Seminarska naloga

Umetna inteligenca

David Šeruga, Žan Hočevar

December 2021

Kazalo

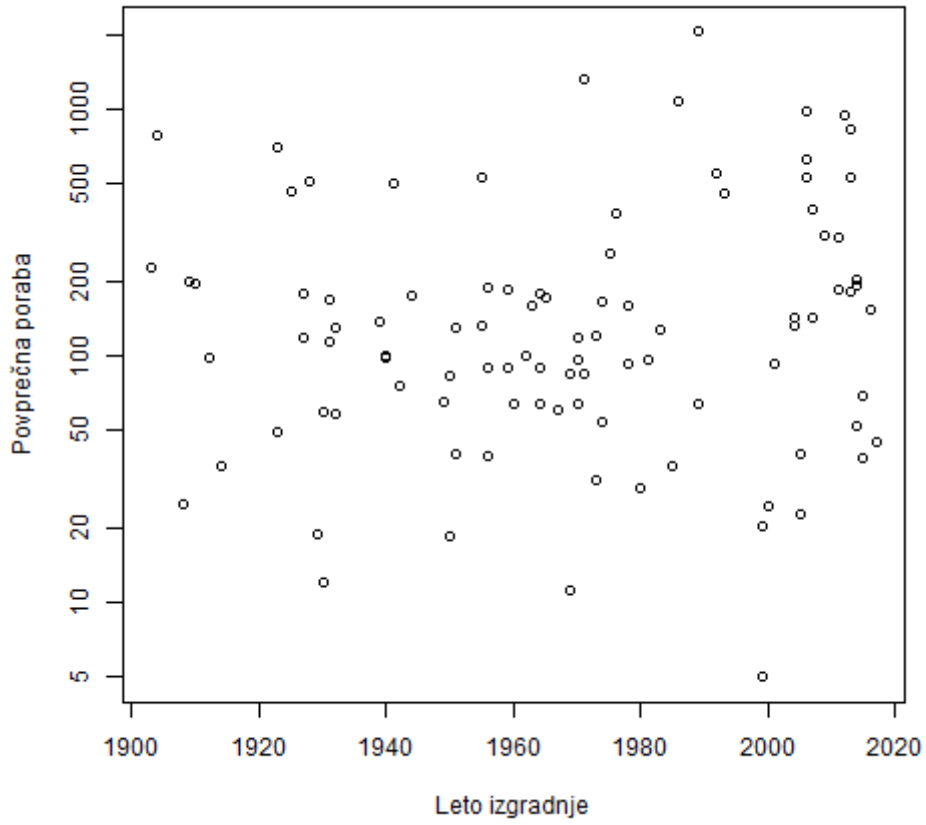
Uvod

To je seminarska naloga v kateri bova predstavila uporabo strojnega učenja za gradnjo modelov, ki bodo napovedovali porabo električne energije in namembnosti stavb. V sklopu prve naloge bova naredila vizualizacijo podatkov, ki smo jih prejeli, v obliki grafov. Nato bova pogledala in ocenila najine atribute z različnimi algoritmi, ter dodala še nekaj svojih izpeljanih atributov. In ko imava pripravljene vse nove atribute in podatke, lahko uporabiva različne algoritme za gradnjo modelov. Ti bodo napovedovali tako klasifikacijo kot regresijo. Po določeni napovedi, ki jo bomo naredili pa bova tudi testirala in ocenjevala kvaliteto napovedovanja.

