Seminarska naloga 1 (Umetna Inteligence 2021-2022)

Bartolomej Kozorog (63200152)

December 5, 2021

Contents

Cilj seminarske naloge je uporabiti metode strojnega učenja za gradnjo modelov za napovedovanje porabe električne energije (regresijski problem) in namembnosti stavbe (klasifikacijski problem), ustrezno ovrednotiti modele in jasno predstaviti dobljene rezultate.

Knjiznice in orodja

Vecina uporabljenih knjiznic je ze privzeto namescenih. Potrebno pa bo namestiti tudi nekaj zunanjih knjicnic, kot sta `ggplot2` in `ggcorrplot` (za risanje grafov).

Vecina pomoznih metod se nahaja v zunanji R skripti `common.R`.

```
library(lubridate) # delo z datumi
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(stringr) # delo z znakovnimi nizi
library(ggplot2)
library(ggcorrplot)
library(rpart)
library(rpart.plot)
library(CORElearn) # za ucenje
library(nnet)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(ipred) # bagging
library(adabag) # boosting
```

```
## Loading required package: caret

## Loading required package: lattice

## Loading required package: foreach

## Loading required package: doParallel

## Loading required package: iterators

## Loading required package: parallel

##
## Attaching package: 'adabag'

## The following object is masked from 'package:ipred':
##
##     bagging
```

```
source("./common.R") # pomozne metode

set.seed(0) # nastavimo random seed
```

Vizualizacija podatkov

Uvoz podatkov

Najprej uvozimo in na kratko preglejmo podatke.

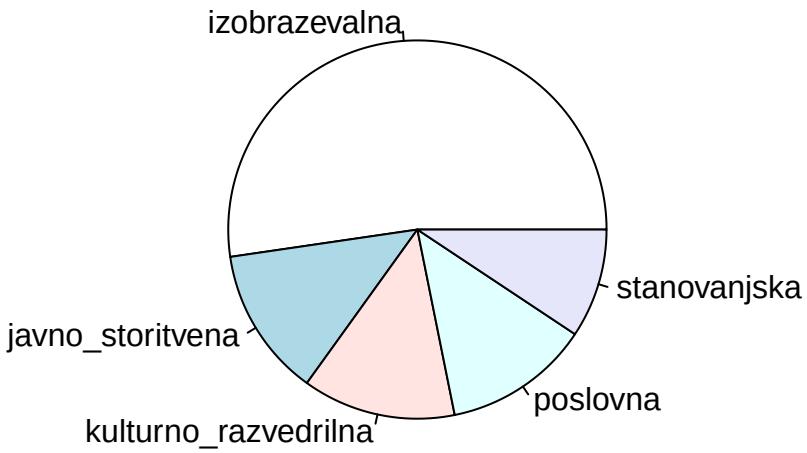
Opazimo, da imamo 3 atribute tipa "character": `datum`, `regija` in `namembnost`. Atributa `regija` in `namembnost` (z indeksi 2 in 4) imata le majhno stevilo vrednosti, zato jih bomo faktorizirali. Datum bomo pa kasneje preuredili v bolj smiselno obliko.

```
train <- read.table("trainset.txt", header=T, sep=",")
test <- read.table("testset.txt", header=T, sep=",")

# zmanjsamo mnozici za potrebo razvoja
# TODO: odstrani pred zadnjim buildom
trainSel <- sample(1:nrow(train), as.integer(nrow(train) * 0.1), replace=T)
testSel <- sample(1:nrow(test), as.integer(nrow(test) * 0.1), replace=T)
train <- train[trainSel,]
test <- test[testSel,]

train <- Factorize(train)
test <- Factorize(test)

allData <- rbind(test, train)
```

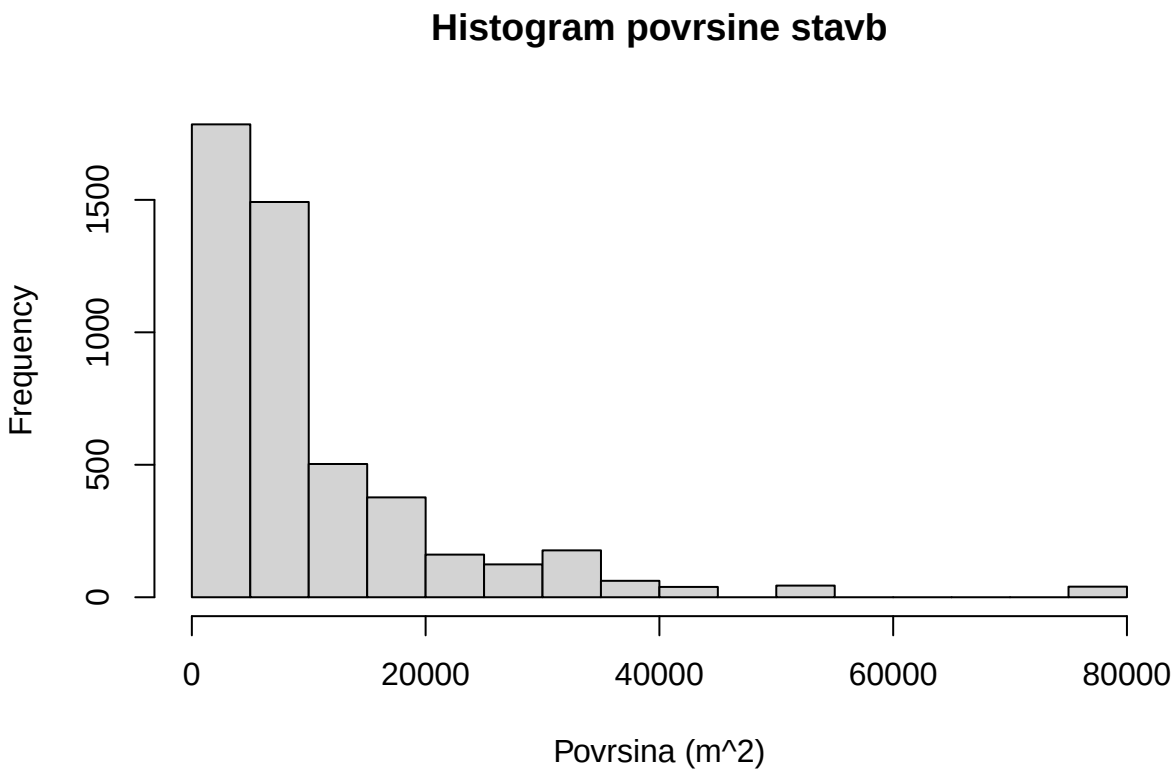
Izris grafov

Porazdelitvene vrednosti Vizualizirajmo porazdelitvene vrednosti posameznih atributov, da dobimo boljsi vpogled v vsak atribut posebej.

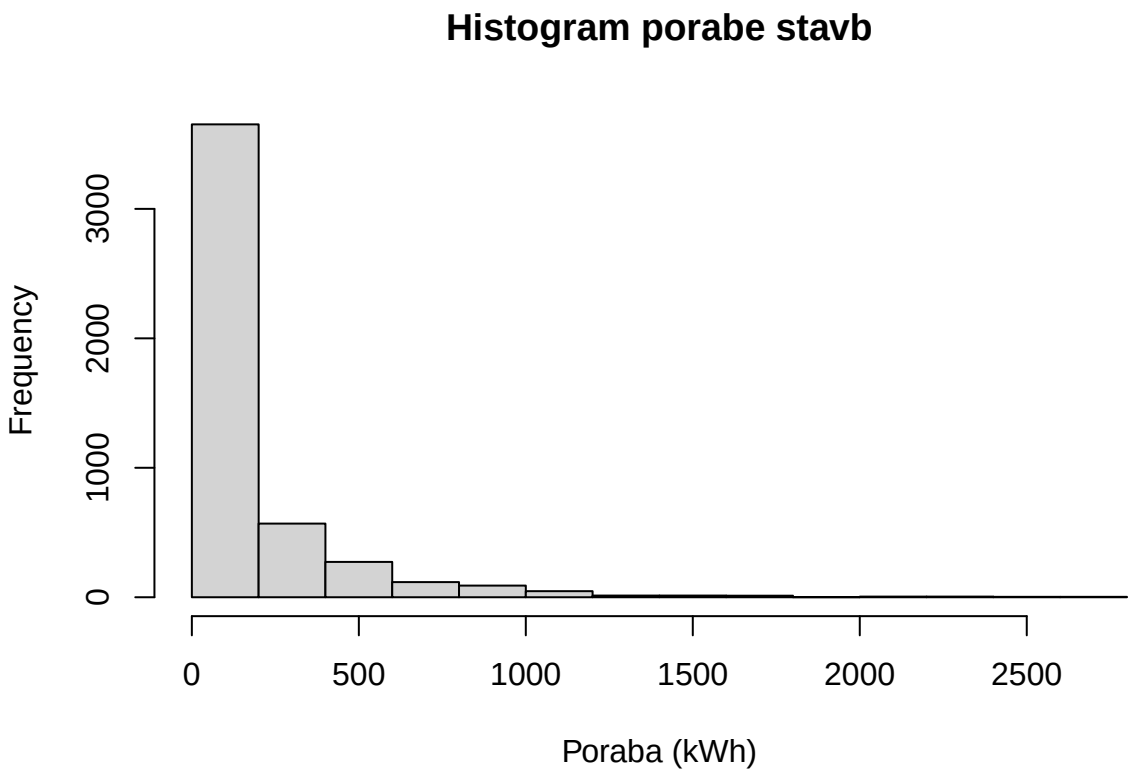
```
pie(table(allData$namembnost), xlab="Namembnost")
```



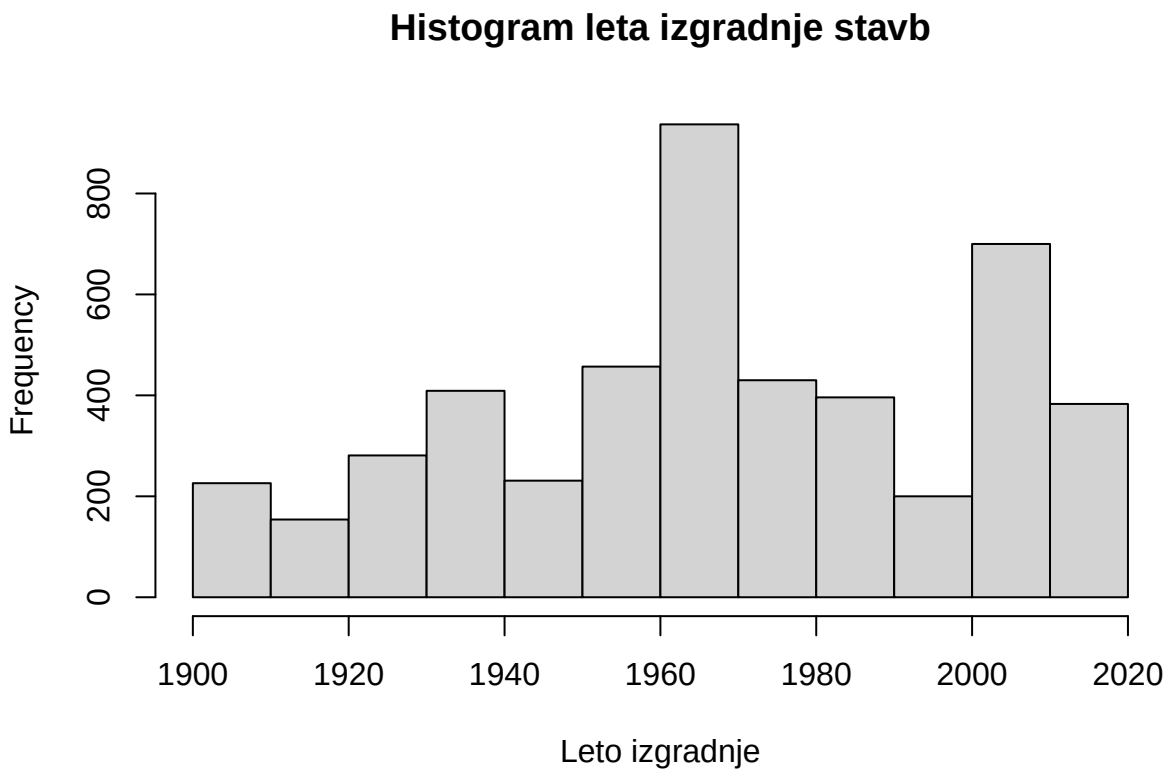

```
hist(allData$povrsina, xlab="Povrsina (m^2)", main="Histogram povrsine stavb")
```



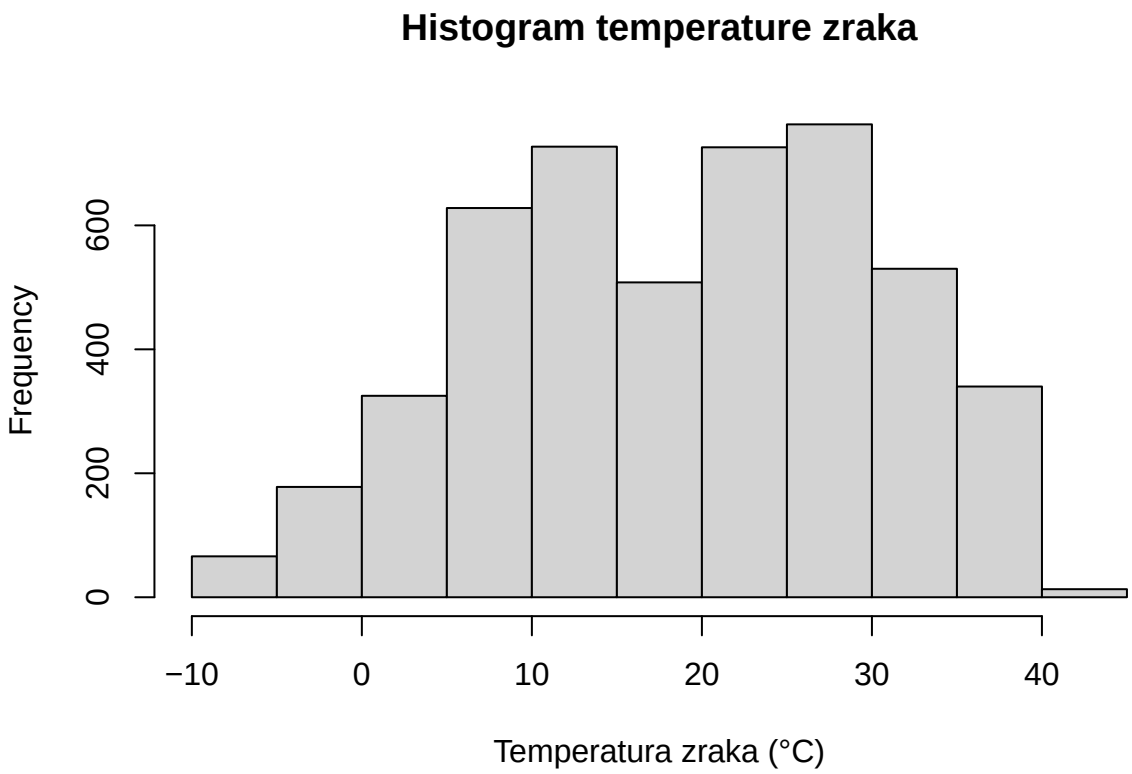

```
hist(allData$poraba, xlab="Poraba (kWh)", main="Histogram porabe stavb")
```



