

ocena opłacalności działań windykacyjnych

Przykłady zastosowań

Contents lists available at ScienceDirect

Urban Forestry & Urban Greening

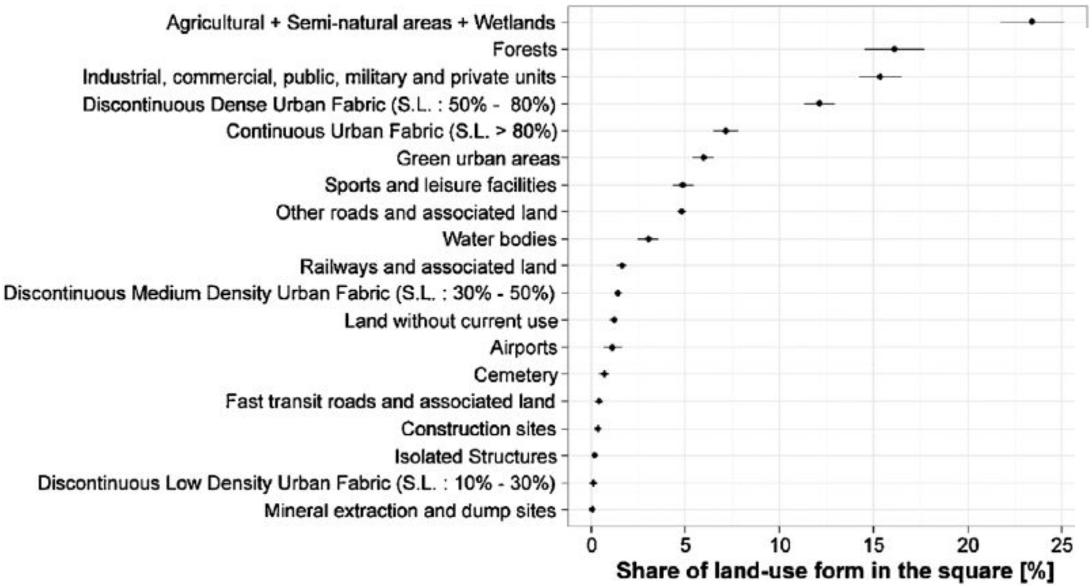
journal homepage: www.elsevier.com/locate/ufug

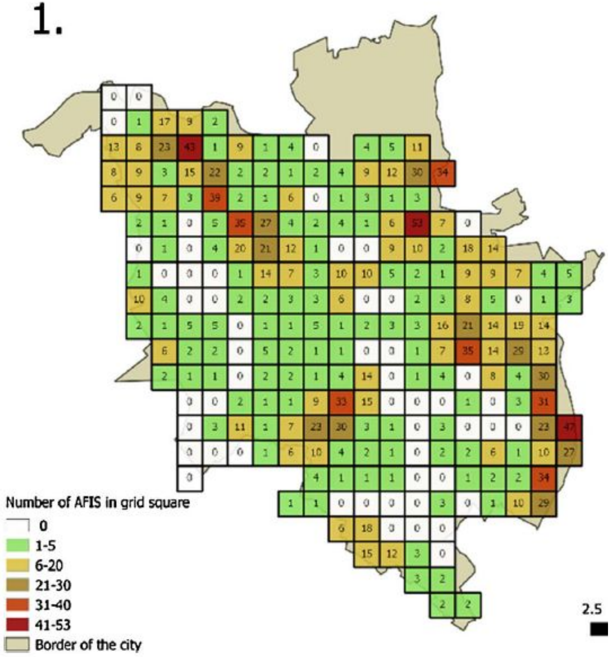
Original article

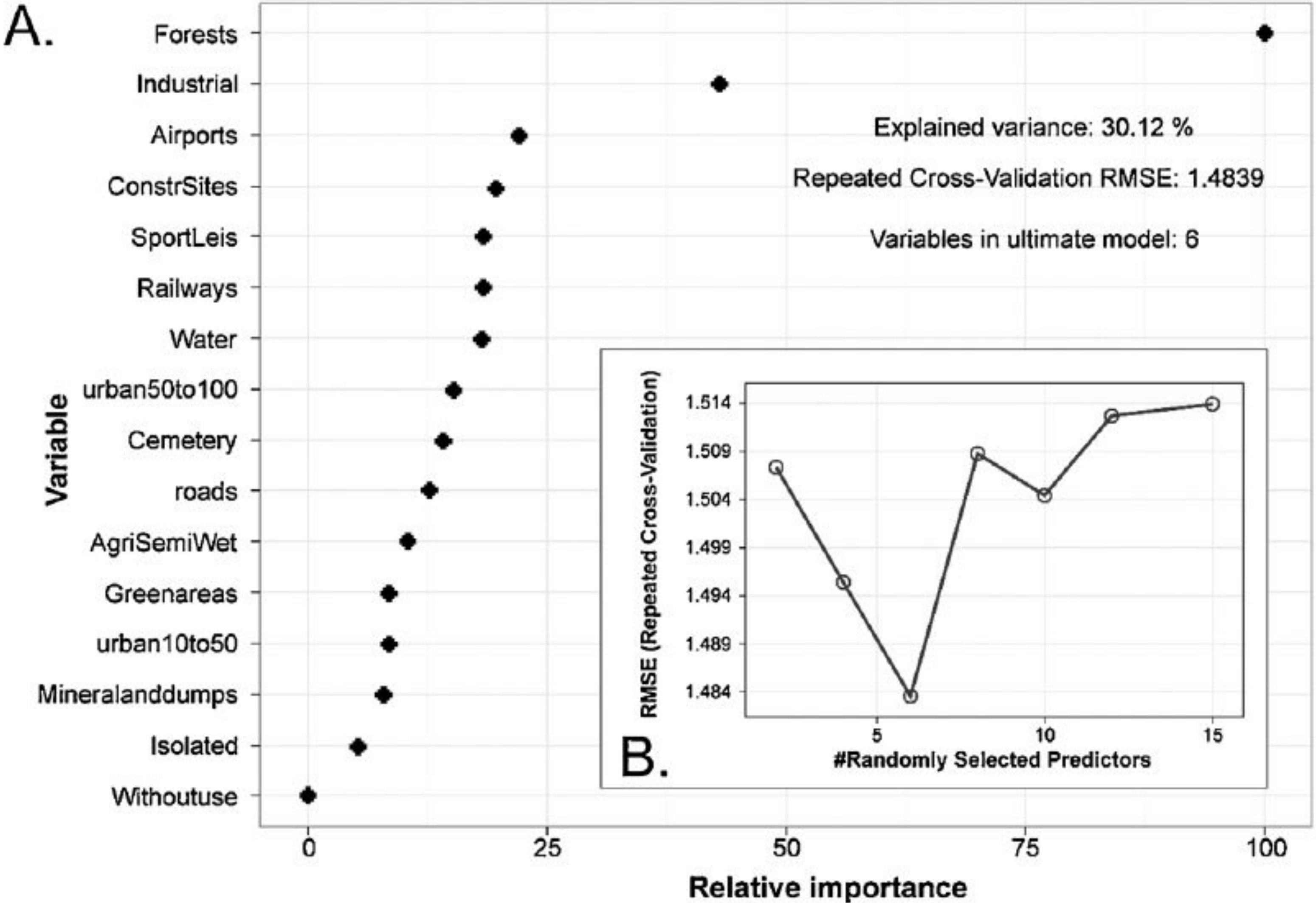
The utility of ancient forest indicator species in urban environments: A case study from Poznań, Poland

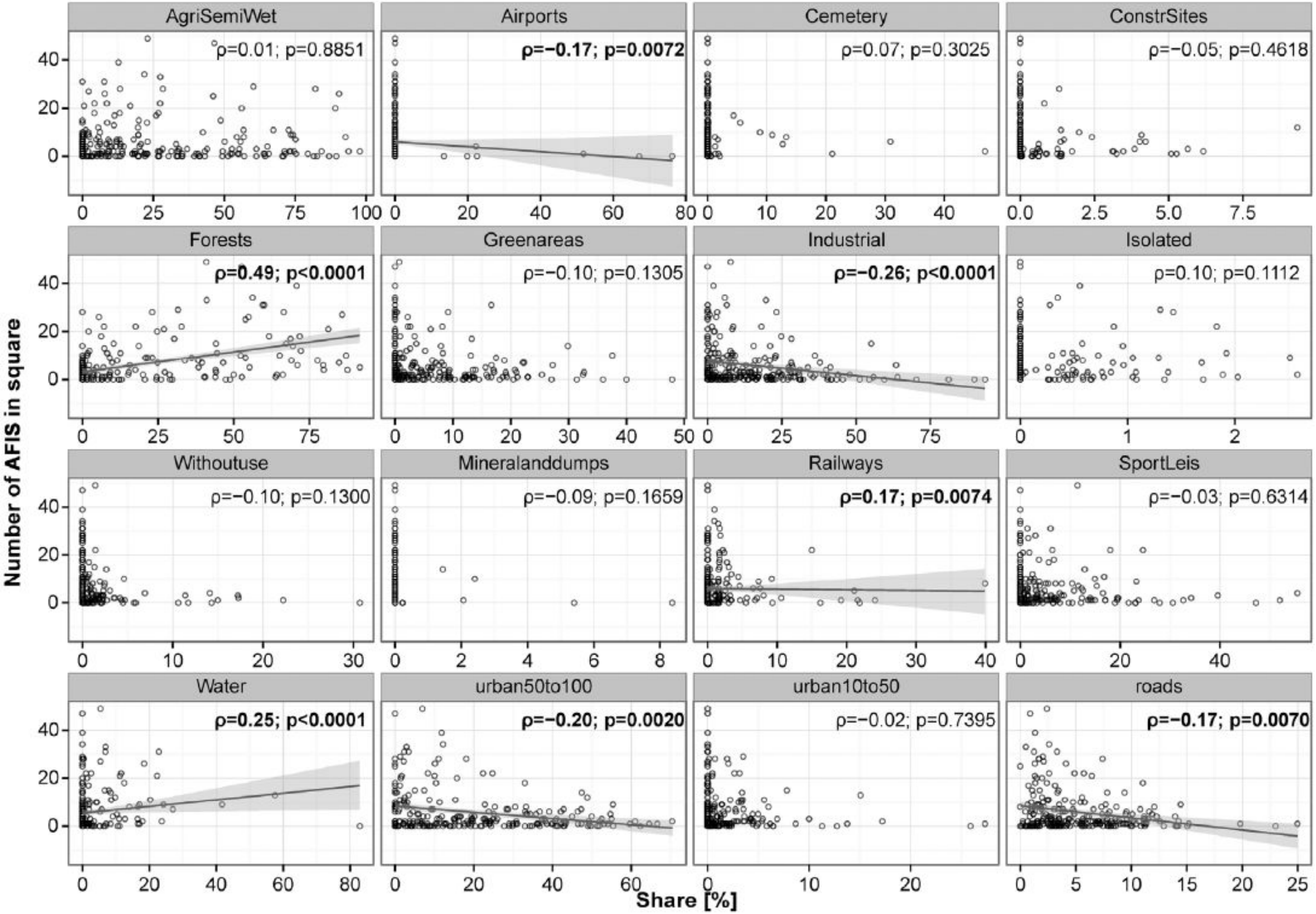
Marcin K. Dyderski[a,b], Jarosław Tyborski[c], Andrzej M. Jagodziński[a,b,*]

[a] Institute of Dendrology, Polish Academy of Sciences, Parkowa 5, 62-035 Kórnik, Poland
[b] Poznań University of Life Sciences, Department of Game Management and Forest Protection, Wojska Polskiego 71c, 60-625 Poznań, Poland
[c] Poznań University of Life Sciences, Faculty of Forestry, Wojska Polskiego 28, 60-637 Poznań, Poland


Agricultural + Semi-natural areas + Wetlands
Forests
Industrial, commercial, public, military and private units
Discontinuous Dense Urban Fabric (S.L. : 50% - 80%)
Continuous Urban Fabric (S.L. > 80%)
Green urban areas
Sports and leisure facilities
Other roads and associated land
Water bodies
Railways and associated land
Discontinuous Medium Density Urban Fabric (S.L. : 30% - 50%)
Land without current use
Airports
Cemetery
Fast transit roads and associated land
Construction sites
Isolated Structures
Discontinuous Low Density Urban Fabric (S.L. : 10% - 30%)
Mineral extraction and dump sites
0
5
10
15
20
25
Share of land-use form in the square [%]


