

Umetna inteligenca 2021-2022

Seminarska naloga 1

Janez Sedeljšak, 63200261  
Marko Vrečer, 63200329

Vsebina

[1 Priprava podatkov za konstrukcijo modelov 3](#_Toc89632517)

[1.1 Analiza in izbor novih atributov 3](#_Toc89632518)

[1.2 Priprava atributov v Javi 3](#_Toc89632519)

[1.3 Fakturiranje podatkov 3](#_Toc89632520)

[1.4 Odpravljanje napak v podatkih 3](#_Toc89632521)

[2 Vizualizacija podatkov 4](#_Toc89632522)

[2.1 Grafi za analizo korelacije med atributi 4](#_Toc89632523)

[2.1.1 Korelacija med že obstoječimi atributi 4](#_Toc89632524)

[2.1.2 Prikaz porazdelitev podatkov 6](#_Toc89632525)

[2.1.3 Porazdelitev podatkov z novimi atributi 8](#_Toc89632526)

[2.1.4 Korelacija med porabo in novimi atributi 9](#_Toc89632527)

[2.1.5 Vizualizacije ocenjevanja atributov 10](#_Toc89632528)

[3 Klasifikacija 11](#_Toc89632529)

[3.1 Trivialen model 11](#_Toc89632530)

[3.2 Ocenjevanje atributov in izris dreves 11](#_Toc89632531)

[3.3 Priprava modelov z različnimi algoritmi 13](#_Toc89632532)

[4 Regresija 14](#_Toc89632533)

[4.1 Linearni regresijski model – lm 14](#_Toc89632534)

[4.2 Regresijsko drevo – rpart 14](#_Toc89632535)

[4.3 Support-vector machine – svm 16](#_Toc89632536)

[4.4 K-najbližjih sosedov – knn 16](#_Toc89632537)

[5 Učenje na ločenih regijah 17](#_Toc89632538)

[5.1 Klasifikacija 17](#_Toc89632539)

[5.2 Regresija 17](#_Toc89632540)

[6 Učenje po mesecih 17](#_Toc89632541)

[6.1 Klasifikacija 17](#_Toc89632542)

[6.2 Regresija 17](#_Toc89632543)

# Priprava podatkov za konstrukcijo modelov

## Analiza in izbor novih atributov

Po krajši analizi podatkov in korelaciji sva opazila, da lahko pripraviva nove atribute, ki bodo bistveno izboljšale same napovedi modelov, prvi atribut ki sva ga razcepila je bil »datum«. Pridobila sva nove atribute – »mesec«, »vikend« (ali gre za podatek porabe med vikendom ali med tednom) in »letni čas«. Kot drugi atribut, pa sva uporabila še poraba, in sicer statistiko za nazaj. Za vsako posamezno stavbo sva dodala podatek »včerajšnja poraba« in »tedenska poraba«.

## Priprava atributov v Javi

Pri izvedbi generiranja novih atributov sva sprva uporabljala kar R. Ampak, pri samem izračunu porabe za pretekli teden je sama izvedba v R dosti »nerodna«, zato sva se odločila za izvedbo v Javi, ki izkorišča hitrost podatkovne strukture »HashMap«, ki za vsako stavbo hrani vrednosti o preteklih porabah na določen dan. Tako se sama konstrukcija atributov, zgodi v ločenem programu in generira nove datoteke, katere potem uporabljava za konstrukcijo modelov.

## Fakturiranje podatkov

V naši množici podatkov imamo, kar nekaj atributov, ki imajo določeno »zalogo vrednosti«, v ta namen sva pripravila metodo, ki te podatke fakturira kar ob branju datoteke. Fakturirala sva »namembnost«, »sezona« – letni čas, »regija«, »oblacnost« - [1,10] »vikend« (1/0), »mesec«- [1,12].

## Odpravljanje napak v podatkih

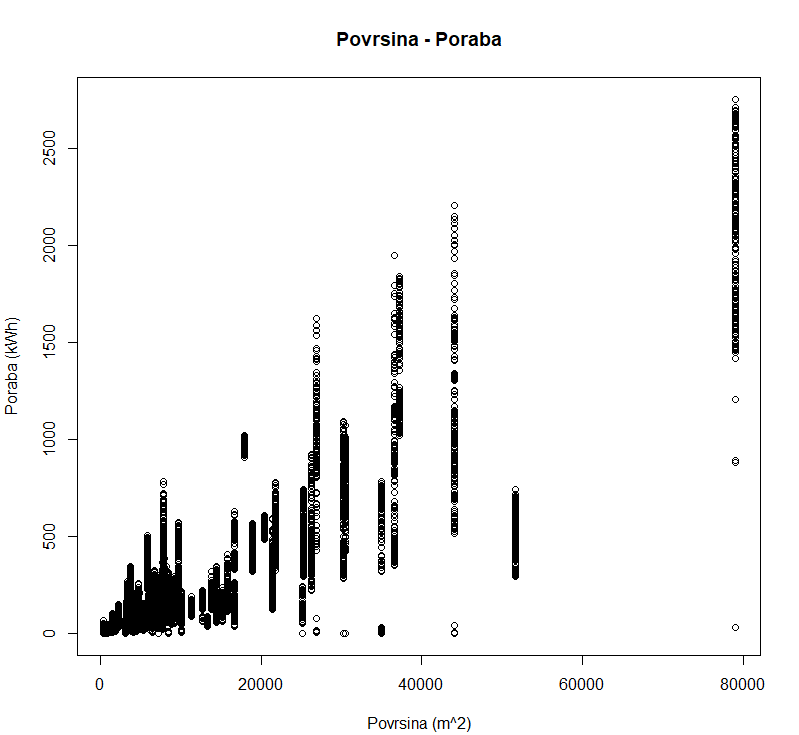
Pri konstrukciji atributov, sva opazila nekaj manjših napak, najprej sva morala podatke, ki smo jih iz Jave dobili, kot »-1« (zapisi, ki zaradi manjkajočih podatkov za nazaj nimajo izračunane zgodovine porabe). Poleg tega, pa ima vrednost padavin, kadar gre za »pršenje« vrednost -1, kar ni v skladu s podatki, saj »0« pomeni »popolnoma jasno«, zato sva vrednost »-1« preslikala v vrednost »1«.

# Vizualizacija podatkov

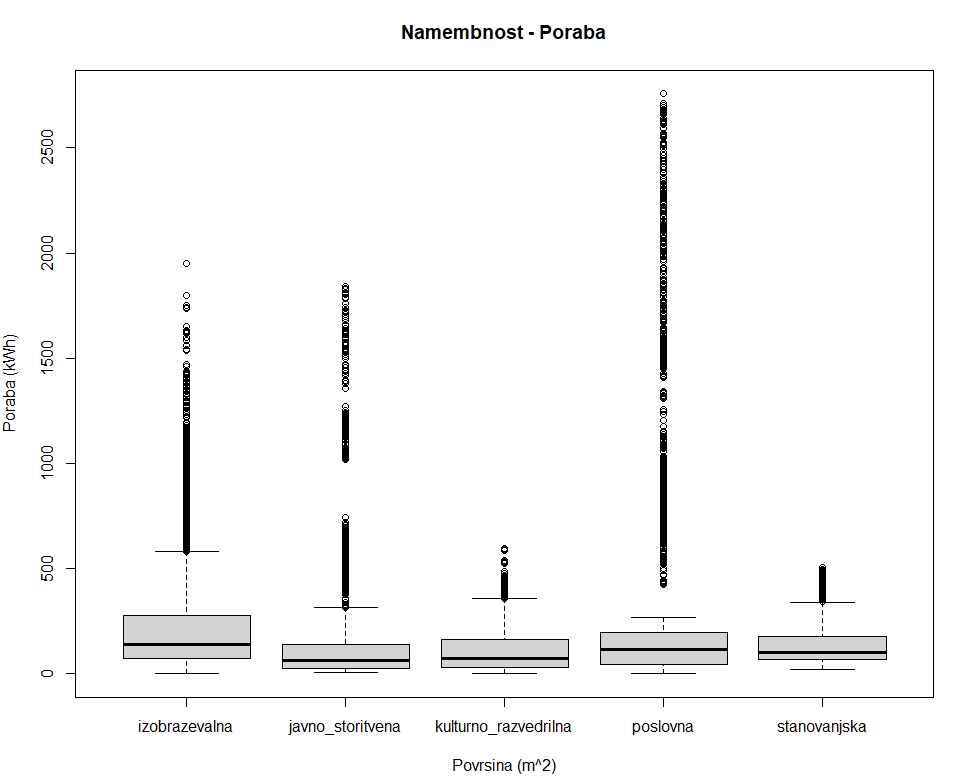
## Grafi za analizo korelacije med atributi

