

Seminarska naloga 1 (Umetna Inteligence 2021-2022)

Bartolomej Kozorog (63200152)

December 5, 2021

Contents

Knjiznice in orodja	2
Vizualizacija podatkov	3
Uvoz podatkov	3
Izris grafov	3
Priprava atributov	19
Pomozne metode	19
Izboljsava mnozice atributov	19
Evalvacija atributov	20
Klasifikacija	23
Vecinski klasifikator	23
Odlocitveno drevo	24
Odlocitveno drevo z rezanjem	24
Naivni Bayes	26
K-bliznjih sosedov	26
Naključni gozd	26
Regresija	27
Trivialni model	27
Linearna regresija	28
Naključni gozd	31
Nevronske mreze	33
Izboljsava klasifikacijskih modelov	35
Metoda ovojnica	35
Glasovanje	35
Utezeno glasovanje	36
Bagging	37
Boosting	37

Primerjava po regijah	38
Priprava podatkov	38
Evalvacija	38
Evalvacija po mesecih	44
Ocene klasifikacije	45
Ocene regresije	46
Zakljucek	47

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 že 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(ggcrrplot)
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=",")

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

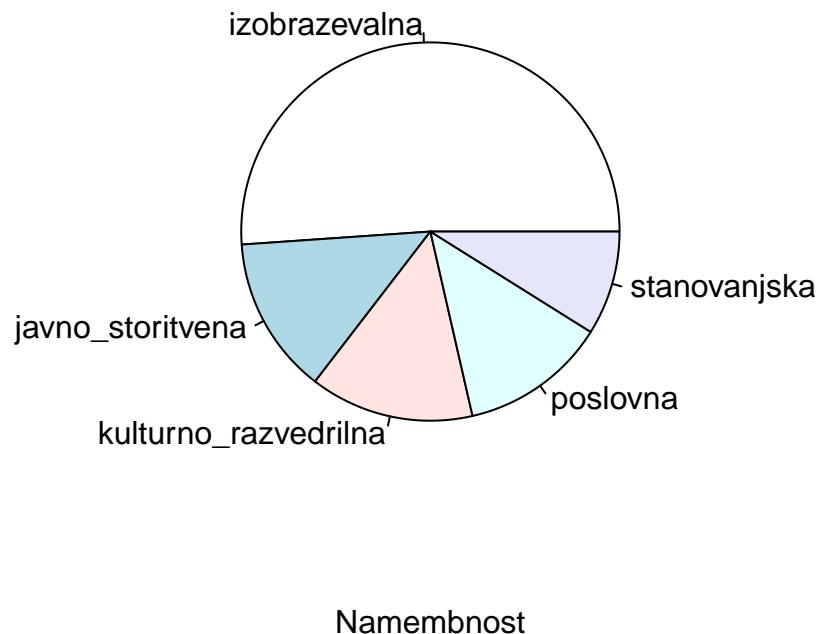
allData <- rbind(test, train)

```

Izris grafov

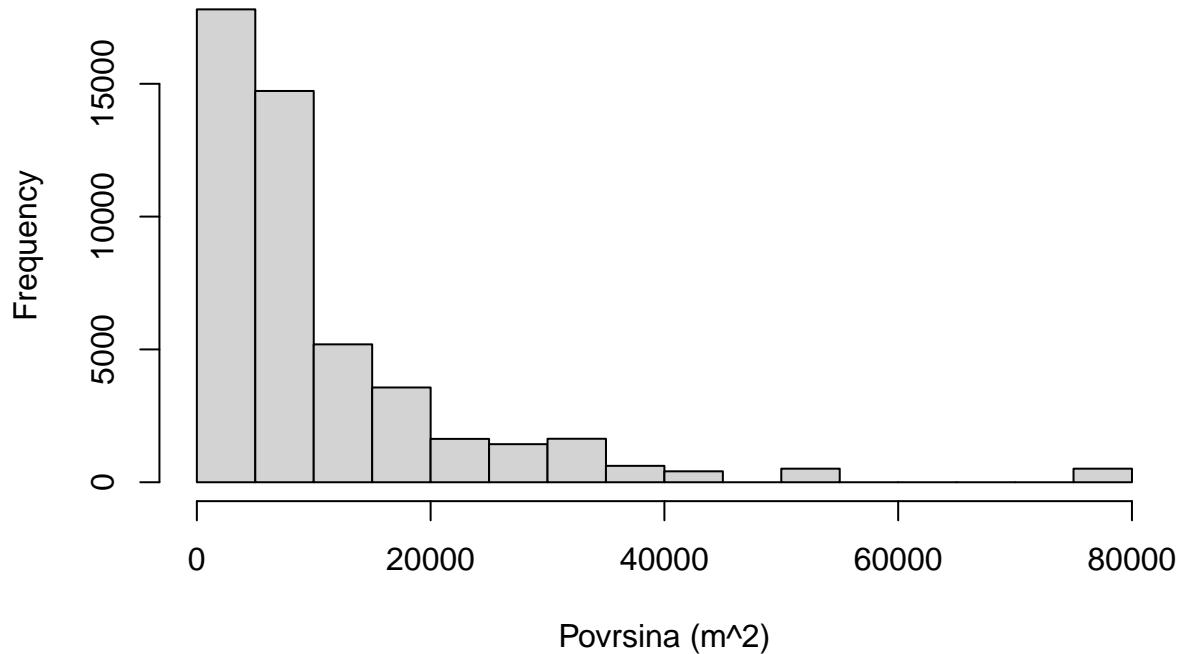
Porazdelitvene vrednosti Vizualizirajmo porazdelitvene vrednosti posameznih atributov, da dobimo boljsi vpogled v vsak atribut posebej.

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



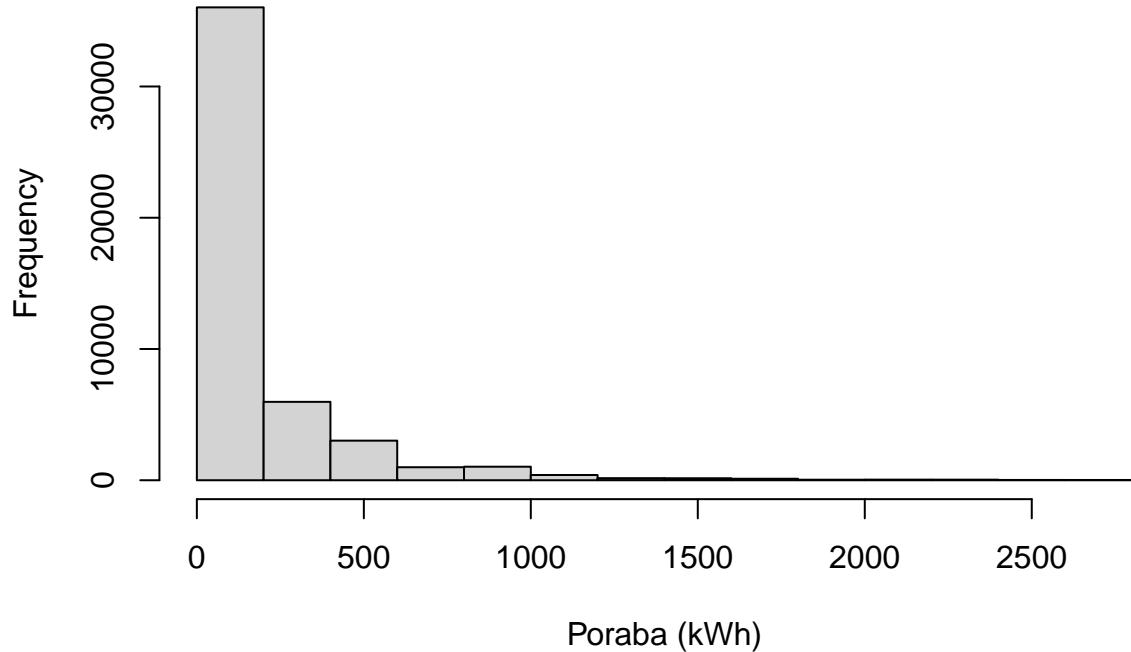
```
hist(allData$povrsina, xlab="Povrsina (m^2)", main="Histogram povrsine stavb")
```

Histogram povrsine stavb



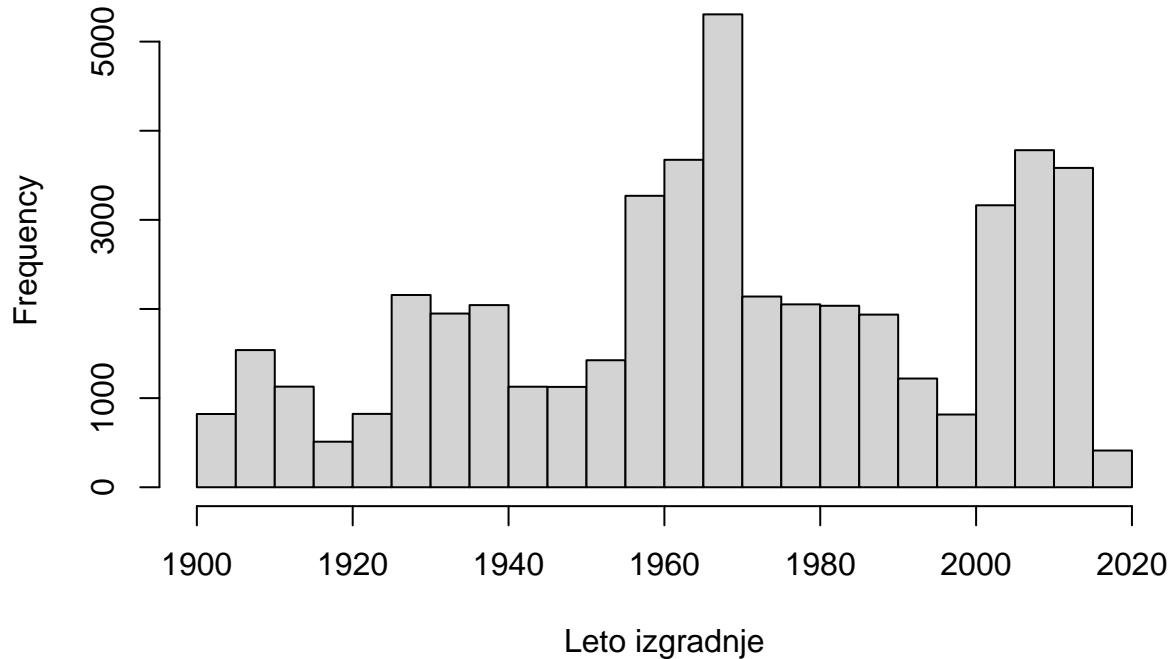
```
hist(allData$poraba, xlab="Poraba (kWh)", main="Histogram porabe stavb")
```

