Contents

[2. Експериментални подаци 2](#_Toc497379453)

[2.1. Опис скупа података 2](#_Toc497379454)

[2.2. Корелациона матрица 3](#_Toc497379455)

[3. Регресиона анализа 5](#_Toc497379456)

[3.1. Линеарна регресија 5](#_Toc497379457)

[3.2. Примена 7](#_Toc497379458)

[4. Недостајуће вредности 8](#_Toc497379459)

[4.1. Механизми недостајућих вредности 8](#_Toc497379460)

[4.2. Технике уметања података 10](#_Toc497379461)

[4.2.1. Уметање средње вредности 10](#_Toc497379462)

[4.2.2. Преношење задњег запажања 11](#_Toc497379463)

[4.2.3. Уметање података коришћењем линеарне регресије 12](#_Toc497379464)

[4.2.4 Уметање података стохастичком регресијом 14](#_Toc497379465)

[4.2.5 Уметање коришћењем (шума) стабала одлучивања 15](#_Toc497379466)

[5. Експеримент 21](#_Toc497379467)

[5.1. Конструисање тренинг скупа 21](#_Toc497379468)

[5.2. Интерпретација резултата импутације података 22](#_Toc497379469)

[5.2.1. Средња квадратна грешка импутације 24](#_Toc497379470)

[5.2.2. Корен средње квадратне грешке 24](#_Toc497379471)

[5.2.3. Просечна релативна грешка 25](#_Toc497379472)

[5.2.4. Корен средње квадратне грешке линеарне регресије 27](#_Toc497379473)

Uvod

Ovaj master rad se bavi problemom nedostajućih vrednosti u podacima koji predstavljaju ulaz za algoritme mašinskog učenja. Ukoliko podaci koji služe za trening nisu potpuni, može se desiti da istreniran model nije dovoljno pouzdan i da kao rezultat donosi pogrešne zaključke.

Skup podataka koji će se koristiti u radu je pogodan za regresionu analizu. Pokazaće se kako različiti procenti

# 2. Експериментални подаци

## 2.1. Опис скупа података

Иако ће сам експеримент бити заступљен у другом делу рада, неопходно је да се сада објасни скуп података који ће се том приликом користити. Сви примери у раду ће се ослањати на овај скуп података (везани за екпсеримент или као помоћно средство за разумевање теорије).

Подаци се односе на пацијенте који болују од дијабетеса. У табели (1) је визуелно представљен само подскуп података. Иницијални скуп података је садржао податке о 442 пацијента, што је за потребе овог рада подељено у два подскупа: први са 400 обсервација (тренинг подскуп) и други са 42 обсервације (тестни подскуп).

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Старост | Пол | ИТМ[[1]](#footnote-1) | КП[[2]](#footnote-2) | Резултати серумских мерења | | | | | | резултат |
| Пацијент |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 59 | 2 | 32.1 | 101 | 157 | 93.2 | 38 | 4 | 4.9 | 87 | 151 |
| 2 | 48 | 1 | 21.6 | 87 | 183 | 103.2 | 70 | 3 | 3.9 | 69 | 75 |
| 3 | 72 | 2 | 30.5 | 93 | 156 | 93.6 | 41 | 4 | 4.7 | 85 | 141 |
| 4 | 24 | 1 | 25.3 | 84 | 198 | 131.4 | 40 | 5 | 4.9 | 89 | 206 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 399 | 52 | 1 | 27.8 | 85 | 219 | 136 | 49 | 4 | 5.1 | 75 | 242 |
| 400 | 65 | 2 | 28.5 | 109 | 201 | 123 | 46 | 4 | 5.1 | 96 | 232 |