1. Vizualizacija podatkov

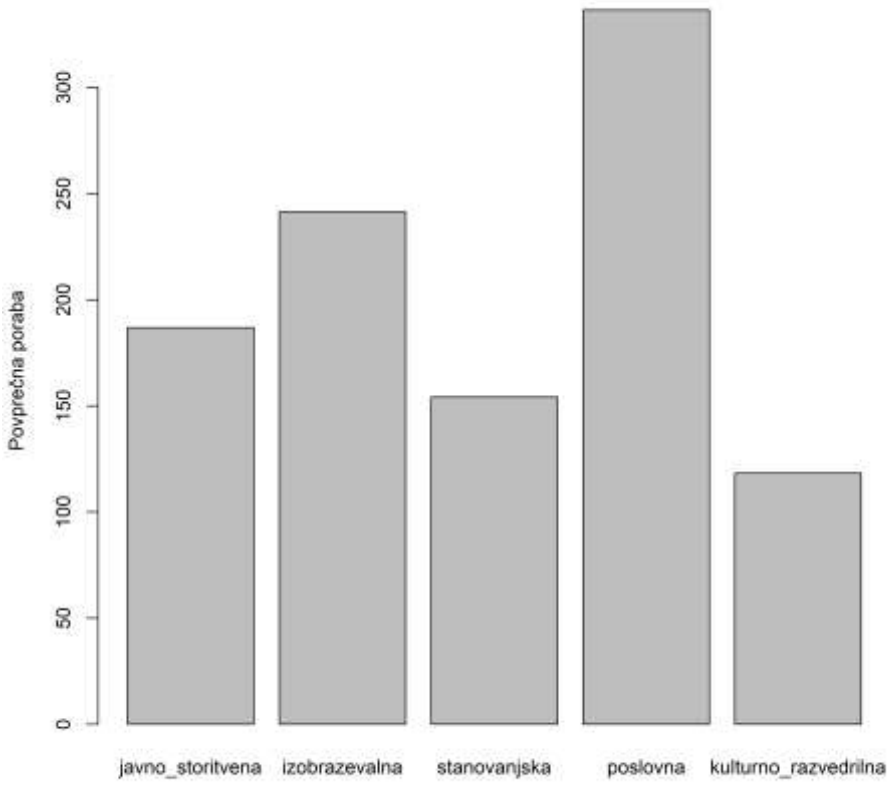
Spodaj bova analizirala podatke, tako da nariševa nekaj grafov, ki so se nama zdeli smiselni. Vsak graf bo imel tudi svoj opis.

1.1 Povprečna poraba za vsako stavbo urejena po letih



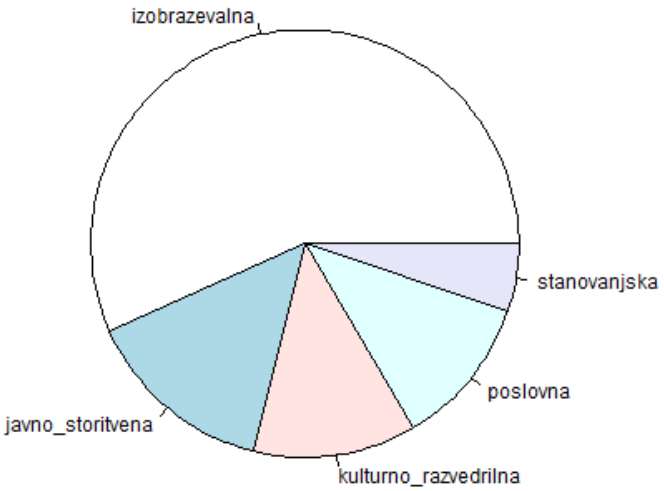
Poraba elektrike je velikokrat povezana s klimatiziranimi prostori in na tem grafu sva hotela pokazati, da starejše stavbe, ki imajo slabšo izolacijo kot novejše, imajo tudi večjo porabo, ampak sva pozabila, da se stavbe razlikujejo po namembnosti in po površini, ki znata biti faktorja pri temu.