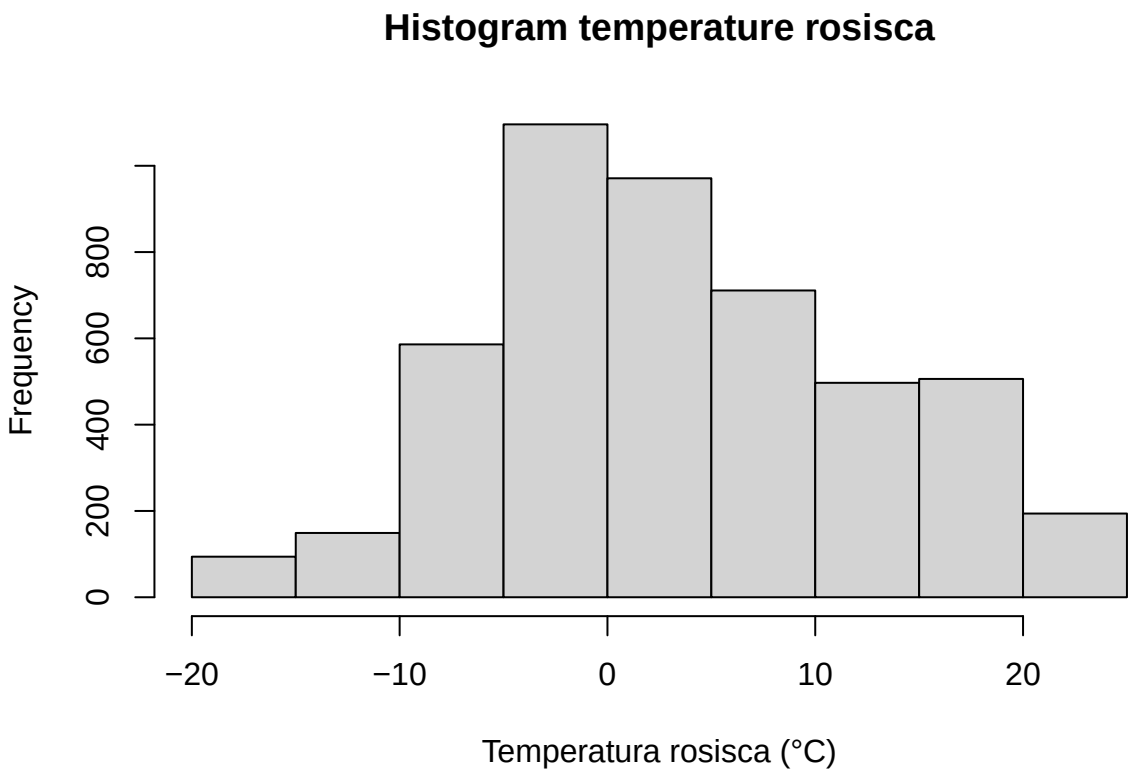

```
hist(allData$leto_izgradnje, xlab="Leto izgradnje", main="Histogram leta izgradnje stavb")
```



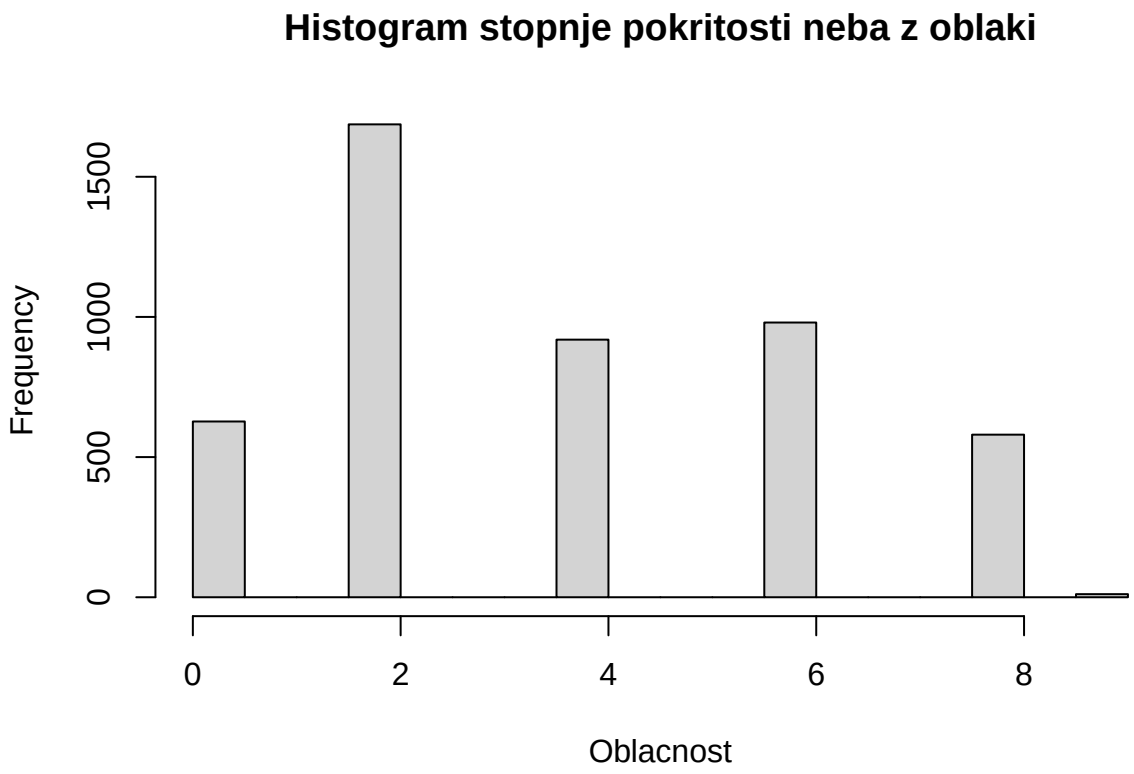

```
hist(allData$temp_zraka, xlab="Temperatura zraka (°C)", main="Histogram temperature zraka")
```



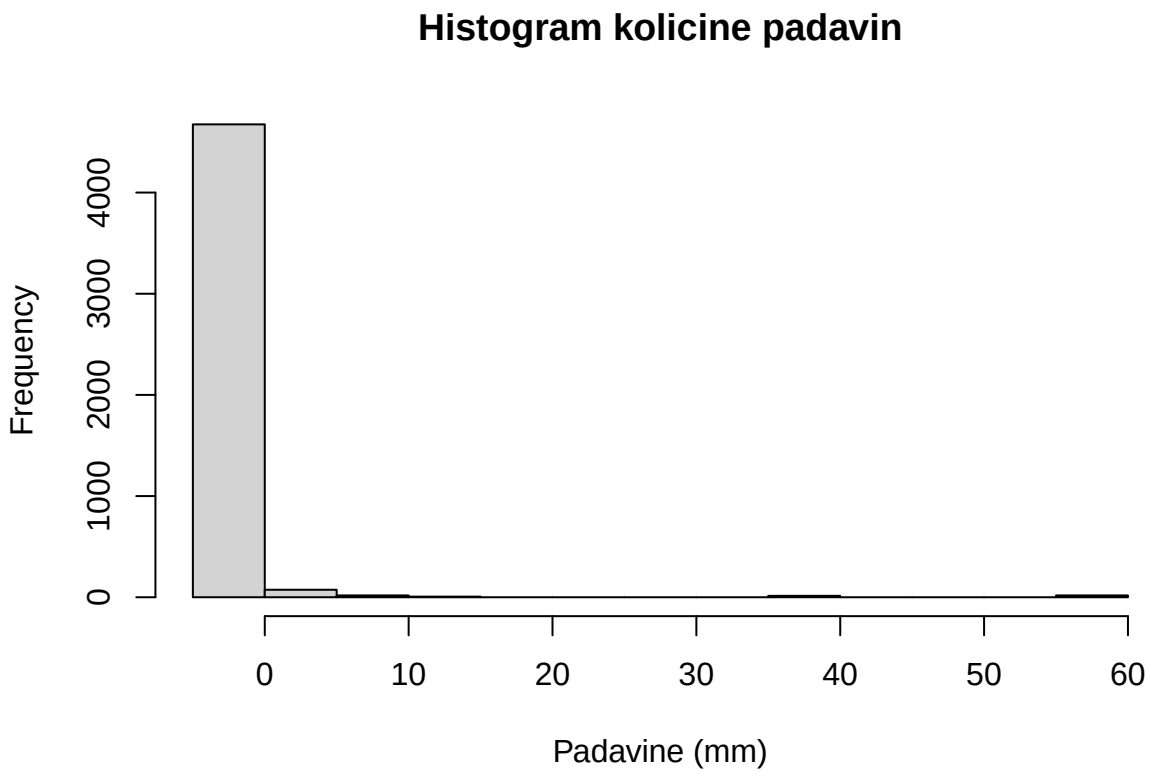

```
hist(allData$temp_rosisca, xlab="Temperatura rosisca (°C)", main="Histogram temperature rosisca")
```



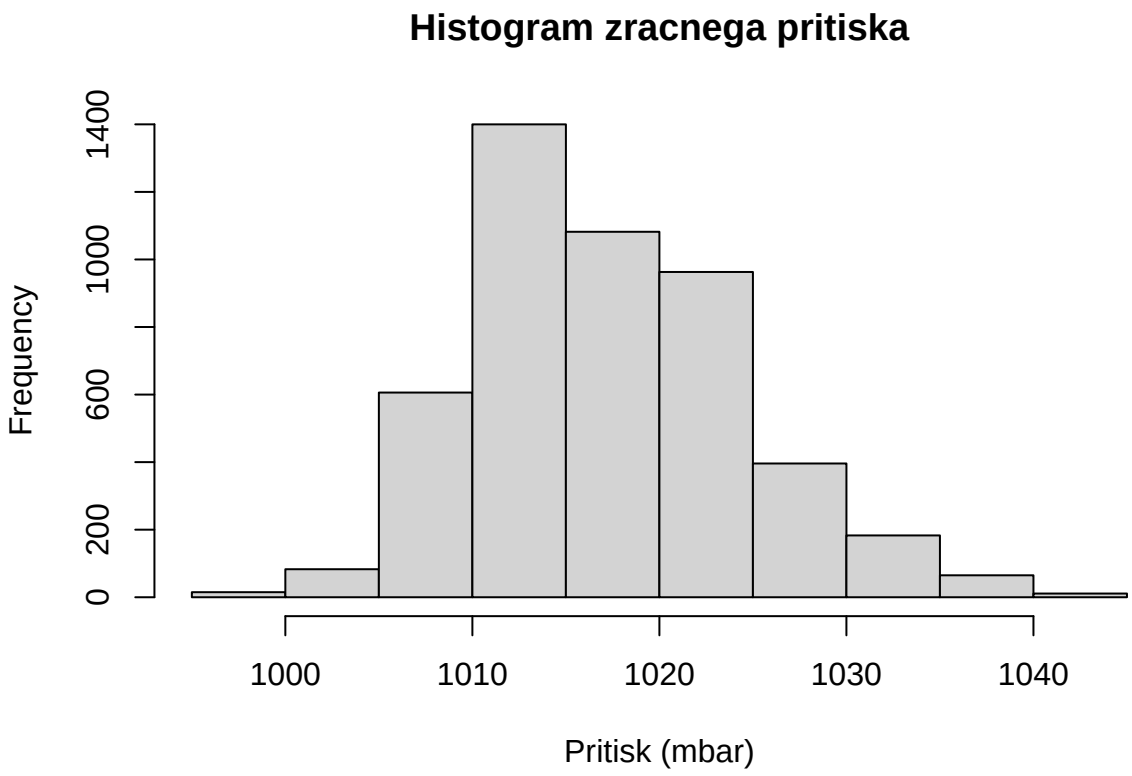

```
hist(allData$oblacnost, xlab="Oblacnost", main="Histogram stopnje pokritosti neba z oblaki")
```



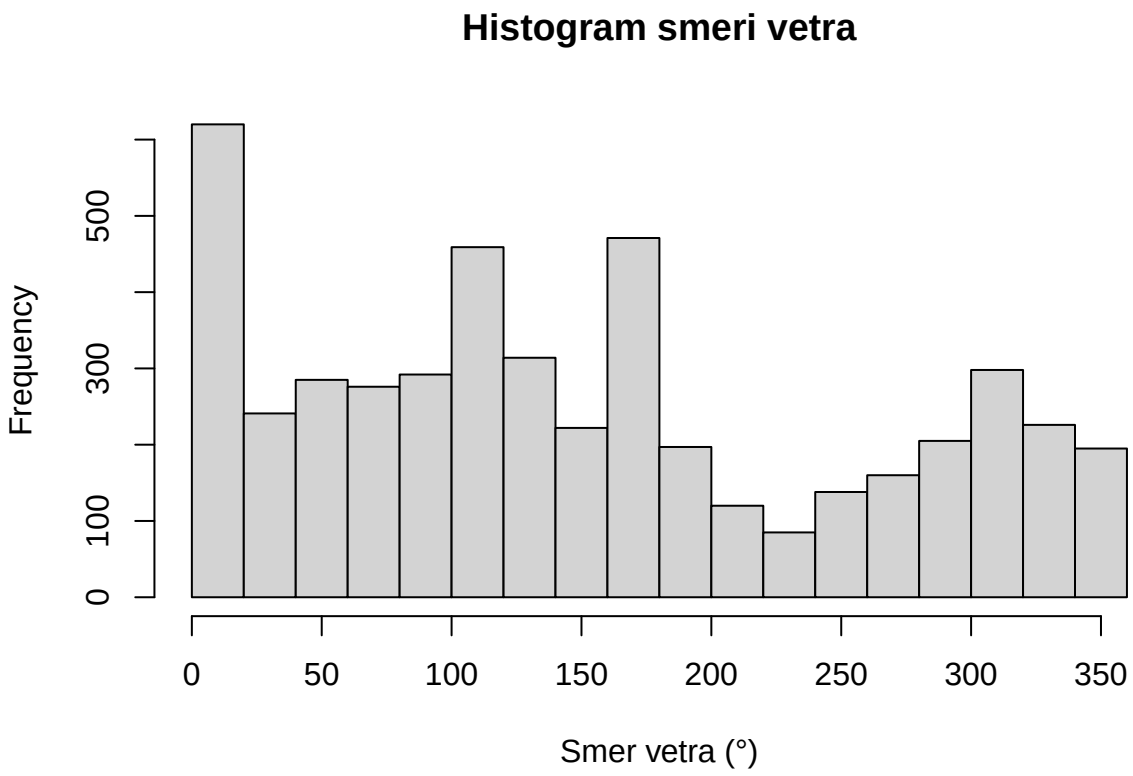

```
hist(allData$padavine, xlab="Padavine (mm)", main="Histogram kolicine padavin")
```



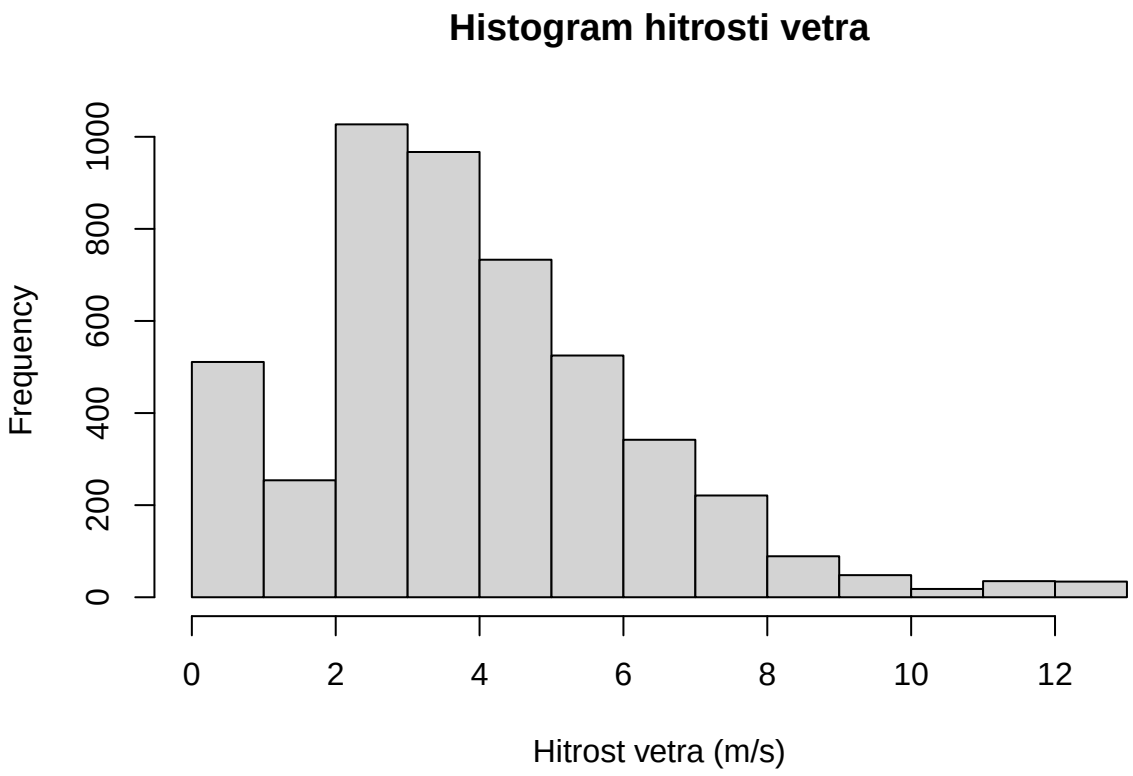

```
hist(allData$pritisk, xlab="Pritisk (mbar)", main="Histogram zracnega pritiska")
```




```
hist(allData$smer_vetra, xlab="Smer vetra (°)", main="Histogram smeri vetra")
```