1.
Number of AFIS in grid square
0
1-5
6-20
21-30
31-40
41-53
Border of the city
2.5


A.
Forests
Industrial
Airports
ConstrSites
SportLeis
Railways
Water
urban50to100
Cemetery
roads
AgriSemiWet
Greenareas
urban10to50
Mineralanddumps
Isolated
Withoutuse
Variable
0
25
50
75
100
Relative importance
Explained variance: 30.12 %
Repeated Cross-Validation RMSE: 1.4839
Variables in ultimate model: 6
B.
RMSE (Repeated Cross-Validation)
1.514
1.509
1.504
1.499
1.494
1.489
1.484
5
10
15
#Randomly Selected Predictors

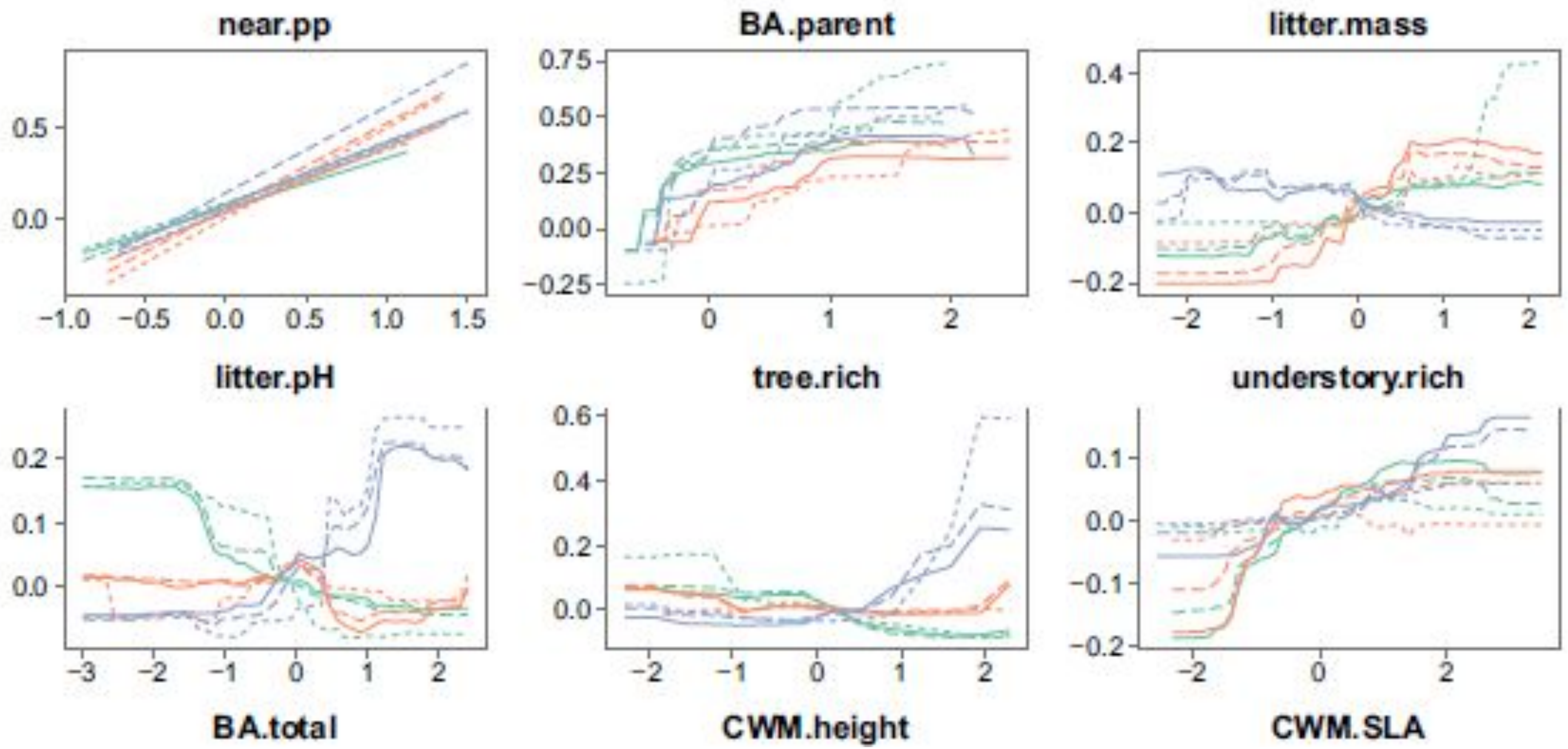

AgriSemiWet
ρ=0.01; p=0.8851
Airports
ρ=−0.17; p=0.0072
Cemetery
ρ=0.07; p=0.3025
ConstrSites
ρ=−0.05; p=0.4618
Forests
ρ=0.49; p<0.0001
Greenareas
ρ=−0.10; p=0.1305
Industrial
ρ=−0.26; p<0.0001
Isolated
ρ=0.10; p=0.1112
Withoutuse
ρ=−0.10; p=0.1300
Mineralanddumps
ρ=−0.09; p=0.1659
Railways
ρ=0.17; p=0.0074
SportLeis
ρ=−0.03; p=0.6314
Water
ρ=0.25; p<0.0001
urban50to100
ρ=−0.20; p=0.0020
urban10to50
ρ=−0.02; p=0.7395
roads
ρ=−0.17; p=0.0070
Number of AFIS in square
Share [%]

Biol Invasions
https://doi.org/10.1007/s10530-018-1706-3

ORIGINAL PAPER

Drivers of invasive tree and shrub natural regeneration in temperate forests

Marcin K. Dyderski · Andrzej M. Jagodziński


near.pp
BA.parent
litter.mass
litter.pH
tree.rich
understory.rich
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0
1.5
0.75
0.50
0.25
0.00
−0.25
0
1
2
0.4
0.2
0.0
−0.2
−2
−1
0
1
2
0.2
0.1
0.0
−3
−2
−1
0
1
2
0.6
0.4
0.2
0.0
−2
−1
0
1
2
0.1
0.0
−0.1
−0.2
−2
0
2
BA.total
CWM.height
CWM.SLA

plant biology

Plant Biology ISSN 1435-8603

RESEARCH PAPER

Do understorey or overstorey traits drive tree encroachment on a drained raised bog?

A. M. Jagodziński[1,2], P. Horodecki[1], K. Rawlik[1] & M. K. Dyderski[1,2]

1 Institute of Dendrology, Polish Academy of Sciences, Kórnik, Poland

2 Faculty of Forestry, Department of Game Management and Forest Protection, Poznań University of Life Sciences, Poznań, Poland

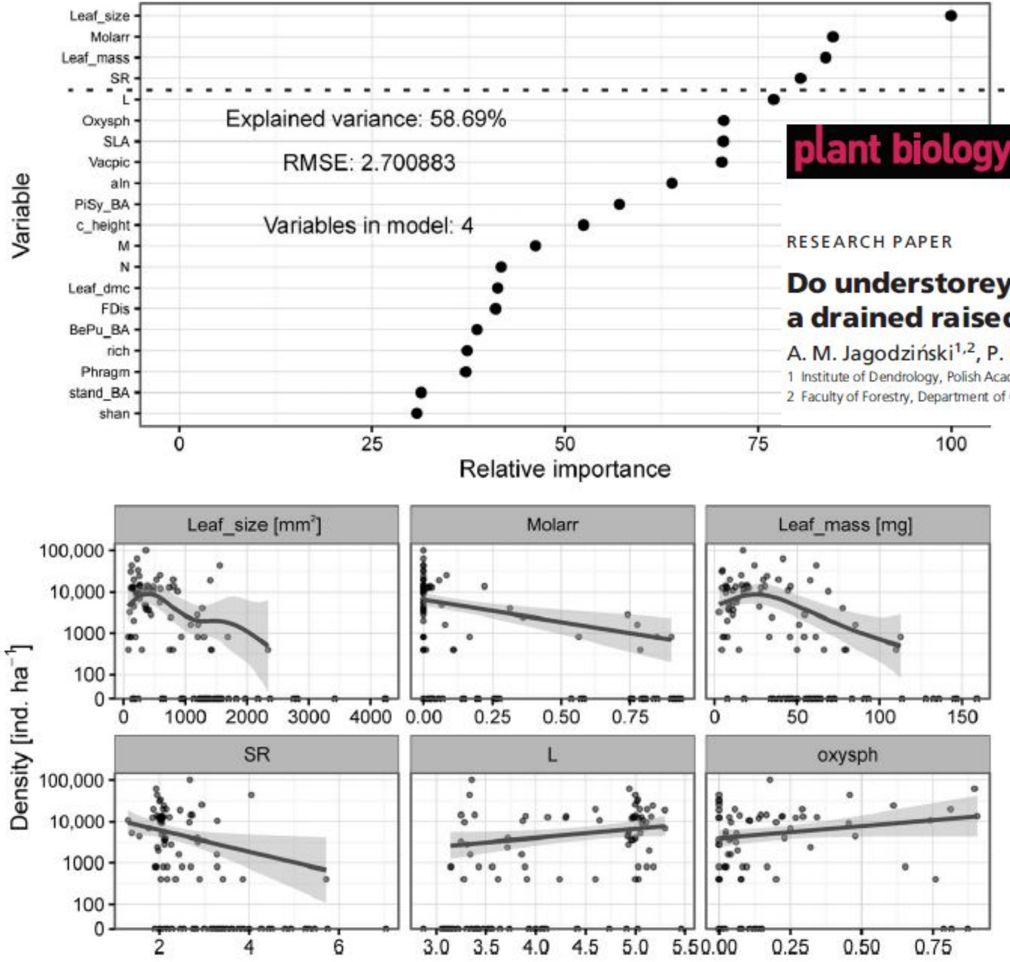


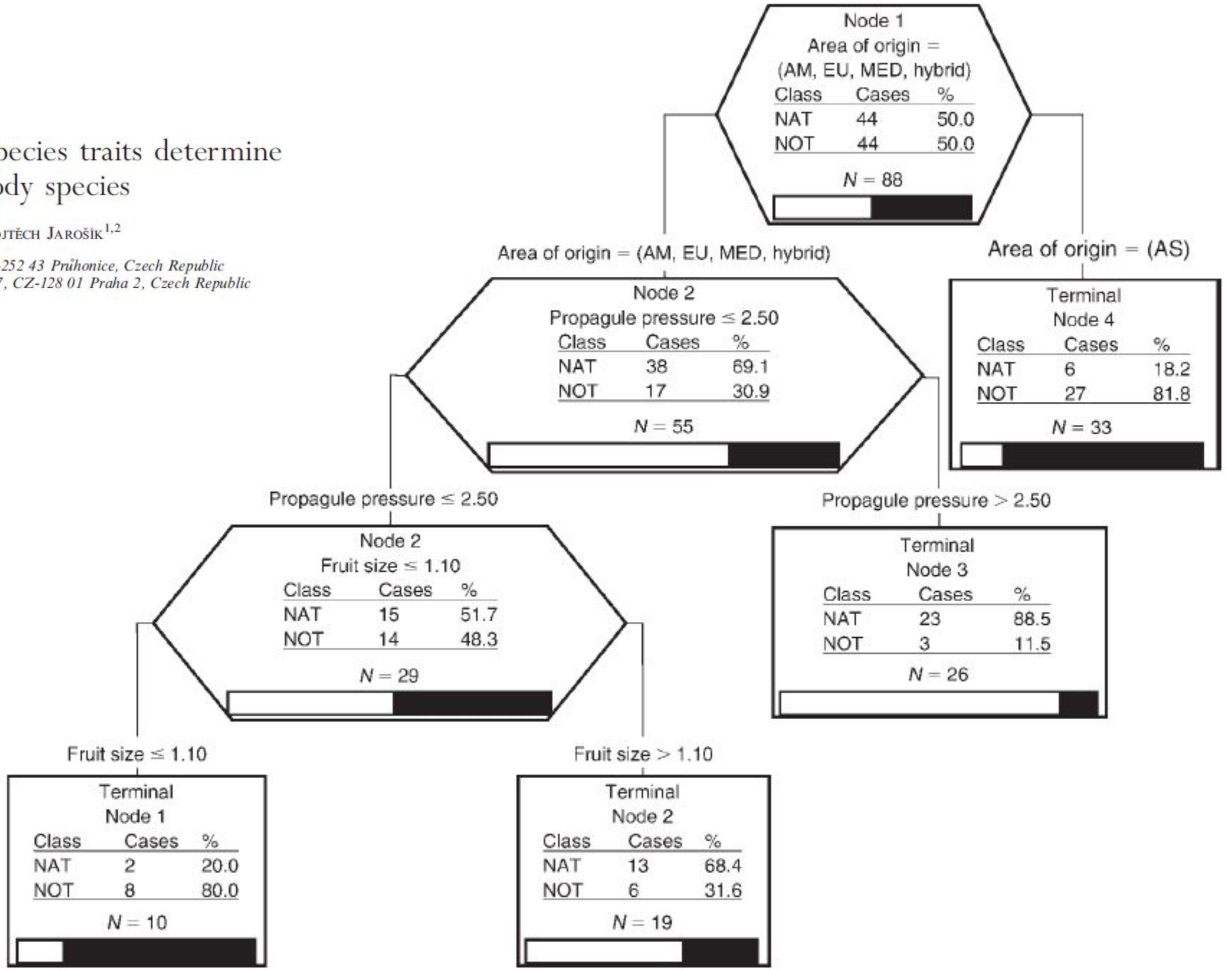
Fig. 4. Results of a random forest model (n = 104) for natural regeneration density of *Pinus sylvestris* ≥1 year old (up to 0.5-m height). Above: relative importance of parameters taken into account during model building; below: partial dependence plots of the six most important variables. Abbreviations: see Table S2. Dashed line cuts variables included in the final model.