Histogram porabe stavb



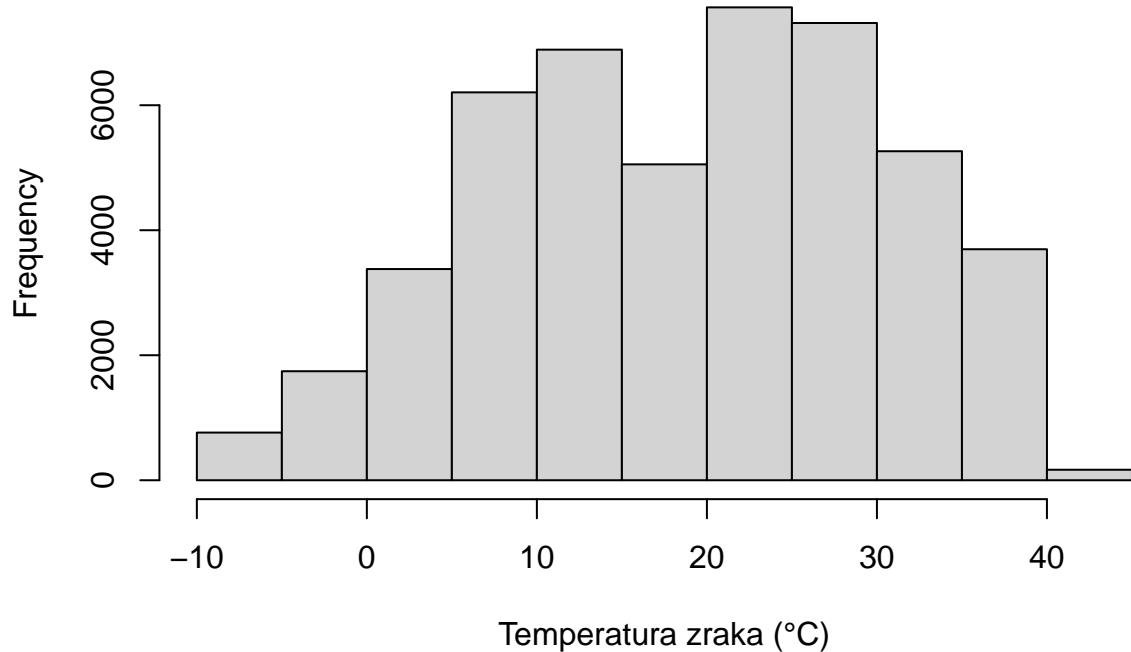
```
hist(allData$leto_izgradnje, xlab="Leto izgradnje", main="Histogram leta izgradnje stavb")
```

Histogram leta izgradnje stavb



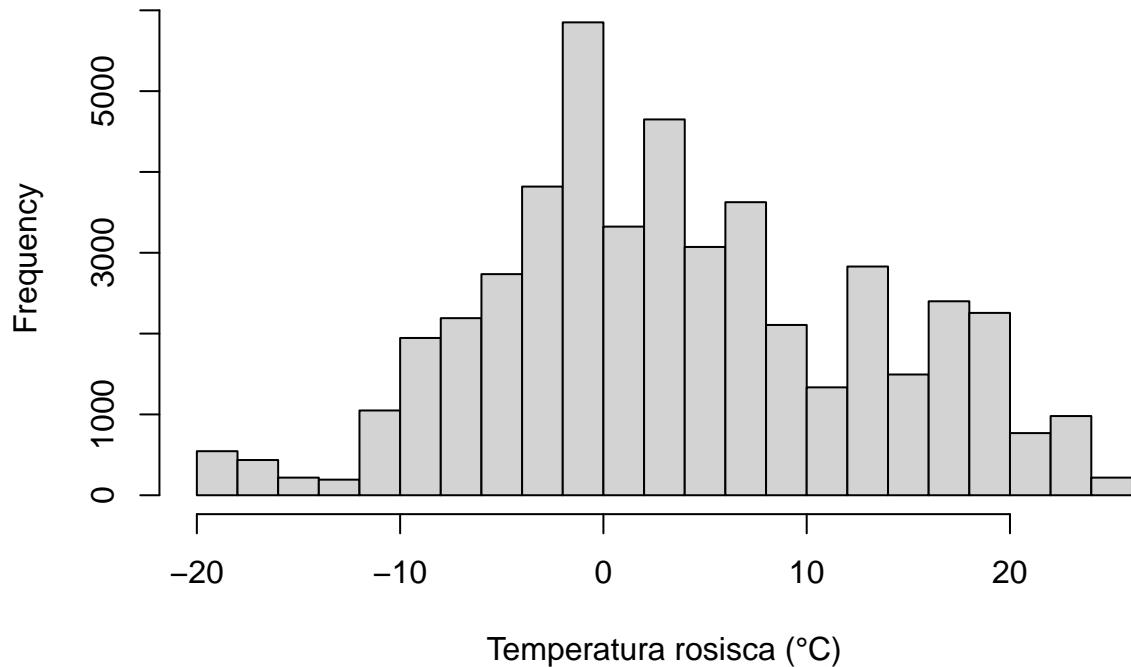
```
hist(allData$temp_zraka, xlab="Temperatura zraka (°C)", main="Histogram temperature zraka")
```

Histogram temperature zraka



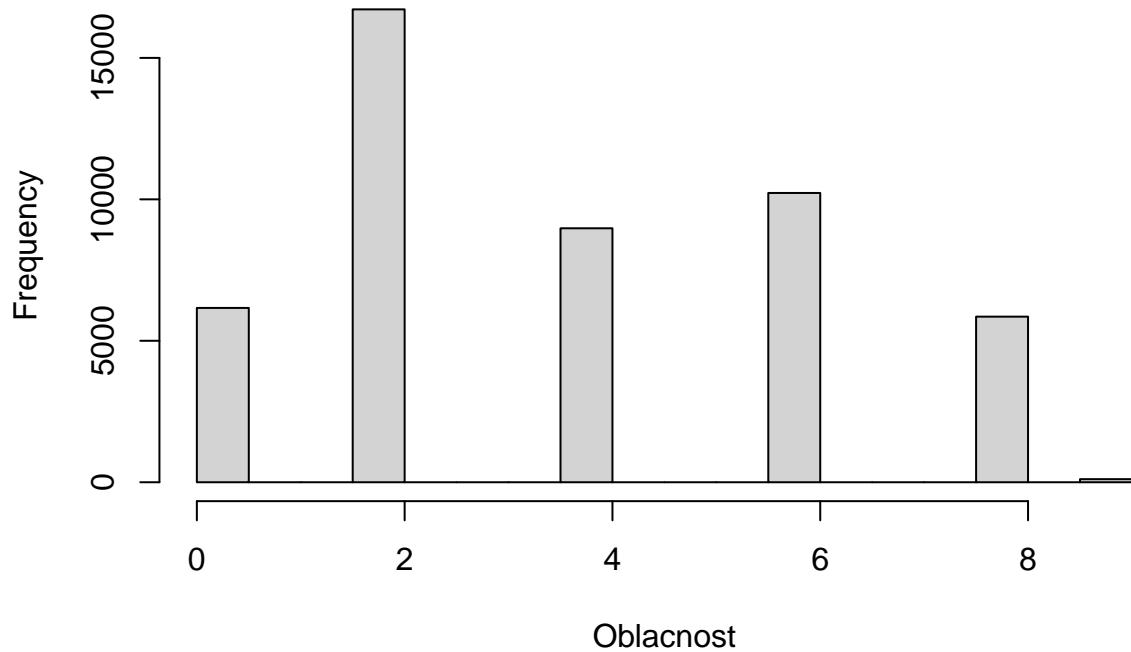
```
hist(allData$temp_rosisca, xlab="Temperatura rosisca (°C)", main="Histogram temperature rosisca")
```

Histogram temperature rosisca



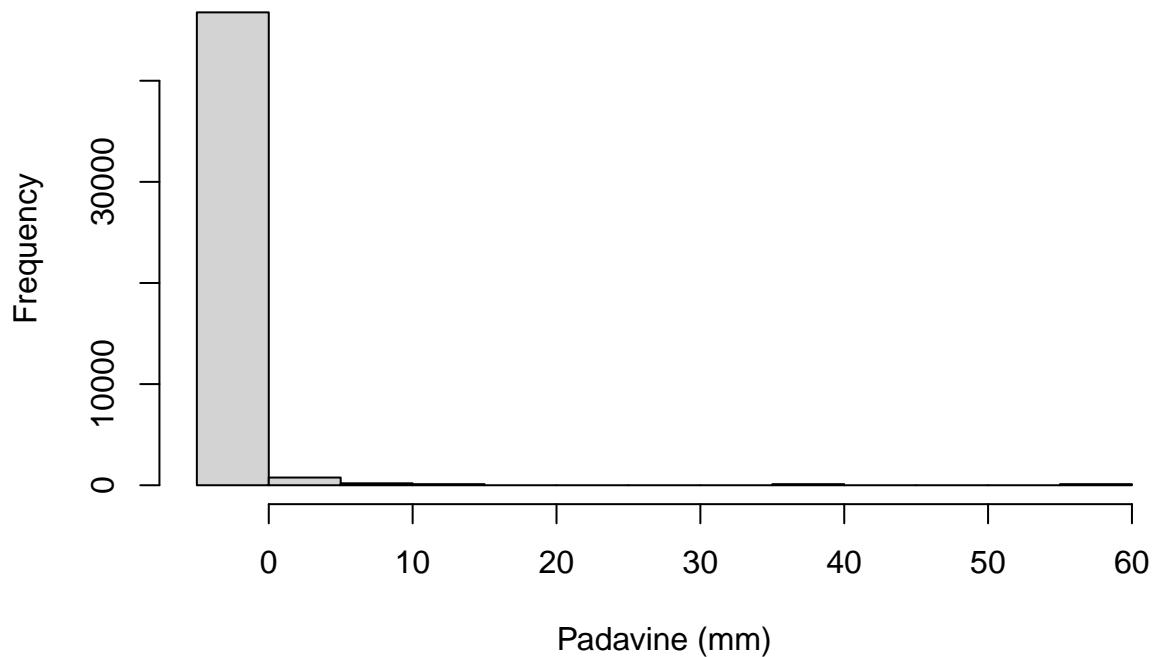
```
hist(allData$oblacnost, xlab="Oblacnost", main="Histogram stopnje pokritosti neba z oblaki")
```

Histogram stopnje pokritosti neba z oblaki



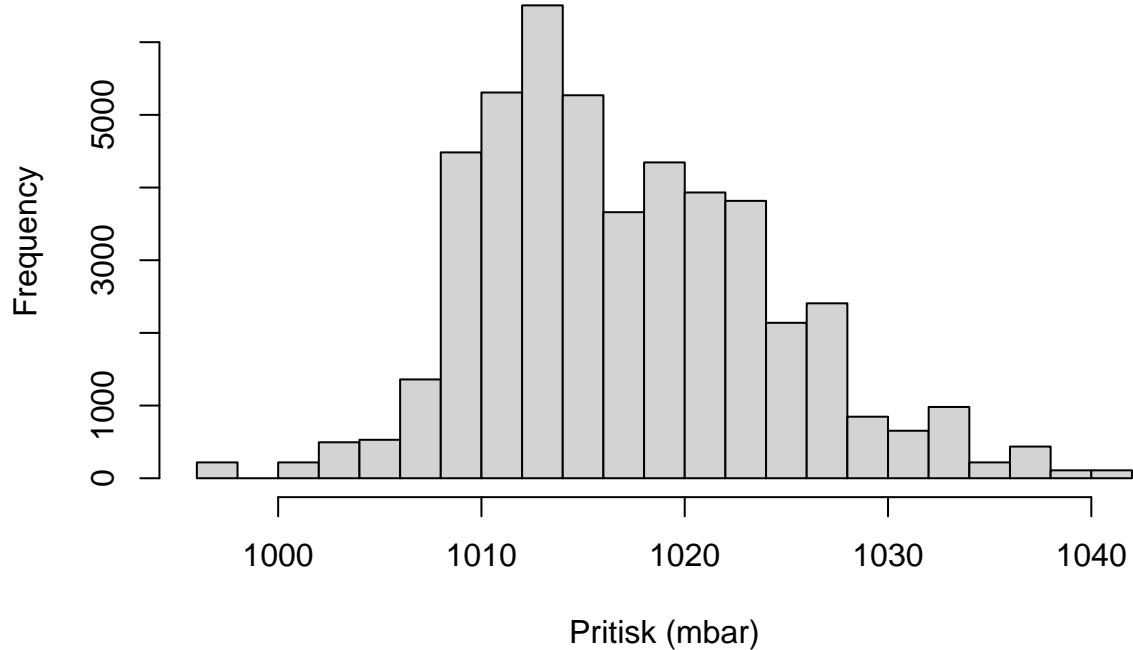
```
hist(allData$padavine, xlab="Padavine (mm)", main="Histogram kolicine padavin")
```

Histogram kolicine padavin



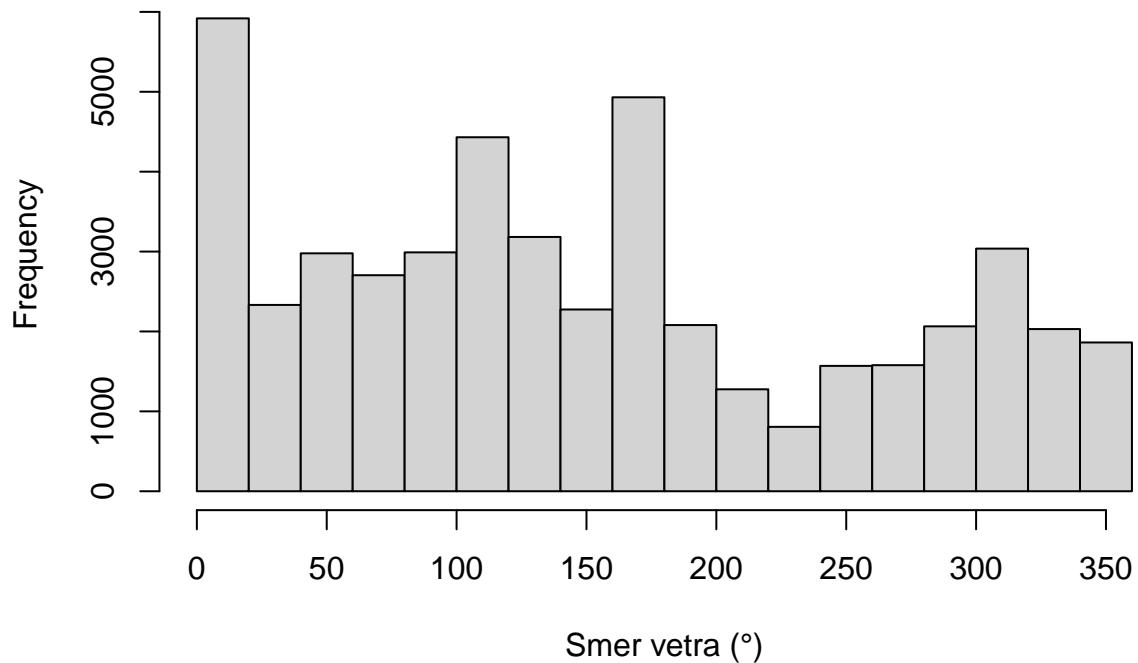
```
hist(allData$pritisk, xlab="Pritisk (mbar)", main="Histogram zracnega pritiska")
```

Histogram zracnega pritiska



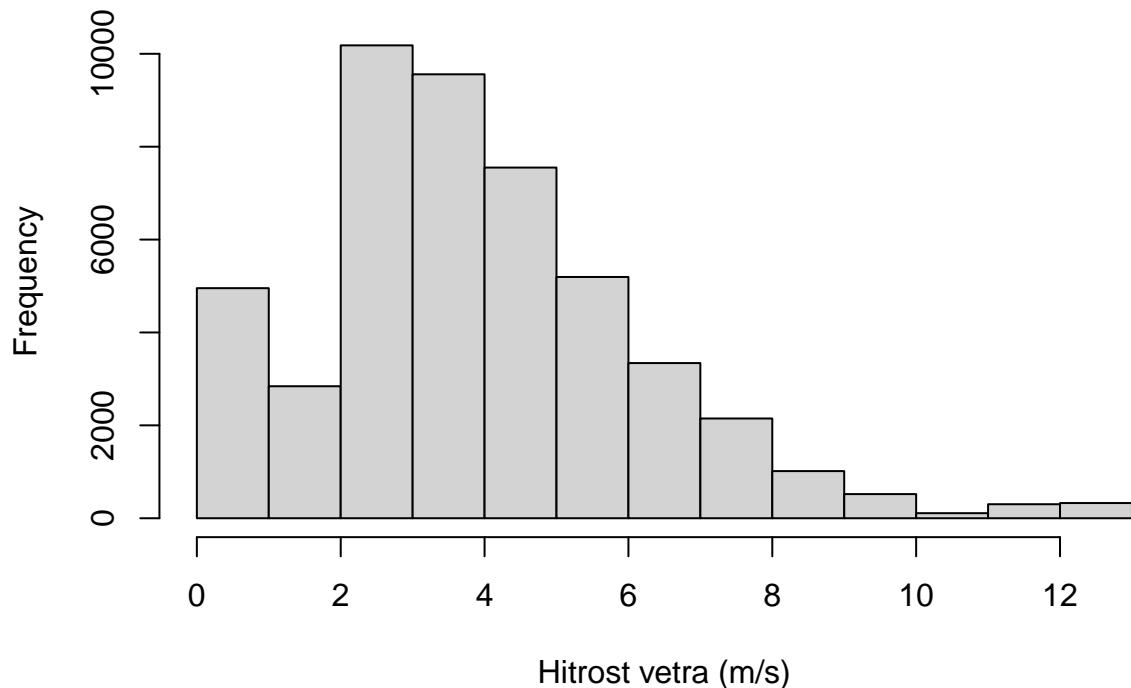
```
hist(allData$smer_vetra, xlab="Smer vetra (°)", main="Histogram smeri vetra")
```

Histogram smeri vetra



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

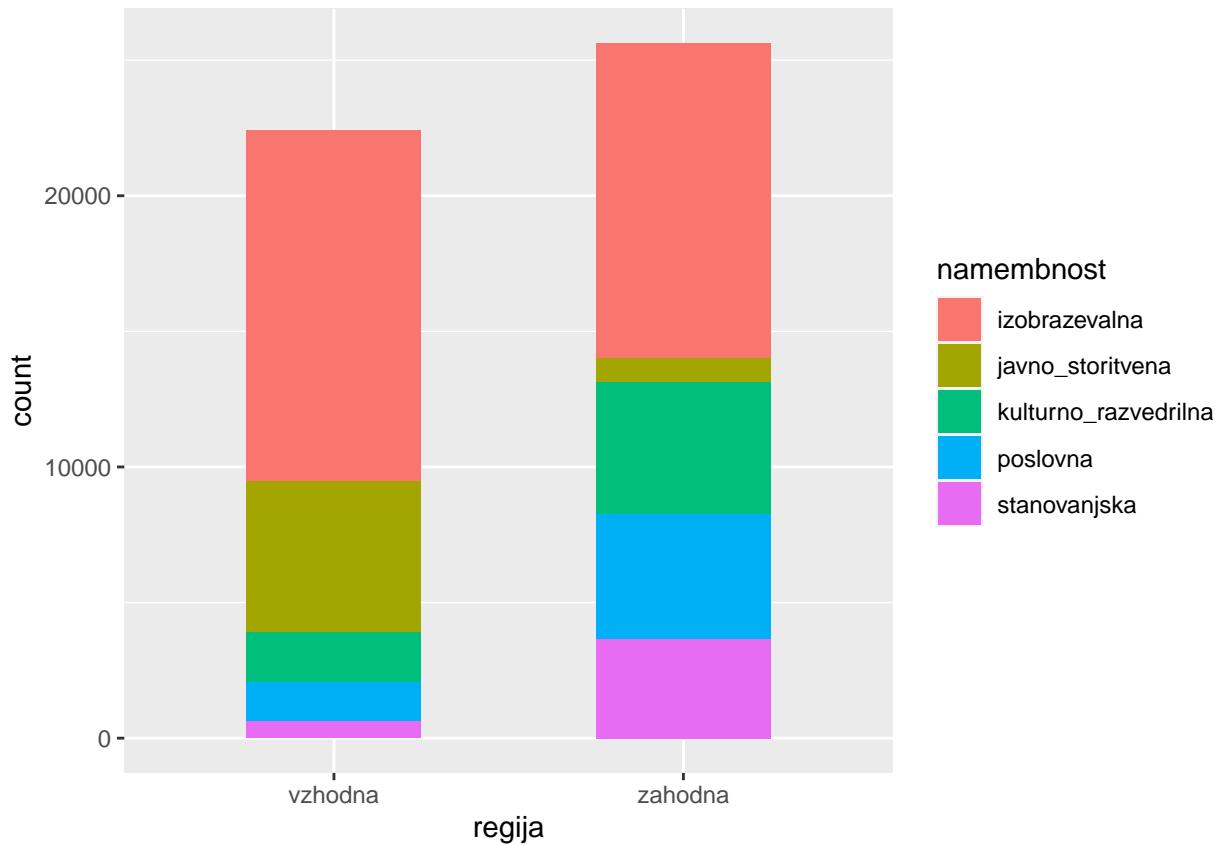
Histogram hitrosti vetra



Namembnost in regija

Namembnost stavb glede na regijo *Ugotovitve:* - priblizno polovica stavb služi izobrazevalnemu namenu - stavb z zahodno lego je malo več kot stavb z zahodno lego - stavbe z vzhodno lego imajo za skoraj 13% več 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 0.577881828316611"
```