### Korelacija med že obstoječimi atributi

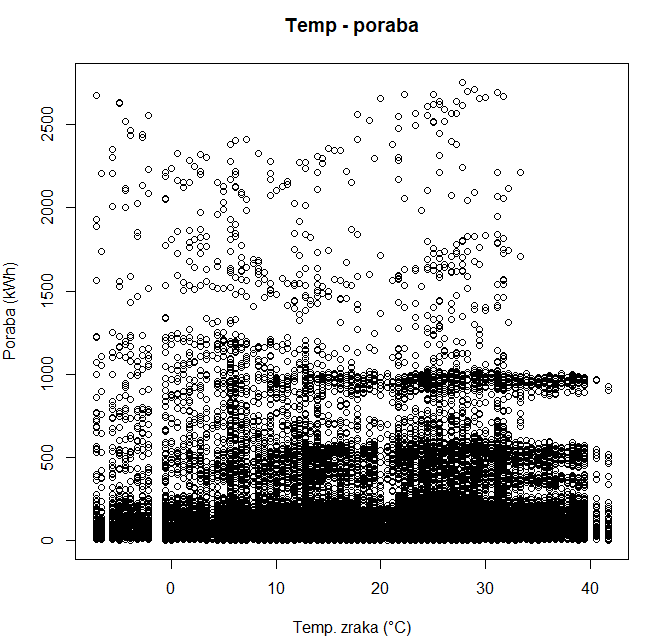
Spodnji graf prikazuje korelacijo med Površino n Porabo. Pričakovano se poraba veča skupaj z površino imamo nekaj manjših odstopanj, ampak v primeru maksimalnih porab, za neko površino, pa imamo zelo lepo naraščanje.



Naslednji graf predstavlja porabe glede na namembnost neke stavbe. Iz grafa lahko vidimo, da imajo tiste najvišje porabe »poslovne« stavbe. »Kulturno razvedrilne« in »Stanovanje« stavbe, pa imajo nekako bistveno manjšo porabo. Iz grafa tako vidimo, da bo površina dober atribut za napoved »namembnosti« stavbe.

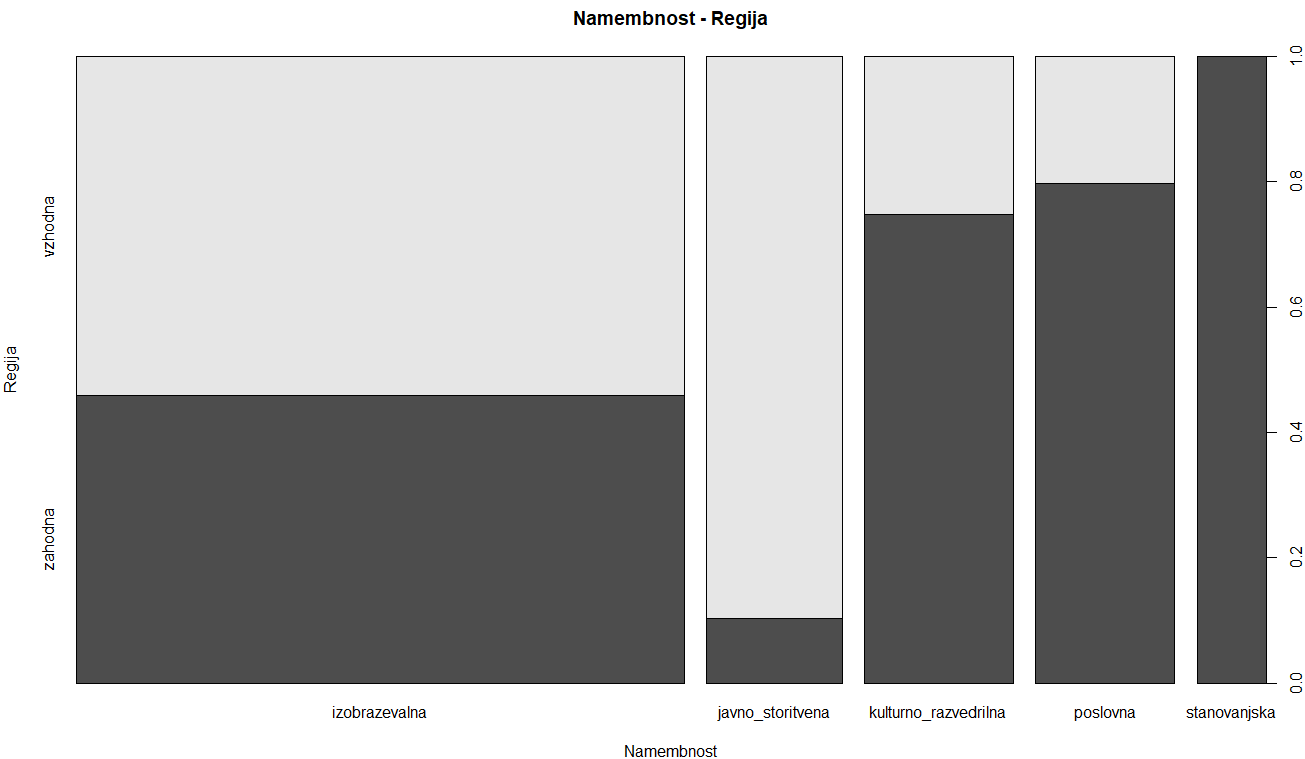


Po prvem pregledu učnih podatkov sva pričakovala, da bodo podatki za »temperaturo zraka« močno vplivali na porabo elektrike, saj ob nižjih temperaturah porabimo nekaj več elektrike (npr. za ogrevanje). Ta hipoteza, pa je bila kaj hitro zavrnjena, saj lahko vidimo, da med porabo in »temperaturo zraka« nimamo dobrega nivoja korelacije.