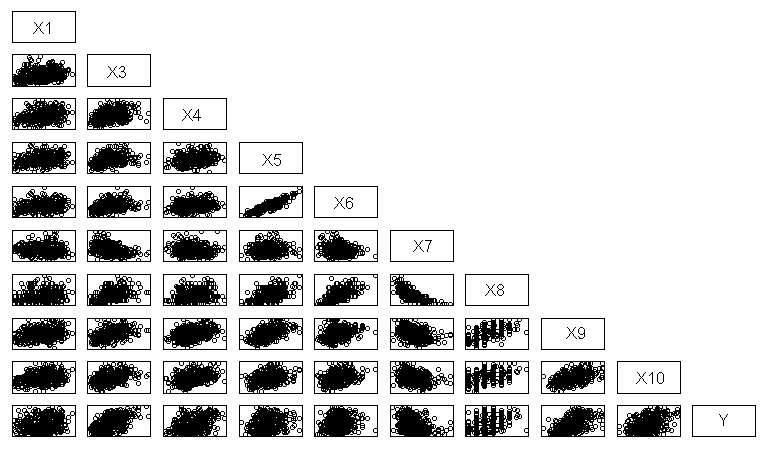
Табела 1 Тренинг подскуп. Поред демографских података (старост и пол), на пацијентима је вршено 8 мерења који заједно утичу на ниво дијабетеса годину дана касније

Скуп се састоји од 10 атрибута (варијабли) од који су 8 нумеричког типа () и једна номиналног карактера (). Резултат представља ниво дијабетеса код људи годину дана након мерења. [5] Такође, резултујућа колона је нумеричког карактера.

## 2.2. Корелациона матрица

Укључујући резултатску променљиву (колону), скуп података се сатоји од 10 колона нумеричког типа, што је одличан предуслов за анализу корелационе матрице. Корелациона матрица, са друге стране, може да укаже на линеаран однос између две колоне. Уколико је испостави да такве зависноти (лиенарне) постоје, у даљем раду ће линеарна регресија бити један од затупљенијих алгоритама машинског учења.

Свакако, пре даље анализе, одличан показатељ било које анализе може да донесе визуелизација самог скупа података (Слика 1).



Слика Визуелни приказ скупа података

Са слике 1 се јасно види да неке променљиве имају изражену линеарну зависност, што је добар показатељ да корелациону матрицу треба креирати и да је могуће из ње извући одређене закључке. Приметимо да променљива није део графичког приказа јер номинални тип податка те променљиве није идеалан за визуелизацију.

За креирање корелационе матрице, потребно је израчунати 45 различитих вредности или:

|  |  |
| --- | --- |
|  | (1.1) |

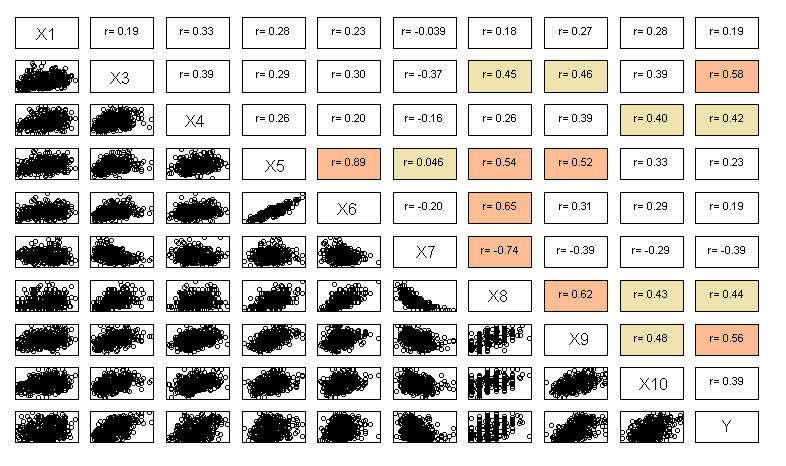
У једначини (1.1), представља број различитих вредности, док је укупан број променљивих у скупу података. У овом случају , јер колона пол није узета у обзир.

Корелација између две променљиве се рачуна по формули:

|  |  |
| --- | --- |
|  | (1.2) |

|  |  |
| --- | --- |
|  | (1.3) |

У једначинама изнад представља вредност корелације за променљиве и . Додатно коваријанса, , чија вредност не говори ништа о степену зависноти између и , се користи приликом рачунања корелације. Са друге стране корелација има вредности у опсегу , где екстремне врдности означавају јаку корелацију (јаку зависност), а вредности блиске нули значе да таква зависнот уопште не постоји.