Ecology, 90(10), 2009, pp. 2734–2744

Planting intensity, residence time, and species traits determine invasion success of alien woody species

Petr Pyšek,[1,2,3] Martin Křivánek,[1,4] and Vojtěch Jarošík[1,2]

[1]*Institute of Botany, Academy of Sciences of the Czech Republic, CZ-252 43 Průhonice, Czech Republic*
[2]*Department of Ecology, Faculty of Science, Charles University, Viničná 7, CZ-128 01 Praha 2, Czech Republic*



Klasyfikacja a regresja

charakter zmiennej - ciągła/dyskretna

metody mogą być dopasowane do klasyfikacji, regresji lub obu

lista modeli do wytrenowania za pomocą caret::train()

http://topepo.github.io/caret/train-models-by-tag.html

library(caret)

Przykład zoologiczny - klasyfikacja

Wildlife Biology 21(5):254-262. 2015

https://doi.org/10.2981/wlb.00105

A morphometric modeling approach to distinguishing among bobcat, coyote and gray fox scats

http://www.bioone.org/doi/full/10.2981/wlb.00105

dane z Appendixu

Figure A2. Examples of scats coded as 'flat' because they lack other distinctive and/or measurable morphological traits, including samples: (A) 012712ANNU22, (B) 012712ANNU21, (C) 012712ANNU23 and (D) 012712ANNU20.

```
> View(Scat)
> summary(Scat)
     Species        Location    Age     Number     Length      
 bobcat   :50   edge     :34   1:13    1:17    Min.   : 3.100  
 coyote   :24   middle   :34   2: 4    2:30    1st Qu.: 6.500  
 gray fox:17   off edge:23   3:32    3:20    Median : 9.000  
                               4:16    4:13    Mean   : 9.553  
                               5:26    5: 6    3rd Qu.:11.750  
                                       6: 3    Max.   :20.500  
                                       7: 2                    
    Diameter         Taper             TI              Mass       
 Min.   : 7.80   Min.   : 2.30   Min.   :0.230   Min.   : 0.940  
 1st Qu.:15.95   1st Qu.:16.90   1st Qu.:0.990   1st Qu.: 5.715  
 Median :18.00   Median :25.80   Median :1.430   Median : 9.750  
 Mean   :18.36   Mean   :27.47   Mean   :1.606   Mean   :12.508  
 3rd Qu.:21.25   3rd Qu.:37.50   3rd Qu.:1.940   3rd Qu.:16.970  
 Max.   :30.00   Max.   :91.50   Max.   :8.680   Max.   :53.700  

       CN         ropey   segmented flat   scrape
 Min.   : 4.50   0:35   0:36      0:91   0:86  
 1st Qu.: 6.15   1:56   1:55      1: 0   1: 5  
 Median : 7.00                                 
 Mean   : 7.83                                 
 3rd Qu.: 8.10                                 
 Max.   :23.60                                 

>
```

preProcessing

pprec<-preProcess(dane, method=c(...))

methods: center, scale, BoxCox, YeoJohnson, …

można podać kilka

rekomendowane:

center + scale

center + scale + YeoJohnson

http://topepo.github.io/caret/pre-processing.html

```
Scat.oryg<-Scat #kopia danych oryginalnych
prec<-preProcess(Scat, method=c(‘center’,’scale’))
Scat<-predict(prec,Scat)
```

Jak przygotować dane?

zbiór treningowy i testowy

reguła podziału - brak 60, 66, 75, 80, 90%

im więcej do testów - tym wiarygodniejsza ocena niezależności

podział - wg zmiennej objaśnianej lub wg predyktorów

http://topepo.github.io/caret/data-splitting.html

Jak przygotować dane?

wg zmiennej objaśnianej

set.seed(1111) #ziarno generatora liczb pseudolosowych - powtarzalność!

podzielscat<-createDataPartition(Scat$Species,p=.75,list=FALSE)

scat.train<-Scat[podzielscat,]

scat.train<-Scat[-podzielscat,]

http://topepo.github.io/caret/data-splitting.html

```
> summary(scat.train$Species)
  bobcat   coyote gray fox
      38       18       13
> summary(scat.test$Species)
  bobcat   coyote gray fox
      12        6        4
```

Drzewa decyzyjne

```
library(rpart)
```

```
set.seed(11)
```

```
rtr<-train(scat.train[,2:14],scat.train$Species,method = 'ctree',metric='Accuracy')
```

train - klucz do data mining

składnia: predyktory, zmienna zależna, metoda, metric - wg czego ma dobrać najlepszy model

Accuracy, Kappa

Accuracy - trafność klasyfikatora

maleje z liczbą klas, nie zawsze daje dobry obraz

punkt odniesienia - null model - no-information rate

jako no-information rate - proporcja najliczniejszej kategorii

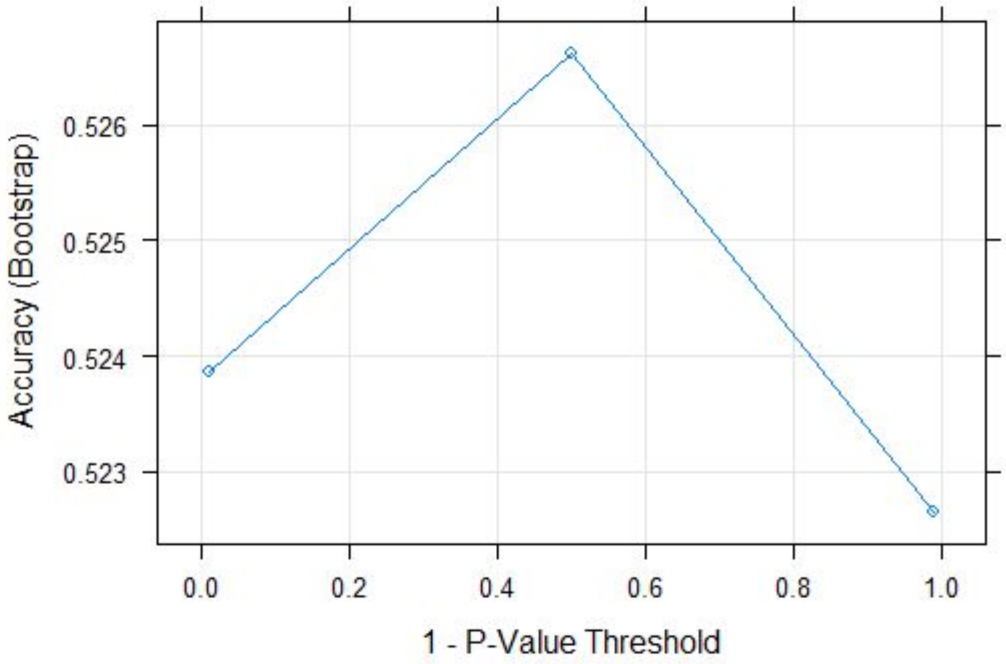
-co by było gdybym zawsze przewidywał najczęstszy przypadek

Kappa Cohena - współczynnik zbieżności, zakres -1 do 1

w praktyce 0 = losowość, <0 się nie powinno zdarzyć

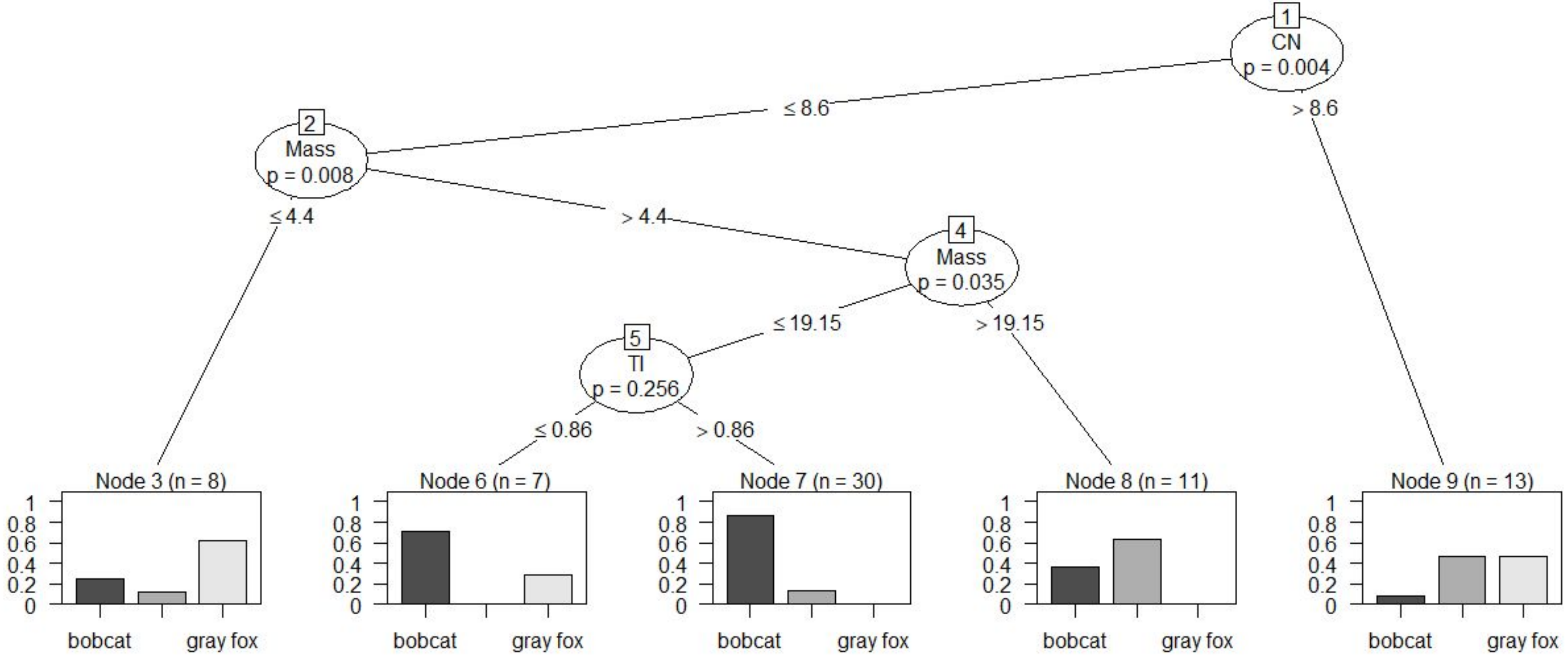
```
> rtr
Conditional Inference Tree
69 samples
13 predictors
 3 classes: 'bobcat', 'coyote', 'gray fox'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 69, 69, 69, 69, 69, 69, ...
Resampling results across tuning parameters:

  mincriterion  Accuracy   Kappa
  0.01          0.5238610  0.1919810
  0.50          0.5266196  0.2005055
  0.99          0.5226484  0.1550557

Accuracy was used to select the optimal model using the
 largest value.
The final value used for the model was mincriterion = 0.5.
```

plot(rtr)



Accuracy (Bootstrap)
0.526
0.525
0.524
0.523
0.0
0.2
0.4
0.6
0.8
1.0
1 - P-Value Threshold

plot(rtr$finalModel)