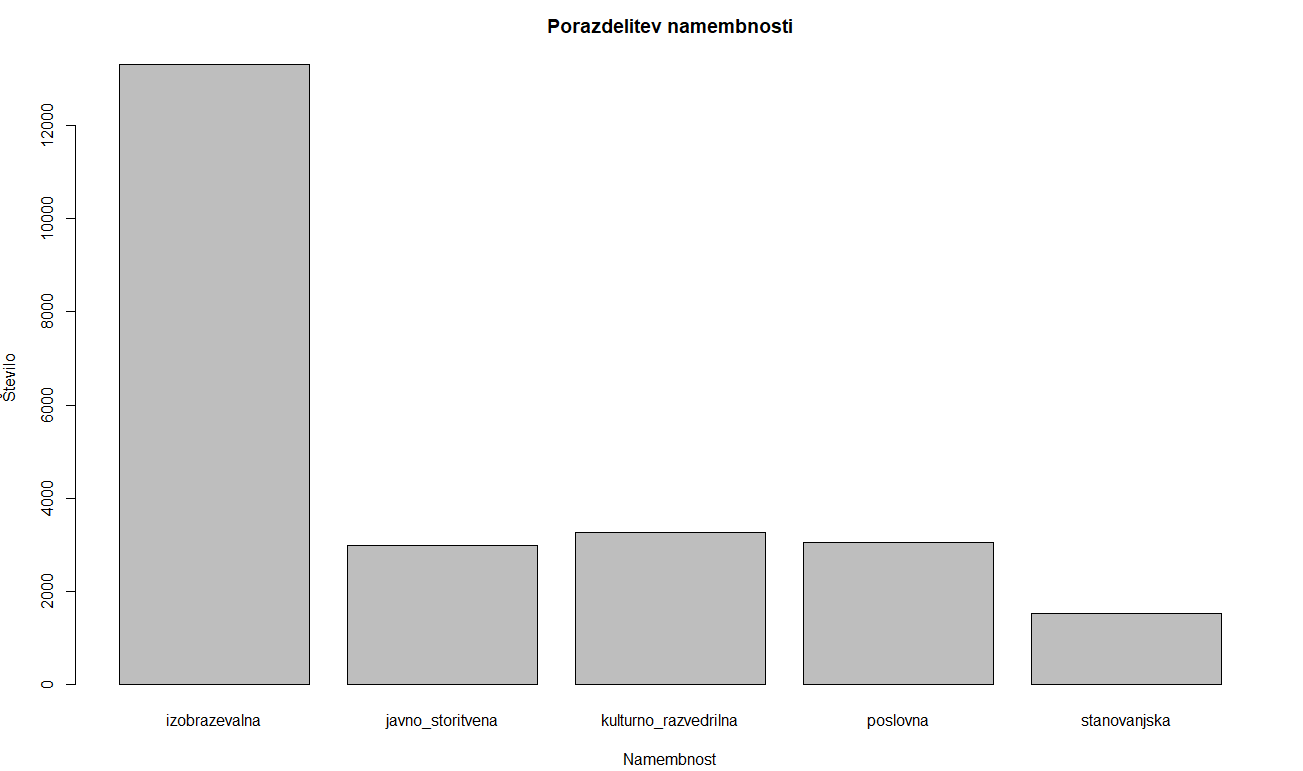


### Prikaz porazdelitev podatkov

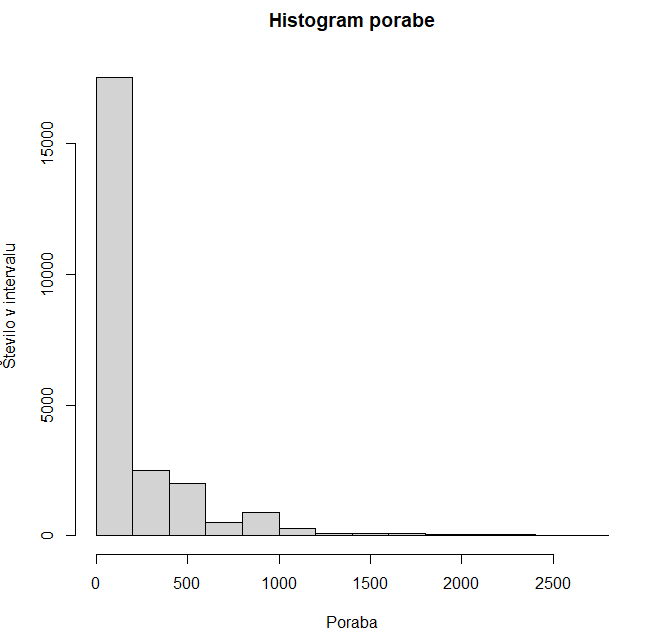
Prvi graf prikazuje porazdelitev glede na regijo in namembnost, razberemo lahko v katerih regijah prevladujejo določene stavbe. Vidimo, da imamo v učnih podatkih »stanovanjske« stavbe le v zahodni regiji, kar je lahko malo zavajajoč podatek, saj lahko, da v učni množici podatki ne sledijo enaki porazdelitvi.



Spodnji graf, pa prikazuje splošno porazdelitev »namembnosti« stavb, razberemo lahko, da je zelo velik del »izobraževalnih« stavb, kar je pozneje pri učenju privedlo, v »zelo dober«, trivialen model, podatki v testni množici sledijo podobni porazdelitvi, in tako dobimo oceno natančnosti (0.47).

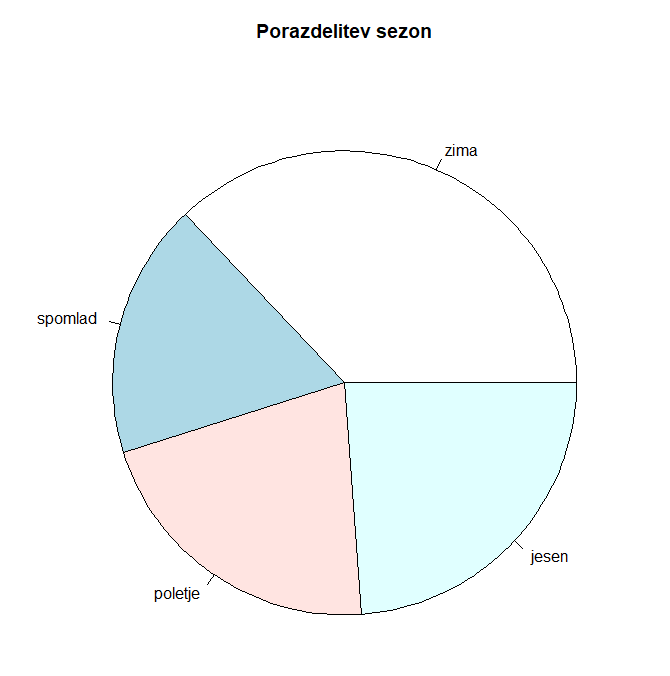


Spodnji histogram prikazuje razporeditev porabe v učnih podatkih vidimo lahko, da prevladujejo zapisi z relativno nizko porabo, torej skupino od 0-200 kWh. Po tej skupini število v skupini močno upade, ta trend se tudi nadaljuje in opazimo, da so podatki z porabo višjo od 100 kWh zelo redki.

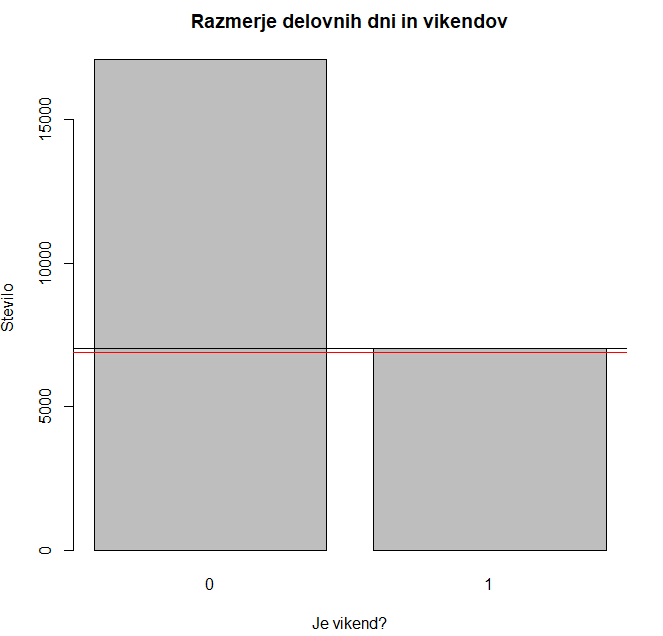


### Porazdelitev podatkov z novimi atributi

Pri porazdelitvi podatkov glede na letni čas opazimo, da je večji delež podatkov iz »zimskega« letnega časa.