Слика Визуелни приказ скупа података са корелационим вредностима

На слици 2 су приказане корелационе вредности где су поља у којима је апсолутна вредност корелације изнад 0.4 осенчена, тако да се лакше могу уочити променљиве које имају израженију међузависност.

По мом мишљењу, овакав графички приказ заједно са вредностима корелационе матрице су добар предуслов за регресиону анализу.

# 3. Регресиона анализа

## 3.1. Линеарна регресија

Регресиона анализа коцептуално представља једноставан метод проналажења функционалних зависности између променњљивих. [4] Та зависност је приказана у облику формуле у коме се са једне стране налази зависна променљива, а са друге стране скуп независних променљивих. Полази се од претпоставке да вредности независних променљивих утичу на вредности зависних променљивих.

Означимо ли зависну променљиву са *Y*, а остале променљиве *X1*, *X2*, … , *Xp* линеарну регресију можемо представити једначином.

|  |  |
| --- | --- |
|  | (2.1) |

Ознаком се означава грешка апроксимације. Уколико са означимо апроксимирану вредност (2.2), једначина (2.1) постаје:

|  |  |
| --- | --- |
|  | (2.2) |
|  | (2.3) |

Из једначине (2.3) је јасно да грешка представља разлику између очекиване и апроксимиране вредности, и пожељно је да та разлика буде што ближа нули[[3]](#footnote-3).

Овај рад ће се фокусирати на линерану регресију, и због тога се (2.1) се представља:

|  |  |
| --- | --- |
|  | (2.4) |

Или векторски:

|  |  |
| --- | --- |
|  | (2.5) |
|  | (2.6) |

Проналажењем параметара вектора , таквих да је вредност минимална, одређује се зависност између и . Величина вектора и је где представља број независних промељивих. Разлог за додавање вредности у вектор је једноставан. Како параметри тог вектора одређују апрксимирајућу функцију, она би без вредности засигурно пролазила кроз координатни почетак. Да би вредност увек била присутна, додата је вредност у вектор , и као што је приказано у једначини (1.4), она увек има исту вредност.

## 3.2. Примена

Линеарну регресију из претходног поглавља применићемо на скупу података описаном раније.

Као што је напоменуто раније, скуп података се састоји од 10 улаза и излаза (ниво дијабатеса). Другим речима, линеарна регресија би требало да процени ниво дијабетеса годину дана након мерења. За ту потребу, пронађени су параметри коришћењем тренинг подскупа, и затим је цео модел проверен тестним подскупом. Резултати се налазе у табели 2.

|  |  |
| --- | --- |
|  | Вредност |
|  | -321.14 |
|  | 0.02 |
|  | -22.73 |
|  | 5.62 |
|  | 1.03 |
|  | -1.03 |
|  | 0.70 |
|  | 0.31 |
|  | 6.85 |
|  | 64.40 |
|  | 0.37 |
| Средња квадратна грешка (MSE) | 1668.71 |
| Корен средње квадратне грешке | 40.85 |

Табела 2 Резултати линеарне регресије

Горњи део табеле 2 показује све вредности вектора . Очигледно је да је параметар додат са разлогом јер има вредност другачију од нуле, што значи да има утицаја на предвиђењу вредност. Доњи део табеле показује колико истренирани модел добро предвиђа.

|  |  |
| --- | --- |
|  | (1.5) |

Разлика између предвиђене вредности и тражене вредности се квадрира, а затим се израчуна просечна вредност такве разлике међу свим пацијентима. Добијена вредност се зове средња квадратна грешка (MSE). Додатно се рачуна и корен ове грешке јер он даје бољу слику колико модел греши. У овом случају, модел приликом процене нивоа дијабетеса за следећу годину греши за око 40 јединица. Важно је приметити да средња квадратна решка не даје никакве информације о знаку, тј. не зна се да ли модел предвиђа више вредности или ниже од очекиваних.