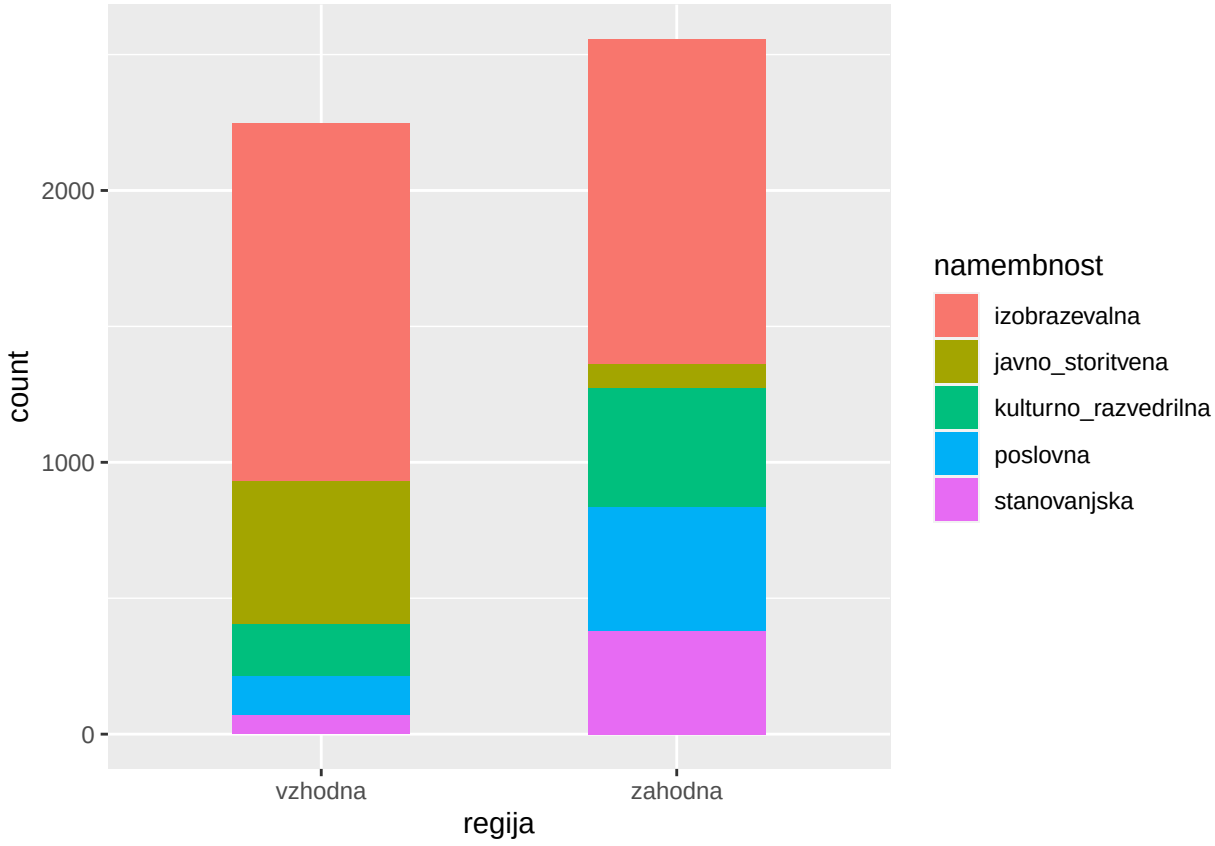

```
hist(allData$hitrost_vetra, xlab="Hitrost vetra (m/s)", main="Histogram hitrosti vetra")
```



Namembnost in regija

Namembnost stavb glede na regijo *Ugotovitve:* - priblizno polovica stavb sluzi izobrazevalnemu namenu - stavb z zahodno lego je malo vec kot stavb z zahodno lego - stavbe z vzhodno lego imajo za skoraj 13% vec stavb za izobrazevalne namene kot stavbe z zahodno lego

```
CalcEducationalPercentage <- function(regija)
{
  filtered <- allData[allData$regija == regija,]
  nrow(filtered[filtered$namembnost == "izobrazevalna",]) / nrow(filtered)
}
p <- ggplot(allData, aes(regija))
p + geom_bar(aes(fill=namembnost), width = 0.5)
```




```
paste("Odstotek izobrazevalnih stavb z vzhodno regijo", CalcEducationalPercentage("vzhodna"))
```

```
## [1] "Odstotek izobrazevalnih stavb z vzhodno regijo 0.586743772241993"
```

```
paste("Odstotek izobrazevalnih stavb z zahodno regijo", CalcEducationalPercentage("zahodna"))
```

```
## [1] "Odstotek izobrazevalnih stavb z zahodno regijo 0.467136150234742"
```

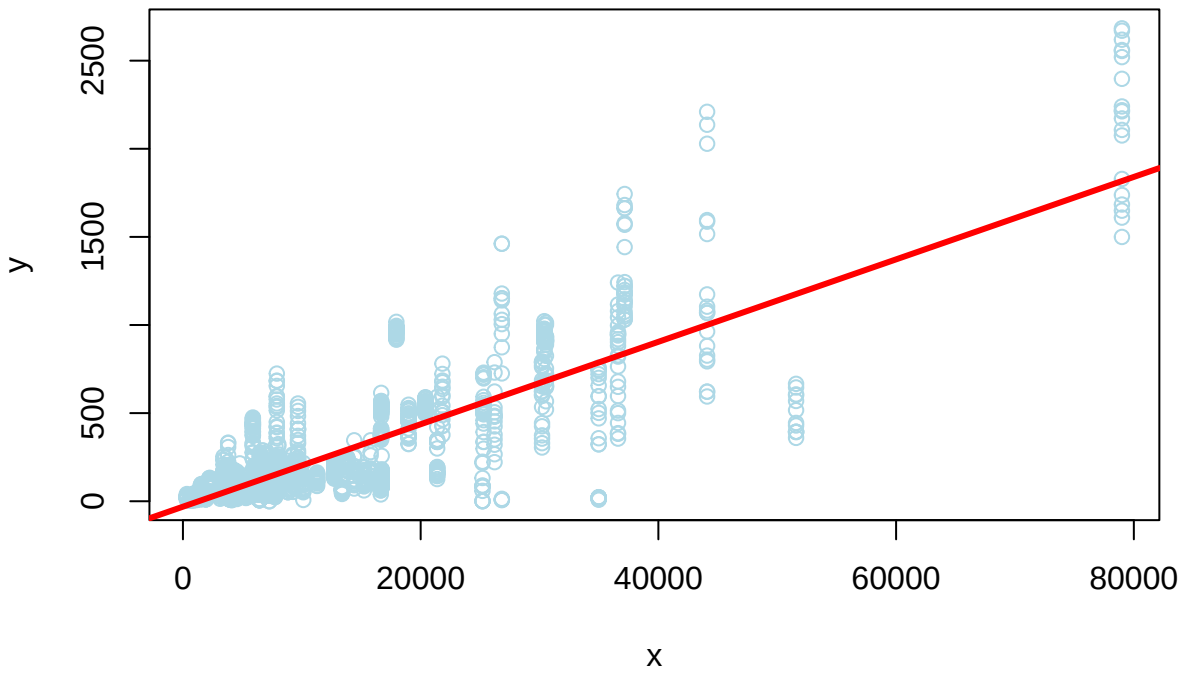
Soodvisnost atributov Pri nadalnji predikciji nam bo koristilo tudi nekaj intuicije o soodvisnosti med doloceni atributi.

Ze samo po sebi je logicno, da bodo nekateri atributi (npr. povrsina train <-> poraba energije) v vecji medsebojni odvisnosti, kot nekateri drugi atributi (npr. smer vetra <-> poraba energije);

Naso hipotezo lahko dodatno potrdimo z nekaj grafi, kjer prikazemo korelacijo med izbranimi pari atributov.

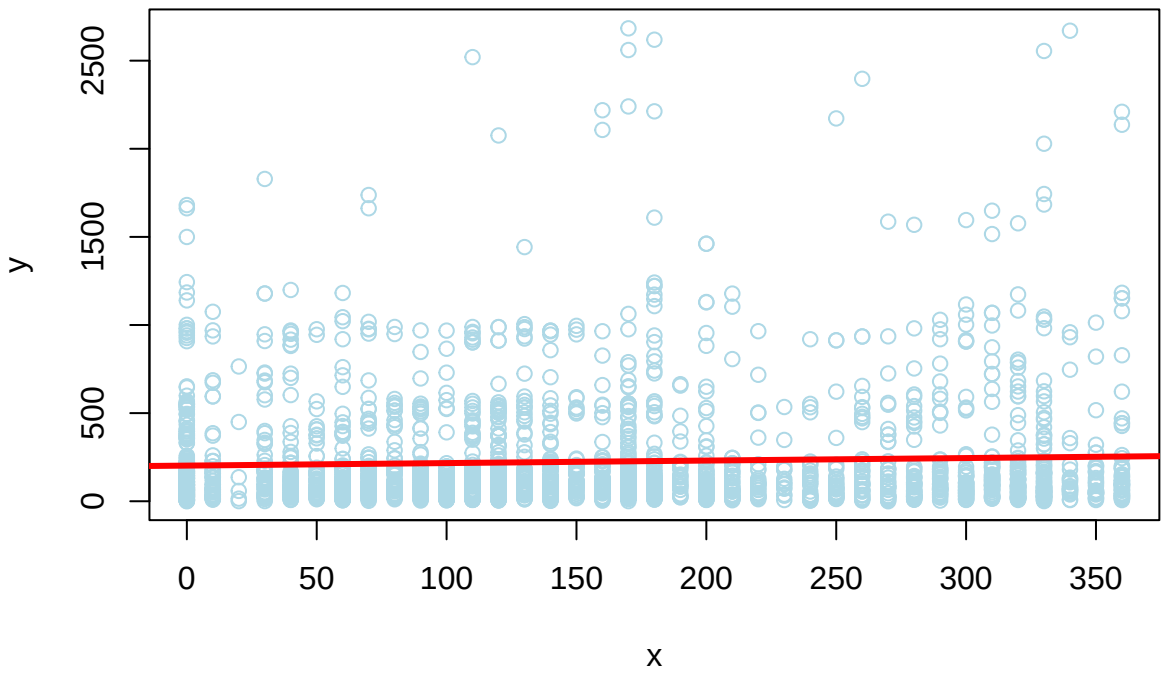
Pri porabi elektricne energije v odvisnosti z povrsino train vidimo, da obstaja jasen pozitiven trend.

```
x <- train$povrsina
y <- train$poraba
plot(x, y, col="lightblue")
abline(lm(y ~ x), col = "red", lwd = 3)
```



Medtem ko pri grafu porabe energije v odvisnosti od smeri vetra jasne korelacije ni.

```
x <- train$smer_vetra
y <- train$poraba
plot(x, y, col="lightblue")
abline(lm(y ~ x), col = "red", lwd = 3)
```



Najboljse bi bilo primerjati vse (numericne) atribute z vsemi drugimi atributi, ter prikazati medsebojne odvisnosti, tako bi pridobili visoko nivojski pogled na odvisnosti med atributi.

Za to vrstno vizualizacijo bomo uporabili dve zunanji knjiznici `ggplot2` in `ggcorrplot`, ki jih moramo prenesti in namestiti.

Ta graf nam izpise korelacijsko matriko, iz katere lahko razberemo korelacije med vsemi numericni atributi. Opazimo, da sta v najvecji medsebojni korelaciji res atributa `poraba` in `povrsina`.

```
data(train, package="mosaicData")
```

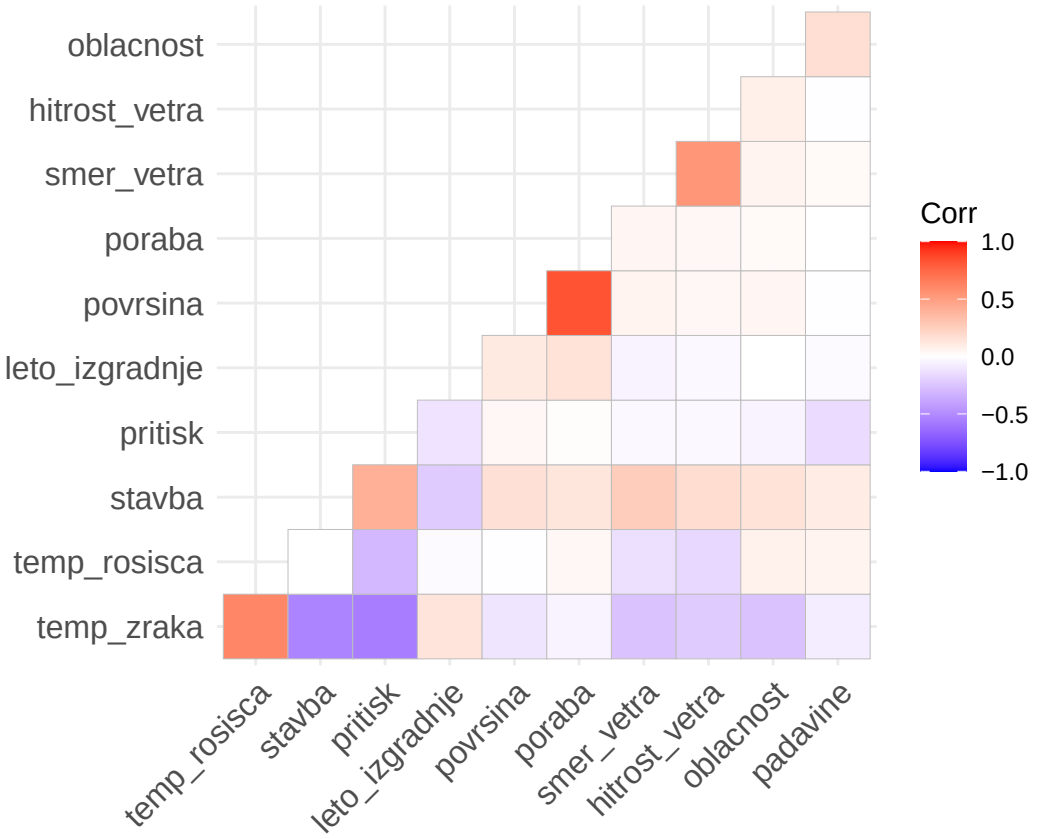
```
## Warning in data(train, package = "mosaicData"): data set 'train' not found
```

```
# izberemo samo numericne atribute
df <- dplyr::select_if(train, is.numeric)

# izracunamo korelacije z metodo cor
r <- cor(df, use="complete.obs")
round(r,2)
```

```
##                stavba povrsina leto_izgradnje temp_zraka temp_rosisca oblacnost
## stavba           1.00     0.16          -0.22      -0.53         0.00      0.15
## povrsina         0.16     1.00           0.11      -0.11        -0.01      0.05
## leto_izgradnje  -0.22     0.11           1.00       0.14        -0.02      0.00
## temp_zraka      -0.53    -0.11           0.14       1.00         0.61     -0.26
## temp_rosisca     0.00    -0.01          -0.02       0.61         1.00      0.07
```

```
## oblacnost        0.15     0.05           0.00      -0.26         0.07      1.00
## padavine         0.10    -0.01          -0.02      -0.08         0.06      0.17
## pritisk          0.41     0.04          -0.12      -0.56        -0.31     -0.05
## smer_vetra       0.26     0.06          -0.05      -0.26        -0.13      0.06
## hitrost_vetra    0.18     0.04          -0.03      -0.22        -0.17      0.08
## poraba           0.13     0.83           0.15      -0.05         0.04      0.03
##                padavine pritisk smer_vetra hitrost_vetra poraba
## stavba             0.10    0.41       0.26          0.18   0.13
## povrsina          -0.01    0.04       0.06          0.04   0.83
## leto_izgradnje    -0.02   -0.12      -0.05         -0.03   0.15
## temp_zraka        -0.08   -0.56      -0.26         -0.22  -0.05
## temp_rosisca       0.06   -0.31      -0.13         -0.17   0.04
## oblacnost          0.17   -0.05       0.06          0.08   0.03
## padavine           1.00   -0.15       0.03         -0.01   0.00
## pritisk           -0.15    1.00      -0.03         -0.03   0.01
## smer_vetra         0.03   -0.03       1.00          0.54   0.05
## hitrost_vetra     -0.01   -0.03       0.54          1.00   0.04
## poraba             0.00    0.01       0.05          0.04   1.00
```

```
ggcorrplot(r,
           hc.order=T, # uredi po korelaciji
           type="lower") # prikazi samo v spodnjem trikotniku
```



Priprava atributov

Pomozne metode

Sedaj bomo poskusali izboljsati kvaliteto posameznih atributov. Pri tem bomo uporabili nekaj pomoznih metod za evaluacijo.