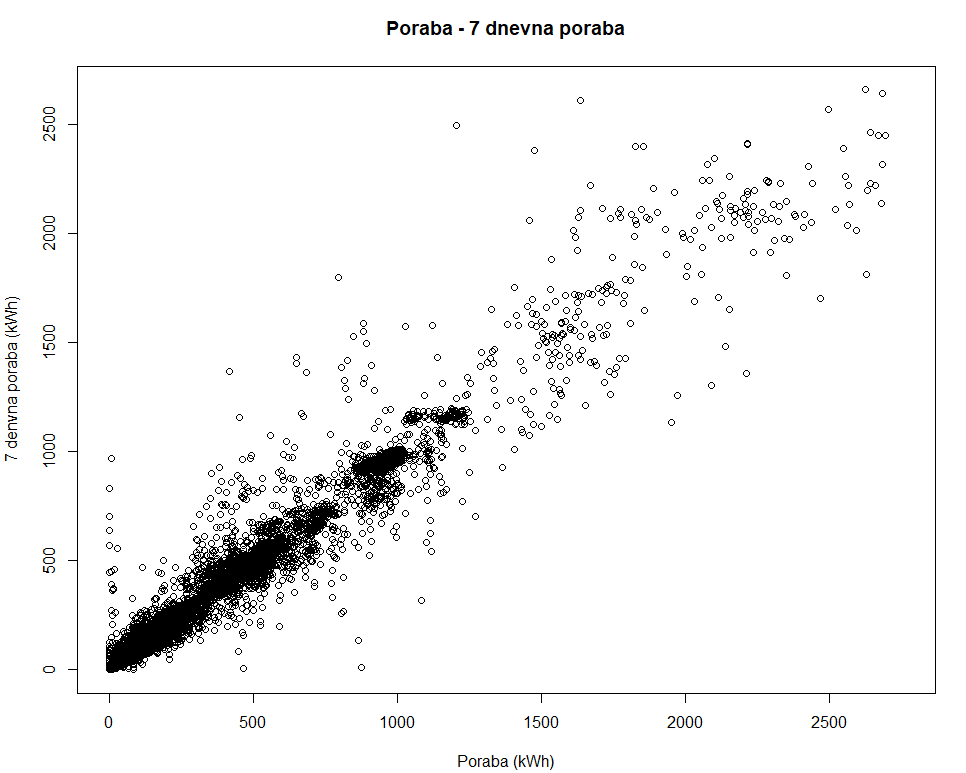


Na spodnjem grafu, pa vidimo porazdelitev »dni med tednu« in »vikendov«, samo razmerje pričakujemo, da bo 2:5, kar prikazuje rdeča črta, črna črta pa prikazuje našo porazdelitev, tako vidimo, da je razmerje podatkov med dnevi dobro uravnoteženo.

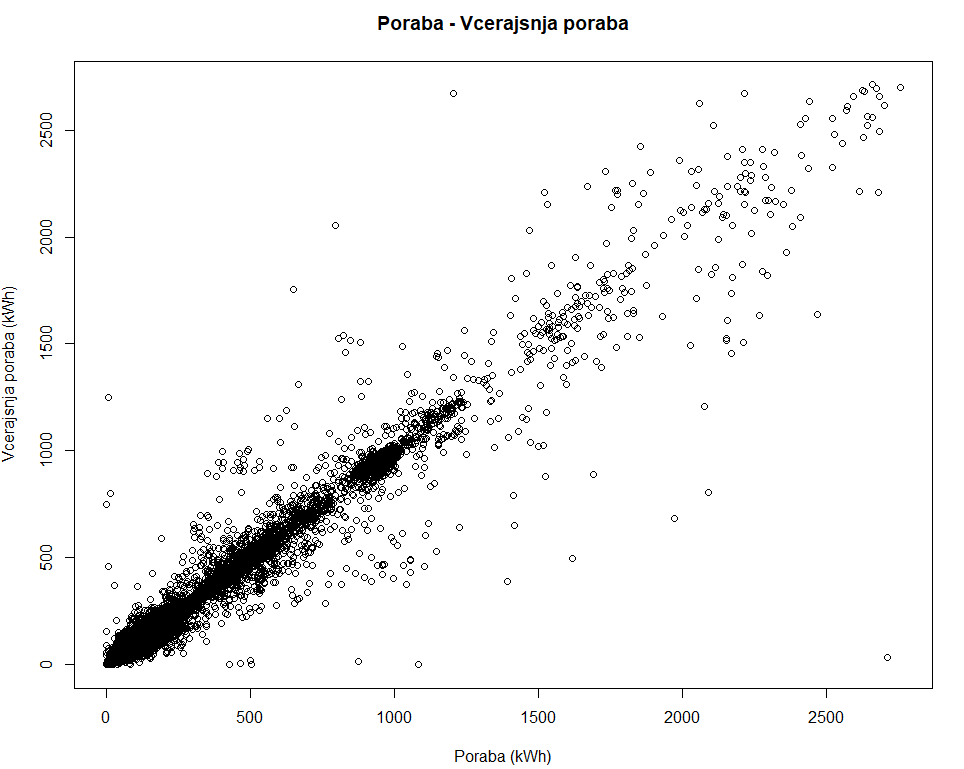


### Korelacija med porabo in novimi atributi

Pri konstrukciji novih atributov, je pri prvih nekaj zapisih računanje pretekle porabe onemogočene, saj teh podatkov nimamo na voljo, zato pri sami konstrukciji grafa te podatke izpustimo (vrednosti »tedenske porabe« za prve zapise imajo vrednost NA). Vidimo pa zelo lepo linearno korelacijo, ki bo napovedi za regresijsko napoved porabe bistveno izboljšala.

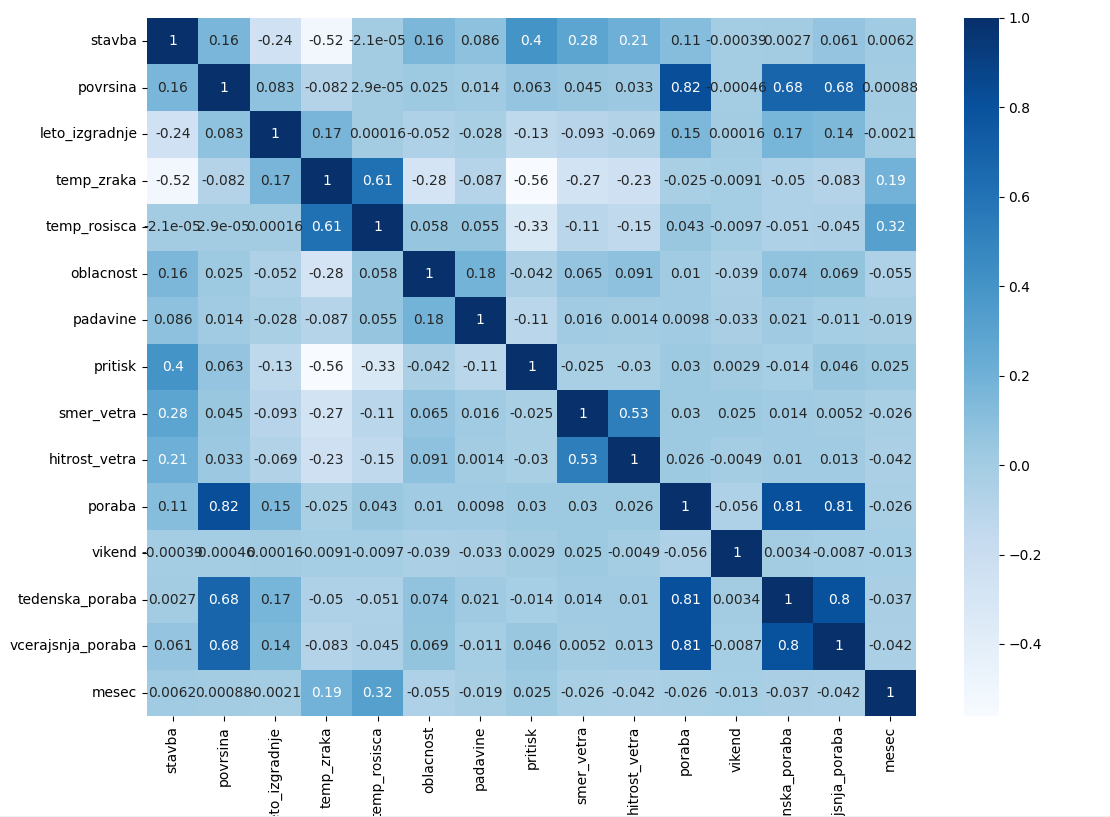


Še boljša je korelacija je med »včerajšnjo« in »današnjo« porabo. To je seveda pričakovano, imamo pa vseeno nekaj manjših »izpadanj podatkov«, ki ne sledijo korelaciji, so pa ti zelo redki. Včerajšnja poraba, bo tako postal ključen atribut, pri napovedi »porabe«.