1.2 Povprečna poraba na namembnost



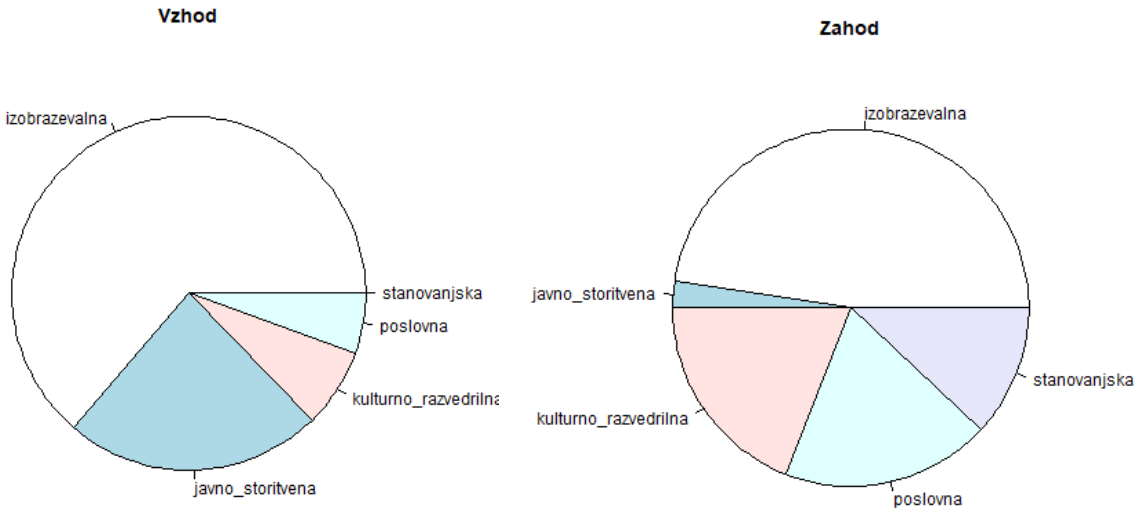
Ta graf nam pokaže kako se stavbe različne namembnosti primerjajo po porabi. Tu lahko vidimo, da imajo stavbe s poslovno namembnostjo najvišjo porabo.

1.3 Delež stavb po namembnosti



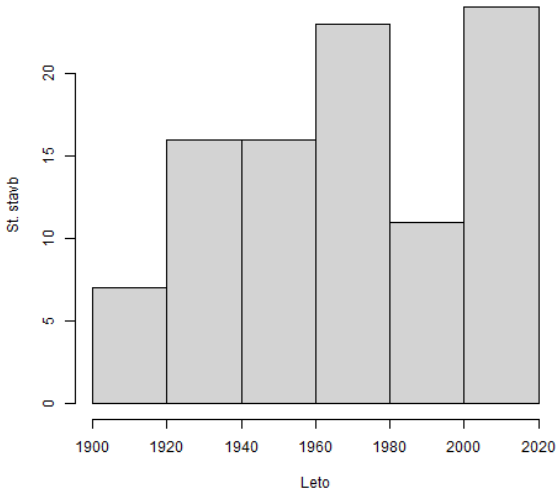
Na temu grafu vidimo, kakšen delež stavb je v določeni kategoriji namembnosti. Za to je tortni graf najboljši.