Metoda `evalClassFeatures` bo evaluirala podatke z dano formulo z vsemi definiranimi ocenami za klasifikacijske probleme. Prav tako bo metoda `evalRegrFeatures` evaluirala atribute z definiranimi ocenami za regresijske probleme.

```
evalFeatures <- function (formula, data, estimators)
{
  for (estimator in estimators) {
    score = attrEval(formula, data, estimator);

    cat(paste(estimator, "\n"))
    print(sort(score, decreasing=T))
    cat("\n\n")
  }
}

evalClassFeatures <- function (formula, data)
{
  shortSighted <- list("InfGain", "GainRatio", "Gini", "MDL")
  nonShortSighted <- list("Relief", "ReliefFequalK", "ReliefFexpRank")
  estimators <- c(shortSighted, nonShortSighted)
  evalFeatures(formula, data, estimators)
}

evalRegrFeatures <- function (formula, data)
{
  estimators <- list("MSEofMean", "RReliefFexpRank")
  evalFeatures(formula, data, estimators)
}
```

Izboljsava mnozice atributov

Poskusimo izboljsati prvotno podatkovno mnozico z dodajanjem / odstranjevanjem atributov. Namen je najti cim manjso mnozico atributov ki maksimizira kvaliteto modela.

```
# atributi za klasifikacijski problem
classSetBase <- list(train=train, test=test)
classSetExt <- list(train=train, test=test)

ExtendClassSet <- function (set)
{
  set$oblacnost <- log1p(set$oblacnost)
  set$poraba <- log1p(set$poraba)
  set$povrsina <- log1p(set$povrsina)
  set$datum <- NULL
  set
}
```

```
classSetExt$train <- ExtendClassSet(classSetExt$train)
classSetExt$test <- ExtendClassSet(classSetExt$test)

# atributi za regresijski problem
regSetBase <- list(train=train, test=test)
regSetExt <- list(train=train, test=test)

ExtendRegSet <- function (set)
{
  set$letni_cas <- as.factor(ToSeason(set$datum))
  set$mesec <- as.factor(ToMonth(set$datum))
  set$zima <- as.factor(IsWinter(set$datum))
  set$vikend <- as.factor(IsWeekend(set$datum))
  set$pritisk <- log1p(set$pritisk)
  set$hitrost_vetra <- log1p(set$hitrost_vetra)

  set$datum <- NULL
  set$stavba <- NULL
  set$temp_rosisca <- NULL
  set$padavine <- NULL
  set$smer_vetra <- NULL

  set$namembnost <- NULL
  set$temp_zraka <- NULL

  set$oblacnost <- log1p(set$oblacnost)
  set$poraba <- log1p(set$poraba)
  set$povrsina <- log1p(set$povrsina)

  set
}

regSetExt$train <- ExtendRegSet(regSetExt$train)
regSetExt$test <- ExtendRegSet(regSetExt$test)
```

Evalvacija atributov

Poglejmo si vse ocene za prvotni mnozici atributov:

```
evalClassFeatures(namembnost ~ ., classSetBase$train)
```

```
## InfGain
##       povrsina         regija         stavba leto_izgradnje     temp_zraka
##    0.250101491    0.190977025    0.190977025    0.162119934    0.057770952
##         poraba        pritisk     smer_vetra      oblacnost   temp_rosisca
##    0.055103230    0.036248185    0.035262339    0.016920944    0.008302622
##       padavine  hitrost_vetra          datum
##    0.008242992    0.007069367    0.006675803
##
##
## GainRatio
##         stavba       povrsina         poraba         regija leto_izgradnje
```

```
##     0.38984167     0.36987727     0.34551681     0.19171400     0.16259839
##   temp_rosisca        pritisk     temp_zraka  hitrost_vetra       padavine
##     0.09561334     0.08004080     0.07467698     0.07173461     0.06723994
##          datum     smer_vetra      oblacnost
##     0.06017320     0.03583389     0.03192463
##
##
## Gini
##       povrsina leto_izgradnje         regija         stavba         poraba
##    0.074979035    0.046192178    0.028507919    0.028507919    0.019063586
##     temp_zraka        pritisk     smer_vetra      oblacnost          datum
##    0.009468243    0.007142769    0.005958509    0.003026184    0.001867707
##  hitrost_vetra   temp_rosisca       padavine
##    0.001713742    0.001514270    0.001168672
##
##
## MDL
##       povrsina         regija         stavba leto_izgradnje     temp_zraka
##    0.242232153    0.182346126    0.182346126    0.155104530    0.050977478
##         poraba        pritisk     smer_vetra      oblacnost   temp_rosisca
##    0.048718048    0.029652532    0.028674621    0.011039282    0.003219395
##       padavine          datum  hitrost_vetra
##    0.003206210    0.002645080    0.001615114
##
##
## Relief
## leto_izgradnje         stavba       povrsina         poraba         regija
##   0.3391643149   0.2109360604   0.2093195908   0.1433817086   0.0008291874
##       padavine     temp_zraka      oblacnost        pritisk   temp_rosisca
##  -0.0009552627  -0.0693258882  -0.0737976783  -0.0824222392  -0.0835101270
##          datum  hitrost_vetra     smer_vetra
##  -0.0840583990  -0.1075870647  -0.1216141515
##
##
## ReliefFequalK
## leto_izgradnje         stavba       povrsina         regija         poraba
##   0.3748615641   0.3417402901   0.2220847867   0.1771195457   0.1239372049
##     temp_zraka     smer_vetra        pritisk          datum   temp_rosisca
##   0.0619396491   0.0467935444   0.0355393489   0.0239461498   0.0205776214
##      oblacnost  hitrost_vetra       padavine
##   0.0162916729   0.0072822804   0.0007959659
##
##
## ReliefFexpRank
## leto_izgradnje         stavba       povrsina         regija         poraba
##   0.3594335021   0.3329030457   0.2105280298   0.2029986303   0.1138472395
##     temp_zraka     smer_vetra        pritisk          datum   temp_rosisca
##   0.0610981372   0.0302705419   0.0276094367   0.0181816406   0.0170283848
##      oblacnost       padavine  hitrost_vetra
##   0.0100941989  -0.0004917695  -0.0062426174
```

```
evalRegrFeatures(poraba ~ ., regSetBase$train)
```

```
## MSEofMean
```

```
##       povrsina leto_izgradnje         stavba     namembnost         regija
##      -50576.57      -95339.86      -98047.76     -101243.82     -101924.27
##   temp_rosisca     temp_zraka      oblacnost          datum     smer_vetra
##     -102319.77     -102331.27     -102339.53     -102653.93     -103055.20
##        pritisk  hitrost_vetra       padavine
##     -103086.97     -103121.73     -103231.44
##
##
## RReliefFexpRank
##       povrsina leto_izgradnje     namembnost         stavba       padavine
##   0.4413738717   0.1890857951   0.1254954539   0.1065823219   0.0013918771
##         regija        pritisk     temp_zraka      oblacnost   temp_rosisca
##  -0.0001330178  -0.0835504967  -0.0873438431  -0.0880498899  -0.1008200969
##  hitrost_vetra          datum     smer_vetra
##  -0.1086545339  -0.1188974223  -0.1253054663
```

Ponovno evaluiramo atribute za popravljeni mnozici atributov:

```
evalClassFeatures(namembnost ~ ., classSetExt$train)
```

```
## InfGain
##       povrsina         regija         stavba leto_izgradnje     temp_zraka
##    0.250101491    0.190977025    0.190977025    0.162119934    0.057770952
##         poraba        pritisk     smer_vetra      oblacnost   temp_rosisca
##    0.055103230    0.036248185    0.035262339    0.016920944    0.008302622
##       padavine  hitrost_vetra
##    0.008242992    0.007069367
##
##
## GainRatio
##         stavba       povrsina         poraba         regija leto_izgradnje
##     0.38984167     0.36987727     0.34551681     0.19171400     0.16259839
##   temp_rosisca        pritisk     temp_zraka  hitrost_vetra       padavine
##     0.09561334     0.08004080     0.07467698     0.07173461     0.06723994
##     smer_vetra      oblacnost
##     0.03583389     0.03192463
##
##
## Gini
##       povrsina leto_izgradnje         regija         stavba         poraba
##    0.074979035    0.046192178    0.028507919    0.028507919    0.019063586
##     temp_zraka        pritisk     smer_vetra      oblacnost  hitrost_vetra
##    0.009468243    0.007142769    0.005958509    0.003026184    0.001713742
##   temp_rosisca       padavine
##    0.001514270    0.001168672
##
##
## MDL
##       povrsina         regija         stavba leto_izgradnje     temp_zraka
##    0.242232153    0.182346126    0.182346126    0.155104530    0.050977478
##         poraba        pritisk     smer_vetra      oblacnost   temp_rosisca
##    0.048718048    0.029652532    0.028674621    0.011039282    0.003219395
##       padavine  hitrost_vetra
```

```
##    0.003206210    0.001615114
##
##
## Relief
## leto_izgradnje       povrsina         stavba         poraba         regija
##    0.464638309    0.418794058    0.330566496    0.304080488    0.002487562
##       padavine        pritisk      oblacnost     temp_zraka   temp_rosisca
##   -0.004744795   -0.117071963   -0.117369800   -0.120678701   -0.138032933
##  hitrost_vetra     smer_vetra
##   -0.140582928   -0.164532277
##
##
## ReliefFequalK
## leto_izgradnje         stavba       povrsina         poraba         regija
##   0.4752112064   0.4313454768   0.4216174488   0.2715857680   0.1655832174
##     temp_zraka       padavine        pritisk     smer_vetra   temp_rosisca
##   0.0119328704   0.0010015730  -0.0004986693  -0.0035302509  -0.0202235950
##      oblacnost  hitrost_vetra
##  -0.0249773120  -0.0339796996
##
##
## ReliefFexpRank
## leto_izgradnje         stavba       povrsina         poraba         regija
##    0.446276723    0.412888415    0.403565235    0.253276937    0.189818904
##     temp_zraka       padavine        pritisk     smer_vetra   temp_rosisca
##    0.010362965   -0.001088266   -0.006924251   -0.008219028   -0.018061265
##      oblacnost  hitrost_vetra
##   -0.024973839   -0.043946605
```

```
evalRegrFeatures(poraba ~ ., regSetExt$train)
```

```
## MSEofMean
##       povrsina leto_izgradnje          mesec      letni_cas         regija
##     -0.9014081     -1.3289248     -1.4152002     -1.4253287     -1.4302387
##  hitrost_vetra      oblacnost        pritisk           zima         vikend
##     -1.4328816     -1.4329784     -1.4335648     -1.4350186     -1.4357428
##
##
## RReliefFexpRank
##       povrsina leto_izgradnje         regija           zima      letni_cas
##    0.334633512    0.160609243    0.009354531   -0.002923525   -0.024754519
##         vikend        pritisk      oblacnost          mesec  hitrost_vetra
##   -0.037839612   -0.064911551   -0.088202775   -0.095756261   -0.101081968
```

Klasifikacija

Vecinski klasifikator

Vecinski klasifikator uvrsti vsak primer v razred ki se najveckrat pojavi. Ta klasifikator bo predstavljal spodnjo mejo kvalitete ucnih modelov.

```
# najveckrat se ponovi "izobrazevalna" namembnost
sum(test$namembnost == "izobrazevalna") / length(test$namembnost)
```

```
## [1] 0.4824415
```

Odlocitveno drevo

```
# osnovna mnozica atributov
dtBase <- rpart(namembnost ~ pritisk, data=classSetBase$train)
EvaluateClassModel(dtBase, classSetBase$train, classSetBase$test)
```

	brier	ca	infGain
izobrazevalna	0.7105376	0.4824415	0

```
# popravljena mnozica atributov
dtExt <- rpart(namembnost ~ ., data=classSetExt$train)
EvaluateClassModel(dtExt, classSetExt$train, classSetExt$test)
```

brier	ca	infGain
0.9506341	0.5196488	0.5824648

Odlocitveno drevo z rezanjem

Izberemo vrednost parametra cp, ki ustreza minimalni napaki internega presnega preverjanja.

```
dtBase <- rpart(namembnost ~ ., data=classSetBase$train, cp=0)
cpTab <- printcp(dtBase)
```

```
##
## Classification tree:
## rpart(formula = namembnost ~ ., data = classSetBase$train, cp = 0)
##
## Variables actually used in tree construction:
## [1] leto_izgradnje poraba         povrsina       regija         stavba
##
## Root node error: 1053/2412 = 0.43657
##
## n= 2412
##
##          CP nsplit rel error   xerror      xstd
## 1  0.104463      0  1.000000 1.000000 0.0231317
## 2  0.093067      1  0.895537 0.889839 0.0227326
## 3  0.063628      2  0.802469 0.802469 0.0222508
## 4  0.039411      3  0.738841 0.754036 0.0219171
## 5  0.037037      5  0.660019 0.515670 0.0194799
## 6  0.036087      6  0.622982 0.474834 0.0189065
```

```
## 7  0.032605      7  0.586895 0.427350 0.0181693
## 8  0.025641     14  0.301994 0.334283 0.0164660
## 9  0.024691     15  0.276353 0.297246 0.0156733
## 10 0.018044     16  0.251662 0.254511 0.0146576
## 11 0.017569     17  0.233618 0.169991 0.0122251
## 12 0.017094     22  0.116809 0.163343 0.0120025
## 13 0.012979     23  0.099715 0.105413 0.0097724
## 14 0.011871     27  0.045584 0.075024 0.0083014
## 15 0.008547     29  0.021842 0.036087 0.0058078
## 16 0.000000     30  0.013295 0.011396 0.0032816
```

```
row <- which.min(cpTab[,"xerror"])
th <- mean(c(cpTab[row, "CP"], cpTab[row-1, "CP"]))
dtBase <- prune(dtBase, cp=th)
EvaluateClassModel(dtBase, classSetBase$train, classSetBase$test)
```

brier	ca	infGain
0.9577598	0.5196488	0.5830208

```
dtExt <- rpart(namembnost ~ ., data=classSetExt$train, cp=0)
cpTab <- printcp(dtExt)
```

```
##
## Classification tree:
## rpart(formula = namembnost ~ ., data = classSetExt$train, cp = 0)
##
## Variables actually used in tree construction:
## [1] leto_izgradnje poraba         povrsina       regija         stavba
##
## Root node error: 1053/2412 = 0.43657
##
## n= 2412
##
##          CP nsplit rel error   xerror      xstd
## 1  0.104463      0  1.000000 1.000000 0.0231317
## 2  0.093067      1  0.895537 0.884141 0.0227057
## 3  0.063628      2  0.802469 0.806268 0.0222749
## 4  0.039411      3  0.738841 0.747388 0.0218674
## 5  0.037037      5  0.660019 0.547009 0.0198852
## 6  0.036087      6  0.622982 0.520418 0.0195432
## 7  0.032605      7  0.586895 0.473884 0.0188926
## 8  0.025641     14  0.301994 0.342830 0.0166387
## 9  0.024691     15  0.276353 0.306743 0.0158838
## 10 0.018044     16  0.251662 0.272555 0.0151009
## 11 0.017569     17  0.233618 0.190883 0.0128907
## 12 0.017094     22  0.116809 0.152896 0.0116408
## 13 0.012979     23  0.099715 0.105413 0.0097724
## 14 0.011871     27  0.045584 0.048433 0.0067099
## 15 0.008547     29  0.021842 0.031339 0.0054180
## 16 0.000000     30  0.013295 0.017094 0.0040140
```

```
row <- which.min(cpTab[,"xerror"])
th <- mean(c(cpTab[row, "CP"], cpTab[row-1, "CP"]))
dtExt <- prune(dtExt, cp=th)
EvaluateClassModel(dtExt, classSetExt$train, classSetExt$test)
```

brier	ca	infGain
0.9577598	0.5196488	0.5830208

Naivni Bayes

```
nbBase <- CoreModel(namembnost ~ ., data=classSetBase$train, model="bayes")
EvaluateClassModel(nbBase, classSetBase$train, classSetBase$test)
```

	brier	ca	infGain
izobrazevalna	0.7902098	0.4243311	0.4519204

```
nbExt <- CoreModel(namembnost ~ ., data=classSetExt$train, model="bayes")
EvaluateClassModel(nbExt, classSetExt$train, classSetExt$test)
```