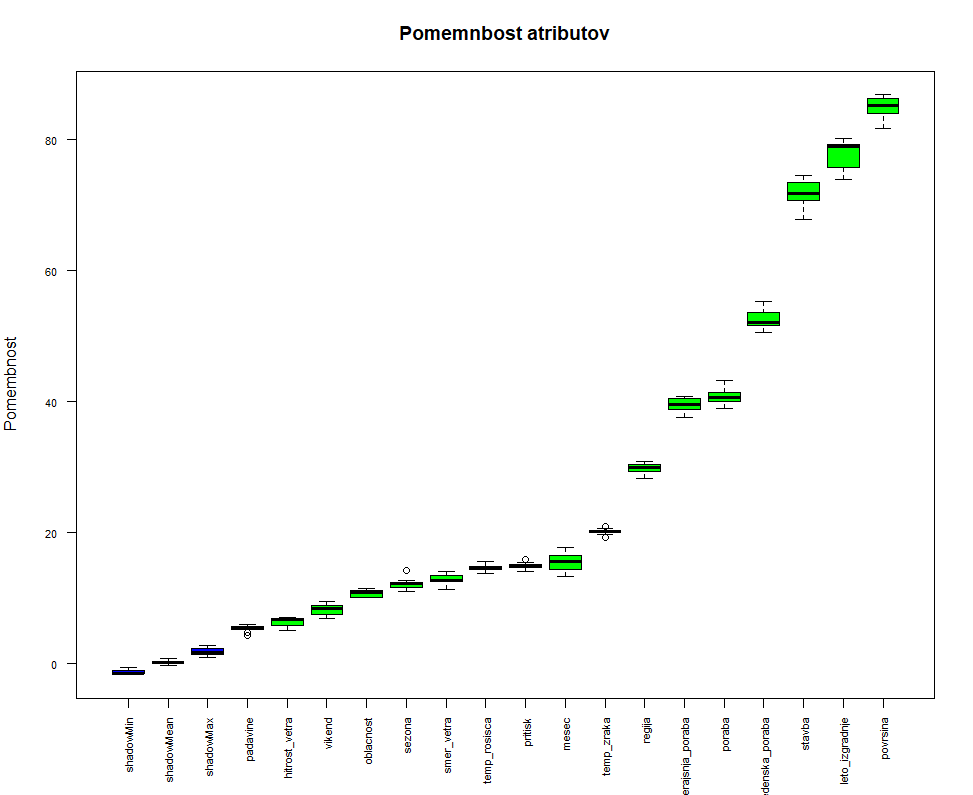
### Vizualizacije ocenjevanja atributov

Za lažje razumevanje, kako se atributi med sabo povezujejo, sva pripravila skripto v »Pythonu«, ki pokaže korelacijo med nekaterimi atributi za regresijo. Opazimo lahko na primer močno korelacijo med »porabo« in »tedensko porabo«, »tedensko porabo« ter »površino«.



Zgornji prikaz sploh dobro prikaže, kateri atributi bodo koristni za regresijski model, ki napoveduje »porabo«. Graf pa ne prikazuje atributa »namembnost« in korelaciji. V ta namen, sva pripravila vizualizacijo s pomočjo knjižnice »Boruta«, kar v R.

Graf napove, da bodo za napovedovanje »namembnosti« najboljši naslednji atributi (napisani od najboljšega proti slabšim) – »poraba«, »leto izgradnje«, »stavba«, »poraba«, »včerajšnja poraba«, »regija«…



# Klasifikacija

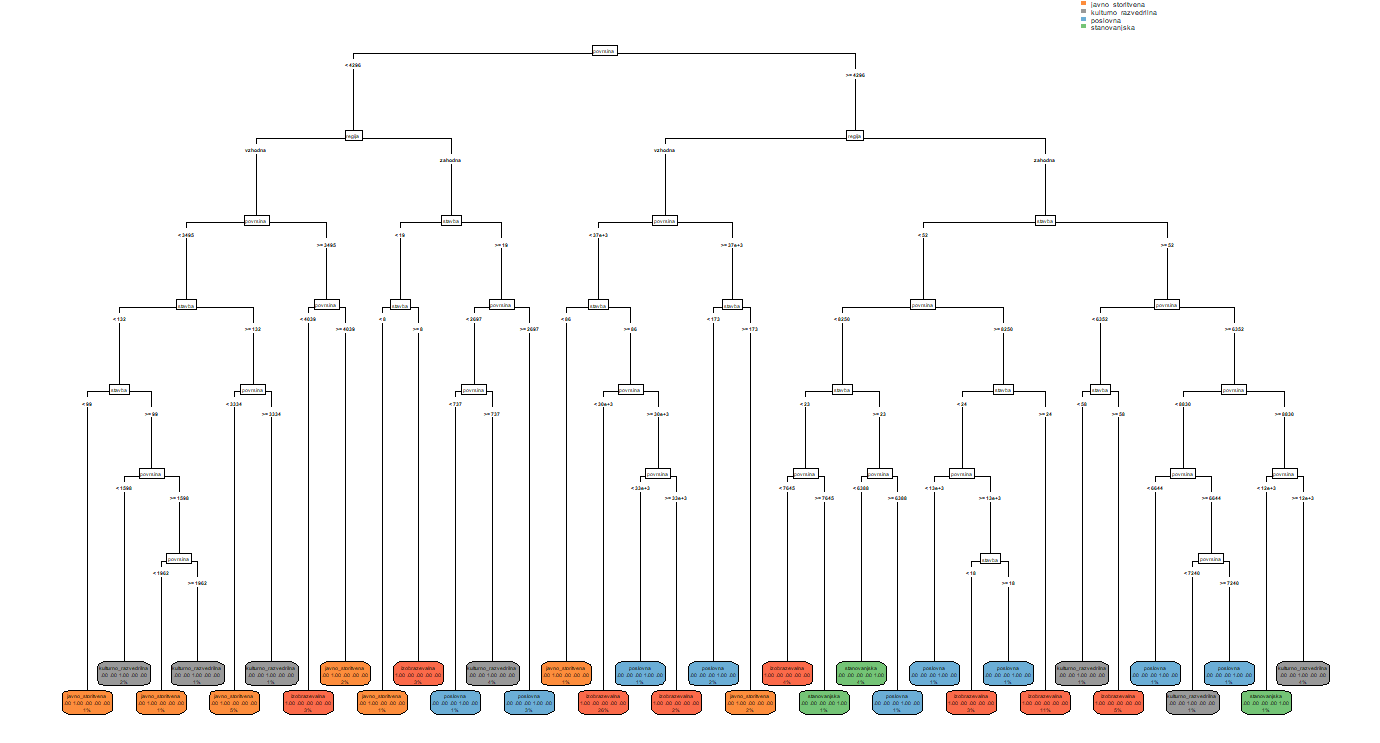
## Trivialen model

Kot prvi kreiran model sem izbral »trivialnega«, ki na podlagi učne množice poišče najbolj pogost tip stavbe - »namembnost«. Ta je dobil, »klasifikacijsko natančnost« 0.47, kar je predstavljal delež »izobraževalnih stavb« v testni množici.