Sprawdzam!

```
confusionMatrix(predict(rtr,scat.test),scat.test$Species)
Confusion Matrix and Statistics

          Reference
Prediction bobcat coyote gray fox
  bobcat        8      0        1
  coyote        2      6        2
  gray fox      2      0        1
```

Overall Statistics

Accuracy **: 0.6818**

95% CI : (0.4513, 0.8614)

No Information Rate : 0.5455

P-Value [Acc > NIR] : 0.1419

Kappa : 0.4934

Mcnemar's Test P-Value : 0.2276

	Class: bobcat	Class: coyote	Class: gray fox
Sensitivity	0.6667	1.0000	0.25000
Specificity	0.9000	0.7500	0.88889
Pos Pred Value	0.8889	0.6000	0.33333
Neg Pred Value	0.6923	1.0000	0.84211
Prevalence	0.5455	0.2727	0.18182
Detection Rate	0.3636	0.2727	0.04545
Detection Prevalence	0.4091	0.4545	0.13636
Balanced Accuracy	0.7833	0.8750	0.56944

?confusionMatrix

Lasy losowe - random forest

jeśli jedno drzewo nie jest stabilne, to weźmy cały las…

drzewa “głosują” - klasyfikują poszczególne obserwacje

następnie głosy są uśredniane

lepsza stabilność, trudniejsza interpretacja - czarna skrzynka

Jedno drzewo za mało - posadźmy las;)

```
set.seed(12)

mtry.grid=expand.grid(mtry=1:10)

ttt2<-train(scat.train[,2:14],scat.train$Species,method = 'rf', preProcess =
c('center','scale'),tuneGrid=mtry.grid,metric='Accuracy',importance=T,
keep.forest=T)
```

tuneGrid - siatka parametrów po której train szuka najlepszego modelu

preProcess - można zapodać w tym miejscu

keep.forest - przyda się później

Random Forest

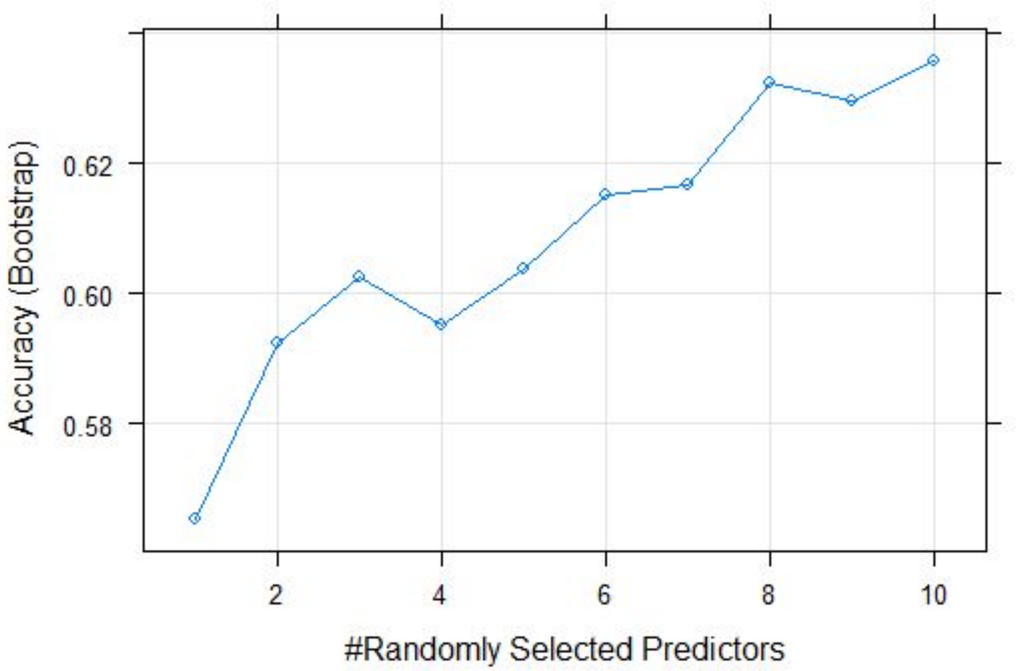
91 samples
13 predictors
3 classes: 'bobcat', 'coyote', 'gray fox'

Pre-processing: centered (6), scaled (6), ignore (7)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 91, 91, 91, 91, 91, 91, ...
Resampling results across tuning parameters:

mtry	Accuracy	Kappa
1	0.5652831	0.1136572
2	0.5922426	0.2388712
3	0.6025285	0.2714902
4	0.5950336	0.2763425
5	0.6037800	0.3002902
6	0.6152177	0.3239840
7	0.6166477	0.3307235
8	0.6323792	0.3591487
9	0.6297370	0.3540056
10	0.6357729	0.3662115

Accuracy was used to select the optimal model using the
largest value.
The final value used for the model was mtry = 10.

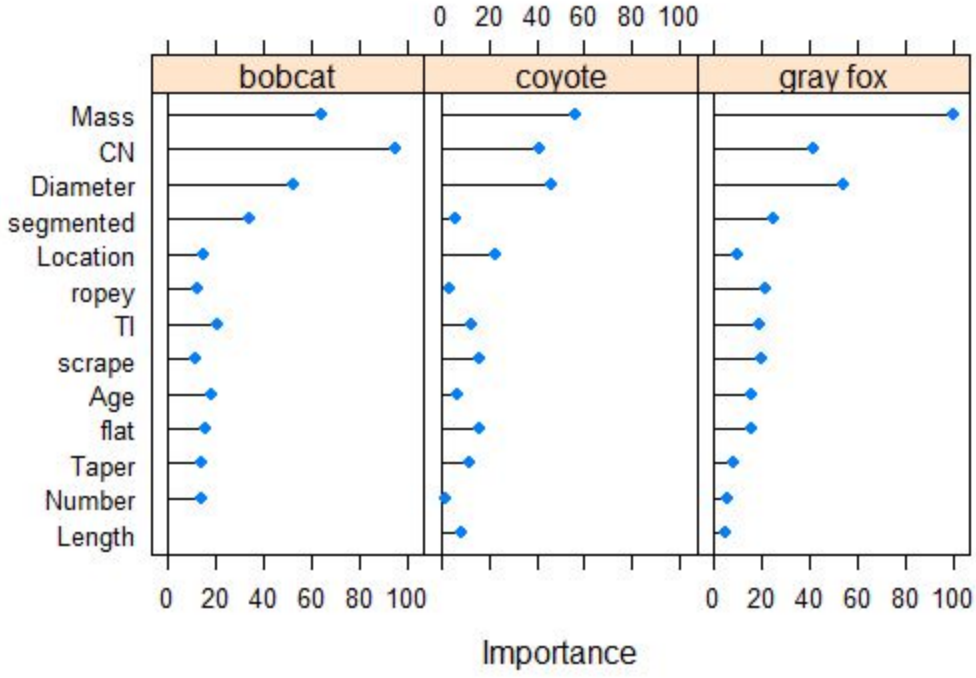
plot(ttt)



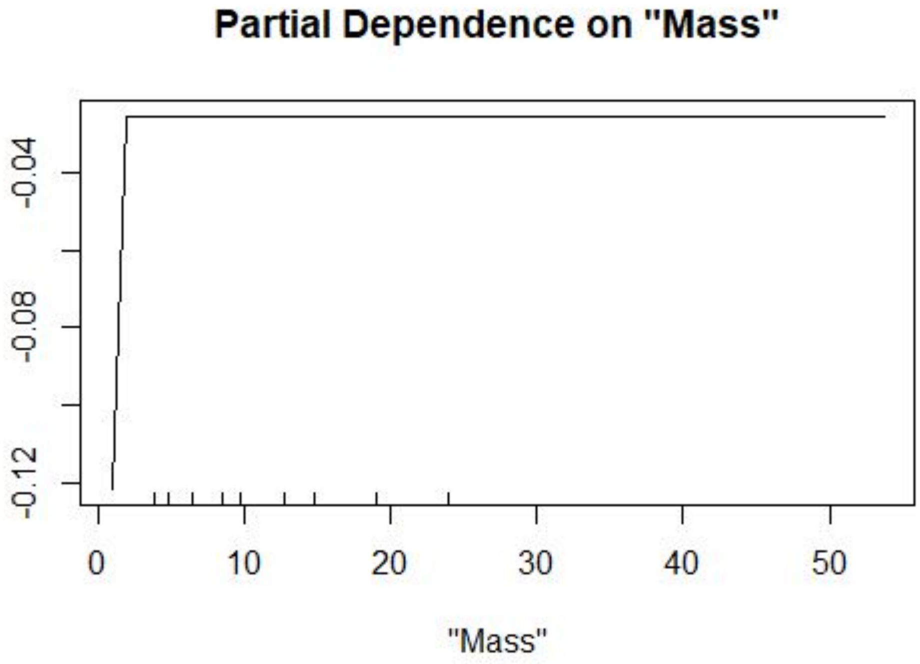
Accuracy (Bootstrap)
0.62
0.60
0.58
2
4
6
8
10
#Randomly Selected Predictors

```
> varImp(ttt)
rf variable importance

  variables are sorted by maximum importance across the classes
          bobcat coyote gray fox
Mass       64.37 55.993  100.000
CN         95.08 41.624   41.254
Diameter   52.96 46.565   53.883
segmented  33.98  5.822   24.556
Location   15.03 23.147    9.720
ropey      12.60  4.086   21.828
TI         20.66 12.953   19.454
scrape     11.58 15.830   20.081
Age        18.67  6.658   15.333
flat       15.83 15.830   15.830
Taper      13.99 12.237    7.819
Number     13.85  1.820    5.930
Length      0.00  8.711    4.459
> |
```

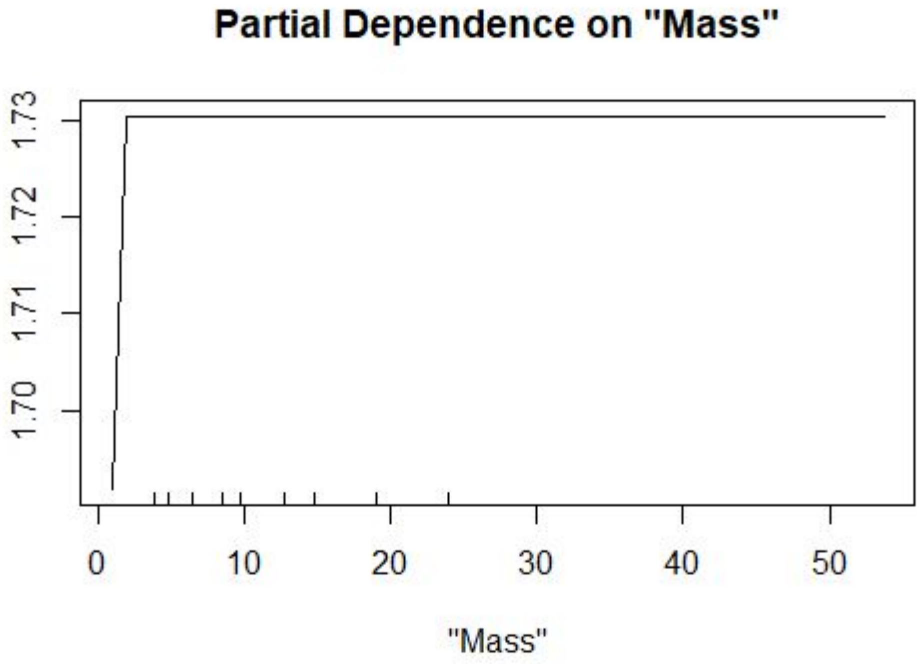
plot(varImp(ttt))