1.4 Delež stavb po namembnosti glede na regijo



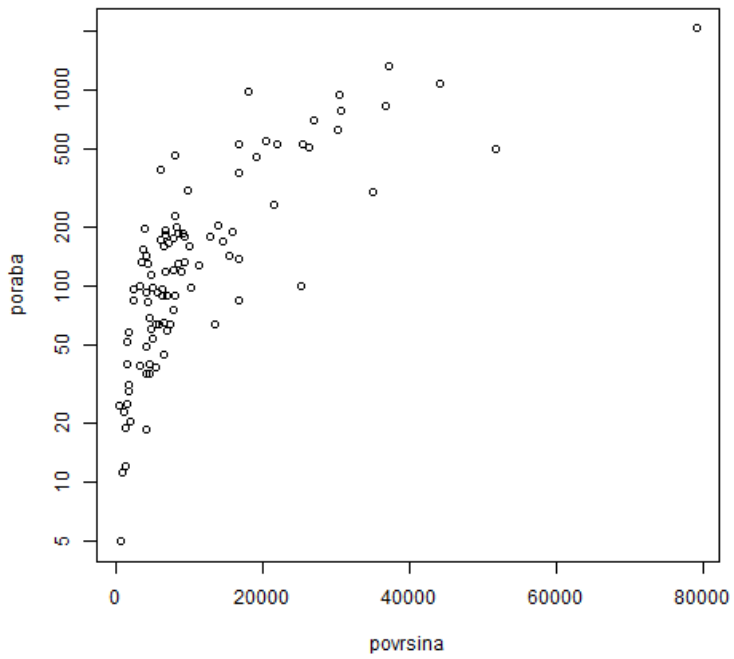
Tu lahko primerjamo deleže stavb po regijah. Kot vidimo je še vedno največ stavb izobraževalnih v obeh regijah. Opazimo pa ekstreme, kot so javno storitvena namembnost, kjer je večina stavb v vzhodni regiji ali pa stavbe z stanovanjsko namembnostjo, ki praktično ne obstajajo v vzhodni regiji.

1.5 Število stavb razdeljenih po letih



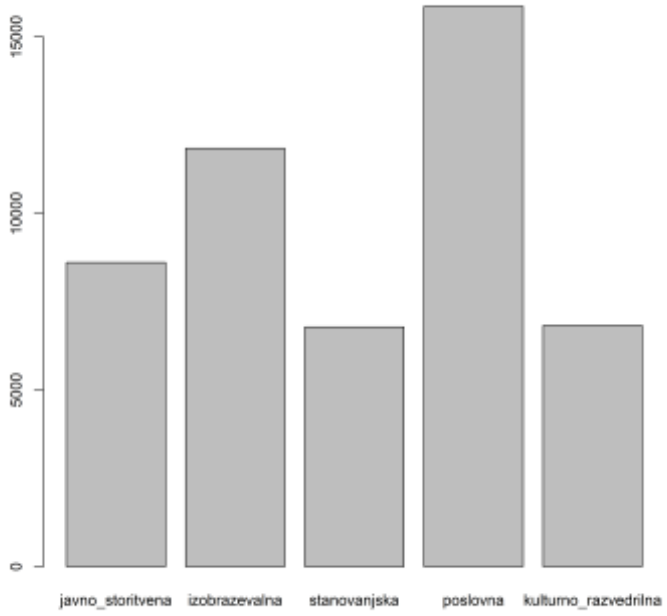
Tu lahko vidimo, koliko stavb je bilo narejenih v nekem časovnem obdobju dvajsetih let. Opazimo, da je največ stavb bilo zgrajenih med letom 2000 in 2020.

1.6 Poraba glede na površino stavbe



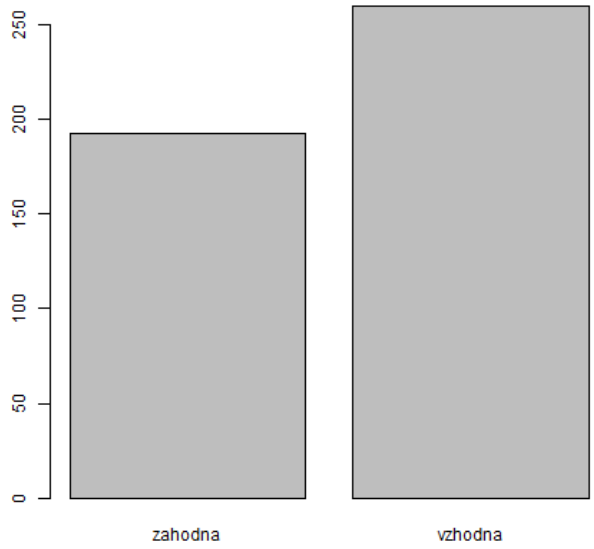
Na temu grafu lahko vidimo, da večja kot je stavba, večja je tudi poraba. Po hitrem pogledu zgleda, zgleda da je poraba logaritmično vezana na površino, ampak ker skala na levi ni linearna, ne moremo tega predvidevati. Poraba gleda bolj linearno vezana na površino.