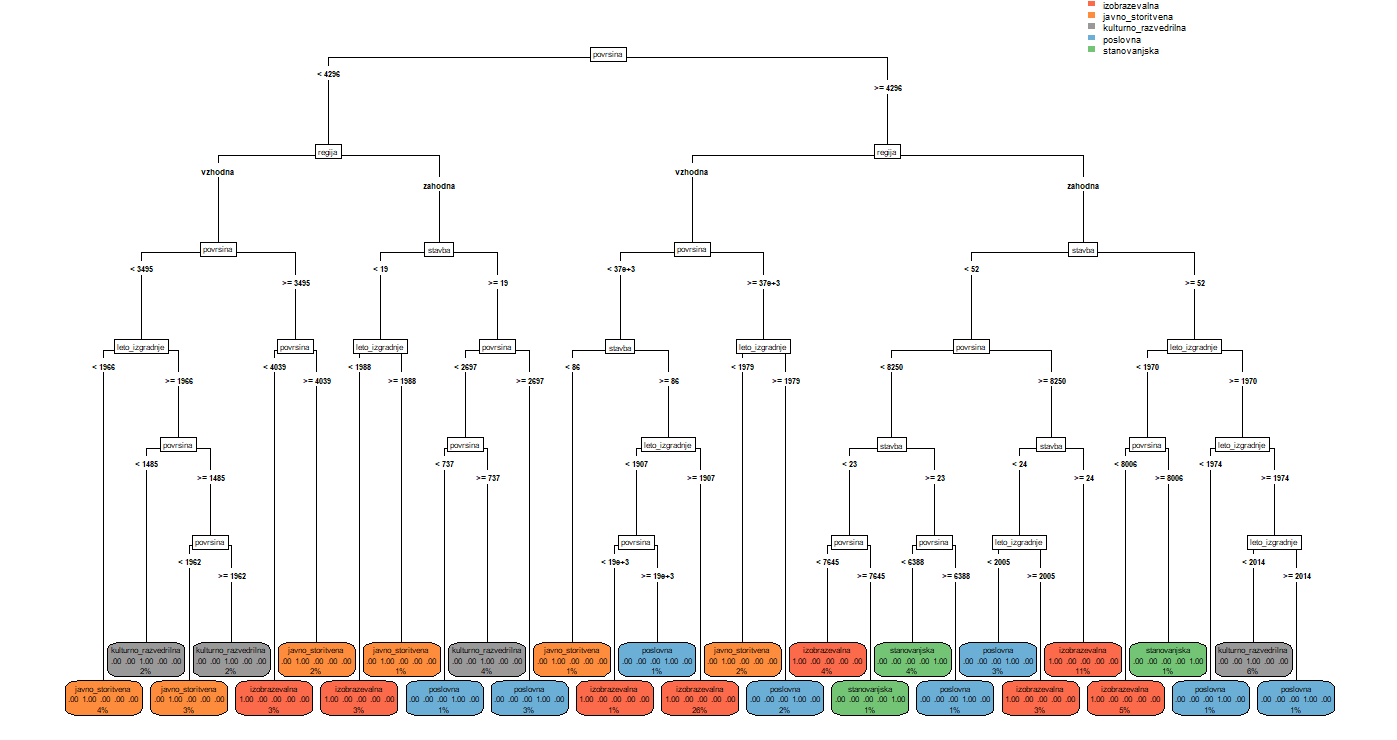
## Ocenjevanje atributov in izris dreves

Za pomoč ocenjevanja atributov sem uporabljal »attrEval«, na podlagi različnih ocenjevanj atributov, sem potem zgradil odločitvena drevesa in izpisal njihovo »klasifikacijsko natančnost« - CA, ki pa je za vse modele bila nekje ~ 0.4451505016.

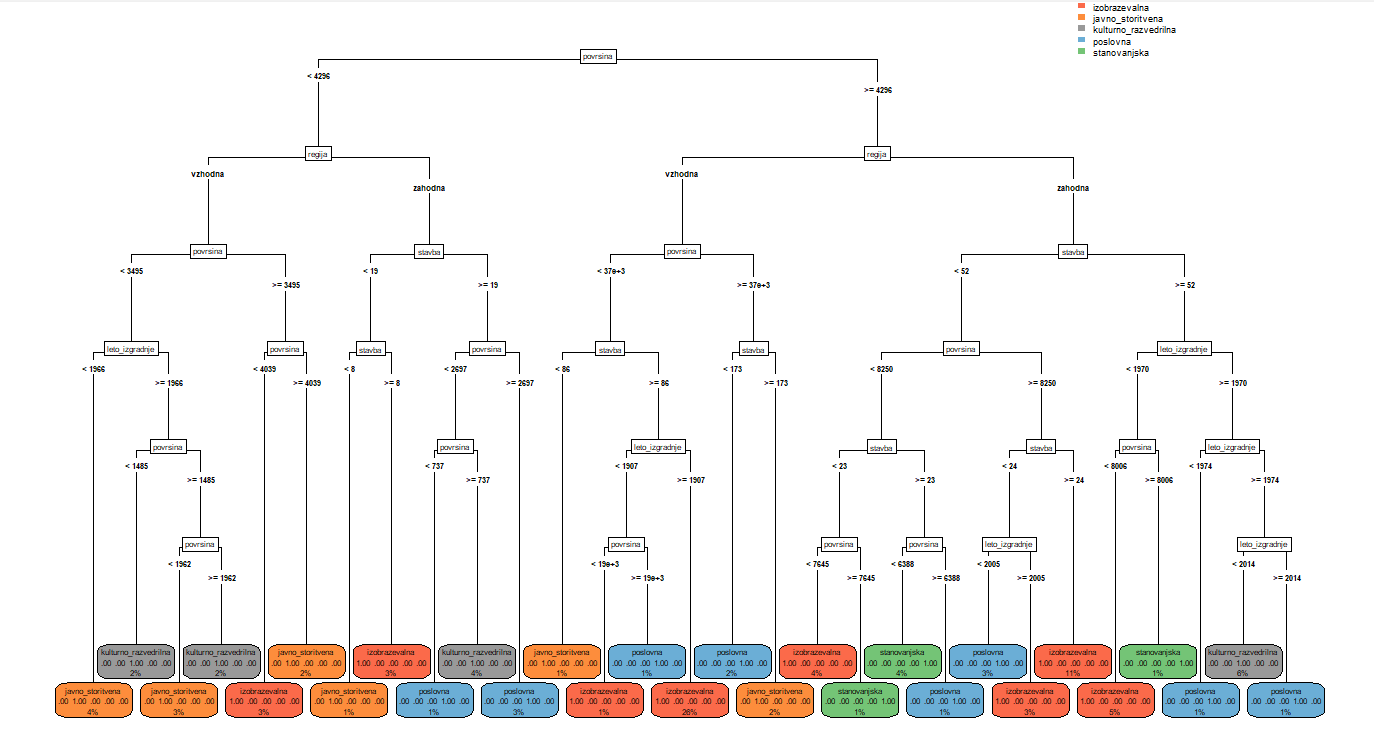
Kot prvi algoritem za ocenjevanje sem izbral »InfGain« in pripravil odločitveno drevo glede na 3 najbolje ocenjene atribute (»površina«, »regija«, »stavba«):



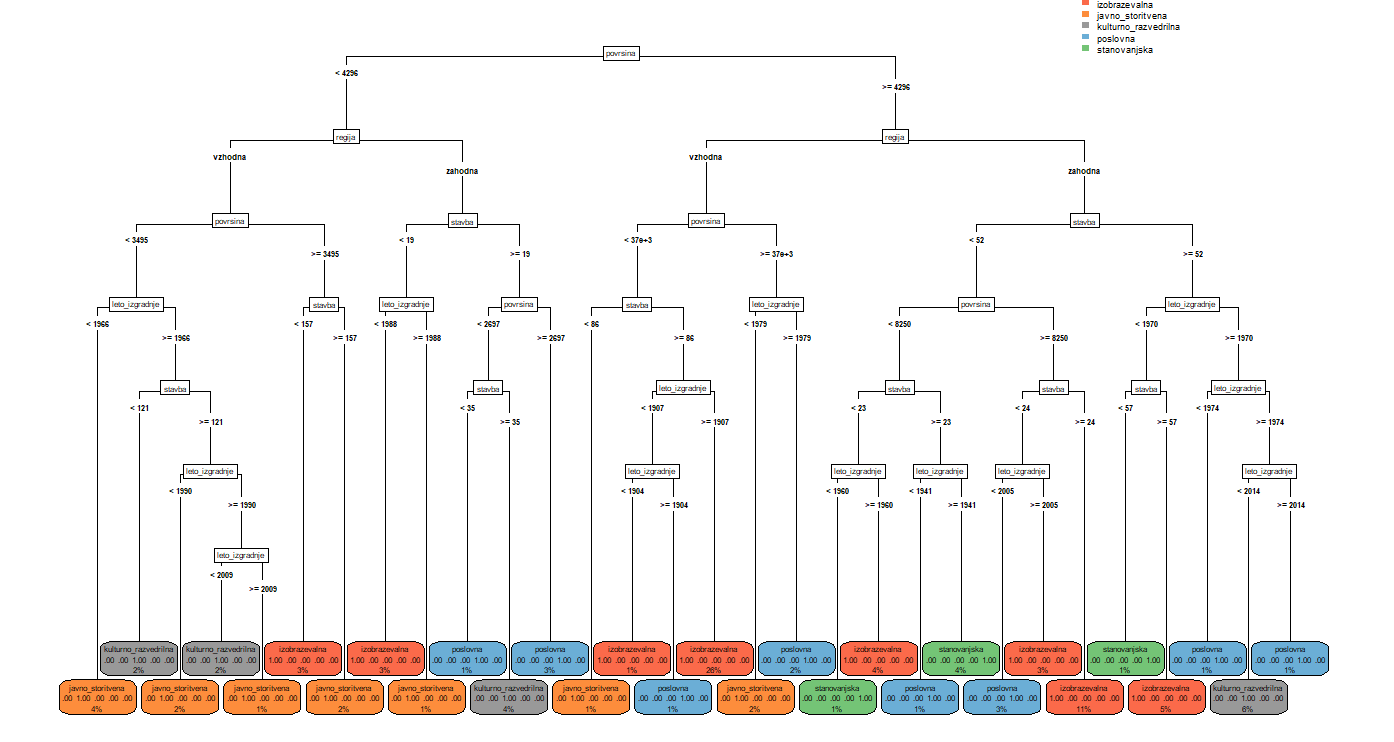
Kot drugi algoritem sem izbral »Gini« od katerega, sem izbral 4 najbolje ocenjene atribute – »povrsina«, »leto izgradnje«, »regija«, »stavba«:



Za tretje drevo sem izbral »MDL« algoritem za izbor atributov, v tem primeru sem izbral najboljših 5 (»povrsina«, »regija«, »stavba«, »leto izgradnje«, »temp. zraka«:



Kot zadnji algoritem za ocenjevanje, pa sem izbral »ReliefFequalK«, za katerega sem izbral 6 najbolje ocenjenih atributov – »leto izgradnje«, »stavba«, »površina«, »poraba«, »regija«, »tedenska poraba«:



## Priprava modelov z različnimi algoritmi

Za kreiranje modelov sem uporabljal »CORELearn« knjižnico, ki že omogoča vgrajeno uporabo algoritmov za ocenjevanje atributov in binarizacijo. Pri izvedbi modelov, sem skušal izbrati modele z čim višjo klasifikacijsko natančnostjo… Testiral sem različne »ocenjevalne algoritme« - »selectionEstimator«, po večini, pa je nekako najbolje ocenjene modele vračalo ocenjevanje z »MDL«.

|  |  |
| --- | --- |
| **Algoritem** | **Klasifikacijska natančnost** |
| Naivni Bayes | 0.444857 |
| Odločitveno drevo | 0.488418 |
| K – najbližjih sosedov | 0.578302 |
| Naključni gozd – izvedba 1 | 0.559740 |
| Naključni gozd – izdedba 2 | 0.542527 |
| Naključni gozd (rfNear) | 0.560576 |
| Bagging | 0.520066 |
| Boosting | 0.501546 |

## Kombiniranje algoritmov z glasovanjem

### Navadno glasovanje

Pri navadnem glasovanju uporabimo več naučenih modelov, in potem njihove »napovedi« uporabimo v »večinskem« glasovanju. Sam sem za to uporabil 4 modele (»naključni gozd«, »knn«, »odločitveno drevo«, »boosting model«. Vse skupaj je na koncu prišlo na »klasifikacijsko natančnost« 0.59765886 ~ 0.6.

### Glasovanje z utežmi in »Brajerjeva mera«

Pri tej vrsti glasovanju, moramo pri ustvarjanju napovedi, spremeniti tip iz »class« v »prob«, torej v tem primeru modeli, več ne bodo napovedali preprosto najbolj verjetnega razreda, ampak bodo napovedali, kakšna je verjetnost, da gre za vsak posamezen razred (najvišja vrednost, bo za tisti razred, ki bi ga vrnilo ocenjevanje z »class«.

Pri ocenjevanju sem za posamezen model uporabljal »Brajerjevo mero«, ki nam oceni različnost med ciljno (idealno) distribucijo in dejansko napovedano distribucijo. Tukaj želimo čim nižje vrednosti sama zaloga vrednosti pa je [0,2]:

|  |  |
| --- | --- |
| Učni algoritem | Brajerjeva mera |
| Odločitveno drevo | 1.02316 |
| Naključni gozd | 0.65531 |
| KNN | 0.65749 |
| Bagging | 0.89268 |

Potem, sem vse te napovedni združil s preprosto formulo: »RF« + »KNN« + »KNNK«, kjer je »klasifikacijska natančnost« 0.586413.

### Uporaba dodatnih uteži

Pri kreiranju modelov, lahko uporabljamo tudi dodatne uteži, torej pri formulah za glasovanje, lahko nekemu modelu dodamo »večjo moč«. Prvi način je, da s oceno napake posameznega modela otežimo – uporabimo formulo: (1 – napaka modela) \* napoved modela.

Jaz sem za izvedbo dodatnih uteži k posameznemu modelu dodal še »klasifikacijsko natančnost«, torej naslednja formula: CA Modela \* napoved modela. Tako je model z višjo klasifikacijsko natančnostjo dobil večji delež v glasovanju.

Na ta način sem pripravil model z naslednjo formulo: CA \* RF + CA \* KNN + CA \* KNNK, ki je vrnil klasifikacijsko natančnost: 0.586873

# Regresija

Pri regresijskem problemu je bilo potrebno določiti porabo elektrike v stavbi.

Najprej sva se odločila oceniti atribute, da bi videla, ali so ocene konsistentne z najino prejšnjo analizo in vizualizacijo. V ta namen sva uporabila knjižnico »CORElearn« in njeno funkcijo »attrEval«. Podatke sva evalvirala na večih ocenah npr. »MSEofMean«, »RReliefFequalK«, »RReliefFbestK«, »MSEofModel« itd., rezultati pa so bili bolj ali manj konsistentni z najino napovedjo; da bodo najmočnejši atributi za napovedovanje »včerajšnja poraba«, »tedenska poraba« in »površina«.

Nato sva se lotila konstrukcije modelov, kjer sva to, kako dobro »napovedujejo« ocenila z kalkulacijami RMSE (relative mean squared error) in MAE (mean absolute error), ki povesta, za koliko smo se pri napovedi zmotili. Pri obeh kalkulacijah želimo, da sta njune vrednosti čim manjše.

Prav tako sva za modele uporabila funkcijo »wrapper«, ki gre čez vse možnosti uporabljenih atributov in predlaga, katera izbira bi bila najboljša.

## Linearni regresijski model – lm

Linearni regresijski model deluje tako, da poskuša čez učne podatke narisati premico, ki služi kot napoved, glede na gledane atribute.

Tako sva linearni model testirala na atributih:

|  |  |  |
| --- | --- | --- |
| **Gledani atributi** | **RMSE** | **MAE** |
| Vsi | 0.06519753 | 19.31359 |
| Včerajšnja poraba, tedenska poraba, površina | 0.05745394 | 17.88321 |
| Včerajšnja poraba, tedenska poraba | 0.05731692 | 17.54313 |
| Včerajšnja poraba | 0.072729 | 17.85158 |

Opazimo lahko, da smo, če za napovedovanje uporabimo vse atribute, dosti manj natančni, kot če uporabimo atribute, ki so bili dobro ocenjeni z »attrEval«. Prav tako opazimo, da če uporabimo premalo atributov, nam natančnost pade.

Nato sva pognala funkcijo »wrapper« in dobila atribute, ki naj bi najbolje napovedovali porabo:

|  |  |  |
| --- | --- | --- |
| **Gledani atributi** | **RMSE** | **MAE** |
| Včerajšnja poraba, tedenska poraba, vikend, površina, oblačnost, padavine, temperatura rosišča, leto izgradnje, sezona, temperatura zraka | 0.05426703 | 19.21306 |
| Včerajšnja poraba, tedenska poraba, vikend | 0.0541383 | 19.04113 |

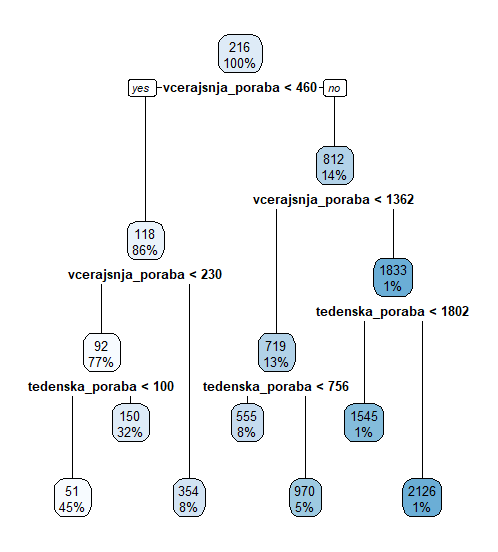
Glede na to, da je atributov precej, se lahko odločimo tudi, da skrajšamo gledane atribute in dobimo dokaj primerljiv rezultat.