	brier	ca	infGain
izobrazevalna	0.7852299	0.4276756	0.4578558

K-bliznjih sosedov

```
knnBase <- CoreModel(namembnost ~ ., data=classSetBase$train, model="knn", kInNN=5)
EvaluateClassModel(knnBase, classSetBase$train, classSetBase$test)
```

	brier	ca	infGain
izobrazevalna	0.7898662	0.4423077	0.3652095

```
knnExt <- CoreModel(namembnost ~ ., data=classSetExt$train, model="knn", kInNN=5)
EvaluateClassModel(knnExt, classSetExt$train, classSetExt$test)
```

	brier	ca	infGain
izobrazevalna	0.7610033	0.4849498	0.4564758

Nakljucni gozd

```
rfBase <- randomForest(namembnost ~ ., data=classSetBase$train)
EvaluateClassModel(rfBase, classSetBase$train, classSetBase$test)
```

brier	ca	infGain
0.6562056	0.5656355	0.6030941

```
rfExt <- randomForest(namembnost ~ ., data=classSetExt$train)
EvaluateClassModel(rfExt, classSetExt$train, classSetExt$test)
```

brier	ca	infGain
0.6520244	0.5681438	0.6208229

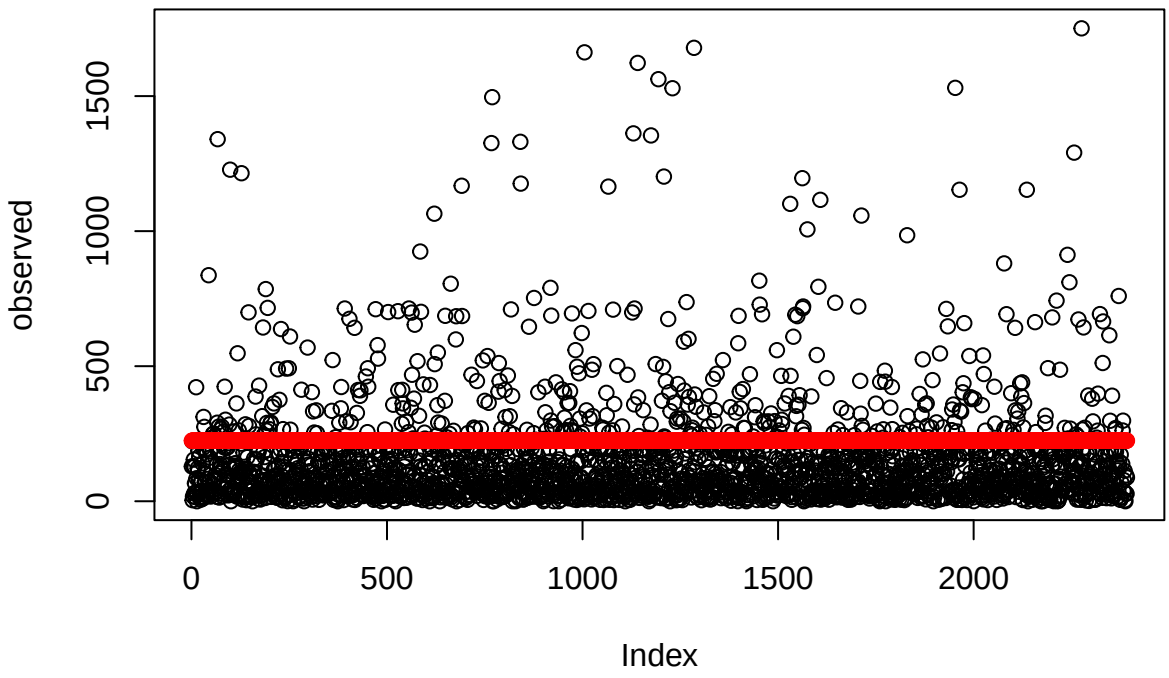
Regresija

Trivialni model

Trivialni model vedno vraca povprecno vrednost ciljne spremenljivke, glede na vse ucne primere. Ta model bo predstavljal spodnjo mejo kvalitete ucnih modelov.

```
meanValue <- mean(regSetBase$train$poraba)
predicted <- rep(meanValue, nrow(regSetBase$test))
observed <- regSetBase$test$poraba

EvaluateTrivialRegModel(observed, predicted)
```

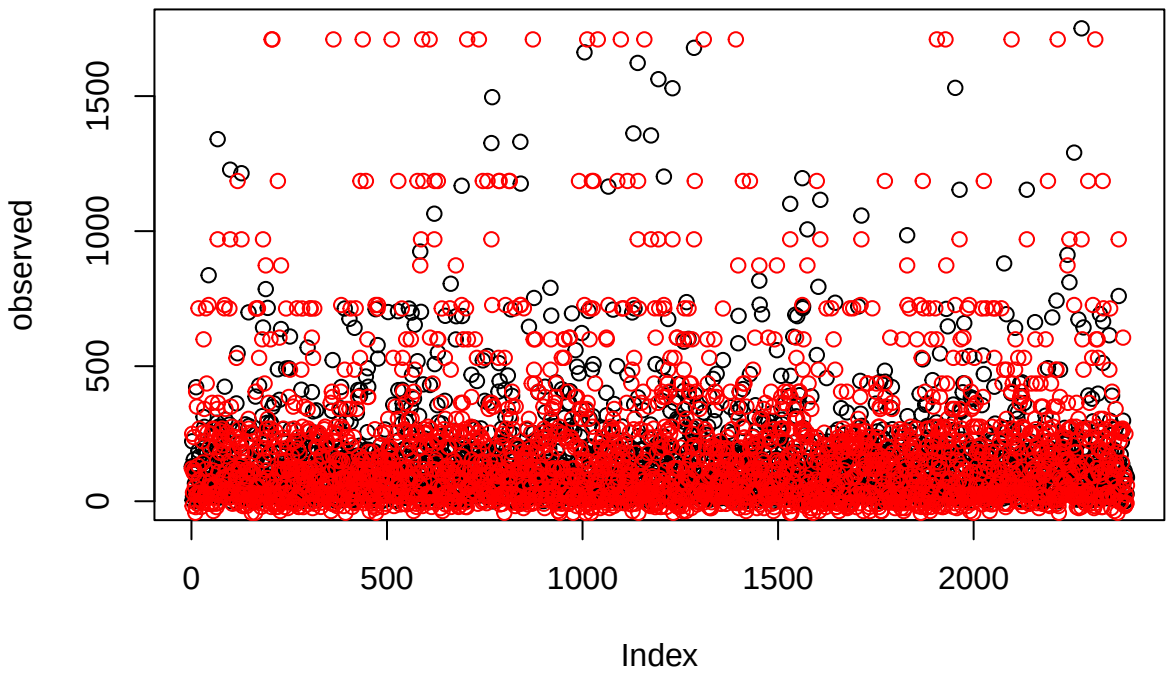
```
## [1] "Srednja absolutna napaka: 156.129714419125"
## [1] "Srednja kvadratna napaka: 43242.2118842803"
## [1] "Relativna srednja absolutna napaka: 1"
## [1] "Relativna srednja kvadratna napaka: 1"
```



Linearna regresija

```
# osnovna mnozica atributov
lmBase <- lm(poraba ~ povrsina + leto_izgradnje, regSetBase$train)
EvaluateRegBaseModel(lmBase, regSetBase$train, regSetBase$test)
```

```
## [1] "Srednja absolutna napaka: 107.298030128528"
## [1] "Srednja kvadratna napaka: 47465.6068612006"
## [1] "Relativna srednja absolutna napaka: 0.687236446487628"
## [1] "Relativna srednja kvadratna napaka: 1.09766833824834"
```

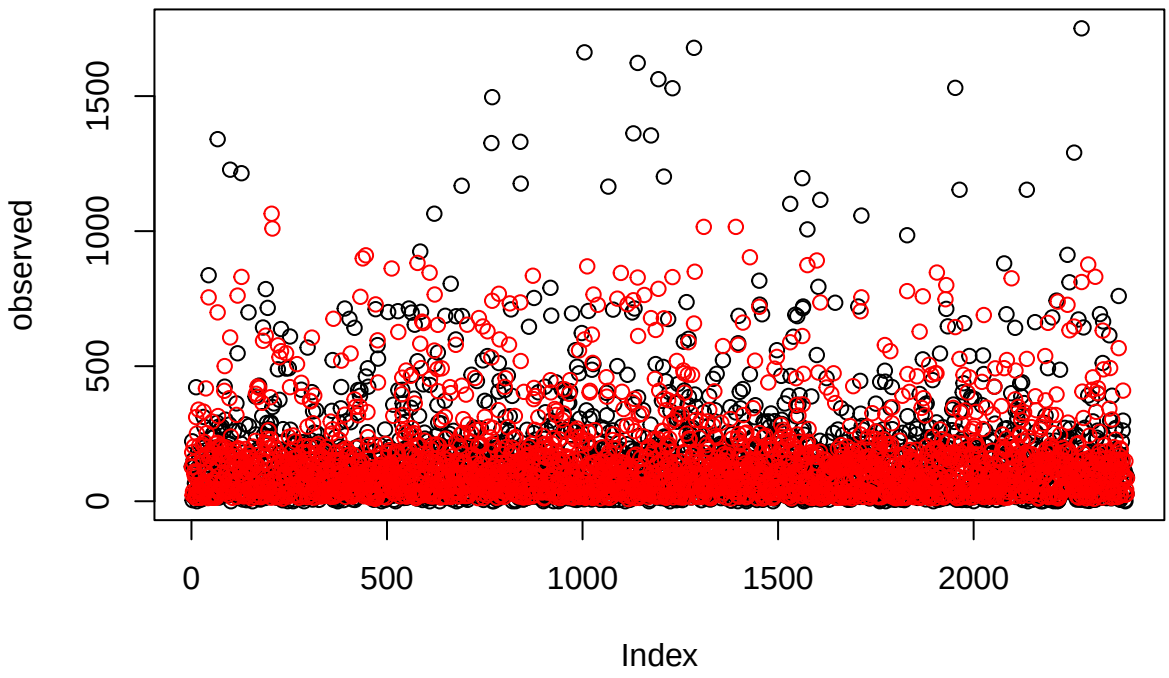


| mae| mse| rmae| rmse| |——-:|———:|———-:|———:| | 107.298| 47465.61| 0.6872364| 1.097668|

```
# popravljena mnozica atributov
lmExt <- lm(poraba ~ ., regSetExt$train)
EvaluateRegExtModel(lmExt, regSetExt$train, regSetExt$test)
```

```
## Warning in predict.lm(model, test): prediction from a rank-deficient fit may be
## misleading
```

```
## [1] "Srednja absolutna napaka: 74.0012295363632"
## [1] "Srednja kvadratna napaka: 19916.7077316699"
## [1] "Relativna srednja absolutna napaka: 0.495351783374952"
## [1] "Relativna srednja kvadratna napaka: 0.328755734403337"
```

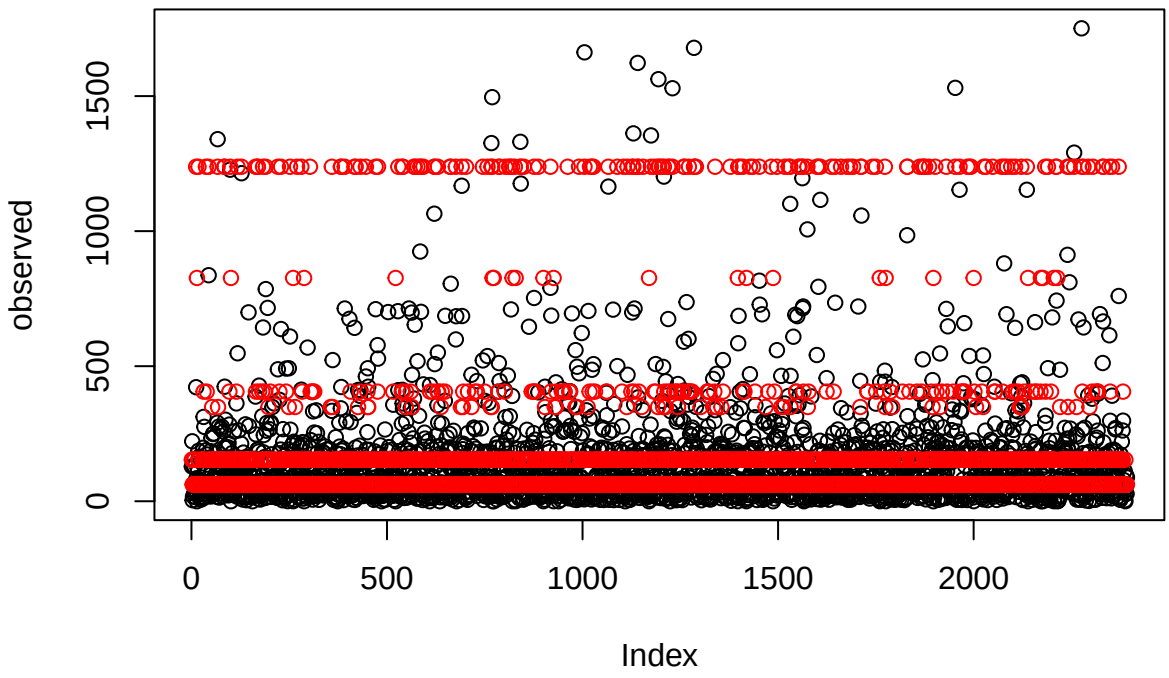


| mae| mse| rmae| rmse| |———:|———:|———:|———:| | 74.00123| 19916.71| 0.4953518| 0.3287557|

```
# osnovna mnozica atributov
baseModel <- rpart(poraba ~ ., data=regSetBase$train)
EvaluateRegBaseModel(baseModel, regSetBase$train, regSetBase$test)
```

Regresijsko drevo

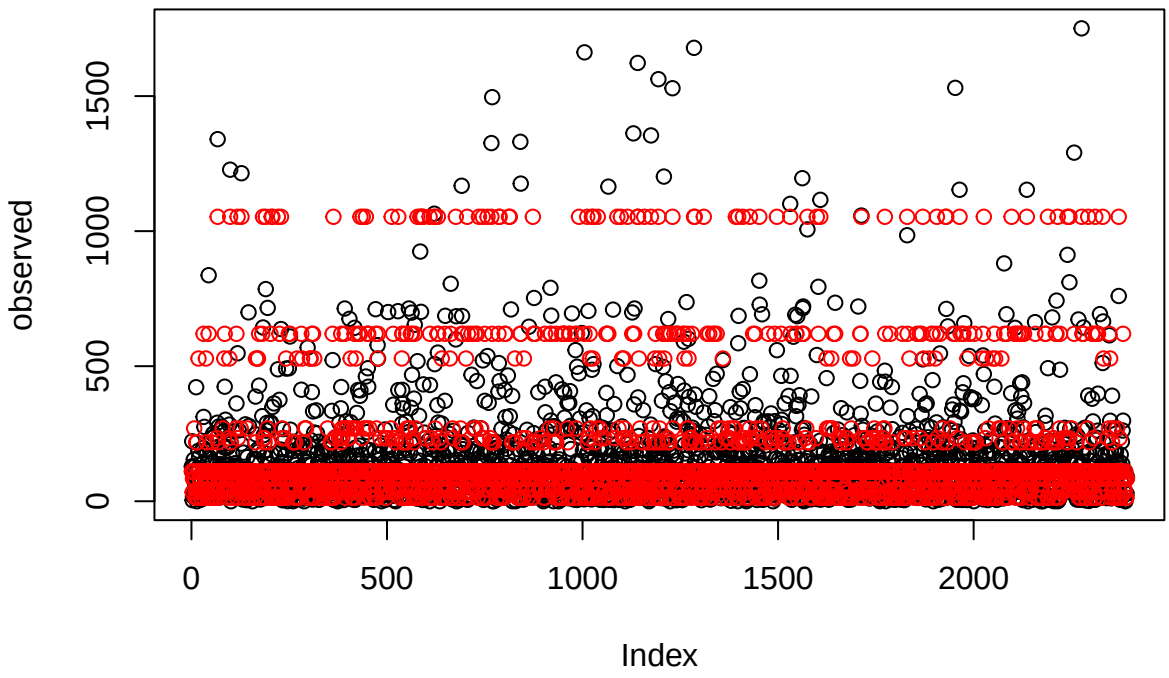
```
## [1] "Srednja absolutna napaka: 131.411079679798"
## [1] "Srednja kvadratna napaka: 90642.5989139421"
## [1] "Relativna srednja absolutna napaka: 0.841678857664654"
## [1] "Relativna srednja kvadratna napaka: 2.09616009367211"
```



| mae| mse| rmae| rmse| |———:|——-:|———:|——-:| | 131.4111| 90642.6| 0.8416789| 2.09616|

```
# popravljena mnozica atributov
extModel <- rpart(poraba ~ ., data=regSetExt$train)
EvaluateRegExtModel(extModel, regSetExt$train, regSetExt$test)
```

```
## [1] "Srednja absolutna napaka: 97.6424658945491"
## [1] "Srednja kvadratna napaka: 31421.5743820927"
## [1] "Relativna srednja absolutna napaka: 0.653602243057674"
## [1] "Relativna srednja kvadratna napaka: 0.518661161335819"
```