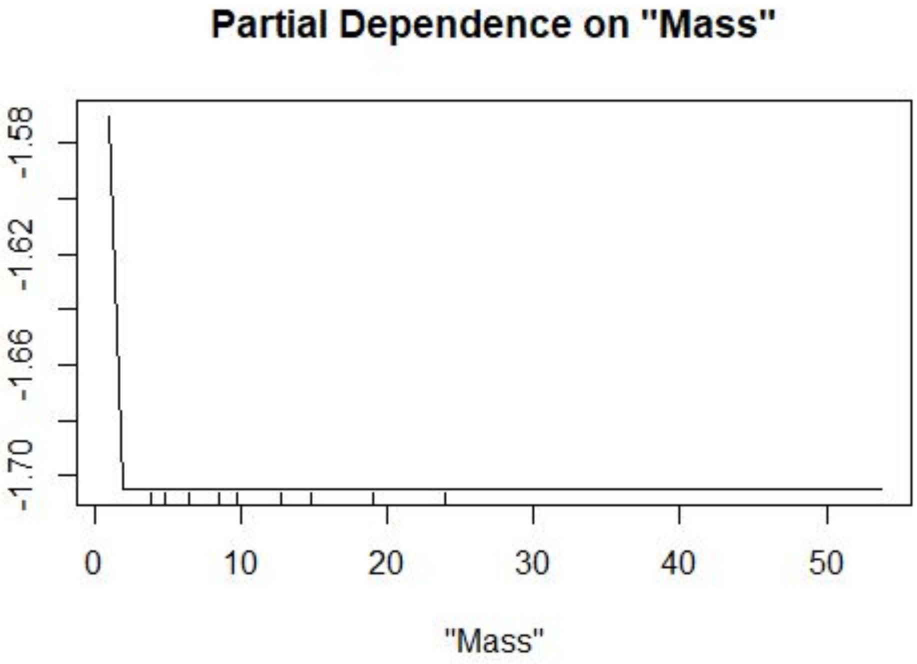
partialPlot(ttt$finalModel,Scat,'Mass')



partialPlot(ttt2$finalModel,Scat,'Mass','coyote')



partialPlot(ttt2$finalModel,Scat,'Mass','gray fox')



Confusion Matrix and Statistics

	Reference		
Prediction	bobcat	coyote	gray fox
bobcat	9	2	2
coyote	1	3	0
gray fox	2	1	2

Overall Statistics

Accuracy : 0.6364
95% CI : (0.4066, 0.828)
No Information Rate : 0.5455
P-Value [Acc > NIR] : 0.2622

Kappa : 0.3803
Mcnemar's Test P-Value : 0.7212

Statistics by Class:

	Class: bobcat	Class: coyote	Class: gray fox
Sensitivity	0.7500	0.5000	0.50000
Specificity	0.6000	0.9375	0.83333
Pos Pred Value	0.6923	0.7500	0.40000
Neg Pred Value	0.6667	0.8333	0.88235
Prevalence	0.5455	0.2727	0.18182
Detection Rate	0.4091	0.1364	0.09091
Detection Prevalence	0.5909	0.1818	0.22727
Balanced Accuracy	0.6750	0.7188	0.66667

randomForest - łatwiej przedstawić przy regresji lub binomial

Gradient Boosted Modeling

GBM, BRT, jak zwał tak zwał

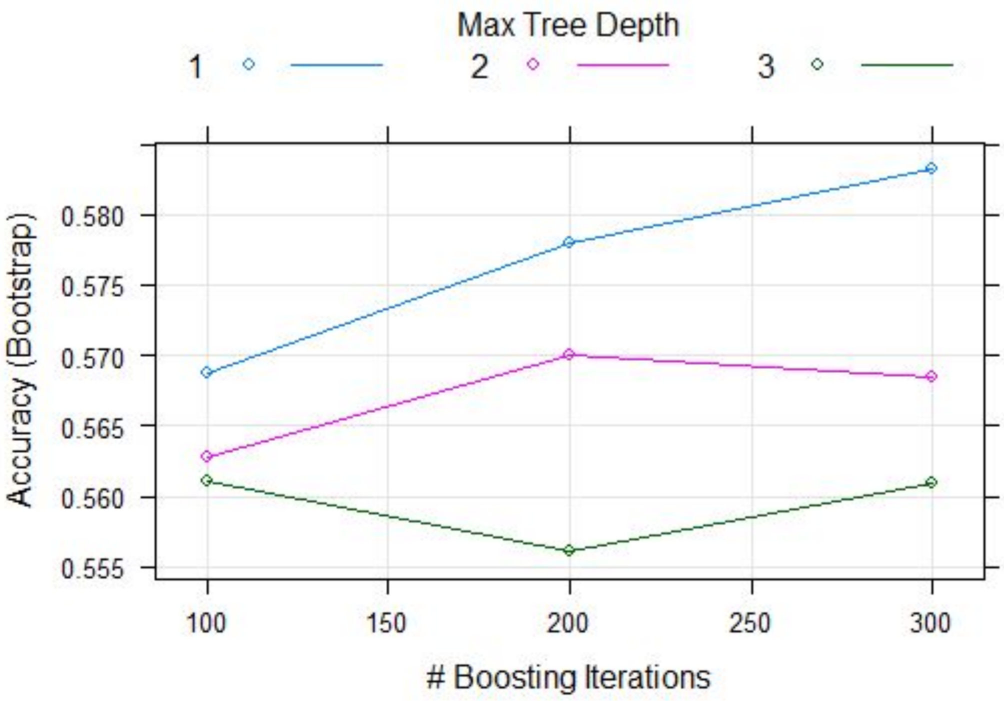
wrzucamy dane, drzewa je przetwarzają - część z nich zaklasyfikowana dobrze

źle zaklasyfikowane - nowe drzewa

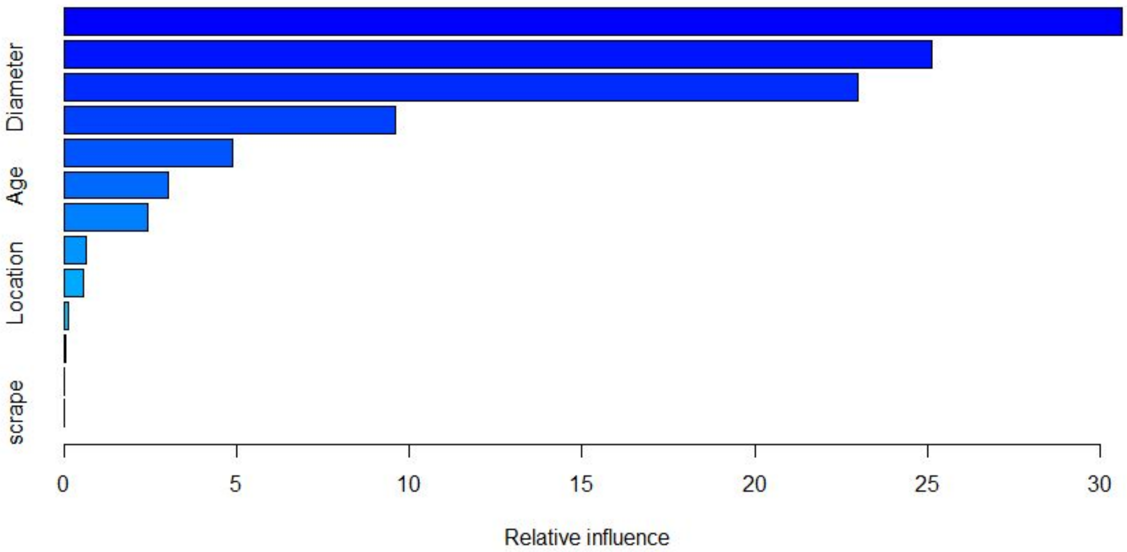
i tak do skutku;)

```
set.seed(111)

gbm.grid=expand.grid(n.trees=c(100,200,300),interaction.depth=1:3,shrinkage=.01,n.minobsinnode=5)

gbmscat<-train(scat.train[,2:14],scat.train$Species,method = 'gbm',preProcess = c('center','scale'),tuneGrid=gbm.grid,metric='Accuracy')
```

plot(gbmscat)

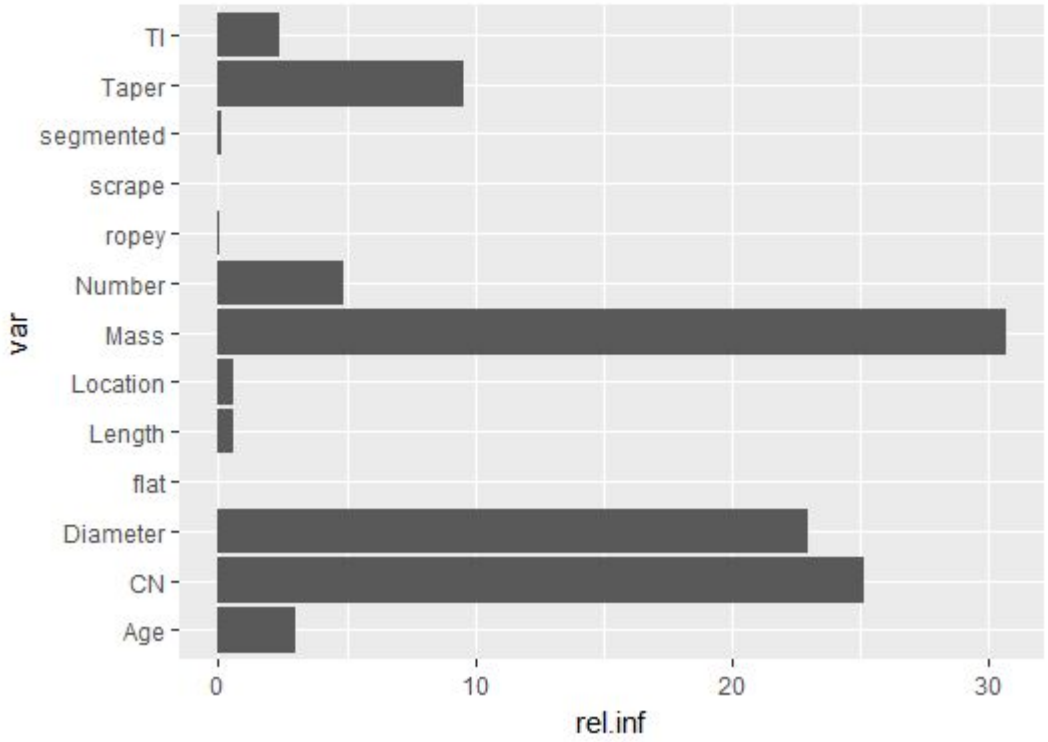


summary(gbmscat)



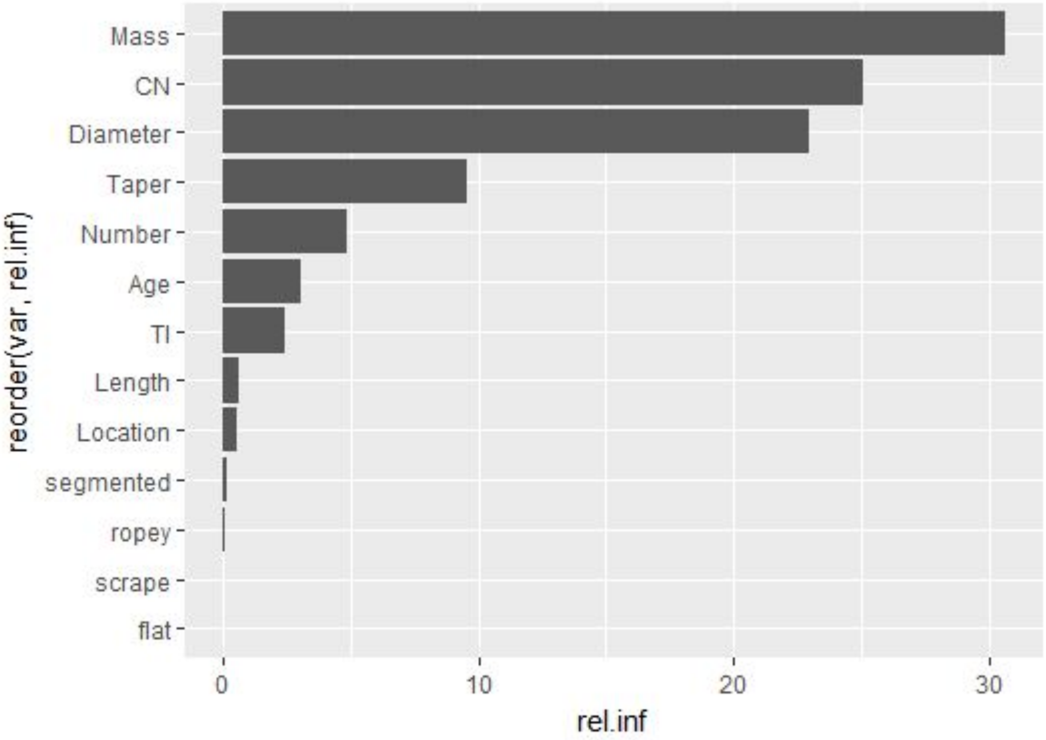
lepiej ggplotem

ggplot(**as.data.frame(summary(gbmscat))**,aes(x=var,y=rel.inf))+geom_col()+coord_flip()