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

```
## [1] "Odstotek izobrazevalnih stavb z zahodno regijo 0.452380952380952"
```

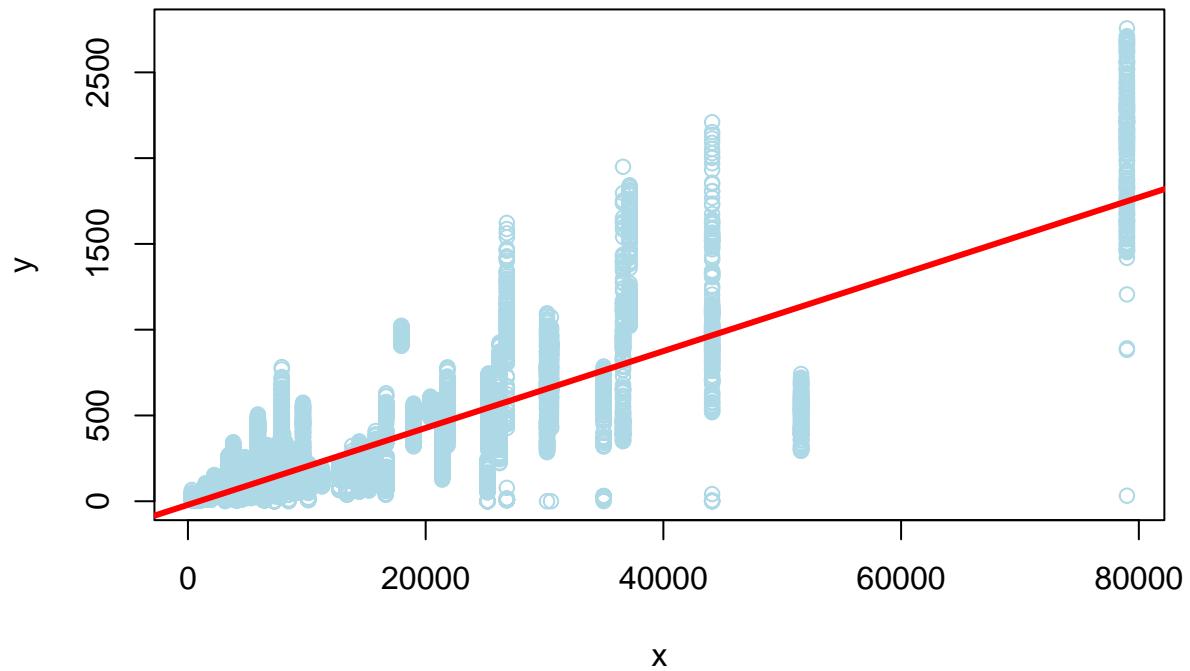
Soodvisnost atributov Pri nadalnji predikciji nam bo koristilo tudi nekaj intuicije o soodvisnosti med določeni atributi.

Ze samo po sebi je logично, 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.

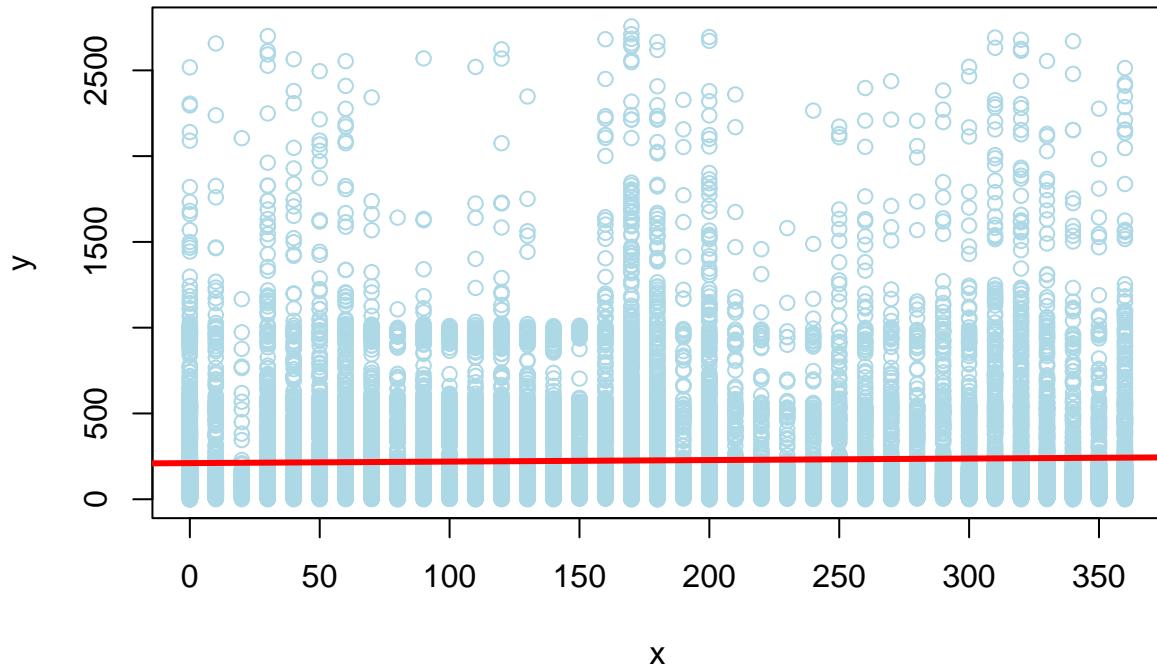
Pri porabi električne energije v odvisnosti z površino train vidimo, da obstaja jasen pozitiven trend.

```
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)
```