1.7 Povprečna površina stavbe glede na namembnost



Graf pokaže povprečno površino stavbe glede na namembnost in iz njega jasno razberemo, da so poslovne stavbe največje. Tu lahko opazimo povezavo iz prejšnjih grafov, kjer vidimo, da so poslovne stave največje in večja kot je stavba, večjo ima porabo in zato imajo tudi poslovne stavbe največjo porabo.

1.8 Povprečna poraba na regijo



Vidimo, da ima vzhodna regija večjo povprečno porabno glede na stavbo, kar je zanimivo saj imajo manj poslovnih stavb, je pa res, da imajo več izobraževalnih stavb, ki pa so 2 največji porabniki, takoj za poslovnimi stavbami.

2. Ocenjevanje in konstrukcija atributov

Dani podatki imajo 14 atributov. Te sva se odločila, da bova prvo uredila, tako da sva nekaj atributov tudi faktorizirala. Faktorizirala sva atribute, ki imajo omejeno število vrednosti, kot so regije ali namembnost in ostali…

2.1 Ocenjevanje atributov

Potem pa sva te atribute ocenila z funkcijo 'attrEval', v kateri sva uporabila različne funkcije za ocenjevanje, kot so MDL, Gini, Gain ratio, …

```
library(CORElearn)

sort(attrEval(namembnost ~ ., ucni_podatki, "InfGain"), decreasing = TRUE)

sort(attrEval(namembnost ~ ., ucni_podatki, "Gini"), decreasing = TRUE)

sort(attrEval(namembnost ~ ., ucni_podatki, "GainRatio"), decreasing =
TRUE)

sort(attrEval(namembnost ~ ., ucni_podatki, "MDL"), decreasing = TRUE)

sort(attrEval(namembnost ~ ., ucni_podatki, "ReliefFequalK"), decreasing =
TRUE)
```

Vse funkcije nam atribute razvrstijo podobno, z majhnimi razlikami. Tri od petih funkcij označijo površino, kot najbolj pomemben atribut.

2.2 Konstrukcija novih atributov

Ustvarila sva nove atribute kot so:

- Mesec

```
ucni_podatki$mesec <- as.numeric(format(ucni_podatki$datum, "%m"))

ucni_podatki$mesec <- as.factor(ucni_podatki$mesec)
```

- Vikend

```
library(chron)

ucni_podatki$vikend <- is.weekend(ucni_podatki$datum)

ucni_podatki$vikend <- as.factor(ucni_podatki$vikend)
```

- Letni časi

```
for (i in 1:nrow(ucni_podatki)) {

  if (ucni_podatki$mesec[i] == 12 || ucni_podatki$mesec[i] == 1 ||
ucni_podatki$mesec[i] == 2) {

    ucni_podatki$letni_cas[i] <- "zima"

  } else if (ucni_podatki$mesec[i] == 3 || ucni_podatki$mesec[i] == 4 ||
ucni_podatki$mesec[i] == 5) {

    ucni_podatki$letni_cas[i] <-"pomlad"

  } else if (ucni_podatki$mesec[i] == 6 || ucni_podatki$mesec[i] == 7 ||
ucni_podatki$mesec[i] == 8) {

    ucni_podatki$letni_cas[i] <- "poletje"

  } else {

    ucni_podatki$letni_cas[i] <- "jesen"

  }

}

ucni_podatki$letni_cas <- as.factor(ucni_podatki$letni_cas)
```

- Povprečna poraba prejšnjega dne za določeno stavbo

```
povp_prejsni_dan <- NULL

for (i in 1:nrow(ucni_podatki)) {

  povp_prejsni_dan[i] <-
mean(as.numeric(ucni_podatki$poraba[ucni_podatki$stavba ==
ucni_podatki$stavba[i] & (ucni_podatki$datum < ucni_podatki$datum[i] |
ucni_podatki$datum > ucni_podatki$datum[i]-2)]))

}

ucni_podatki$povp_prejsni_dan <- NULL

ucni_podatki$povp_prejsni_dan <- povp_prejsni_dan
```