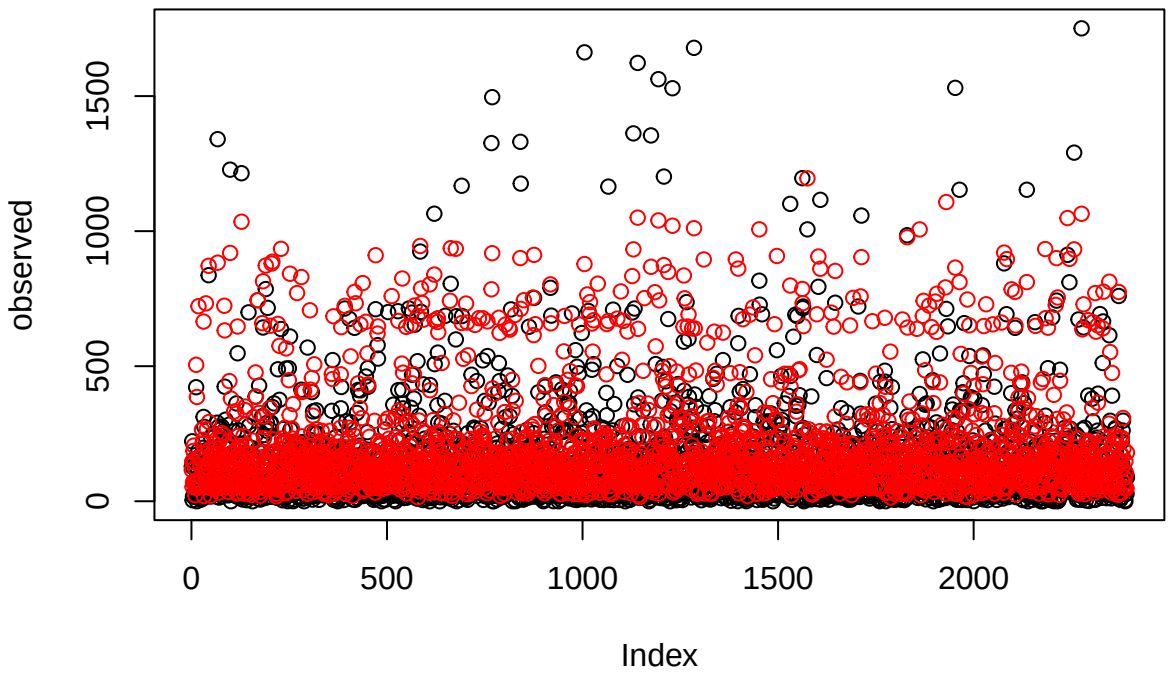


| mae| mse| rmae| rmse| |———:|———:|———:|———:| | 97.64247| 31421.57| 0.6536022| 0.5186612|

Nakljucni gozd

```
# osnovna mnozica atributov
baseModel <- randomForest(poraba ~ ., data=regSetBase$train)
EvaluateRegBaseModel(baseModel, regSetBase$train, regSetBase$test)
```

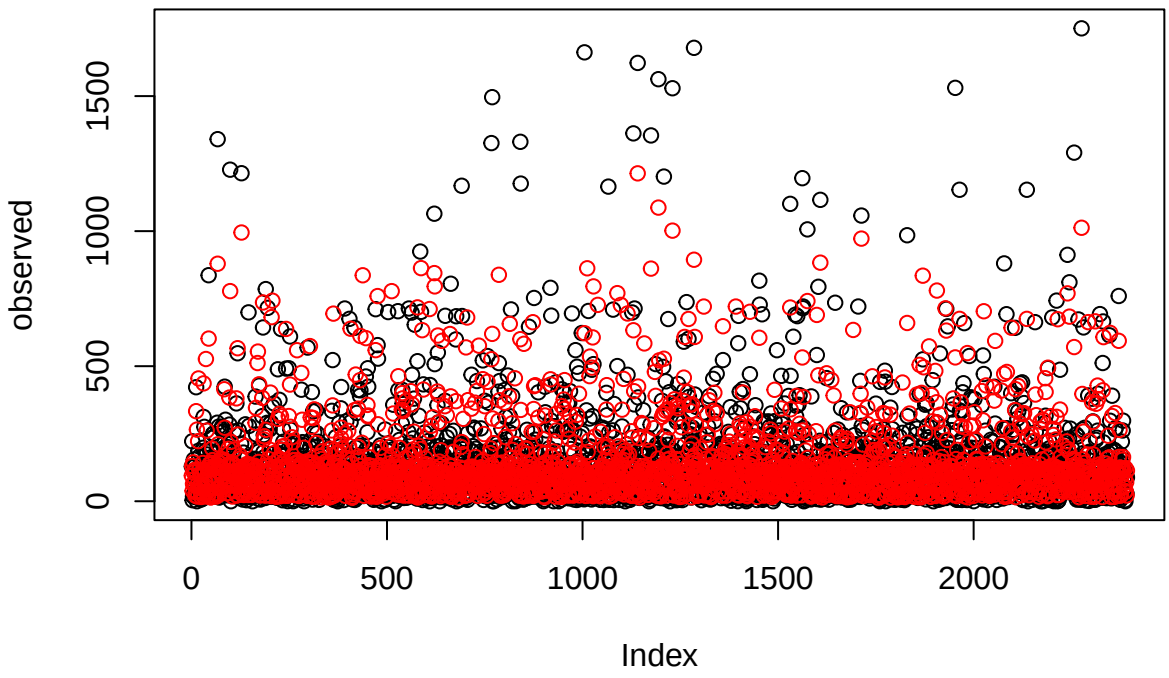
```
## [1] "Srednja absolutna napaka: 96.5330809300327"
## [1] "Srednja kvadratna napaka: 25582.2719345468"
## [1] "Relativna srednja absolutna napaka: 0.618287692955694"
## [1] "Relativna srednja kvadratna napaka: 0.591604148349465"
```



| mae| mse| rmae| rmse| |———:|———:|————:|————:| | 96.53308| 25582.27| 0.6182877| 0.5916041|

```
# popravljena mnozica atributov
extModel <- randomForest(poraba ~ ., data=regSetExt$train)
EvaluateRegExtModel(extModel, regSetExt$train, regSetExt$test)
```

```
## [1] "Srednja absolutna napaka: 78.3879489789711"
## [1] "Srednja kvadratna napaka: 21329.4179641908"
## [1] "Relativna srednja absolutna napaka: 0.524715745469577"
## [1] "Relativna srednja kvadratna napaka: 0.35207467828948"
```



| mae| mse| rmae| rmse| |———:|———:|———:|———:| | 78.38795| 21329.42| 0.5247157| 0.3520747|

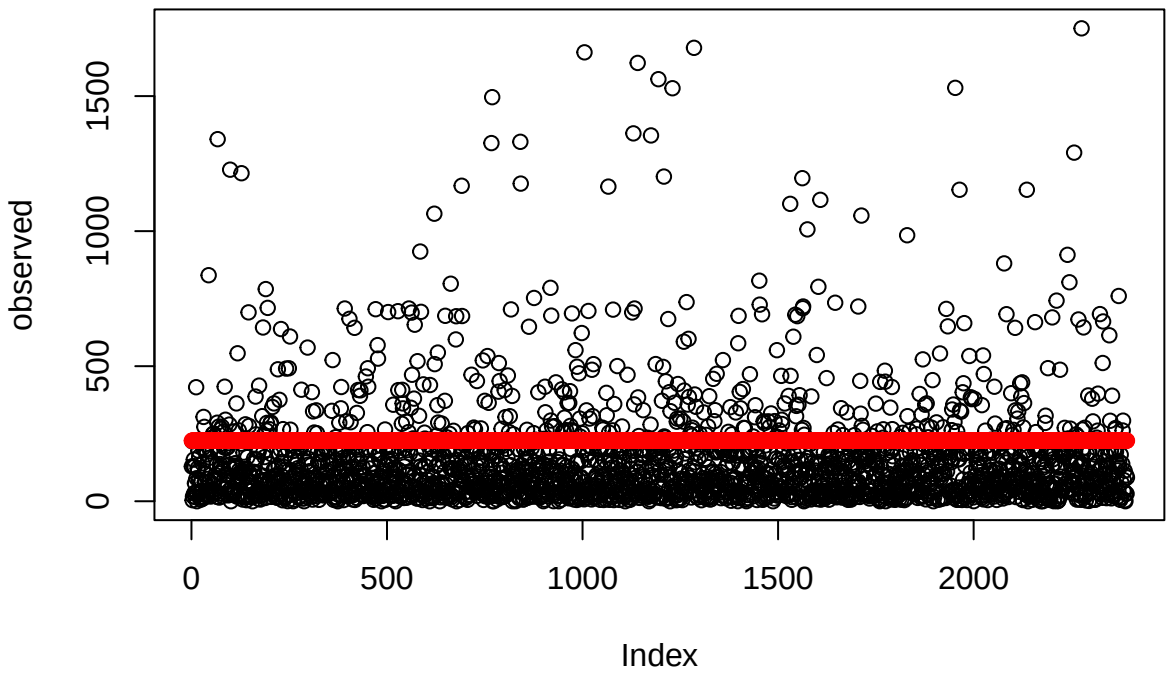
Nevronske mreze

```
# osnovna mnozica atributov
baseModel <- nnet(poraba ~ ., regSetBase$train, size=5, decay=0.001, maxit=10000, linout=T)
```

```
## # weights:  91
## initial  value 370994700.372666
## final  value 249206001.413665
## converged
```

```
EvaluateRegBaseModel(baseModel, regSetBase$train, regSetBase$test)
```

```
## [1] "Srednja absolutna napaka: 156.129621657455"
## [1] "Srednja kvadratna napaka: 43242.1913023466"
## [1] "Relativna srednja absolutna napaka: 0.999999405867931"
## [1] "Relativna srednja kvadratna napaka: 0.999999524031428"
```



| mae| mse| rmae| rmse| |———:|———:|———:|———:| | 156.1296| 43242.19| 0.9999994| 0.9999995|

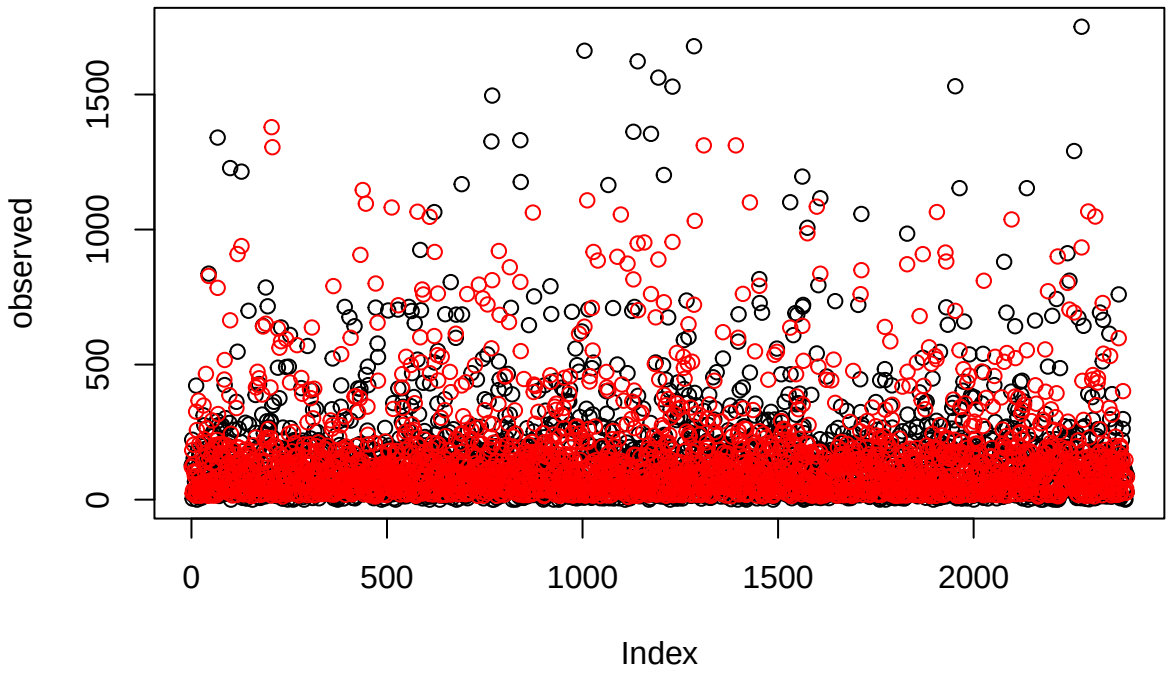
```
# popravljena mnozica atributov
extModel <- nnet(poraba ~ ., regSetExt$train, size=5, decay=0.001, maxit=10000, linout=T)
```

```
## # weights:  121
## initial  value 33786.792735
## iter  10 value 3463.168298
## iter  20 value 3463.150458
## iter  30 value 3453.852050
## iter  40 value 2884.543830
## iter  50 value 2483.606368
## iter  60 value 2012.801329
## iter  70 value 1789.832926
## iter  80 value 1604.558993
## iter  90 value 1435.595895
## iter 100 value 1411.877983
## iter 110 value 1408.862694
## iter 120 value 1407.505882
## iter 130 value 1405.345119
## iter 140 value 1404.731765
## iter 150 value 1404.317679
## iter 160 value 1404.041677
## iter 170 value 1403.315886
## iter 180 value 1403.170145
## iter 190 value 1403.115655
## iter 200 value 1403.056179
```

```
## iter 210 value 1403.032289
## iter 220 value 1403.025350
## iter 230 value 1403.023959
## iter 240 value 1403.022255
## final  value 1403.022119
## converged
```

```
EvaluateRegExtModel(extModel, regSetExt$train, regSetExt$test)
```

```
## [1] "Srednja absolutna napaka: 77.938639121452"
## [1] "Srednja kvadratna napaka: 23484.1965754327"
## [1] "Relativna srednja absolutna napaka: 0.521708140858081"
## [1] "Relativna srednja kvadratna napaka: 0.387642596158205"
```



| mae| mse| rmae| rmse| |———:|——-:|———:|———:| | 77.93864| 23484.2| 0.5217081| 0.3876426|

Izboljsava klasifikacijskih modelov

Metoda ovojnice

Izboljsava klasifikacijskega modela z izbiro optimalne podmnozice atributov, ki minimizira doloceno oceno.

```
runWrapper(namembnost ~ ., classSetBase$train)
```

```
## best model: estimated error =  0.003732725 , selected feature subset =  namembnost ~ povrsina + leto
```

```
dtBase <- rpart(namembnost ~ povrsina + leto_izgradnje + stavba + datum + regija + temp_zraka + temp_ros
EvaluateClassModel(dtBase, classSetBase$train, classSetBase$test)
```

brier	ca	infGain
0.9636537	0.5146321	0.5997643

Glasovanje

Zgradimo modele z osnovno in popravljeno mnozico atributov:

```
dtBase <- rpart(namembnost ~ pritisk, data=classSetBase$train)
knnBase <- CoreModel(namembnost ~ ., data=classSetBase$train, model="knn", kInNN=5)
rfBase <- randomForest(namembnost ~ ., data=classSetBase$train)

dtExt <- rpart(namembnost ~ pritisk, data=classSetExt$train)
knnExt <- CoreModel(namembnost ~ ., data=classSetExt$train, model="knn", kInNN=5)
rfExt <- randomForest(namembnost ~ ., data=classSetExt$train)
```

Glasovanje z osnovno mnozico atributov:

```
predDtBase <- predict(dtBase, classSetBase$test, type="class")
predKnnBase <- predict(knnBase, classSetBase$test, type="class")
predRfBase <- predict(rfBase, classSetBase$test, type="class")

modelsDf <- data.frame(
  predDtBase,
  predKnnBase,
  predRfBase
)

runVoting(modelsDf, classSetBase$test$namembnost)
```

```
## [1] "Classification accuracy: 0.5"
```