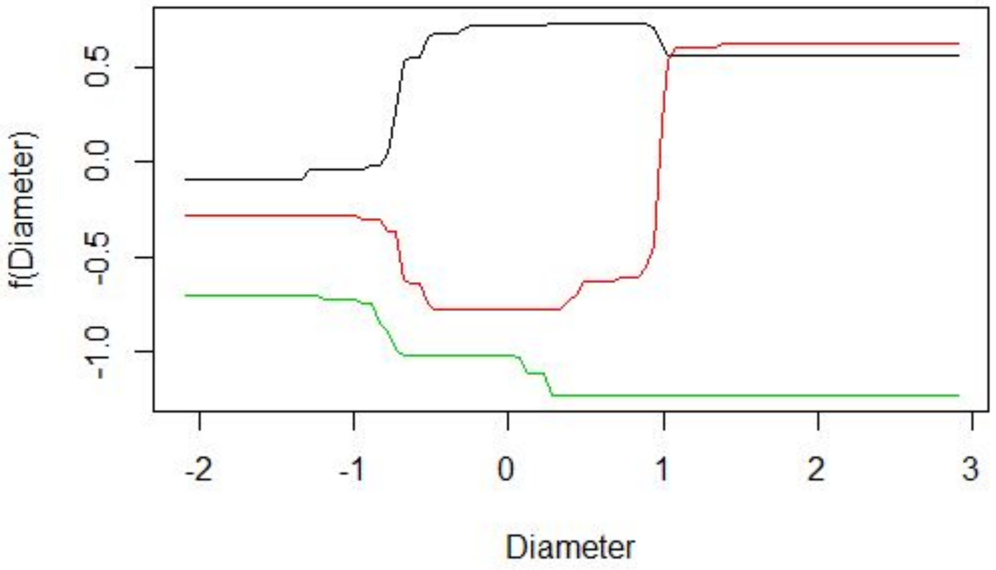


lepiej ggplotem

ggplot(as.data.frame(summary(gbmscat)),aes(x=**reorder(var,rel.inf)**,y=rel.inf))+geom_col()+coord_flip()



```
library(gbm)

plot.gbm(gbmscat$finalModel,5)
```



```
> confusionMatrix(predict(gbmscat,scat.test),scat.test$Species)
Confusion Matrix and Statistics

          Reference
Prediction bobcat coyote gray fox
  bobcat       11      3        3
  coyote        1      3        0
  gray fox      0      0        1
```

Accuracy : 0.6818

95% CI : (0.4513, 0.8614)

No Information Rate : 0.5455

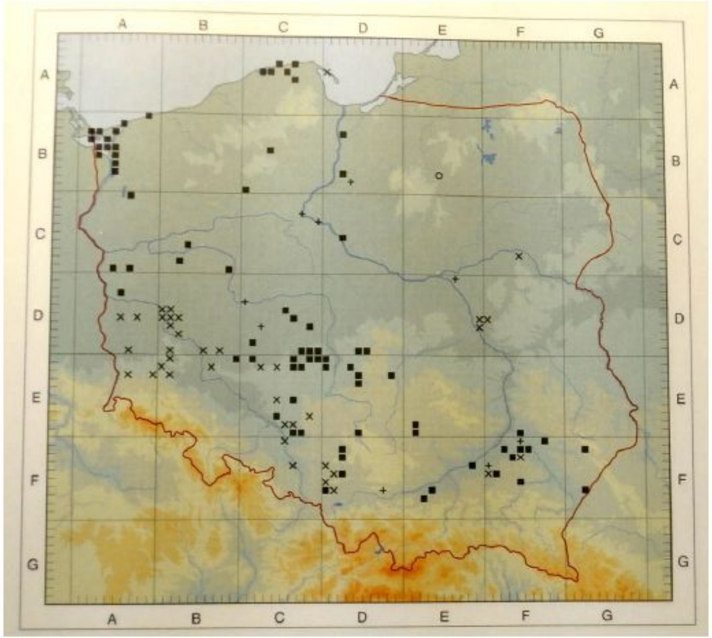
P-Value [Acc > NIR] : 0.1419

Kappa : 0.3889

Mcnemar's Test P-Value : NA

	Class: bobcat	Class: coyote	Class: gray fox
Sensitivity	0.9167	0.5000	0.25000
Specificity	0.4000	0.9375	1.00000
Pos Pred Value	0.6471	0.7500	1.00000
Neg Pred Value	0.8000	0.8333	0.85714
Prevalence	0.5455	0.2727	0.18182
Detection Rate	0.5000	0.1364	0.04545
Detection Prevalence	0.7727	0.1818	0.04545
Balanced Accuracy	0.6583	0.7188	0.62500

Klasyfikacja binarna - *Osmunda regalis*



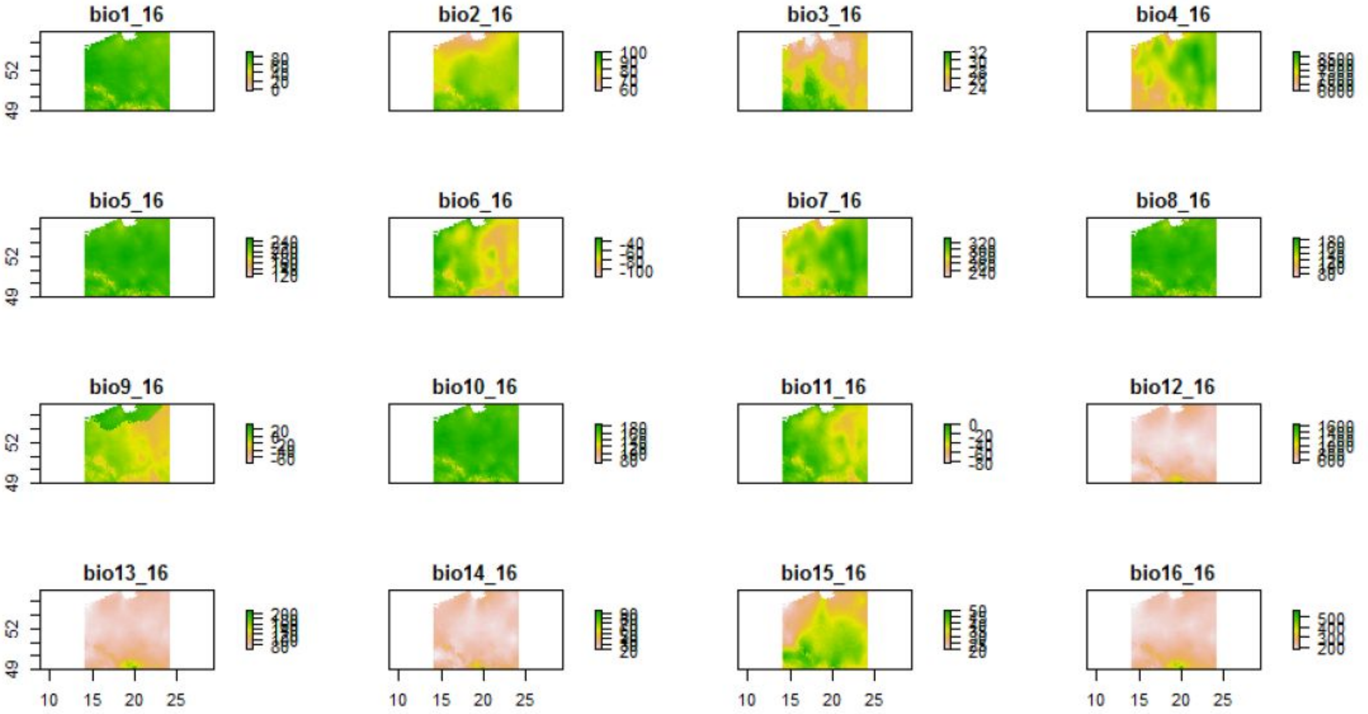
(Zając & Zając 2001)

- Atlas Rozmieszczenia Roślin Naczyniowych w Polsce [ATPOL]
- <100 stanowisk w Polsce
- Polska Czerwona Lista

Skąd dane o klimacie?

BIO1 = Annual Mean Temperature
BIO2 = Mean Diurnal Range (Mean of monthly (max temp - min temp))
BIO3 = Isothermality (BIO2/BIO7) (* 100)
BIO4 = Temperature Seasonality (standard deviation *100)
BIO5 = Max Temperature of Warmest Month
BIO6 = Min Temperature of Coldest Month
BIO7 = Temperature Annual Range (BIO5-BIO6)
BIO8 = Mean Temperature of Wettest Quarter
BIO9 = Mean Temperature of Driest Quarter
BIO10 = Mean Temperature of Warmest Quarter
BIO11 = Mean Temperature of Coldest Quarter
BIO12 = Annual Precipitation
BIO13 = Precipitation of Wettest Month
BIO14 = Precipitation of Driest Month
BIO15 = Precipitation Seasonality (Coefficient of Variation)
BIO16 = Precipitation of Wettest Quarter
BIO17 = Precipitation of Driest Quarter
BIO18 = Precipitation of Warmest Quarter
BIO19 = Precipitation of Coldest Quarter

można ściągnąć bez problemu funkcją raster::getData()



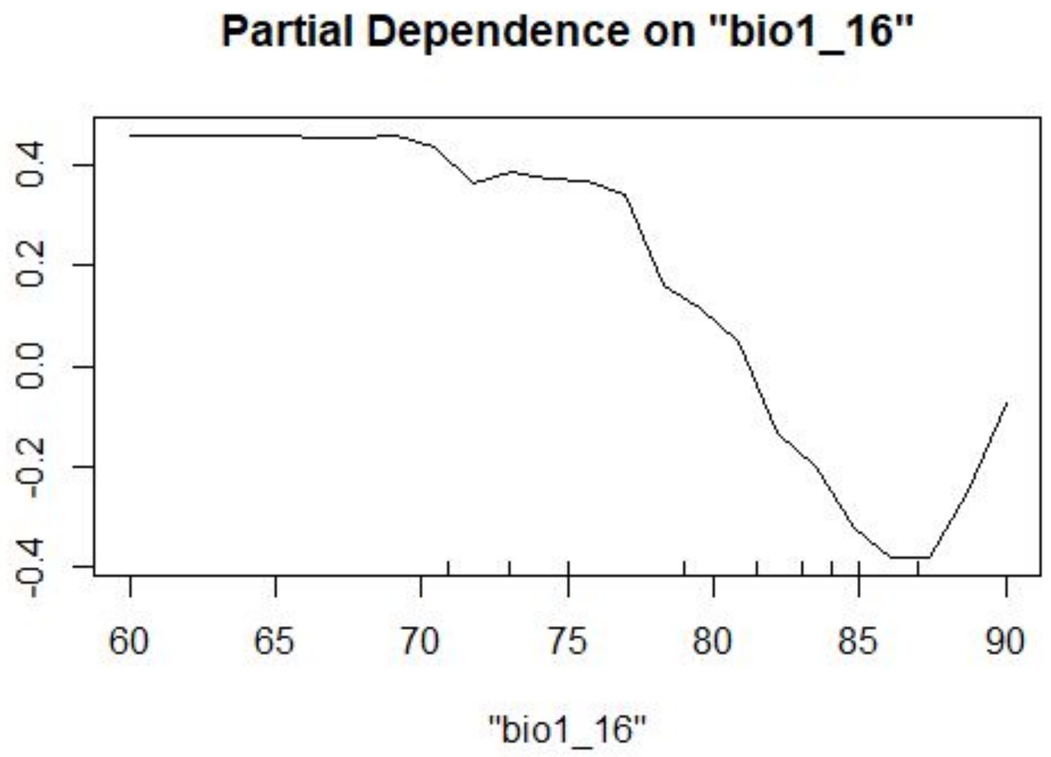
```
set.seed(11)
osm.dziel<-createDataPartition(osmunda$osmunda,p=.75,list=F)
osm.tr<-osmunda[osm.dziel,]
osm.te<-osmunda[-osm.dziel,]

ffitControl <- trainControl(method =
"repeatedcv",number=10,repeats=10,classProbs = TRUE,summaryFunction =
twoClassSummary)
model.osm<-train(osm.tr[,1:14],as.factor(osm.tr$osmunda),method =
'rf',metric='ROC',trControl = fitControl ,tuneGrid = mtry.grid)
```