Овај рад се неће даље бавити линераном регресијом, али ће ови резултати служити као мерило (референтне вредности) у будућим експериментима.

# 4. Недостајуће вредности

## 4.1. Механизми недостајућих вредности

Технике машинског учења као сто су надгледано и ненадгледано учење се могу посматрати као системи у ком су улази представљени као подаци, а излази су истренирани модели (алгоритми). Из овог угла посматрања, квалитет података директно утиче на каснију тачност предвиђања истренираног алгоритма. Међутим, улазни подаци често нису комплетни и као такви често онемогућавају тренинг алгоритама.

Некомплетан скуп података се означава и као скуп података са недостајућим вредностима. У таквим скуповима вредности недостају по једном од три могућа механизма, а механизам представља математички однос између података и недостастка. [1]. Механизми могу бити:

**MCAR** – Missing Completely At Random – Nедостајуће вредности су присутне без икакве законитости. Уколико скуп података има 2 променљиве (колоне), непостојаност податка у првој колони нема никакву повезаност са вредностима из обе колоне. Овај случај је веома чест, обзиром да углавном настаје људском ненамерном грешком. На пример, испитаник је случајно превидео одређено питање и оно је остало неодговорено. [3]

**MAR** – Missing At Random – недостајуће вредности присутне унутар једне променљиве немају никакву законитост (повезаност) са том променљивом. Уколико посматрамо исти скуп података (2 колоне), непостојаност податка у првој колони не зависи од вредности те колоне, али зависи вредности из друге колоне(а). На пример, прва колона садржи податке о просечној оцени током студија, а друга колона резултате теста приликом запослења. Испитаници (редови у скупу података) са ниском просечном оценом неће бити ни узети у разматрање, па њихова оцена са теста је ирелевентана, и не садржи се у скупу података.

**MNAR** – Missing Not At Random – недостајуће вредности једне променљиве су директно зависне од посматране променљиве. У скупу са две колоне, недостајуће вредности прве колоне недостају због ње саме, и немају никакве повезаности са другом колоном. На примеру скупа података који садржи резултате теста као променљиву (назив колоне), подаци те колоне могу да недостају у свим редовима где је резултат теста мањи од одређене оцене.

За потребе експеримената у овом раду посматраће се искључиво MCAR механизам недостајућих вредности. Разлог за ту одлуку је могућност добијања недостајућих вредности синтетичким путем. Скуп података из 1.2 (Примена регресионе анализе) ће бити ''пробушен'' више пута насумично и том приликом ће проценат недостајућих вредности бити различит. На тај начин од првобитног комплетног скупа података добиће се више некомплетних, и као додатна погодност знаће се иницијални вредности (касније ће се те почетне вредности користити за евалуацију технике уметања података).

## 4.2. Технике уметања података

У случају недостајућих података, понекад је најлакше одбацити обсервације које нису потпуне. На пример, уколико скуп садржи 100 обсервација (редова) и 10 атрибута (колона) , и од тога 10 различитих редова има тачно једну недостајућу вредност (тачно једну једну колону непопуњену), овом јендноставном техником остало би 90 редова (инстанци) као улаз за алгоритам машинског учења.

Описан скуп података се састоји од 100 редова и 10 колона. Матрица података потенцијално садржи 1000 вредности (потенцијално јер неке вредности нису присутне). Уколико 10 вредности недостаје, ова матрица ће садржати 990 ненедостајућих вредности, што представља 99%. Уколико применимо технику одбацивања обсервација, избацићемо 10 редова, односно укупно 100 вредности, и коначна матрица ће садржати само 90 редова (900 вредности, 90%).

Јасно се види да се оваквим приступом због 1% недостајућих података, може елиминистаи чак 10% укупних вредности. У сваком случају, избацивање података може касније довести до већих грешака предвиђања, јер се скуп података који служи за тренинг драстично смањује [6].

Стога, у овом делу ће бити описане само технике уметања података, где ће се поља која недостају у матрици података заменити (највероватнијом) вредношћу.