- Povprečna poraba prejšnjega tedna za določeno stavbo

```
povp_prejsni_teden <- NULL

for (i in 1:nrow(ucni_podatki)) {

  povp_prejsni_teden[i] <-
mean(as.numeric(ucni_podatki$poraba[ucni_podatki$stavba ==
ucni_podatki$stavba[i] & (ucni_podatki$datum < ucni_podatki$datum[i] |
ucni_podatki$datum > ucni_podatki$datum[i]-8)]))

}

ucni_podatki$povp_prejsni_teden <- NULL

ucni_podatki$povp_prejsni_teden <- povp_prejsni_teden
```

Ko smo dodala te atribute, jih lahko ponovno ocenimo na isti način kot smo jih prej. Opazimo lahko, da se ni kaj dosti spremenilo. Dodatni atributi bodo najbrž bolj vplivali na regresijski problem.

3. Kreiranje modelov napovedovanja in njihova evaluacija

V tej seminarski nalogi rešujeva 2 problema. Napovedujeva namembnost stavbe, ki je klasifikacijski problem in napovedujemo porabo, ki pa je regresijski problem. Za to se uporabijo različni učni algoritmi in pri vsakem problemu bova uporabila vsaj tri učne algoritme iz tiste kategorije, katero napovedujeva. Za testiranje napovedi modelov, pa prej rabimo definirati nekaj testnih funkcij, ki jih bomo kasneje uporabili.

3.1 Klasifikacija

Tukaj napovedujemo namembnost stavbe. Za napoved namembnosti bomo prvo ugotovili kako natančen je večinski klasifikator. Če naredimo 'summay()' nad učnimi podatki vidimo, da je najbolj pogosta namembnost 'izobraževalna'. Potem pa lahko pogledamo delež zadetka za učno množico in testno množico. Številka se giblje nekje okoli 0,5 natančnosti.

```
> vec_klas <- sum(ucni_podatki$namembnost == "izobrazevalna") / length(ucni_podatki$namembnost)
> print(vec_klas)
[1] 0.5513368
> vec_klas <- sum(testni_podatki$namembnost == "izobrazevalna") / length(testni_podatki$namembnost)
> print(vec_klas)
[1] 0.4702341
```

Izbrala sva tri klasifikacijske algoritme:

- Odločitveno drevo
- Naključni gozd
- Naivni Bayes

3.1.1 Odločitveno drevo

Pri odločitvenem drevesu sva uporabila knjižnico 'rpart', ki nama je zgradila drevo, ki pa sva ga nato uporabila, da sva napovedala namembnost. Napovedovala sva z vsemi atributi, z izbranimi atributi in nato še z porezanim drevesom z izbranimi atributi. Za ocenjevanje točnosti napovedovanja pa sva uporabila funkcije CA, brier.score in inf.score. Rezultati izgledajo nekako tako:

Atributi	CA	Brier.score	Inf.score
Vsi	0.4966555	0.9950272	0.5367325
Izbrani	0.5434783	0.8767163	0.7091357
Izbrani z porezanim drevesom	0.5570234	0.885451	0.7775018

3.1.2 Naključni gozd

Pri naključnem gozdu sva uporabila knjižnico 'CORElearn', s katero sva naredila model, ki je napovedoval namembnost. Pri gradnji modela sva uporabila vse atribute in izbrane atribute,

testirala pa sva z funkcijami CA in brier.score. Ta model se je izkazal za malo boljšega kot odločitveno drevo.