Glasovanje z popravljeno mnozico atributov:

```
predDtExt <- predict(dtExt, classSetExt$test, type="class")
predKnnExt <- predict(knnExt, classSetExt$test, type="class")
predRfExt <- predict(rfExt, classSetExt$test, type="class")

modelsDf <- data.frame(
  predDtExt,
  predKnnExt,
  predRfExt
)

runVoting(modelsDf, classSetExt$test$namembnost)
```

```
## [1] "Classification accuracy: 0.522157190635452"
```

Utezeno glasovanje

Glasovanje z osnovno mnozico atributov:

```
predDtBase <- predict(dtBase, classSetBase$test, type="prob")
predKnnBase <- predict(knnBase, classSetBase$test, type="prob")
predRfBase <- predict(rfBase, classSetBase$test, type="prob")
runWeightedVoting(predDtBase + predKnnBase + predRfBase, classSetBase$test$namembnost)
```

```
## [1] "Classification accuracy: 0.523829431438127"
```

Glasovanje z popravljeno mnozico atributov:

```
predDtExt <- predict(dtExt, classSetExt$test, type="prob")
predKnnExt <- predict(knnExt, classSetExt$test, type="prob")
predRfExt <- predict(rfExt, classSetExt$test, type="prob")
runWeightedVoting(predDtExt + predKnnExt + predRfExt, classSetExt$test$namembnost)
```

```
## [1] "Classification accuracy: 0.538461538461538"
```

Bagging

Bagging z osnovno mnozico atributov:

```
bag <- bagging(namembnost ~ ., classSetBase$train, nbagg=30)
predictions <- predict(bag, classSetBase$test)
ca <- CA(classSetBase$test$namembnost, predictions$class)
print(paste("Classification accuracy:", ca))
```

```
## [1] "Classification accuracy: 0.535535117056856"
```

Bagging z popravljeno mnozico atributov:

```
bag <- bagging(namembnost ~ ., classSetExt$train, nbagg=30)
predictions <- predict(bag, classSetExt$test)
ca <- CA(classSetExt$test$namembnost, predictions$class)
print(paste("Classification accuracy:", ca))
```

```
## [1] "Classification accuracy: 0.525083612040134"
```

Boosting

Boosting z osnovno mnozico atributov:

```
bm <- boosting(namembnost ~ ., classSetBase$train, mfinal=100)
predictions <- predict(bm, classSetBase$test)
ca <- CA(classSetBase$test$namembnost, predictions$class)
print(paste("Classification accuracy:", ca))
```

```
## [1] "Classification accuracy: 0.492892976588629"
```

Boosting z popravljeno mnozico atributov:

```
bm <- boosting(namembnost ~ ., classSetExt$train, mfinal=100)
predictions <- predict(bm, classSetExt$test)
ca <- CA(classSetExt$test$namembnost, predictions$class)
print(paste("Classification accuracy:", ca))
```

```
## [1] "Classification accuracy: 0.469063545150502"
```

Primerjava po regijah

Priprava podatkov

Pripravimo podatke, tako da ucno in testno mnozico razbijemo na dve podmnozici: - mnozica ki vsebuje samo primere z vzhodno regijo - mnozica ki vsebuje samo primere z zahodno regijo

```
selTrain <- classSetExt$train$regija == "vzhodna"
selTest <- classSetExt$test$regija == "vzhodna"

classVzhodnaTrain <- classSetExt$train[selTrain,]
classVzhodnaTest <- classSetExt$test[selTest,]
classVzhodnaTrain$regija <- NULL
classVzhodnaTest$regija <- NULL

classZahodnaTrain <- classSetExt$train[!selTrain,]
classZahodnaTest <- classSetExt$test[!selTest,]
classZahodnaTrain$regija <- NULL
classZahodnaTest$regija <- NULL
```

Podatki za klasifikacijo

```
selTrain <- regSetExt$train$regija == "vzhodna"
selTest <- regSetExt$test$regija == "vzhodna"

regVzhodnaTrain <- regSetExt$train[selTrain,]
regVzhodnaTest <- regSetExt$test[selTest,]
regVzhodnaTrain$regija <- NULL
regVzhodnaTest$regija <- NULL

regZahodnaTrain <- regSetExt$train[!selTrain,]
regZahodnaTest <- regSetExt$test[!selTest,]
regZahodnaTrain$regija <- NULL
regZahodnaTest$regija <- NULL
```

Podatki za regresijo

Evalvacija

Klasifikacija Zgradimo nekaj klasifikacijskih modelov, ki se ucijo iz posamezne podmnozice, ter vsakega posebej se ocenimo glede na testne primere iz ustrezne testne mnozice.

```
runClassification(namembnost ~ ., classVzhodnaTrain, classVzhodnaTest)
```

```
## [1] "Trivial classification accuracy: 0.527433628318584"
## [1] "odlocitveno drevo classification accuracy: 0.708849557522124"
## [1] "naivni bayes classification accuracy: 0.587610619469027"
## [1] "k-najblizjih sosedov classification accuracy: 0.588495575221239"
## [1] "nakljucni gozd classification accuracy: 0.769026548672566"
```

```
runClassification(namembnost ~ ., classZahodnaTrain, classZahodnaTest)
```

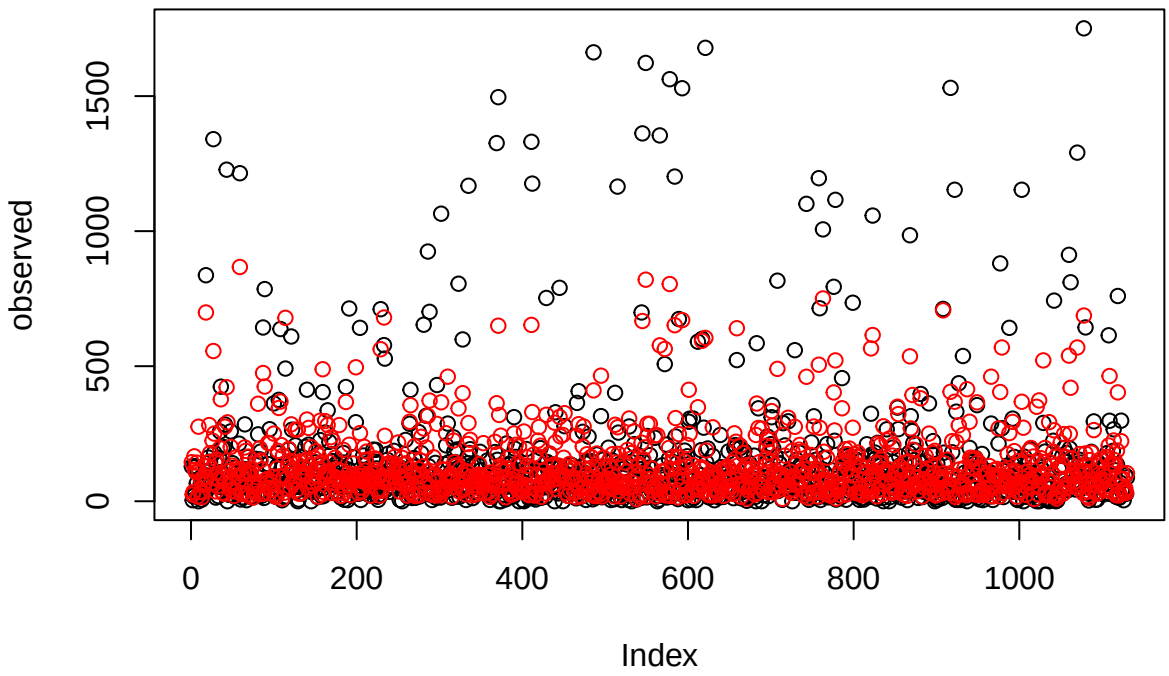
```
## [1] "Trivial classification accuracy: 0.44215530903328"
## [1] "odlocitveno drevo classification accuracy: 0.327258320126783"
## [1] "naivni bayes classification accuracy: 0.329635499207607"
## [1] "k-najblizjih sosedov classification accuracy: 0.389064976228209"
## [1] "nakljucni gozd classification accuracy: 0.38351822503962"
```

Regresija Zgradimo nekaj regresijskih modelov, ki se ucijo iz posamezne podmnozice, ter vsakega posebej se ocenimo glede na testne primere iz ustrezne testne mnozice.

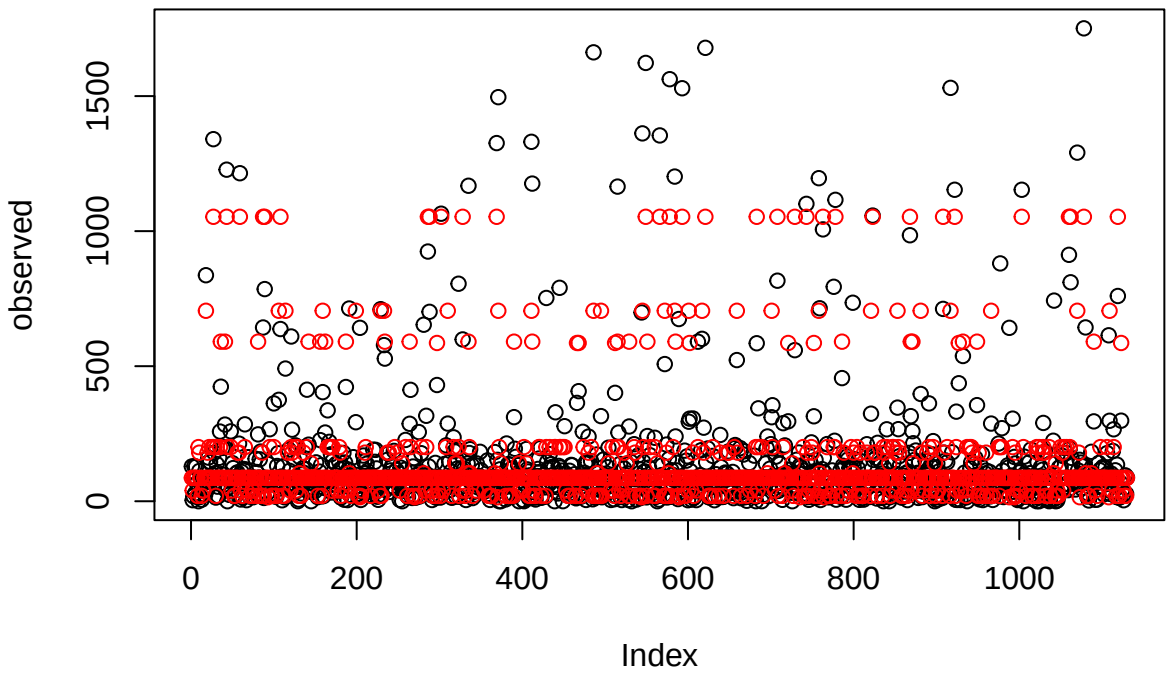
```
runRegression(poraba ~ ., regVzhodnaTrain, regVzhodnaTest)
```

```
## Warning in predict.lm(model, test): prediction from a rank-deficient fit may be
## misleading
```

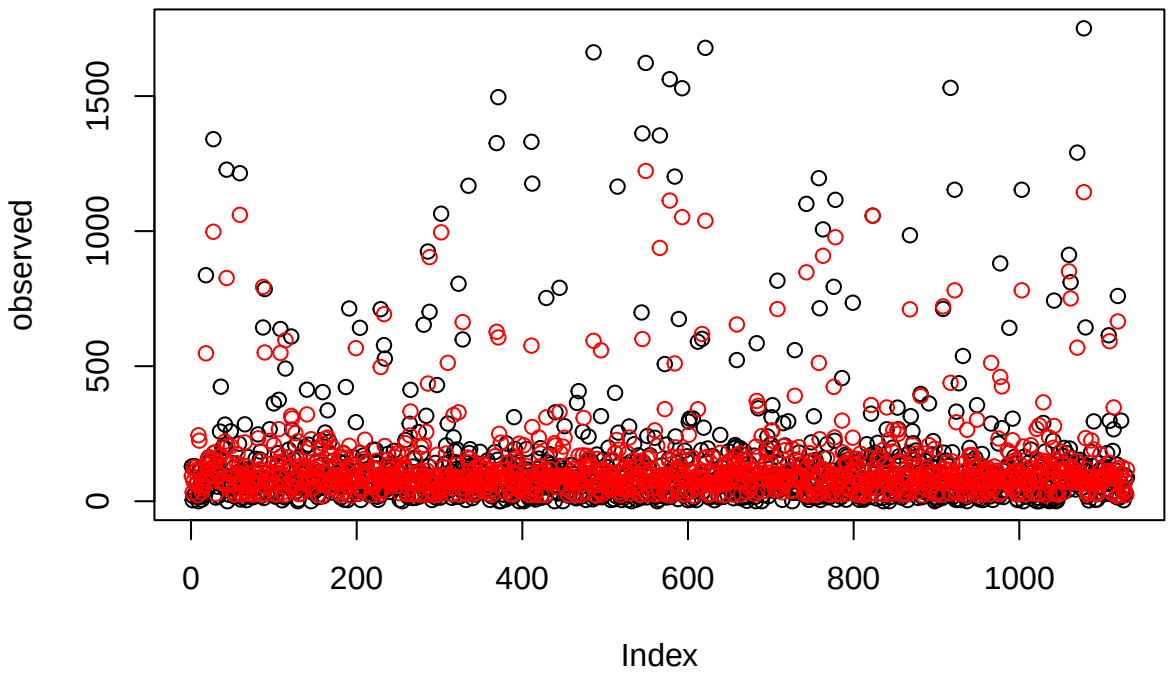
```
## [1] "Srednja absolutna napaka: 68.0364221786923"
## [1] "Srednja kvadratna napaka: 24245.8709018281"
## [1] "Relativna srednja absolutna napaka: 0.482408149805635"
## [1] "Relativna srednja kvadratna napaka: 0.324012921907863"
```




```
## [1] "Srednja absolutna napaka: 69.7672168706825"
## [1] "Srednja kvadratna napaka: 19365.4640776844"
## [1] "Relativna srednja absolutna napaka: 0.494680245226284"
## [1] "Relativna srednja kvadratna napaka: 0.258792955935403"
```




```
## [1] "Srednja absolutna napaka: 62.9844710717455"
## [1] "Srednja kvadratna napaka: 17976.4119691739"
## [1] "Relativna srednja absolutna napaka: 0.446587595044395"
## [1] "Relativna srednja kvadratna napaka: 0.240230173258587"
```

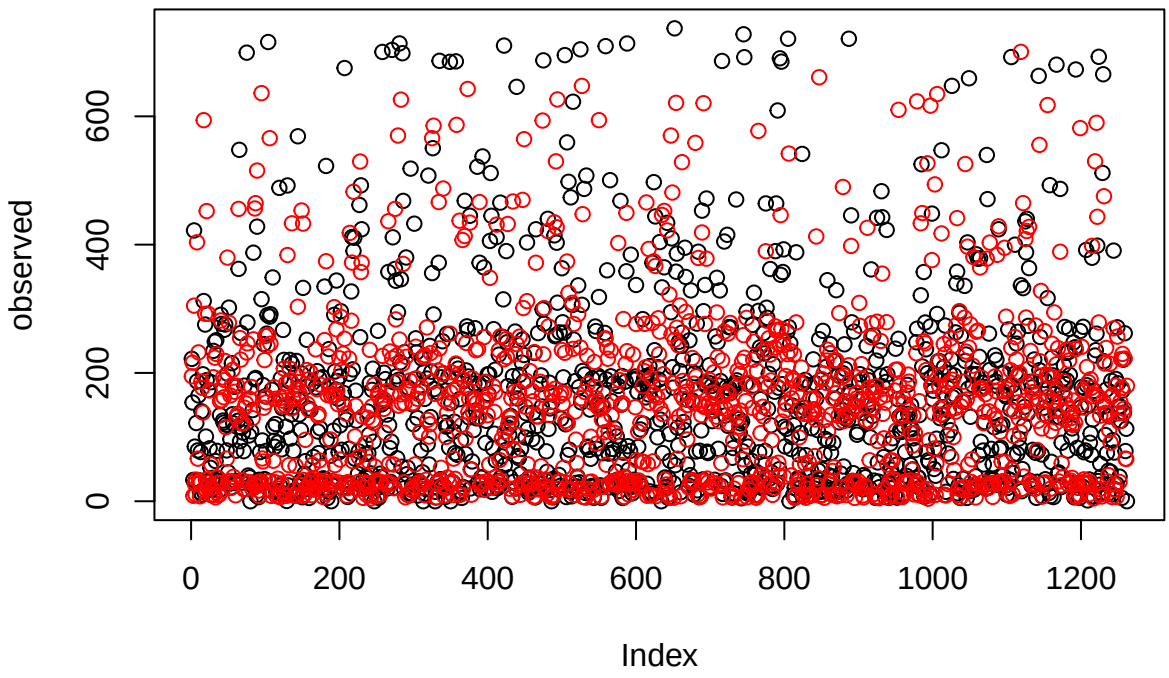


| mae| mse| rmae| rmse| |———:|———:|———:|———:| | 62.98447| 17976.41| 0.4465876| 0.2402302|