### 4.2.1. Уметање средње вредности

Уметање средње вредности је вероватно најједноставнија метода. Она подразумева замену недостајуће вредносте за сваку променљиву (колону) са средњом вредности познатих обсервација у посматрној колони. Овај приступ може бити погодан у случају када мало података недостаје и то по MCAR механизму. У сваком случају смањује се варијанса међу подацима као и корелација између променљивих. [7] Логично је да се варијанса смањује јер до сада непознату вредност замењујемо ''очекиваном'' вредности, и самим тим смањујемо одступање од те ''очекиване'' вредности. Када се каже да ће корелација бити мања мисли се на корелацију између варијабли (колона). Како овим приступом покушавамо да пронађемо везу између свих обсервација (редова) унутарт једне колоне, логично је да смањујемо корелацију између колона. На пример, скуп података садржи две колоне (висина и тежина), и 5 обсервације. (Табела3)

|  |  |
| --- | --- |
| Висина | Тежина |
| 160 | 67 |
| 165 | 65 |
| 158 | 59 |
| ? | 98 |
| 191 | 98 |

Табела Уметање средње вредности

Очигледно је да постоји веза између прве и друге колоне, и да би недостајућа вредност требало бити замењена са 191 (или неком вредношћу блику кој). Међутим, овом техником се та веза (корелација) занемарује и недостајућа вредност ће постати 168.

Уметање средњих вредности се због описаних недостатака неће даље разматрати у раду. Екпериментални скуп података садржи вредности о особама, где неке променљиве (колоне) имају велику варијансу, па свакако није добра идеја сманњивати ту варијансу. Такоће, испитује се утицај разних фактора на ниво дијабетеса, па је корелација један од главних предуслова.

### 4.2.2. Преношење задњег запажања

Преношење задњег запажања је још једна техника која захтева MCAR механизам недостајућих вредности. Веома је популарна у ситуацијама где се посматрају одређене појаве (субјекти) кроз време. Изузетно може бити знимљива у истраживањимакоје садрже номиналне типове атрибута (вредности колона). На пример, посматрајмо медицинско истраживање које прати пацијента (субјекат) кроз време и бележи да ли је узео терапију или није. Другим речима, постоји номинална променљива са могућим вредностима ДА/НЕ. Уколико је пацијент одређеног дана заборавио да унесе да ли је узео лек или није, та вредност ће се попунити са вредношћу из претходног дана.

Међутим, за променљиве нумеричког типа, ова техника није препоручљива јер повећава пристраснот модела, и такође измњује средњу вредност и варијансу (по променљивој) [8]. Ни ова теника неће бити коришћења у експериментима у овом раду, јер се скуп података за тренинг сасатоји углавном од нумеричких типова, и притом павијенти се не посматрају кроз време.

### 4.2.3. Уметање података коришћењем линеарне регресије

Ова техника захтева одређени ниво корелације између променљивих. Идеја је једноставна; у скупу података са две улазне променљиве и једном излазном неке вредности недостају. Потребно је да попунимо све вредности у нпр. користећи вредности из и . Оно што је компликовано у овом примеру је то што приликом уметања података у , очекивано је да се деси да неке од вредности из , такође нису познате. Због тога потребно је направити 3 једначине линеарне регресије како би се адекватно попуниле вредности у колони .