Atributi	CA	Brier.score
Vsi	0.5547659	0.6294118
Izbrani	0.5726589	0.6619525

3.1.3 Naivni Bayes

Prav tako kot pri prejšnjem modelu sva tega zgradila z knjižnico 'CORElearn' in z njim napovedala namembnost. Pri gradnji modela sva uporabila vse atributi in samo izbrane, napovedi pa sva testirala z funkcijama CA in brier.score. Izkazalo se je, da je Naivni Bayes še najslabši izmed vseh teh modelov.

Atributi	CA	Brier.score
Vsi	0.4388796	0.8094718
Izbrani	0.5263796	0.6823086

3.2 Regresija

Pri regresiji bomo napovedovali porabo. Za preverjanje natančnosti napovedi pa bomo tukaj uporabili drugačne funkcije. Te so:

- Mae
- Mse
- Rmae
- Rmse

Tukaj pa bomo uporabili 4 algoritme za gradnjo modelov:

- Linearna regresija
- Regresijsko drevo
- Naključni gozd
- K-najbližjih sosedov

3.2.1 Linearna regresija

Tukaj sva zgradila tri modele. Model z vsemi atributi, z izbranimi atributi in z izbranimi atributi z wrapperjem. Prišla sva do zanimivih odkritij. Wrapper nama je izbral vse atribute in zaradi tega napove isto, kot če preprosto vzameva vse atribute. Če pri wrapperju odstranim vse atribute razen najpomembnejšega, pa še vedno dobim isto oceno. Ta model nama je pri nekaterih vrednostih napovedal 'NaN', in zato sva te vrednosti v napovedani množici spremenila v povprečno porabo iz testnih podatkov. Vse ocene so zelo podobne.

Atributi	Mae	Mse	Rmae	Rmse
Vsi	36.64551	4817.991	0.2293964	0.1065846
Izbrani	33.21662	5127.323	0.207932	0.1134277
Izbrani z wrapperjem	36.64551	4817.991	0.2293964	0.1065846

3.2.2 Regresijsko drevo

Regresijsko drevo se je izkazalo za slabši model kot linearna regresija. Tu smo zgradila drevo z vsemi atributi in preverila njegovo natančnost, nato pa smo še preverila natančnost porezanega drevesa, ki nam je dalo malo boljši rezultat, ampak še vedno slabši od linearne regresije.

Atributi	Mae	Mse	Rmae	Rmse
Vsi	53.87275	7289.31	0.3372368	0.1612556
Porezano	36.16087	5895.206	0.2263626	0.1304149

3.2.3 Naključni gozd

Naključni gozd se je izkazal za najboljši algoritem med vsemi. Bil je najboljši že pri klasifikaciji in pri regresiji ni nič drugače. Ena zanimivost pri njemu pa je, da je delal ta model za napoved regresije neverjetno dolgo. Cca. Uro in pol. Pri tem algoritmu smo naredila model samo z vsemi atributi.

Atributi	Mae	Mse	Rmae	Rmse
Vsi	32.73794	4248.439	0.2049355	0.09398484

3.2.4 K-najbližjih sosedov

Ta algoritem, pa se je med vsemi izkazal za najslabšega. Delal je približno isto dobro kot regresijsko drevo, ki ni porezano. Z tem algoritmom sva prav tako naredila samo en model z vsemi atributi.

Atributi	Mae	Mse	Rmae	Rmse
Vsi	55.49995	7238.434	0.3474229	0.1601301

4. Zaključek

S to seminarsko nalogo sva spoznala kako lahko analizirava podatke in te prikaževa na zanimiv način, kako dodajava in spreminjava atribute, ter kakšno težo nosijo glede na nek drug atribut. No seveda pa na koncu sva se tudi naučila uporabljati različne algoritme za izdelavo modelov za napovedovanje regresijskih in klasifikacijskih vrednosti. Naučila pa sva se tudi izbire določenih atributov za izdelavo modela, ki so nam omogočili narediti boljši model. Te napovedi sva tudi znala preveriti in primerjati med sabo. Sproti pa sva tudi dobro spoznala programski jezik R.