Najboljše 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 zunanjji 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	-0.24	-0.52	0.00	0.16
## povrsina	0.16	1.00	0.08	-0.08	0.00	0.02
## leto_izgradnje	-0.24	0.08	1.00	0.17	0.00	-0.05
## temp_zraka	-0.52	-0.08	0.17	1.00	0.61	-0.28
## temp_rosisca	0.00	0.00	0.00	0.61	1.00	0.06

```

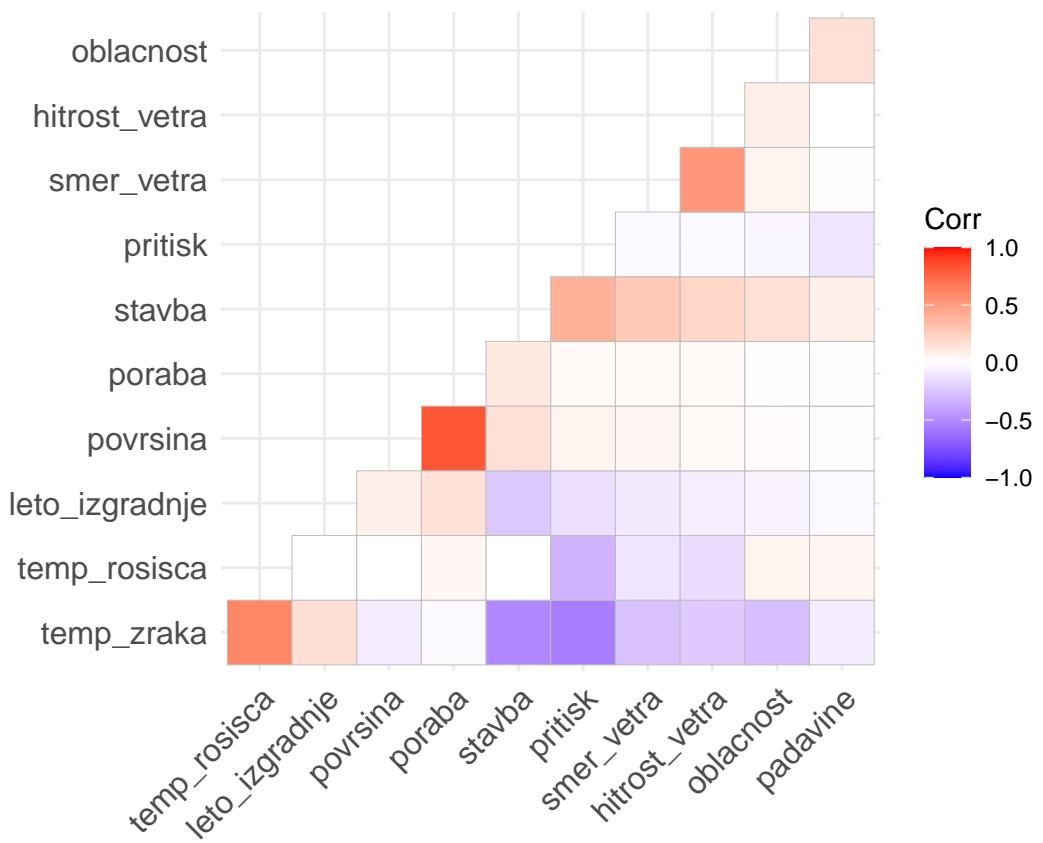
## oblacnost      0.16    0.02     -0.05   -0.28    0.06   1.00
## padavine      0.08    0.01    -0.03   -0.08    0.05   0.16
## pritisk       0.40    0.06    -0.13   -0.56   -0.33  -0.04
## smer_vetra    0.28    0.05    -0.09   -0.27   -0.11  0.06
## hitrost_vetra 0.21    0.03    -0.07   -0.23   -0.15  0.09
## poraba        0.11    0.82     0.15   -0.02    0.04   0.01
##                  padavine pritisk smer_vetra hitrost_vetra poraba
## stavba         0.08    0.40     0.28    0.21    0.11
## povrsina       0.01    0.06     0.05    0.03    0.82
## leto_izgradnje -0.03   -0.13    -0.09   -0.07    0.15
## temp_zraka     -0.08   -0.56    -0.27   -0.23   -0.02
## temp_rosisca   0.05   -0.33    -0.11   -0.15    0.04
## oblacnost      0.16   -0.04     0.06    0.09    0.01
## padavine       1.00   -0.11     0.01    0.00    0.01
## pritisk        -0.11   1.00    -0.03   -0.03    0.03
## smer_vetra     0.01   -0.03     1.00    0.53    0.03
## hitrost_vetra  0.00   -0.03     0.53    1.00    0.03
## poraba         0.01   0.03     0.03    0.03    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 manjšo 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$oblačnost <- log1p(set$oblačnost)
  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
## 0.242867703 0.187635967 0.187635967 0.170489197 0.058759648
##      poraba          pritisk        smer_vetra        oblacnost      hitrost_vetra
## 0.046933702 0.038678565 0.031117122 0.014874294 0.007483121
##      temp_rosisca       padavine         datum
## 0.006136749 0.004304307 0.002677500
##
##
## GainRatio
##      stavba          povrsina          poraba          regija      leto_izgradnje

```

```

##    0.398622264    0.363963455    0.338566263    0.188157512    0.170936745
##    temp_zraka      pritisk     temp_rosisca    smer_vetra      oblacnost
##    0.074257281    0.044118883    0.033563197    0.031192614    0.027426812
##    padavine       hitrost_vetra      datum
##    0.027189466    0.025828283    0.003164738
##
##
## Gini
##    povrsina leto_izgradnje      regija      stavba      poraba
##    0.0812406365  0.0526331906  0.0292248014  0.0292248014  0.0172333605
##    temp_zraka      pritisk     smer_vetra      oblacnost  hitrost_vetra
##    0.0098485629  0.0069052732  0.0055992237  0.0025554217  0.0013654599
##    temp_rosisca    padavine      datum
##    0.0010846709  0.0007024586  0.0004900156
##
##
## MDL
##    povrsina      regija      stavba leto_izgradnje      temp_zraka
##    0.241739833   0.186436871   0.186436871   0.169517637   0.057774668
##    poraba        pritisk     smer_vetra      oblacnost  hitrost_vetra
##    0.046023490   0.037743430   0.030187093   0.014010364   0.006574901
##    temp_rosisca    padavine      datum
##    0.005322963   0.003582116   0.001783835
##
##
## Relief
##    leto_izgradnje    povrsina      poraba      stavba      regija
##    0.144401903   0.141499631   0.081808566   0.038833333   0.000000000
##    padavine       temp_zraka     temp_rosisca    datum  hitrost_vetra
##    -0.001011544  -0.019591581  -0.032974218  -0.034348688  -0.035208424
##    pritisk        oblacnost    smer_vetra
##    -0.037114824  -0.038839378  -0.047187104
##
##
## ReliefFequalK
##    leto_izgradnje      stavba      povrsina      poraba      regija
##    0.340066063   0.281672186   0.244269510   0.162840115   0.096751049
##    temp_zraka      pritisk      datum      smer_vetra  temp_rosisca
##    0.047698521   0.036217820   0.032040063   0.029595735   0.018115182
##    hitrost_vetra    oblacnost    padavine
##    0.013374770   0.012056304   0.004058682
##
##
## ReliefFexpRank
##    leto_izgradnje      stavba      povrsina      poraba      regija
##    0.339488247   0.288581076   0.232952991   0.147563919   0.117536460
##    temp_zraka      pritisk     smer_vetra      datum  temp_rosisca
##    0.059597809   0.049852127   0.049596823   0.040894553   0.029950460
##    hitrost_vetra    oblacnost    padavine
##    0.027001366   0.026127641   0.004677626

evalRegrFeatures(poraba ~ ., regSetBase$train)

## MSEofMean

```

```

##      povrsina leto_izgradnje      stavba      namembnost      regija
##      -49643.37      -93093.85      -96620.67      -98182.44      -99578.23
##      temp_rosisca          datum      smer_vetra      pritisk      temp_zraka
##      -100039.15      -100451.62      -100503.28      -100513.20      -100523.85
##      hitrost_vetra      oblacnost      padavine
##      -100591.84      -100652.53      -100694.52
##
##
## RReliefFexpRank
##      povrsina leto_izgradnje      namembnost      regija      padavine
##      4.521807e-01      1.064202e-01      3.193940e-02      -4.785751e-06      -3.054951e-04
##      stavba      temp_zraka      pritisk          datum      temp_rosisca
##      -1.320170e-02      -3.042607e-02      -3.342632e-02      -4.259575e-02      -4.789351e-02
##      oblacnost      hitrost_vetra      smer_vetra
##      -5.142319e-02      -5.235320e-02      -6.479992e-02

```

Ponovno evaluiramo atributte za popravljeni mnozici atributov:

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

```

## InfGain
##      povrsina      regija      stavba      leto_izgradnje      temp_zraka
##      0.242867703      0.187635967      0.187635967      0.170489197      0.058759648
##      poraba      pritisk      smer_vetra      oblacnost      hitrost_vetra
##      0.046933702      0.038678565      0.031117122      0.014874294      0.007483121
##      temp_rosisca      padavine
##      0.006136749      0.004304307
##
##
## GainRatio
##      stavba      povrsina      poraba      regija      leto_izgradnje
##      0.39862226      0.36396346      0.33856626      0.18815751      0.17093675
##      temp_zraka      pritisk      temp_rosisca      smer_vetra      oblacnost
##      0.07425728      0.04411888      0.03356320      0.03119261      0.02742681
##      padavine      hitrost_vetra
##      0.02718947      0.02582828
##
##
## Gini
##      povrsina      leto_izgradnje      regija      stavba      poraba
##      0.0812406365      0.0526331906      0.0292248014      0.0292248014      0.0172333605
##      temp_zraka      pritisk      smer_vetra      oblacnost      hitrost_vetra
##      0.0098485629      0.0069052732      0.0055992237      0.0025554217      0.0013654599
##      temp_rosisca      padavine
##      0.0010846709      0.0007024586
##
##
## MDL
##      povrsina      regija      stavba      leto_izgradnje      temp_zraka
##      0.241739833      0.186436871      0.186436871      0.169517637      0.057774668
##      poraba      pritisk      smer_vetra      oblacnost      hitrost_vetra
##      0.046023490      0.037743430      0.030187093      0.014010364      0.006574901
##      temp_rosisca      padavine

```

```

##      0.005322963    0.003582116
##
##
## Relief
## leto_izgradnje      povrsina      stavba      poraba      regija
##   3.724301e-01    2.768455e-01    1.834758e-01    1.547306e-01    4.145078e-05
##      padavine      temp_zraka      oblacnost  hitrost_vetra  temp_rosisca
##  -3.145896e-03   -6.260209e-02   -6.478756e-02   -7.349312e-02   -7.874247e-02
##      pritisk      smer_vetra
##  -8.273006e-02   -1.097855e-01
##
## ReliefFequalK
## leto_izgradnje      povrsina      stavba      poraba      regija
##   0.502460904    0.486736654    0.418928096    0.353628014    0.093551631
##      padavine      temp_zraka      pritisk  temp_rosisca      oblacnost
##   0.001039381   -0.019695164   -0.030544031   -0.051203319   -0.054809267
##      hitrost_vetra      smer_vetra
##  -0.057314982   -0.060293058
##
## ReliefFexpRank
## leto_izgradnje      povrsina      stavba      poraba      regija
##   0.498943794    0.466549320    0.431482645    0.329571518    0.112128986
##      padavine      temp_zraka      pritisk  temp_rosisca      smer_vetra
##   0.002204067   -0.006812587   -0.014569871   -0.034587228   -0.036921300
##      oblacnost  hitrost_vetra
##  -0.039027036   -0.043970445

evalRegrFeatures(poraba ~ ., regSetExt$train)

```

```

## MSEofMean
##      povrsina leto_izgradnje      mesec      letni_cas      regija
##  -0.9116275     -1.3488061    -1.4516484    -1.4535388    -1.4572507
##      vikend  hitrost_vetra      pritisk      oblacnost      zima
##  -1.4608254     -1.4609828    -1.4610883    -1.4611062    -1.4614874
##
## RReliefFexpRank
##      povrsina leto_izgradnje      regija      zima      letni_cas
##   3.244337e-01    1.057386e-01    1.406610e-04   -4.497249e-05   -2.979408e-03
##      vikend      mesec      oblacnost      pritisk  hitrost_vetra
##  -1.697988e-02   -4.463985e-02   -5.269498e-02   -6.046437e-02   -9.968089e-02

```

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] 0.4702341
```

Odlocitveno drevo

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

	brier	ca	infGain
izobrazevalna	0.7178714	0.4702341	0

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

	brier	ca	infGain
	0.9950272	0.4966555	0.5367325

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 povrsina      regija      stavba
##
## Root node error: 10824/24125 = 0.44866
##
## n= 24125
##
##          CP nsplit rel error      xerror      xstd
## 1  0.1323910      0  1.000000 1.00000000 0.00713698
## 2  0.1140983      1  0.867609 0.86760902 0.00699673
## 3  0.0423134      2  0.753511 0.75351072 0.00678822
## 4  0.0379712      5  0.612528 0.61252772 0.00640608
## 5  0.0281781      7  0.536585 0.53658537 0.00613506
## 6  0.0211336     16  0.282982 0.28510717 0.00479280
```

```

## 7 0.0190318      20 0.198448 0.28510717 0.00479280
## 8 0.0189394      21 0.179416 0.21895787 0.00427103
## 9 0.0140891      24 0.122598 0.14855876 0.00357912
## 10 0.0095159     30 0.038064 0.03907982 0.00188339
## 11 0.0000000     34 0.000000 0.00018477 0.00013065

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
	1.006689	0.4966555	0.5573658

```

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

```

```

##
## Classification tree:
## rpart(formula = namembnosc ~ ., data = classSetExt$train, cp = 0)
##
## Variables actually used in tree construction:
## [1] letov_izgradnje povrsina      regija      stavba
##
## Root node error: 10824/24125 = 0.44866
##
## n= 24125
##
##          CP nsplits rel_error      xerror      xstd
## 1  0.1323910      0 1.000000 1.00000000 0.00713698
## 2  0.1140983      1 0.867609 0.86760902 0.00699673
## 3  0.0423134      2 0.753511 0.75351072 0.00678822
## 4  0.0379712      5 0.612528 0.62888027 0.00645811
## 5  0.0281781      7 0.536585 0.56448633 0.00624046
## 6  0.0211336     16 0.282982 0.28141168 0.00476616
## 7  0.0190318     20 0.198448 0.28141168 0.00476616
## 8  0.0189394     21 0.179416 0.24039172 0.00445127
## 9  0.0140891     24 0.122598 0.14772727 0.00356981
## 10 0.0095159     30 0.038064 0.03880266 0.00187682
## 11 0.0000000     34 0.000000 0.00027716 0.00016001

```

```

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
	1.006689	0.4966555	0.5573658