Error: At least one of the class levels is not a valid R variable name; This will cause errors when class probabilities are generated because the variables names will be converted to X0, X1 . Please use factor levels that can be used as valid R variable names (see ?make.names for help).

w caret random forest nie wchodzą do ROC 0-1

```
Random Forest

120 samples
 14 predictor
  2 classes: 'brak', 'obecny'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 108, 108, 108, 108, 108, 108, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   1    0.8025000  0.7300000  0.7416667
   2    0.8098611  0.7450000  0.7383333
   3    0.8043056  0.7466667  0.7433333
   4    0.8095833  0.7433333  0.7300000
   5    0.8081944  0.7483333  0.7366667
   6    0.8050000  0.7416667  0.7350000
   7    0.8084722  0.7416667  0.7333333
   8    0.8083333  0.7433333  0.7433333
   9    0.8041667  0.7400000  0.7433333
  10    0.8090278  0.7333333  0.7433333

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

partialPlot(model.osm$finalModel,osm.tr,'bio1_16')



```
confusionMatrix(predict(model.osm$finalModel,osm.te),osm.te$osmunda)
```

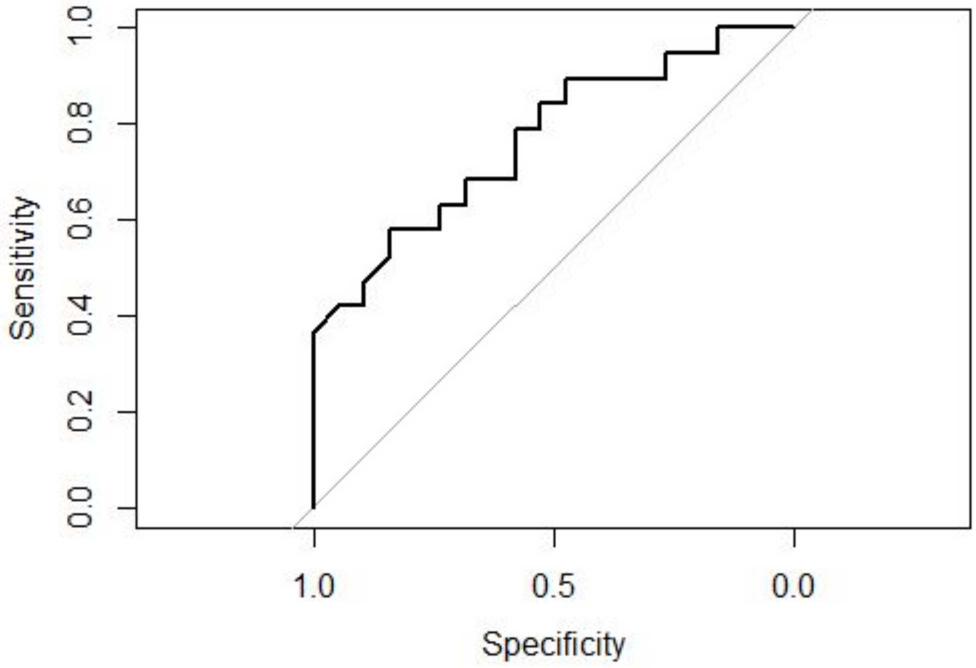
```
        Reference
Prediction brak obecny
   brak     13      6
   obecny    6     13

Accuracy : 0.6842
             95% CI : (0.5135, 0.825)
   No Information Rate : 0.5
   P-Value [Acc > NIR] : 0.01678
```

```
Sensitivity : 0.6842
         Specificity : 0.6842
       Pos Pred Value : 0.6842
       Neg Pred Value : 0.6842
          Prevalence : 0.5000
       Detection Rate : 0.3421
  Detection Prevalence : 0.5000
    Balanced Accuracy : 0.684
```

o co chodzi z tym ROC?

receiver-operator curve - czułość vs. specyficzność

```
library(pROC)
krzywa<-roc(osm.te$osmunda,predykcja[,2])
plot(krzywa)
krzywa
```



Area under the curve: 0.7673

dużo? mało?

ważne że porównujemy zbiór testowy

<0.7 - słabo (ale też publikowalne;)

0.7-0.8 - w miarę

0.8-0.9 dobrze

0.9-0.95 - bardzo dobrze

0.95-0.98 - excellent

>.98 - coś jest nie tak!

od czego zależy AUC?

od zbilansowania klas

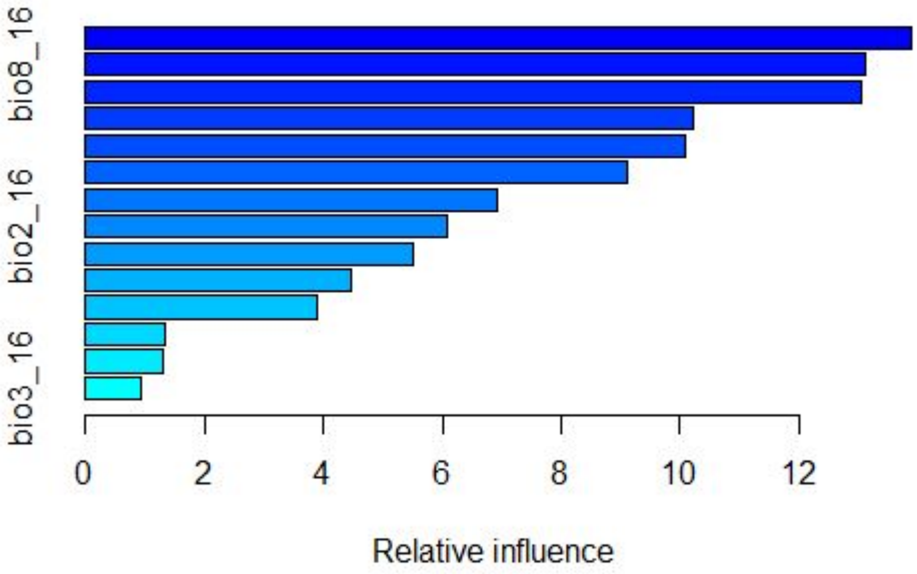
od zbilansowania predyktorów

inne podejście - gbm

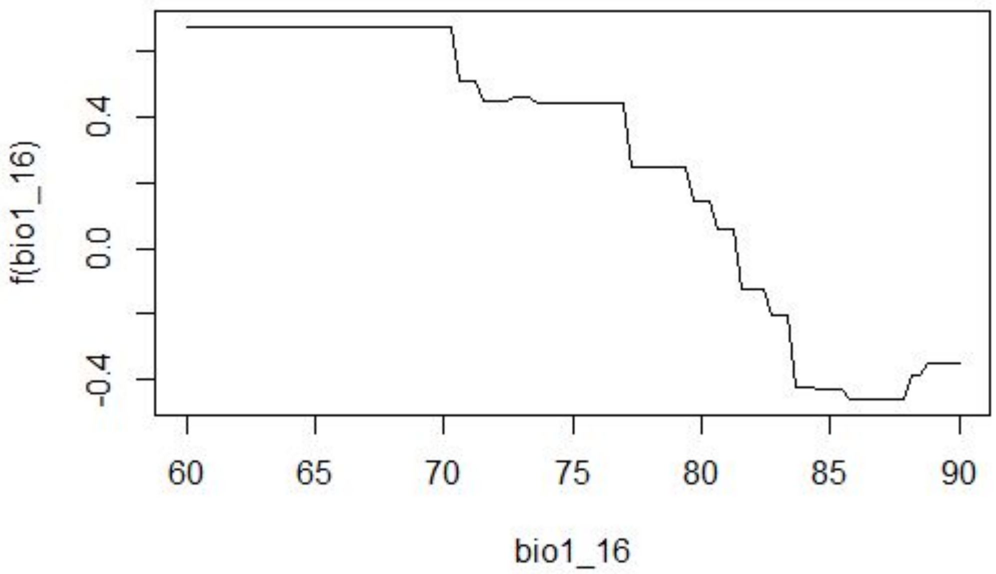
```
set.seed(9)

osmgbm<-train(osm.tr[,1:14],as.factor(osm.tr$osmunda),method =
'gbm',metric='ROC',trControl = fitControl ,tuneGrid = gbm.grid)
```

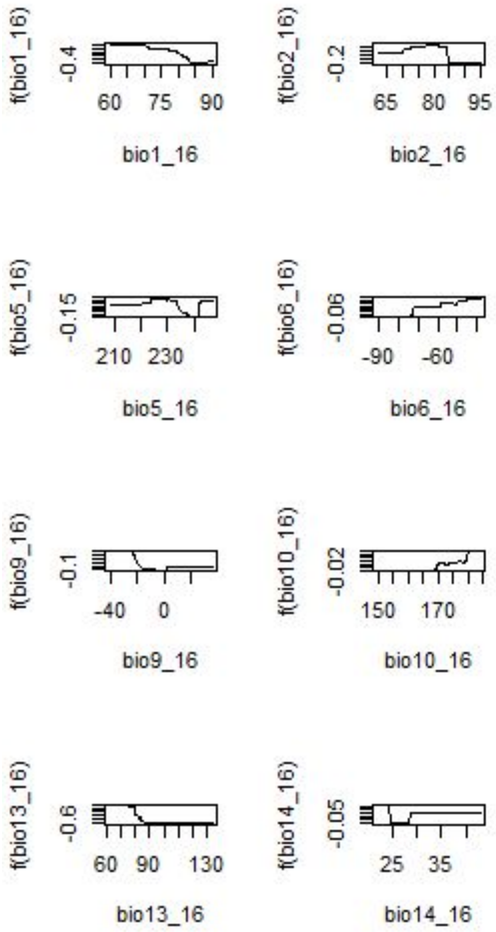
```
No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 108, 108, 108, 108, 108, 108, ...
Resampling results across tuning parameters:

  interaction.depth  n.trees  ROC        Sens       Spec
  1                  100      0.7602778  0.6616667  0.7133333
  1                  200      0.7879167  0.6716667  0.7100000
  1                  300      0.7995833  0.6833333  0.7183333
  2                  100      0.8058333  0.6950000  0.7266667
  2                  200      0.8180556  0.7083333  0.7500000
  2                  300      0.8222222  0.7250000  0.7566667
  3                  100      0.8184722  0.7066667  0.7433333
  3                  200      0.8280556  0.7166667  0.7616667
  3                  300      0.8377778  0.7250000  0.7783333

Tuning parameter 'shrinkage' was held constant at a value of
 0.01
Tuning parameter 'n.minobsinnode' was held constant at
 a value of 5
ROC was used to select the optimal model using the largest value.
The final values used for the model were n.trees =
 300, interaction.depth = 3, shrinkage = 0.01 and
 n.minobsinnode = 5.
>
```

summary(osmgbm)



plot.gbm(osmgbm$finalModel,1)

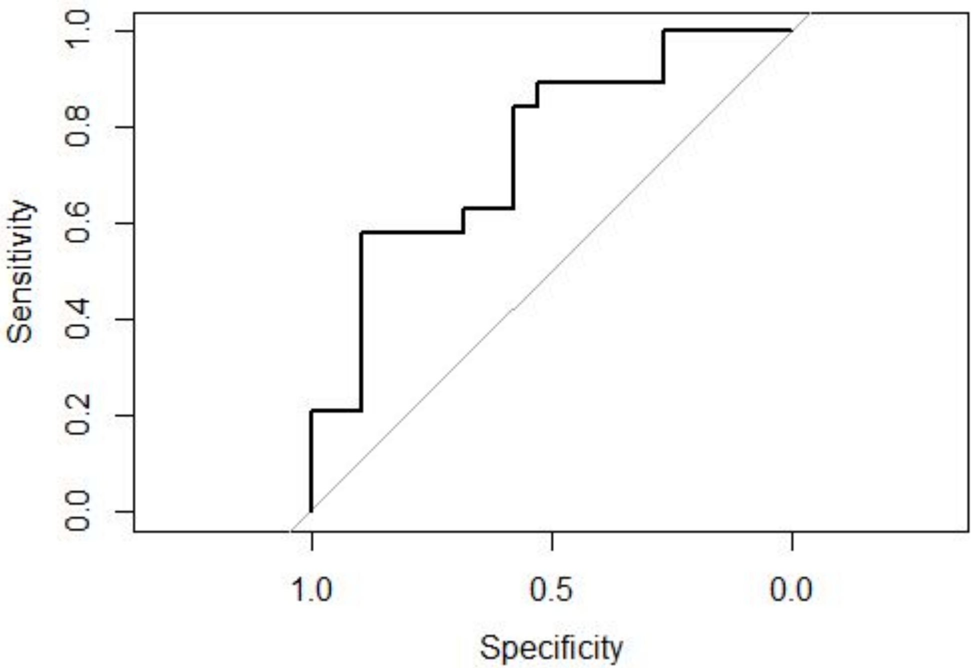