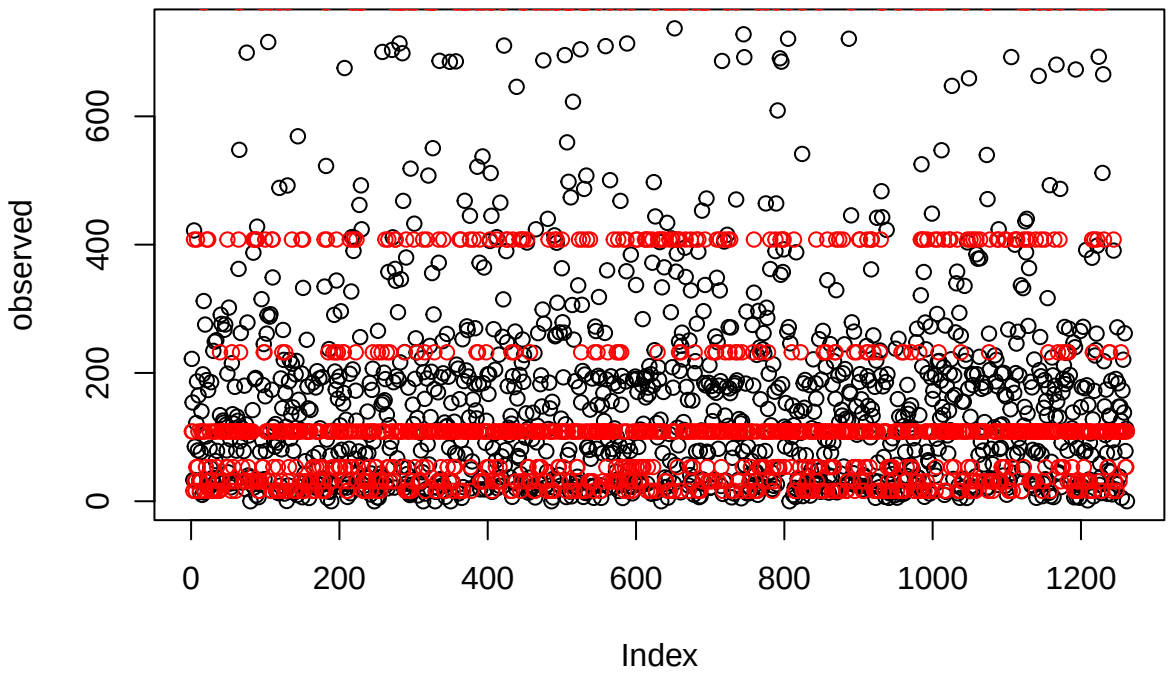
```
runRegression(poraba ~ ., regZahodnaTrain, regZahodnaTest)
```

```
## Warning in predict.lm(model, test): prediction from a rank-deficient fit may be
## misleading
```

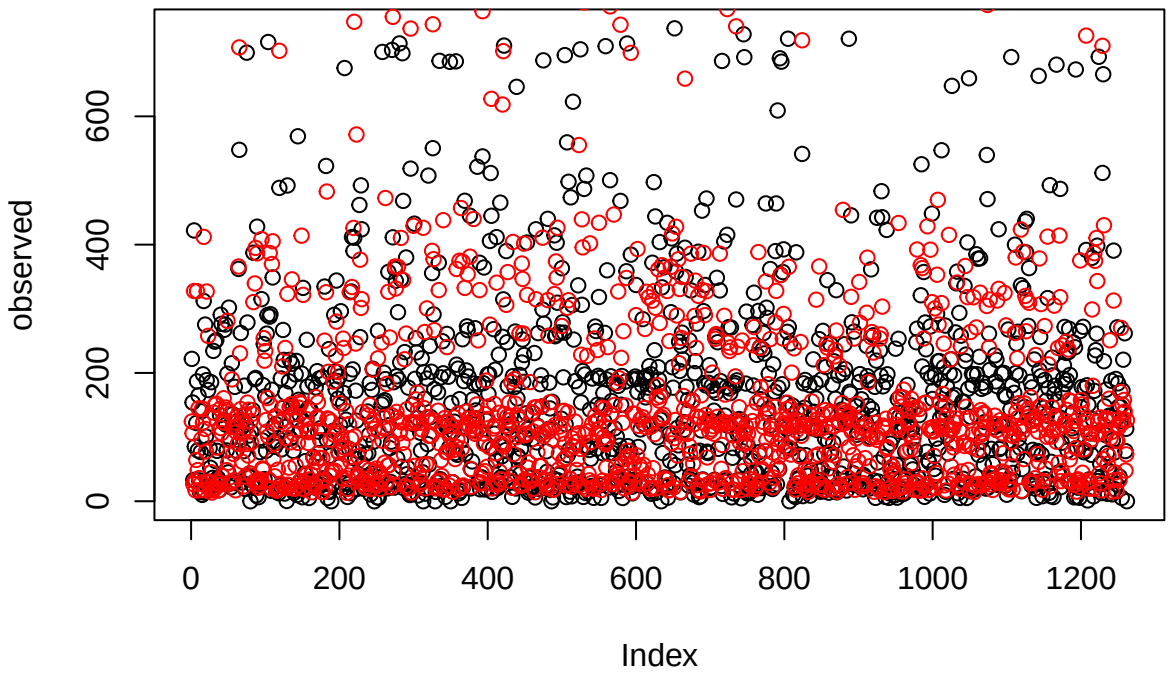
```
## [1] "Srednja absolutna napaka: 102.234797806236"
## [1] "Srednja kvadratna napaka: 40377.2807422382"
## [1] "Relativna srednja absolutna napaka: 0.651691692238129"
## [1] "Relativna srednja kvadratna napaka: 0.844254033480331"
```




```
## [1] "Srednja absolutna napaka: 105.516186325833"
## [1] "Srednja kvadratna napaka: 30080.6996535656"
## [1] "Relativna srednja absolutna napaka: 0.672608774123297"
## [1] "Relativna srednja kvadratna napaka: 0.628961424484114"
```




```
## [1] "Srednja absolutna napaka: 86.0077749517434"
## [1] "Srednja kvadratna napaka: 20390.91090344"
## [1] "Relativna srednja absolutna napaka: 0.548253174131273"
## [1] "Relativna srednja kvadratna napaka: 0.426356318704711"
```



| mae| mse| rmae| rmse| |———:|———:|———:|———:| | 86.00777| 20390.91| 0.5482532| 0.4263563|

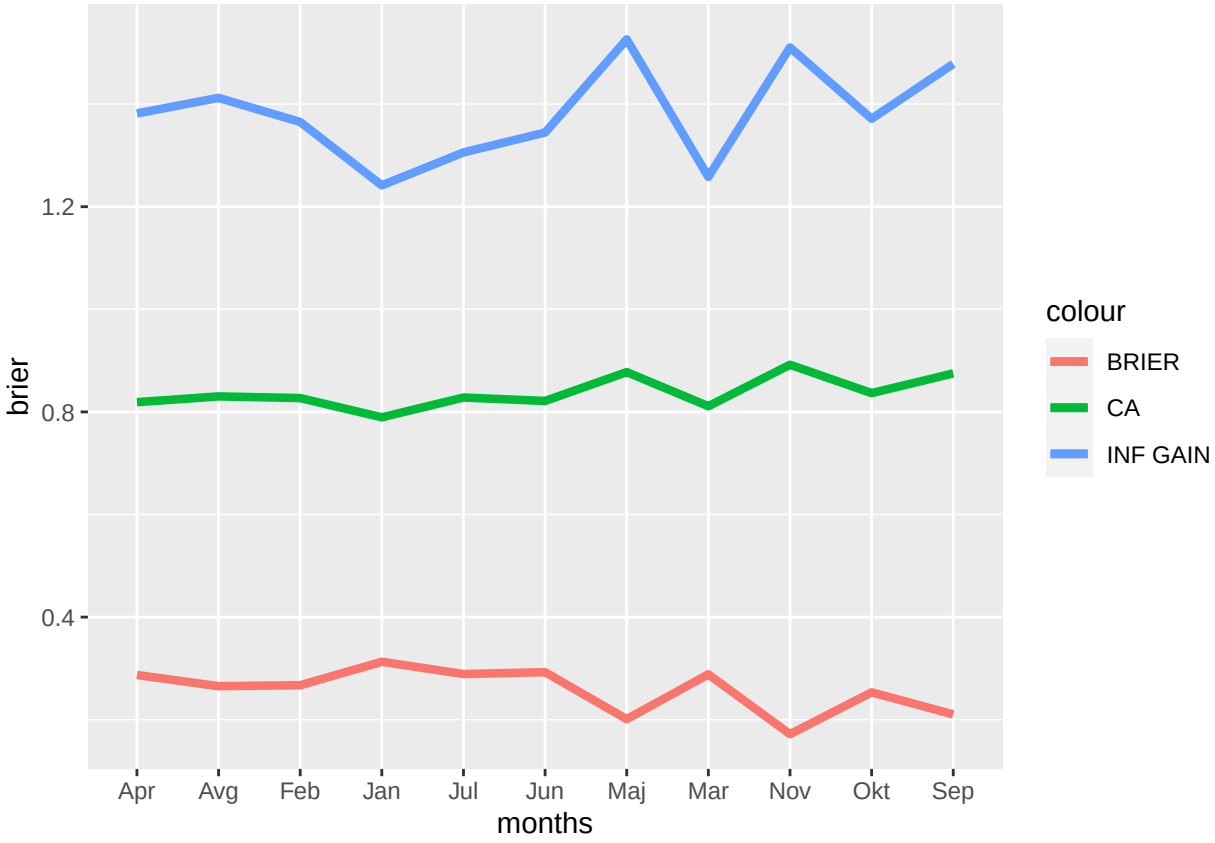
Evalvacija po mesecih

```
regData <- ExtendRegSet(allData)
classData <- ExtendClassSet(allData)
classData$mesec <- as.factor(ToMonth(allData$datum))

regDataByMonth = list()
classDataByMonth = list()

for (i in 1:12)
{
  regDataByMonth[[i]] <- regData[regData$mesec==i,]
  classDataByMonth[[i]] <- classData[classData$mesec==i,]
  classDataByMonth[[i]]$mesec <- NULL
  regDataByMonth[[i]]$mesec <- NULL
  regDataByMonth[[i]]$letni_cas <- NULL
  regDataByMonth[[i]]$zima <- NULL
}

brier <- vector()
ca <- vector()
infGain <- vector()
```

```
mae <- vector()
mse <- vector()
rmse <- vector()
rmae <- vector()


for (i in 1:11)
{
  regTrain <- do.call("rbind", regDataByMonth[1:i])
  regTest <- regDataByMonth[[i + 1]]

  classTrain <- do.call("rbind", classDataByMonth[1:i])
  classTest <- classDataByMonth[[i + 1]]

  dt <- rpart(namembnost ~ ., data=classTrain)
  score <- EvaluateClassModel(dt, classTrain, classTest, F)
  brier[i] <- score$brier
  ca[i] <- score$ca
  infGain[i] <- score$infGain

  lmExt <- lm(poraba ~ ., regTrain)
  score <- EvaluateRegExtModel(lmExt, regTrain, regTest, F, F)
  mae[i] <- score$mae
  mse[i] <- score$mse
  rmse[i] <- score$rmse
  rmae[i] <- score$rmae
}
```

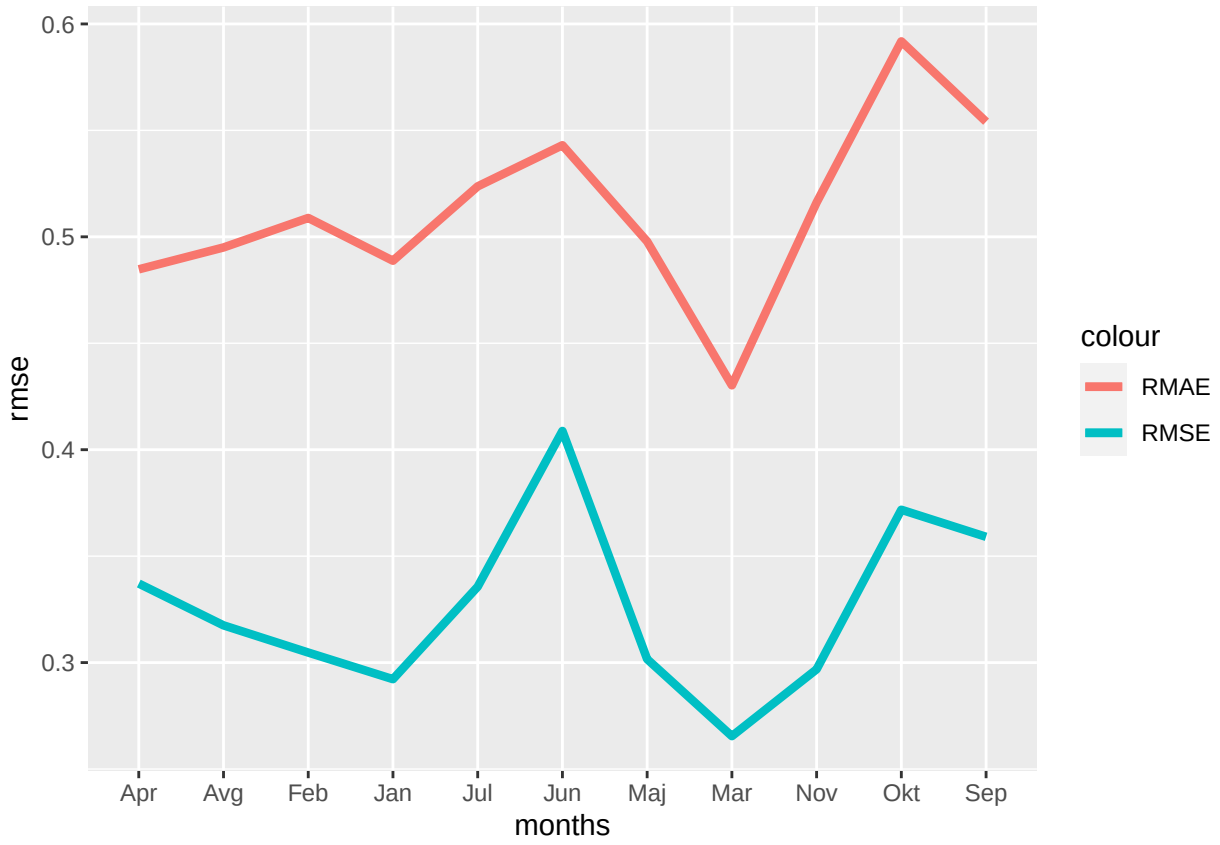
Ocene klasifikacije

```
drawClassEvaluationGraph(brier, ca, infGain)
```



Ocene regresije

```
drawRegrEvaluationGraph(rmse, rmae)
```



Zakljucek

V tej seminarski nalogi sem zgradil in evaluiral nekaj razlicnih klasifikacijskih in regresijskih modelov.

Pri klasifikaciji je bil najbolj kvaliteten model nakljucnega gozda, najslabsi pa naivni bayesov klasifikator. Medtem ko je bila pri regresiji najboljsa metoda linearne regresija, najslabsa pa metoda nevronskih mrez.

Pri locenem ucenju glede na regije smo opazili, da je klasifikacijska in regresijska napoved obcutno uspesnejsa za primere iz podmnozice podatkov z vzhodno regijo. Eden izmed moznih razlogov za to je verjetno tudi dejstvo, da so stavbe za izobrazevalne namene vecinski razred, ter da imajo stavbe z vzhodno lego imajo za skoraj 13% vec stavb za izobrazevalne namene kot stavbe z zahodno lego.