Naivni Bayes

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

	brier	ca	infGain
izobrazevalna	0.7680676	0.4593645	0.5344814

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

	brier	ca	infGain
izobrazevalna	0.7622223	0.4629181	0.5415797

K-bliznjih sosedov

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

	brier	ca	infGain
izobrazevalna	0.6796054	0.5642559	0.6321713

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

	brier	ca	infGain
izobrazevalna	0.6816254	0.5348244	0.6996965

Nakljucni gozd

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

	brier	ca	infGain
	0.6743042	0.5388378	0.6413587

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

	brier	ca	infGain
	0.6757219	0.5352007	0.6455013

Regresija

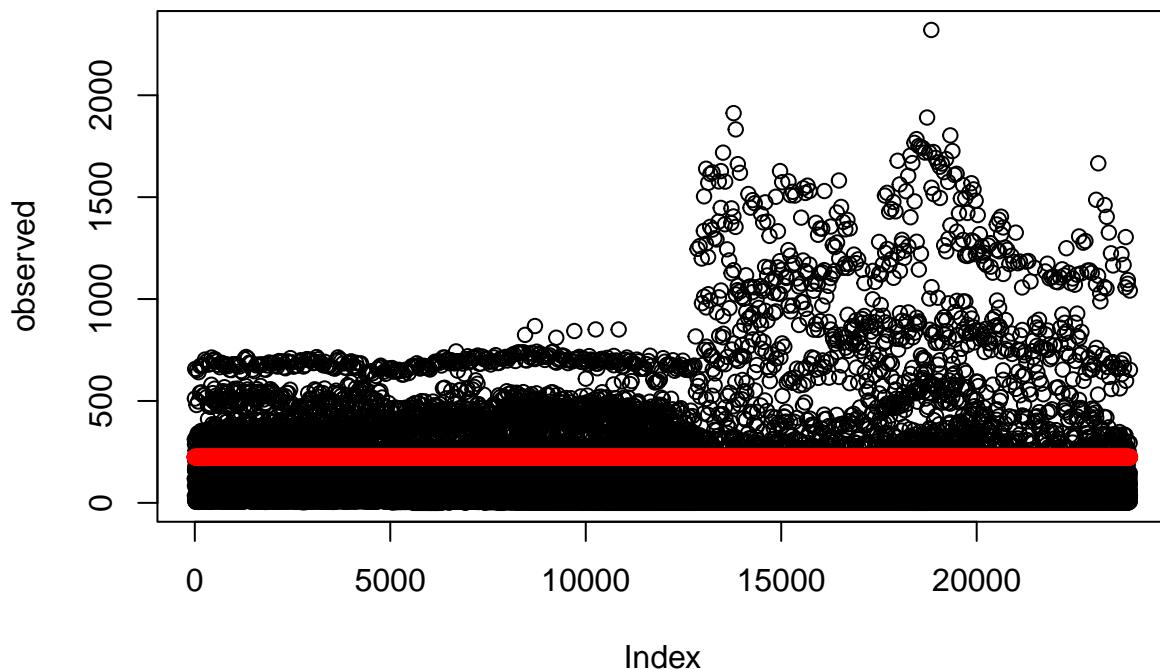
Trivialni model

Trivialni model vedno vraca povprecno vrednost ciljne spremenljivke, glede na vse ucene 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)

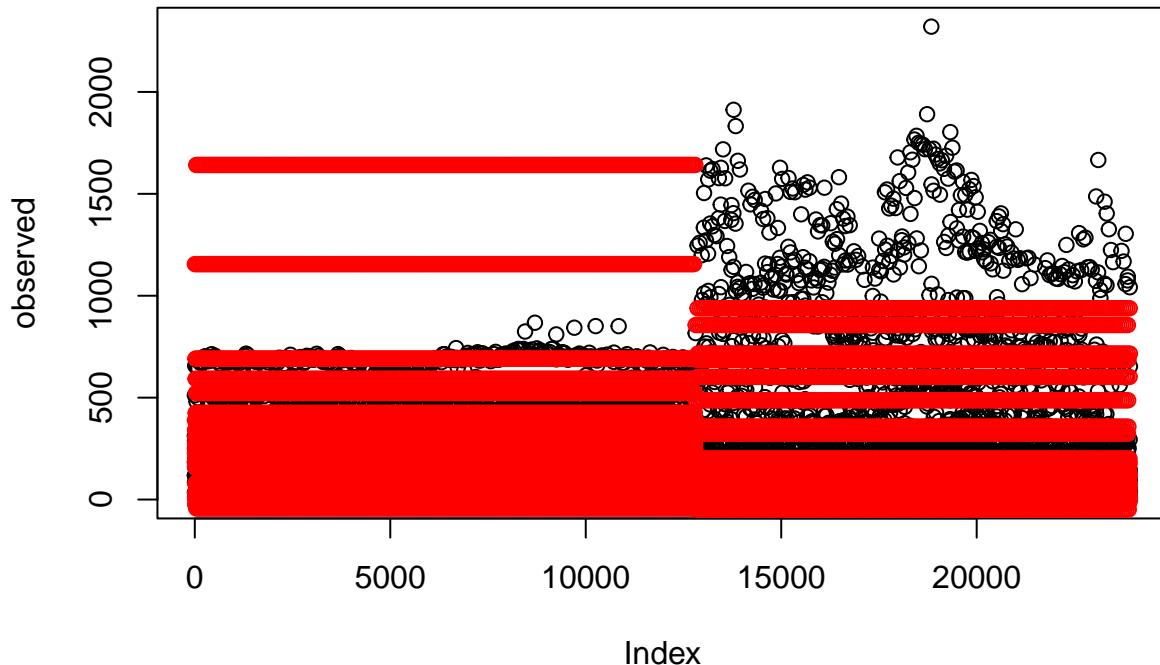
## [1] "Srednja absolutna napaka: 159.747548777531"
## [1] "Srednja kvadratna napaka: 45203.454637348"
## [1] "Relativna srednja absolutna napaka: 1"
## [1] "Relativna srednja kvadratna napaka: 1"
```



Linearna regresija

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

```
## [1] "Srednja absolutna napaka: 110.363642505599"
## [1] "Srednja kvadratna napaka: 53188.6419254433"
## [1] "Relativna srednja absolutna napaka: 0.690862822936296"
## [1] "Relativna srednja kvadratna napaka: 1.17664993421757"
```

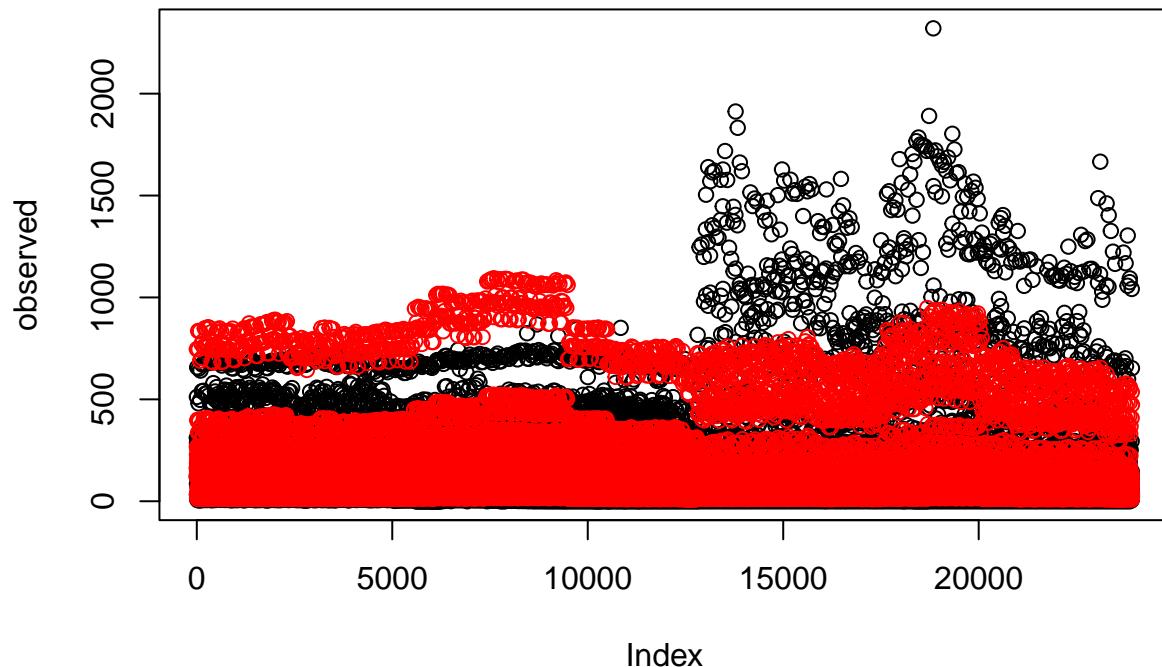


```
| mae| mse| rmae| rmse| |-----:|-----:|-----:|-----:| | 110.3636| 53188.64| 0.6908628| 1.17665|
```

```
# 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: 77.7227673827236"
## [1] "Srednja kvadratna napaka: 22412.6417979647"
## [1] "Relativna srednja absolutna napaka: 0.514398611699836"
## [1] "Relativna srednja kvadratna napaka: 0.354427436673099"
```

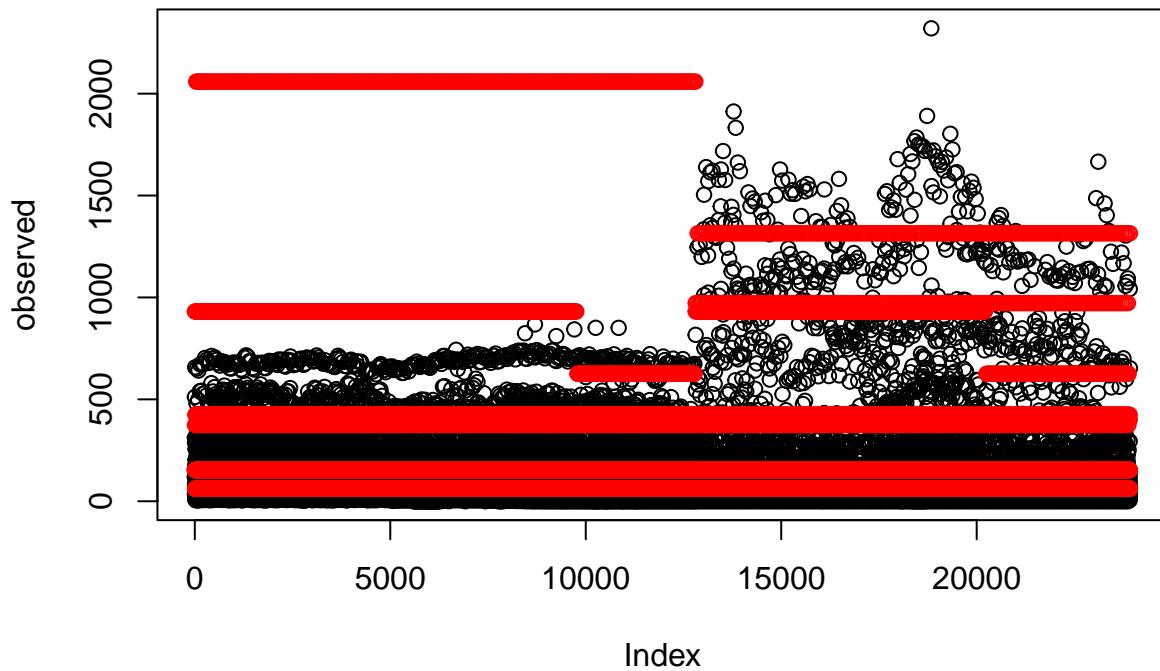


```
| mae| mse| rmae| rmse| |-----:|-----:|-----:| | 77.72277| 22412.64| 0.5143986| 0.3544274|
```

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

Regresijsko drevo

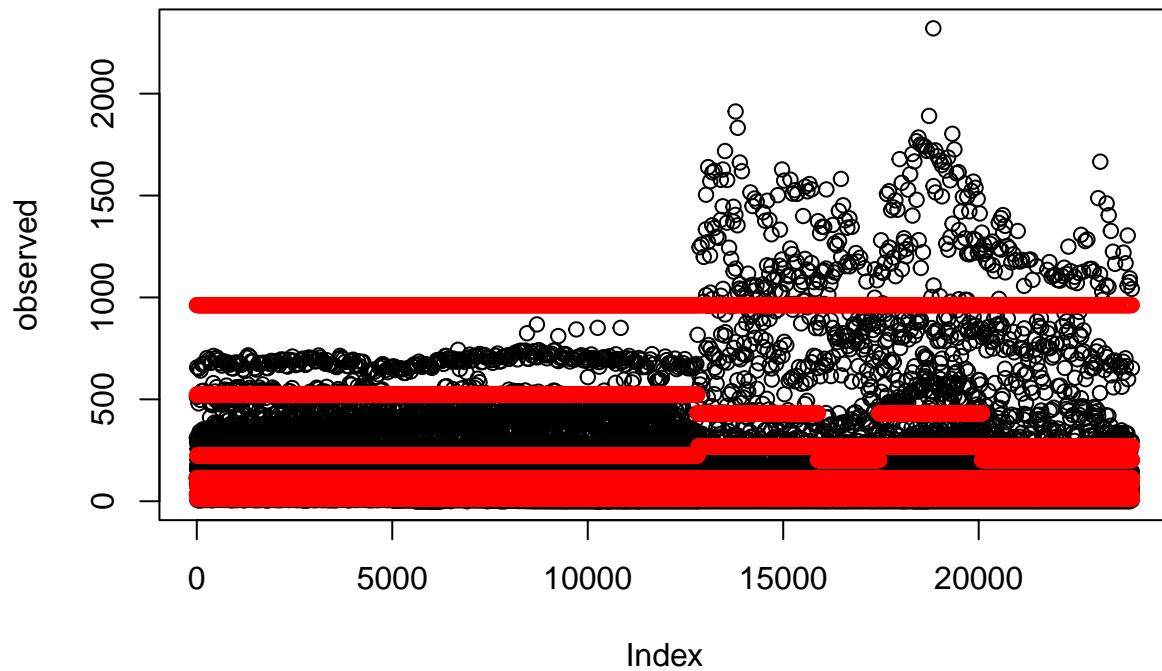
```
## [1] "Srednja absolutna napaka: 121.845330384628"
## [1] "Srednja kvadratna napaka: 83541.7481227028"
## [1] "Relativna srednja absolutna napaka: 0.762736776351506"
## [1] "Relativna srednja kvadratna napaka: 1.84812751133583"
```