```
par(mfrow=c(4,4))
for(i in 1:14) plot.gbm(osmgbm$finalModel,i)
```

```
predykcja<-predict(osmgbm$finalModel,osm.te,n.trees=200)
krzywa<-roc(osm.te$osmunda,predykcja)
krzywa

Call:
roc.default(response = osm.te$osmunda, predictor = predykcja)

Data: predykcja in 19 controls (osm.te$osmunda brak) > 19 cases
(osm.te$osmunda obecny).
Area under the curve: 0.7535
```

plot(krzywa)



Regresja

Zbiór danych z Poznania:

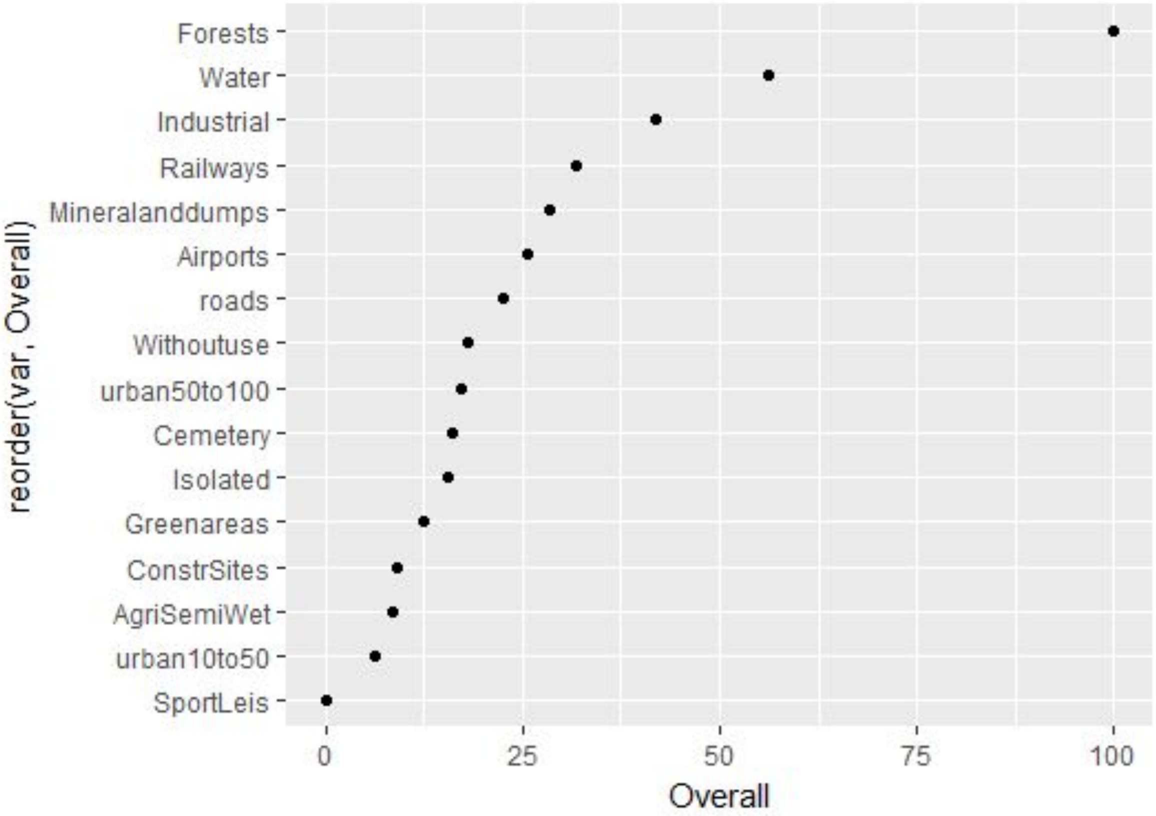
zmienna objaśniana - liczba gatunków starych lasów

predyktory - udział typów zagospodarowania terenu

```
mtryGrid<-expand.grid(mtry=c(2,4,6,8,10,12,15))#parametry RF - ile losowo
dobranych ma być w modelu - który lepszy będzie

poznan<-train(x=zbtrainG[,c(2:14,17:19)],y=zbtrainG$ile,
meth=c('rf'),preProcess=c('YeoJohnson','center','scale'),trControl=ctrl,
importance=TRUE,tuneGrid = mtryGrid)
```

```
ance=TRUE,tuneGrid = mtryGrid)#SoF mowi zeby dodac imp=T)
> poznan#oglądamy model
Random Forest

182 samples
 16 predictor

Pre-processing: Yeo-Johnson transformation (11), centered
 (16), scaled (16)
Resampling: Cross-Validated (10 fold, repeated 1 times)
Summary of sample sizes: 165, 165, 164, 163, 164, 164, ...
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
   2    7.363558  0.2245170  5.204747
   4    7.416160  0.2221702  5.258511
   6    7.474293  0.2183596  5.246684
   8    7.496270  0.2185214  5.244452
  10    7.504952  0.2230687  5.251739
  12    7.549857  0.2182108  5.279518
  15    7.607259  0.2183835  5.281102

RMSE was used to select the optimal model using the
 smallest value.
The final value used for the model was mtry = 2.
>
```

```
poznanimp<-(varImp(poznan)$importance)
poznanimp$var<-rownames(poznanimp)
ggplot(poznanimp,aes(x=reorder(var,Overall),y=Overall))+geom_point()+coord_flip(
```

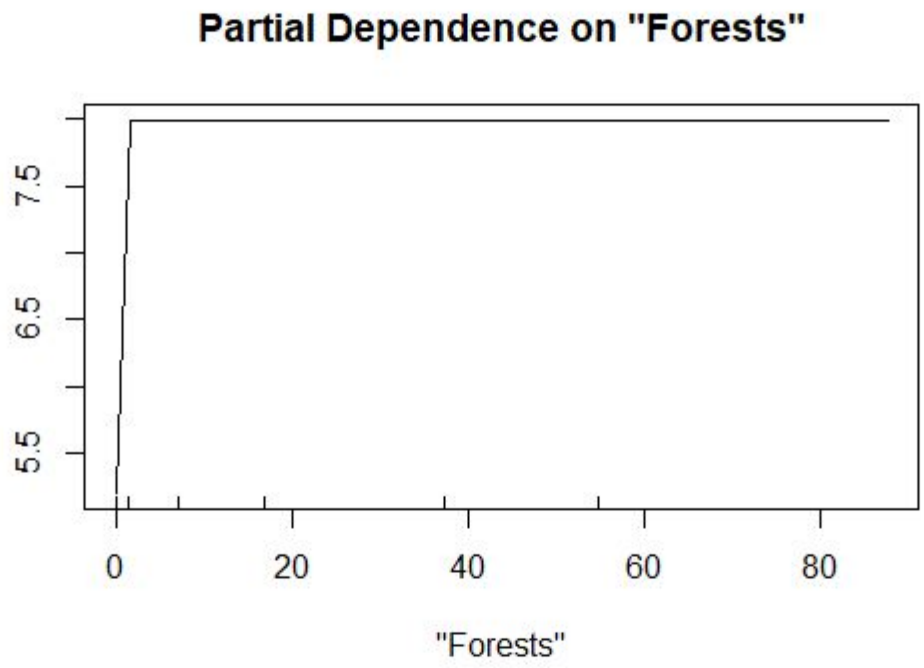


test RMSE

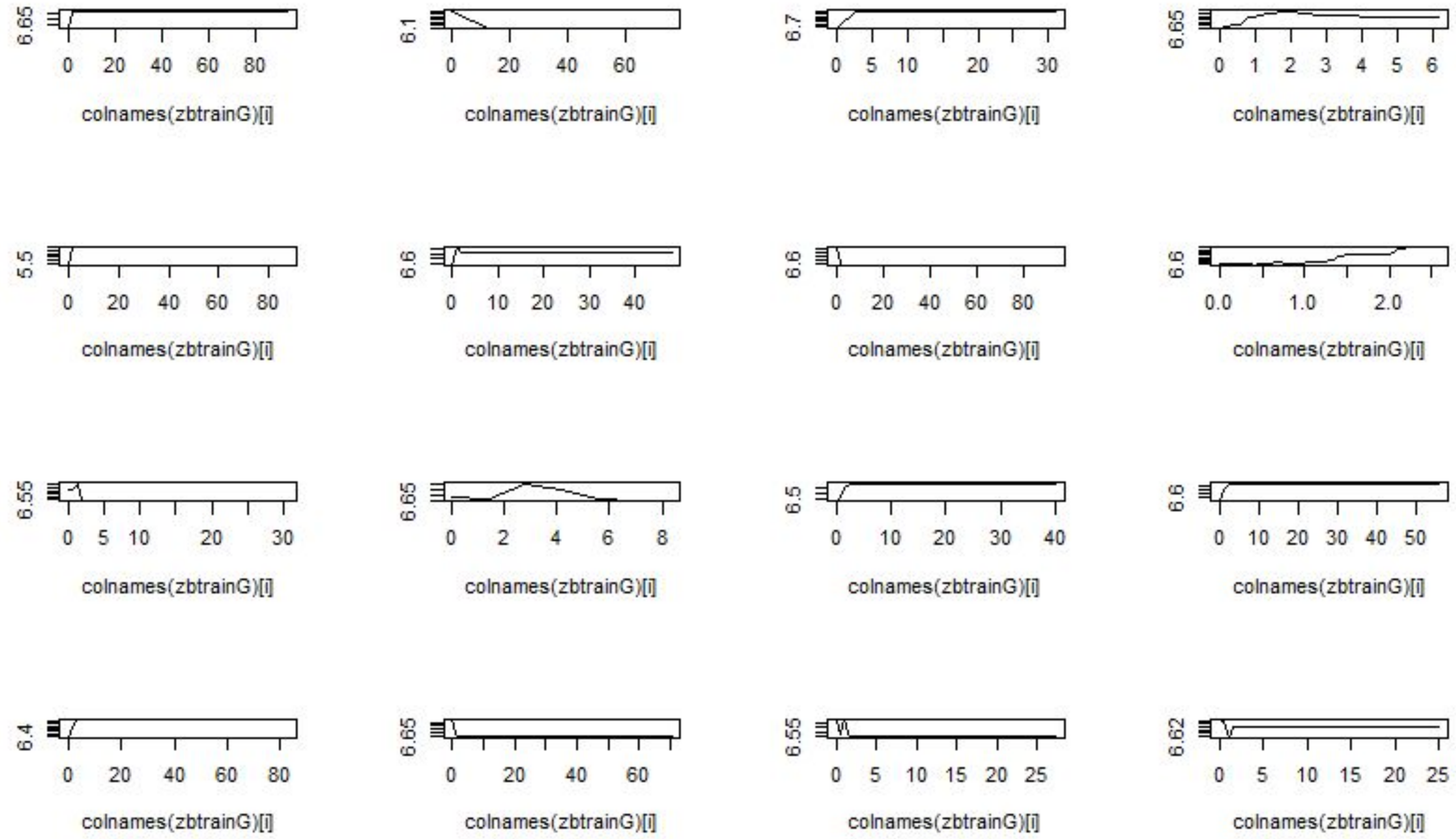
```
> sqrt(mean((zbtrainG$ile-(predict(poznan$finalModel, zbtrainG)))^2))
[1] 7.684401
> sqrt(mean((zbtestG$ile-(predict(poznan$finalModel, zbtestG)))^2))
[1] 8.520902
```

lekki overfitting

partialPlot(poznan$finalModel,zbtrainG,'Forests')



```
par(mfrow=c(4,4))
for(i in c(2:14,17:19)) partialPlot(poznan$finalModel,zbtrainG,colnames(zbtrainG)[i],main='')
```



Dobre źródła informacji

https://nowosad.github.io/files/presentations/pazur12/#1 - świetna prezentacja po polsku

http://topepo.github.io/caret/ - manual careta

publikacje, typy modeli, konkretne zastosowania

BSS

BIAŁOWIESKA SZKOŁA STATYSTYKI