## Regresijsko drevo – rpart

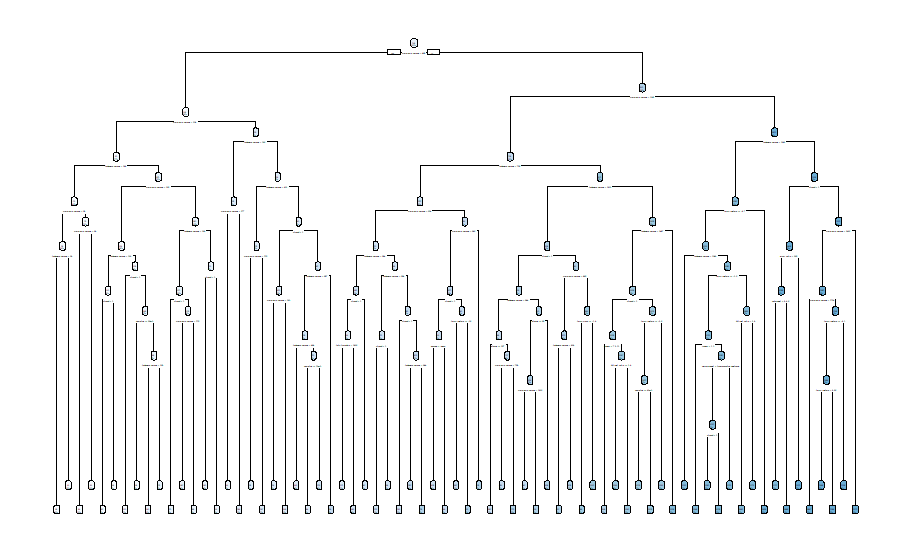
Regresijsko drevo je odločitveno drevo, kjer se na vsakem vozlišču izračuna pogoj glede na enega izmed vhodnih atributov. V korenu drevesa je vedno »najmočnejši atribut«, kasneje pa se lahko pojavljajo tudi ostali.

Za ustvaritev in izris regresijskega drevesa sva uporabila knjižnici »rpart« in »rpart.plot«.

Prvo drevo sva zagnala na vseh atributih.



To drevo je dobilo RMSE 0.1288535 in MAE 44.48812, kar je v primerjavi z linearnim regresijskim modelom dokaj slabo. Zato sva se odločila še za izdelavo celotnega drevesa in rezanje.



Novo drevo izgleda gromozansko, ampak je kljub temu veliko bolj natančno kot prvo, saj ima RMSE 0.06194545 in MAE 19.69517, kar pa je primerljivo tudi s prejšnjim modelom.

## Support-vector machine – svm

SVM oz. metoda podpornih vektorjev je metoda, ki učno množico razdeli v razrede in jih pretvori v n-dimenzionalni prostor in poskusi ta prostor čim bolje razdeliti, da so posamezni razredi med sabo razpoznavni – lahko ločujemo med njimi.

Za izdelavo SVM modela, sva uporabila knjižnico »e1071«. Model sva ustvarila z različnimi atributi:

|  |  |  |
| --- | --- | --- |
| **Gledani atributi** | **RMSE** | **MAE** |
| Vsi | 0.05762129 | 21.38413 |
| Včerajšnja poraba, tedenska poraba, vikend, regija | 0.04510532 | 19.29338 |
| Včerajšnja poraba, tedenska poraba, vikend | 0.04445059 | 19.76115 |
| Včerajšnja poraba, tedenska poraba | 0.06234921 | 21.2505 |

Glede izbire atributov opazimo podobno, kot pri linearnem regresijskem modelu, da moramo biti pri izbiri atributov pazljivi, saj če jih je premalo ali preveč, se rezultat poslabša. Prav tako opazimo, da je SVM model bolje napovedoval že z vsemi atributi in z izbiro pravih atributov močno prekašal linearni regresijski model.

## K-najbližjih sosedov – knn

Metoda z uporabo k-najbližjih sosedov vzame podan primer in poskusi podati odgovor, glede na naučene podatke. Pri tem upošteva več vrednosti (k)

Za izdelavo tega modela sva uporabila knjižnico »kknn« in ga testirala z različnimi atributi in različnim številom sosedov:

|  |  |  |  |
| --- | --- | --- | --- |
| **Gledani atributi** | **K** | **RMSE** | **MAE** |
| Vsi | 5 | 0.164123 | 52.56553 |
| Včerajšnja poraba, tedenska poraba, vikend | 5 | 0.05526724 | 17.97103 |
| Včerajšnja poraba, tedenska poraba, vikend, površina | 5 | 0.06698666 | 23.69035 |
| Včerajšnja poraba, tedenska poraba, vikend | 10 | 0.04661027 | 16.65517 |
| Včerajšnja poraba, tedenska poraba, vikend | 20 | 0.0435993 | 15.86111 |
| Včerajšnja poraba, tedenska poraba, vikend | 30 | 0.04334001 | 15.65017 |

Opazimo lahko, da je model zelo slab, če gledamo vse atribute in se, kot je bilo pri prejšnjih primerih, precej izboljša z uporabo pravilnih atributov. Opazimo lahko tudi, da se natančnost modela veča, če povečujemo k, vendar nam ga, da dobimo zadovoljiv rezultat ni treba preveč povečati (gledanje okoli 20 sosedov je že precej dobro), s tem ko preverjamo manj sosedov, pa prihranimo tudi na času.

# Učenje na ločenih regijah

## Klasifikacija

## Regresija

Pri regresiji je bilo učenje na zahodni množici podatkov veliko bolj natančno, kot za vzhodni. Za preverjanje tega, sva uporabila nekatere izmed prej prestavljenih modelov in dobila za regijo »vzhod«:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Gledani atributi** | **K** | **RMSE** | **MAE** |
| LM | Včerajšnja poraba, tedenska poraba |  | 0.07113724 | 24.56143 |
| SVM | Vsi |  | 0.05861002 | 26.40473 |
| KNN | Včerajšnja poraba, tedenska poraba, vikend | 10 | 0.05317498 | 21.06857 |

In za regijo »zahod«:

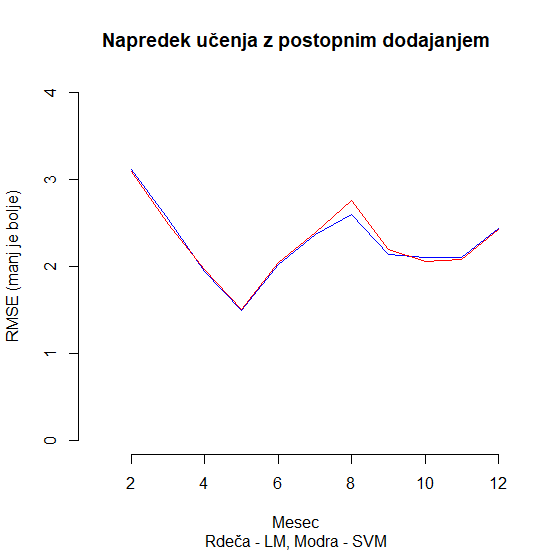
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Gledani atributi** | **K** | **RMSE** | **MAE** |
| LM | Včerajšnja poraba, tedenska poraba |  | 0.02711596 | 12.68869 |
| SVM | Vsi |  | 0.02711596 | 12.68869 |
| KNN | Včerajšnja poraba, tedenska poraba, vikend | 10 | 0.0300223 | 13.8744 |

# Učenje po mesecih

## Klasifikacija

## Regresija

Za regresijo, sva se odločila po mesecih naučiti in nato primerjati linearni regresijski model in SVM model.



Oba sta po mesecih napovedovala dokaj podobno; najbolje sta napovedala podatke v mesecu marcu, nato pa se je natančnost nekoliko poslabšala. Opazimo tudi, da je linearni regresijski model rahlo slabši pri napovedovanju v določenih mesecih (npr. avgust), ampak v glavnem med njima ni prevelikih razlik.