```
| mae| mse| rmae| rmse| |-----:|-----:|-----:|-----:| 121.8453| 83541.75| 0.7627368| 1.848127|
```

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

```
## [1] "Srednja absolutna napaka: 100.631472462449"
## [1] "Srednja kvadratna napaka: 34817.9613083123"
## [1] "Relativna srednja absolutna napaka: 0.666017069015228"
## [1] "Relativna srednja kvadratna napaka: 0.550601793752346"
```

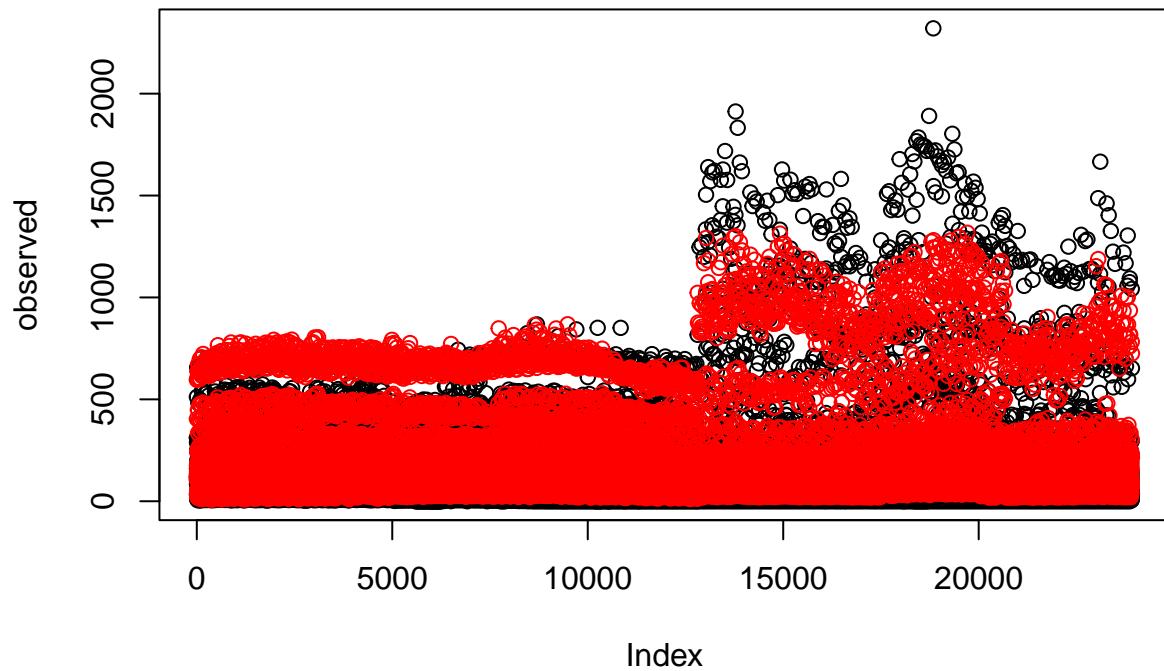


```
| mae| mse| rmae| rmse| |-----:|-----:|-----:|-----:| | 100.6315| 34817.96| 0.6660171| 0.5506018|
```

Nakljucni gozd

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

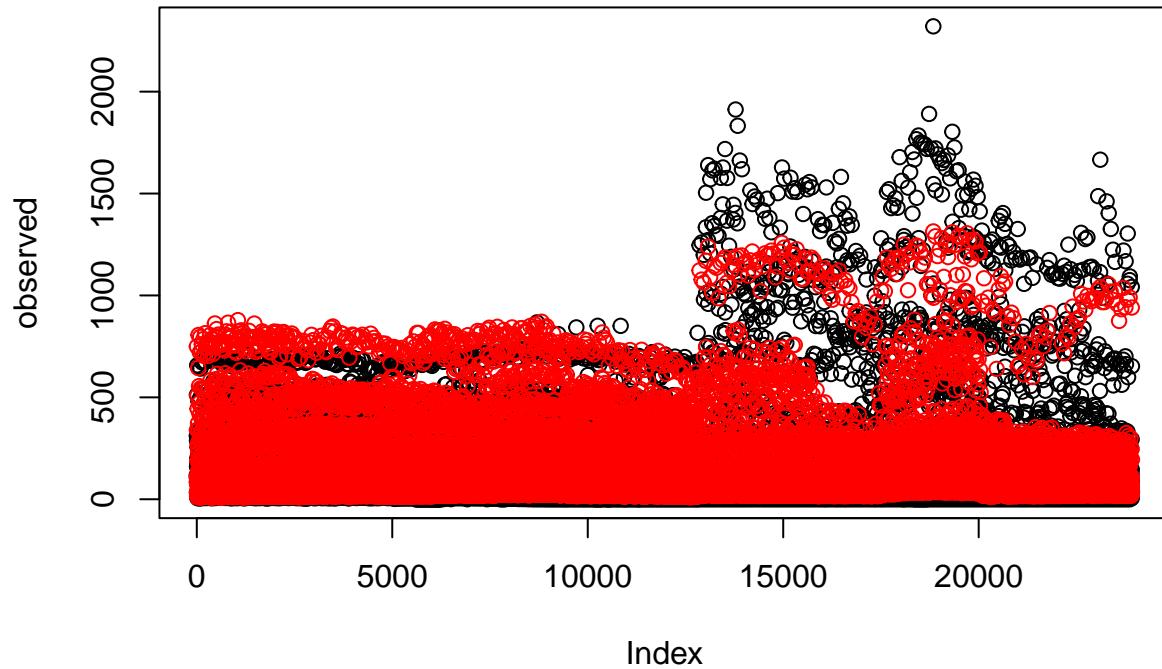
```
## [1] "Srednja absolutna napaka: 93.6193692576685"
## [1] "Srednja kvadratna napaka: 24726.6808763871"
## [1] "Relativna srednja absolutna napaka: 0.586045732620578"
## [1] "Relativna srednja kvadratna napaka: 0.547008653979235"
```



```
| mae| mse| rmae| rmse| |-----:|-----:|-----:| 93.61937| 24726.68| 0.5860457| 0.5470087|
```

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

```
## [1] "Srednja absolutna napaka: 82.7411654815375"
## [1] "Srednja kvadratna napaka: 23988.7139890089"
## [1] "Relativna srednja absolutna napaka: 0.547612264557502"
## [1] "Relativna srednja kvadratna napaka: 0.379351014701916"
```



```
| mae| mse| rmae| rmse| |-----:|-----:|-----:|-----:| 82.74117| 23988.71| 0.5476123| 0.379351|
```

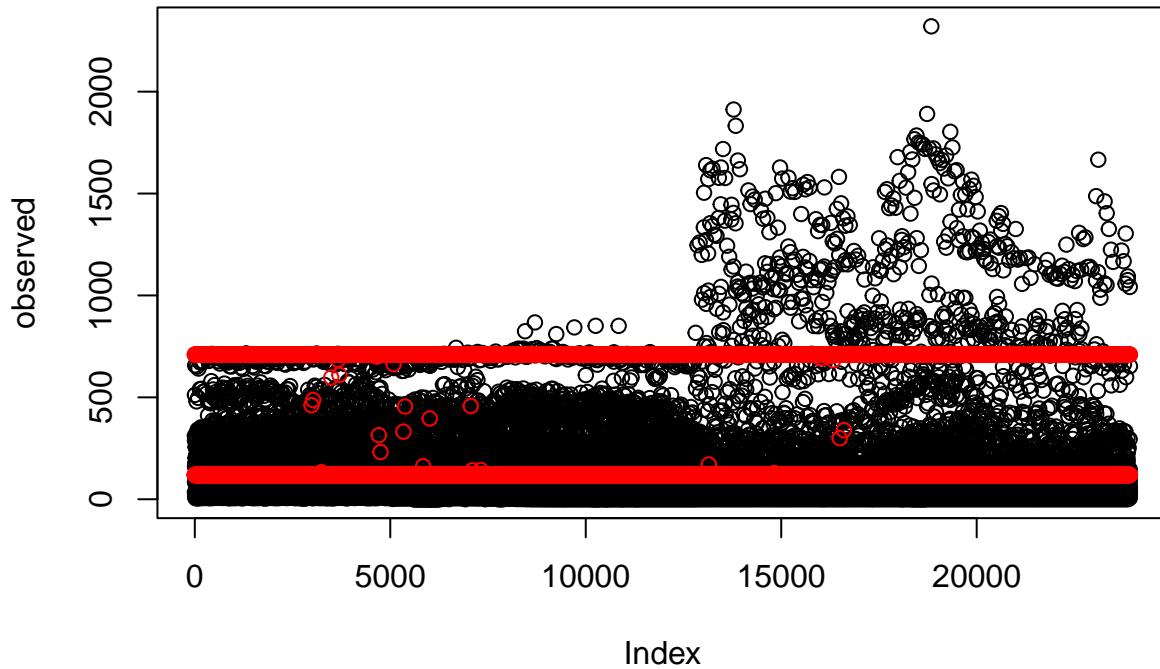
Nevronske mreze

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

## # weights:  91
## initial  value 3632707622.714557
## final    value 1197647050.947654
## converged

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

## [1] "Srednja absolutna napaka: 131.781896546149"
## [1] "Srednja kvadratna napaka: 41878.3380437743"
## [1] "Relativna srednja absolutna napaka: 0.824938457926985"
## [1] "Relativna srednja kvadratna napaka: 0.926441095702754"
```



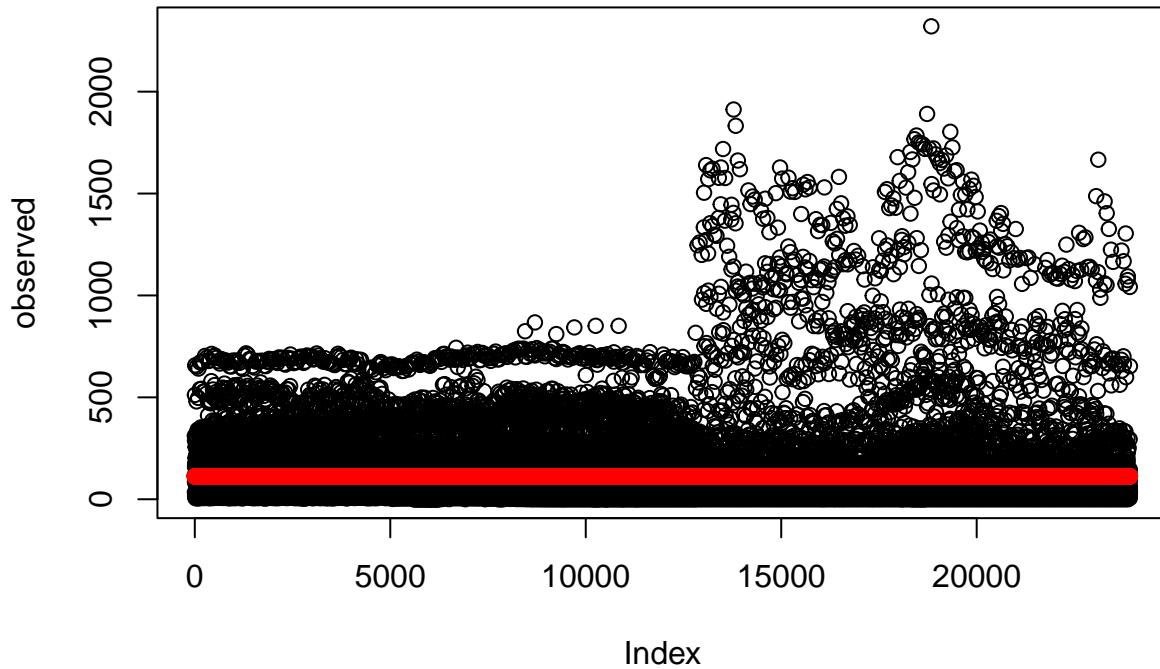
```
| mae| mse| rmae| rmse| |-----:|-----:|-----:|-----:| 131.7819| 41878.34| 0.8249385| 0.9264411|
```

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

```
## # weights: 121
## initial value 509475.804912
## final value 35268.065177
## converged
```

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

```
## [1] "Srednja absolutna napaka: 116.152684020241"
## [1] "Srednja kvadratna napaka: 42339.8300532952"
## [1] "Relativna srednja absolutna napaka: 0.768742305726277"
## [1] "Relativna srednja kvadratna napaka: 0.669550585345399"
```



```
| mae| mse| rmae| rmse| |-----:|-----:|-----:|-----:| 116.1527| 42339.83| 0.7687423| 0.6695506|
```

Izboljsava klasifikacijskih modelov

Metoda ovojnica

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

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

```
## best model: estimated error = 0.007502428 , selected feature subset = namembnost ~ povrsina + leto_izgradnje
```

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

brier	ca	infGain
0.9267998	0.5307692	0.634058

Glasovanje

Zgradimo modele z osnovno in popravljeno mnozico atributov:

```

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

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

```

```
## [1] "Classification accuracy: 0.546989966555184"
```

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$test$namembnosc)

```

```
## [1] "Classification accuracy: 0.56128762541806"
```

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$namembnosc)

## [1] "Classification accuracy: 0.577508361204013"

```

Glasovanje z popravljenim 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$namembnosc)
```

```
## [1] "Classification accuracy: 0.574289297658863"
```

Bagging

Bagging z osnovno mnozico atributov:

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

```
## [1] "Classification accuracy: 0.505476588628763"
```

Bagging z popravljenim mnozico atributov:

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

```
## [1] "Classification accuracy: 0.496655518394649"
```

Boosting

Boosting z osnovno mnozico atributov:

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

```
## [1] "Classification accuracy: 0.509782608695652"
```

Boosting z popravljenim mnozico atributov:

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

```
## [1] "Classification accuracy: 0.501170568561873"
```

Primerjava po regijah

Priprava podatkov

Pripravimo podatke, tako da učno in testno množico razbijemo na dve podmnožici: - množica ki vsebuje samo primere z vzhodno regijo - množica 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

Evaluacija

Klasifikacija Zgradimo nekaj klasifikacijskih modelov, ki se učijo iz posamezne podmnožice, ter vsakega posebej se ocenimo glede na testne primere iz ustrezne testne množice.

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

```
## [1] "Trivial classification accuracy: 0.518271827182718"
## [1] "odlocitveno drevo classification accuracy: 0.703690369036904"
```

```

## [1] "naivni bayes classification accuracy: 0.621962196219622"
## [1] "k-najblizjih sosedov classification accuracy: 0.728442844284428"
## [1] "nakljucni gozd classification accuracy: 0.762916291629163"

```

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

```

## [1] "Trivial classification accuracy: 0.428571428571429"
## [1] "odlocitveno drevo classification accuracy: 0.333333333333333"
## [1] "naivni bayes classification accuracy: 0.325448868071819"
## [1] "k-najblizjih sosedov classification accuracy: 0.413505074160812"
## [1] "nakljucni gozd classification accuracy: 0.35480093676815"

```

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

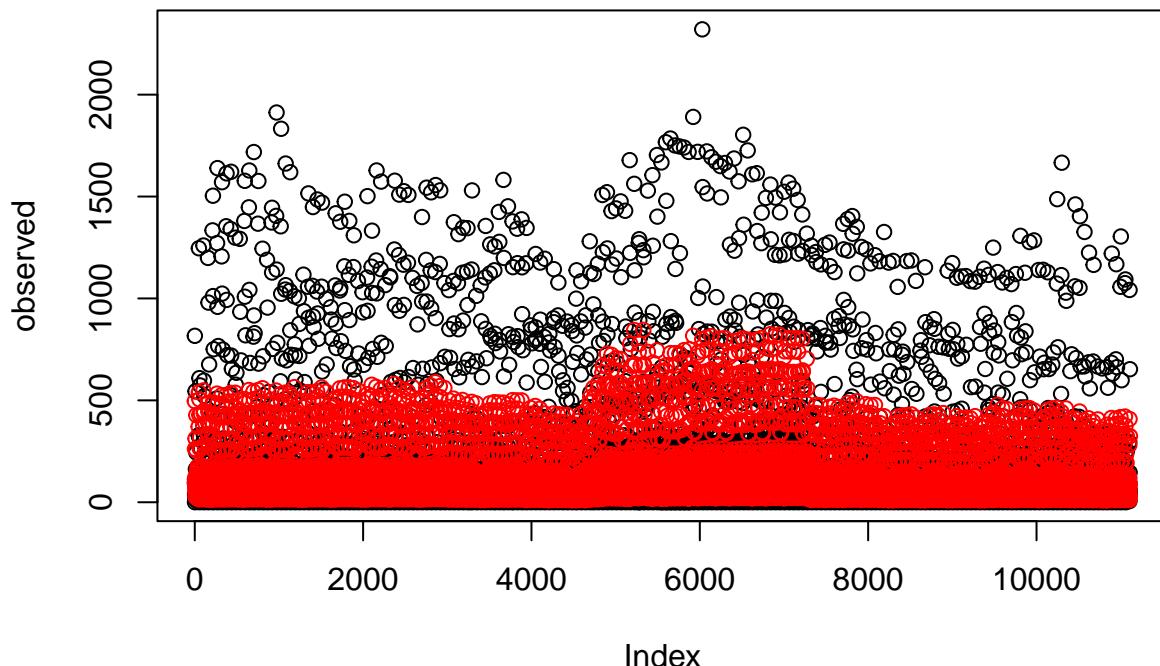
```
runRegression(poraba ~ ., regVzhodnaTrain, regVzhodnaTest)
```

```

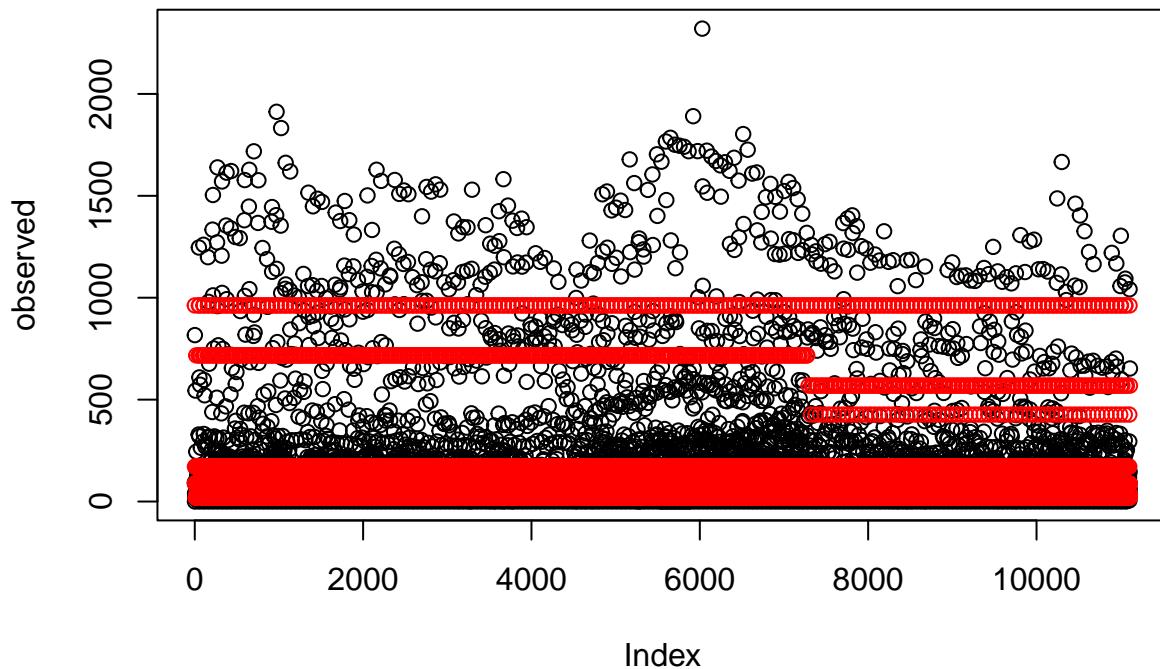
## Warning in predict.lm(model, test): prediction from a rank-deficient fit may be
## misleading

## [1] "Srednja absolutna napaka: 67.7616533496973"
## [1] "Srednja kvadratna napaka: 25517.5726135063"
## [1] "Relativna srednja absolutna napaka: 0.461324563256286"
## [1] "Relativna srednja kvadratna napaka: 0.311131917614226"

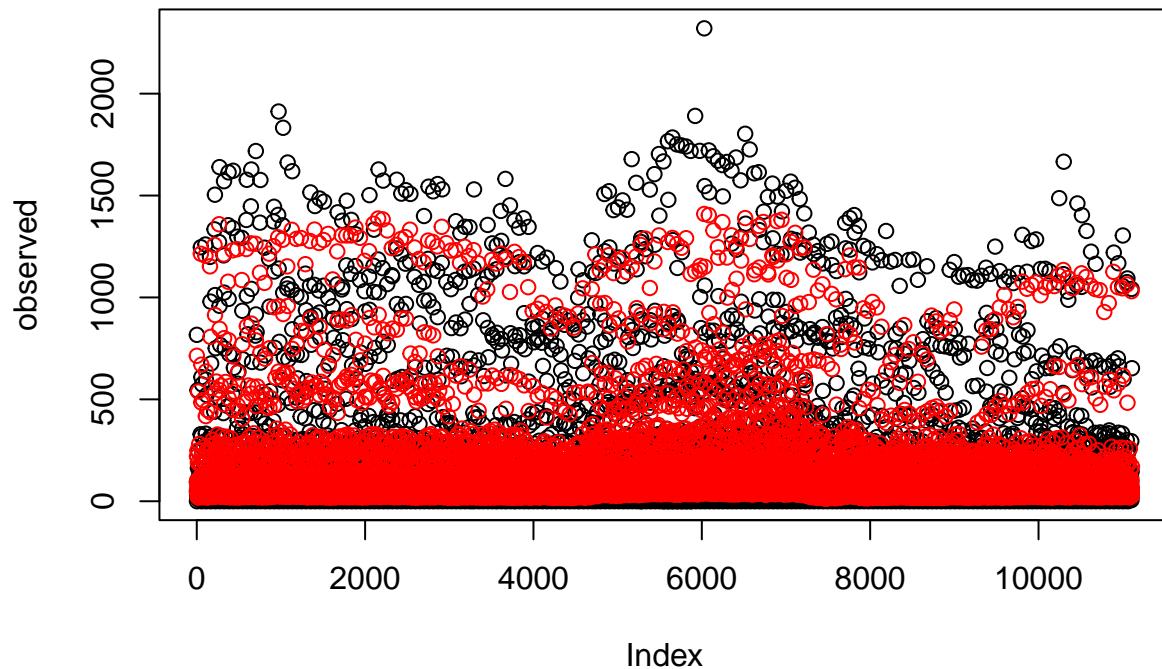
```



```
## [1] "Srednja absolutna napaka: 67.3895816272914"  
## [1] "Srednja kvadratna napaka: 19293.2594760883"  
## [1] "Relativna srednja absolutna napaka: 0.458791481249668"  
## [1] "Relativna srednja kvadratna napaka: 0.235239805476127"
```



```
## [1] "Srednja absolutna napaka: 65.1763723882061"  
## [1] "Srednja kvadratna napaka: 19091.7136390358"  
## [1] "Relativna srednja absolutna napaka: 0.443723847342529"  
## [1] "Relativna srednja kvadratna napaka: 0.232782387456043"
```

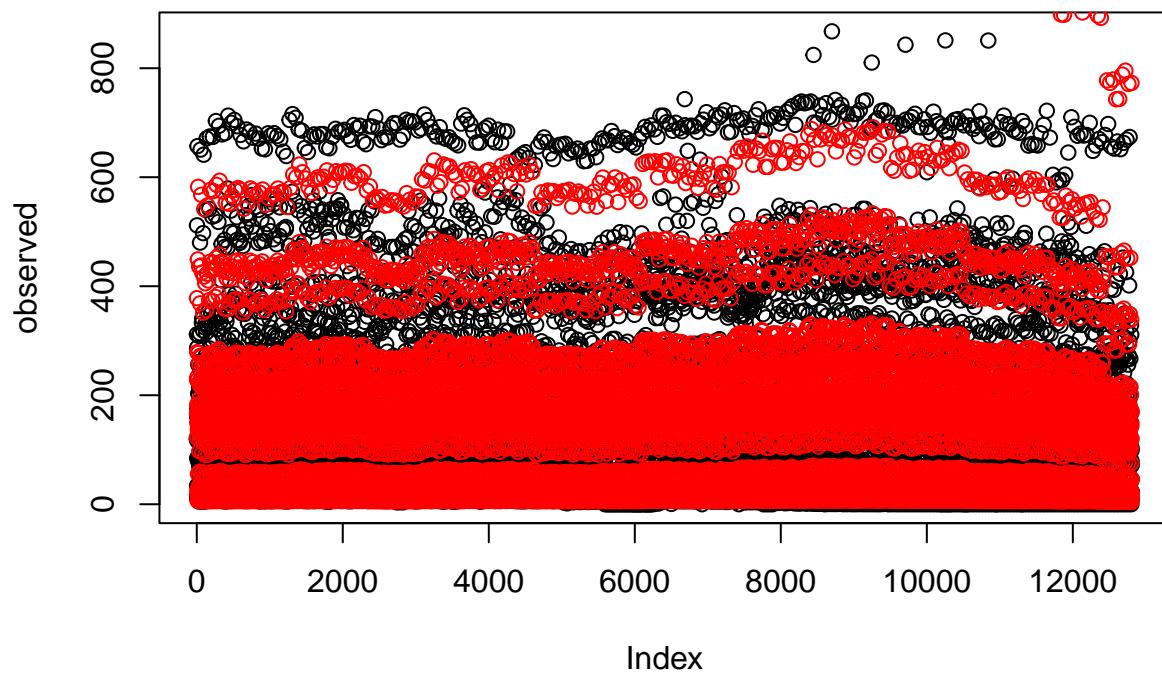


```
| mae| mse| rmae| rmse| |————:|————:|————:|————:| 65.17637| 19091.71| 0.4437238| 0.2327824|
```

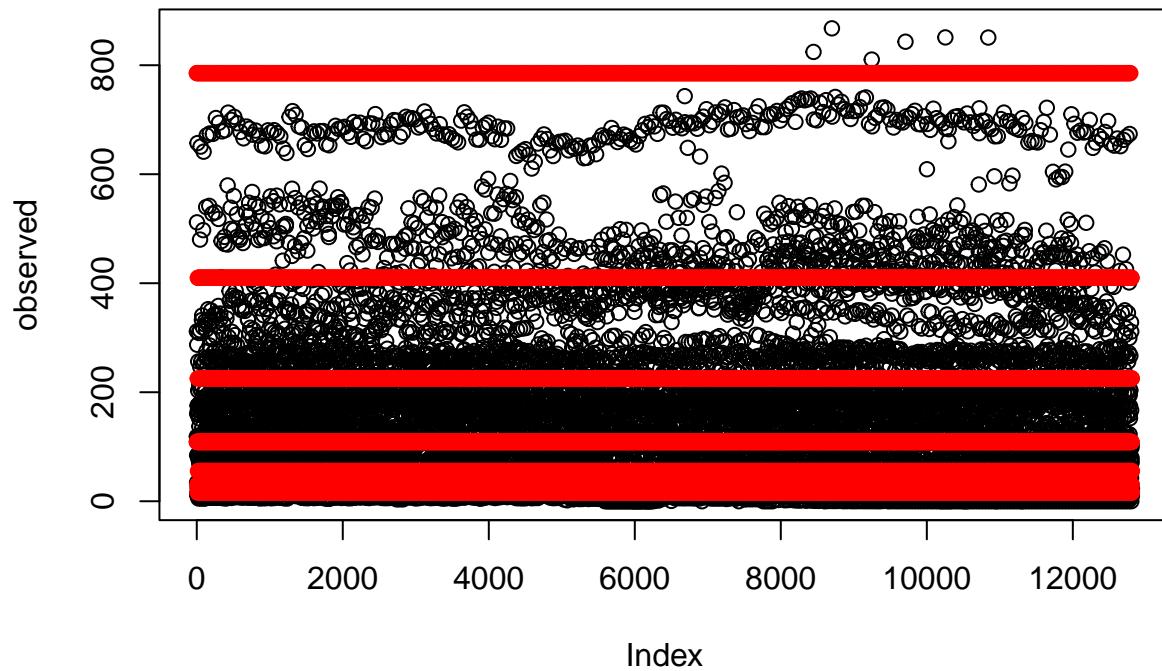
```
runRegression(poraba ~ ., regZahodnaTrain, regZahodnaTest)
```

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

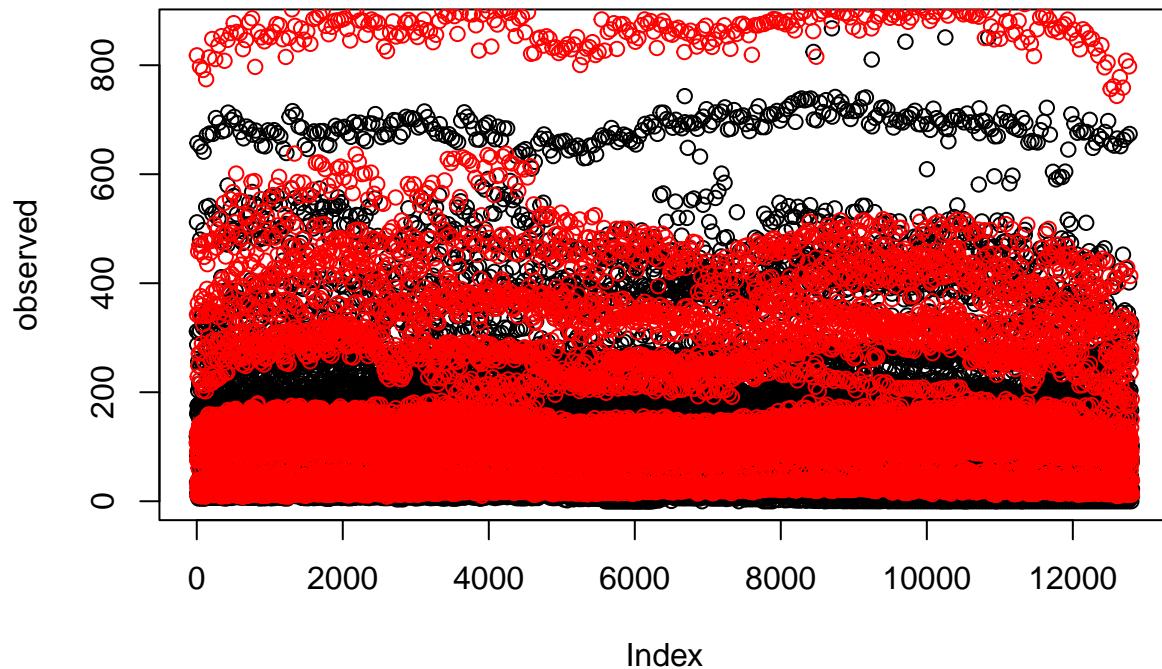
```
## [1] "Srednja absolutna napaka: 110.143235179218"
## [1] "Srednja kvadratna napaka: 48474.6406636499"
## [1] "Relativna srednja absolutna napaka: 0.711753941078423"
## [1] "Relativna srednja kvadratna napaka: 1.03246045221538"
```



```
## [1] "Srednja absolutna napaka: 104.763891123982"
## [1] "Srednja kvadratna napaka: 29439.2602166482"
## [1] "Relativna srednja absolutna napaka: 0.676992211722084"
## [1] "Relativna srednja kvadratna napaka: 0.627026245064239"
```



```
## [1] "Srednja absolutna napaka: 93.925370706793"
## [1] "Srednja kvadratna napaka: 24137.3199303643"
## [1] "Relativna srednja absolutna napaka: 0.606952870587424"
## [1] "Relativna srednja kvadratna napaka: 0.514100319453399"
```



| mae| mse| rmae| rmse| |————:|————:|————:|————:| | 93.92537| 24137.32| 0.6069529| 0.5141003|

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]]

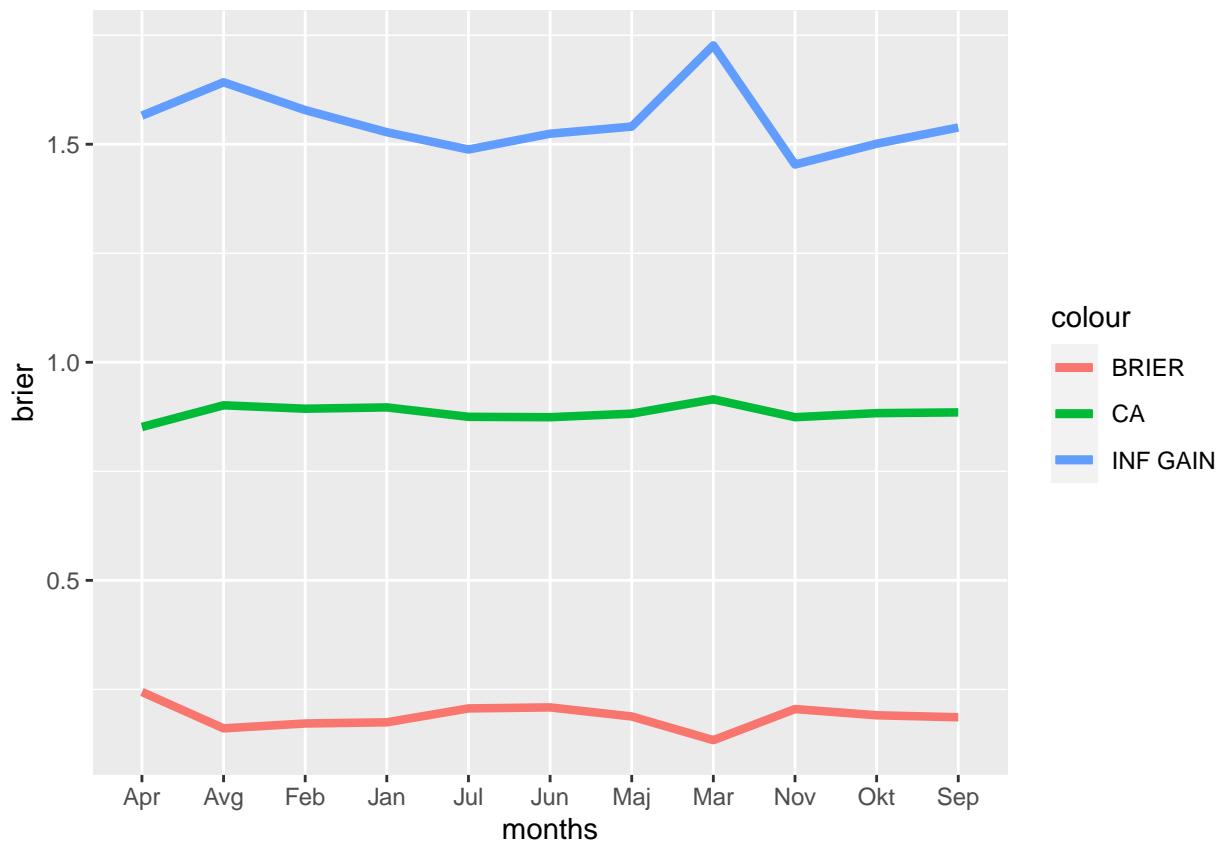
  dt <- rpart(namembnost ~ ., data=classTrain)
  score <- EvaluateClassModel(dt, classTrain, classTest, F)
  brier[i] <- score$brier
  ca[i] <- score$ca
  infGain[i] <- score$infGain

  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
}

```

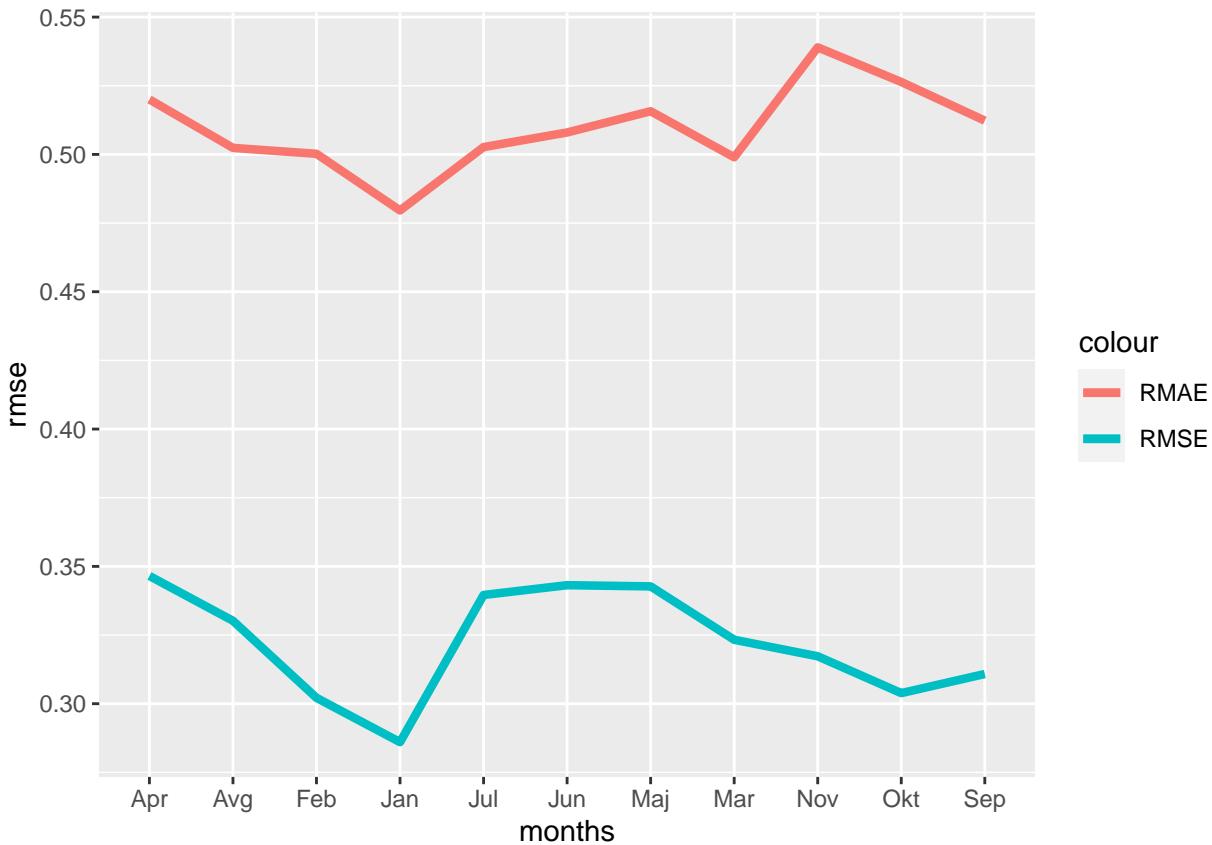
Ocene klasifikacije

```
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 nevronskeih 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.