|  |  |
| --- | --- |
|  | (4.1) |

|  |  |
| --- | --- |
|  | (4.2) |

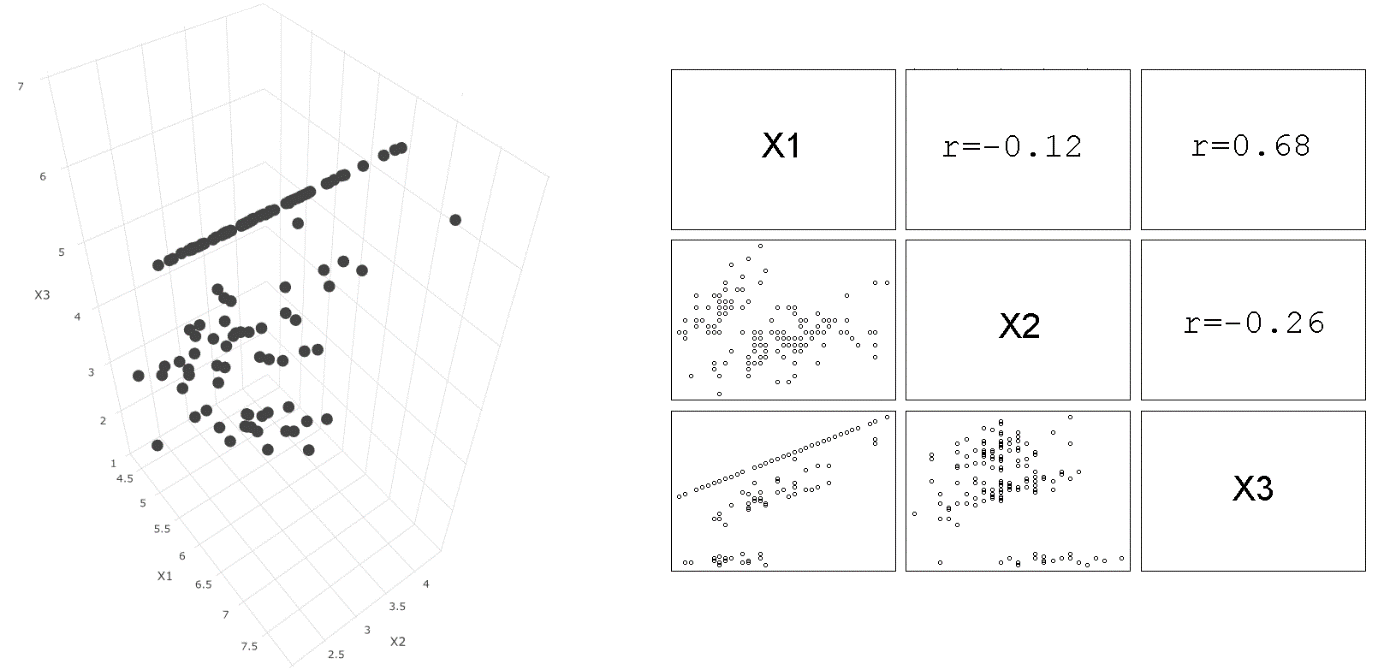
|  |  |
| --- | --- |
|  | (4.3) |

Пример се састоји од само три променљиве и потребно је креирати чак 9 једначина (по 3 једначине за сваку променљиву) како би се адекватно извршила импутација. Поред јасне комплексности проблема, овом техником се повећава корелација између променљивих.

Замислимо да сваки пут када недостаје променљива , такође недостаје и проенљива . Другим речима, подаци недостају по MAR механизму. У том случају, за одређивање вредности користила би се једначина (4.3) и то специјалан облике те једначине:

|  |  |
| --- | --- |
|  | (4.4) |

Овакав начин уметања значајно би повећао вредност корелације између и . На слици (3) је приказана та корелација као и просторни приказ једног скупа података.



Слика 3 Корелација након уметања података линеарном регресијом

Уметање линеарном регресијом захтева да подаци недостају по MCAR механизму, што ће управо бити случај у експериментима у овом раду. Због тога резултујућа корелација између променљивих неће имати вердности као на слици 3. Иако се подаци за експеримент састоје од укупно 11 променљивих () што може да проузрокује велики број једначина линеарне регресије, ова техника ће се користити у даљем раду.

### 4.2.4 Уметање података стохастичком регресијом

У претходном примеру су приказани недостаци уметања линеарном регресијом. Као покушај превзилажења тих недостатака, понекад се користи модификована верзија уметања података која се назива и уметање података стохастичком регресијом. [9].

Идеја је поприлично интуитивна; у једначину линеарне регресије додати још један параметар који ће на случајан начин да промени резултујућу вредност. Самим тим једначина (2.5) постаје:

|  |  |
| --- | --- |
|  | (4.5) |

Како би се једноставније објаснила случајна променљива уместо векторског записа, стохастичку регресију ћемо представити:

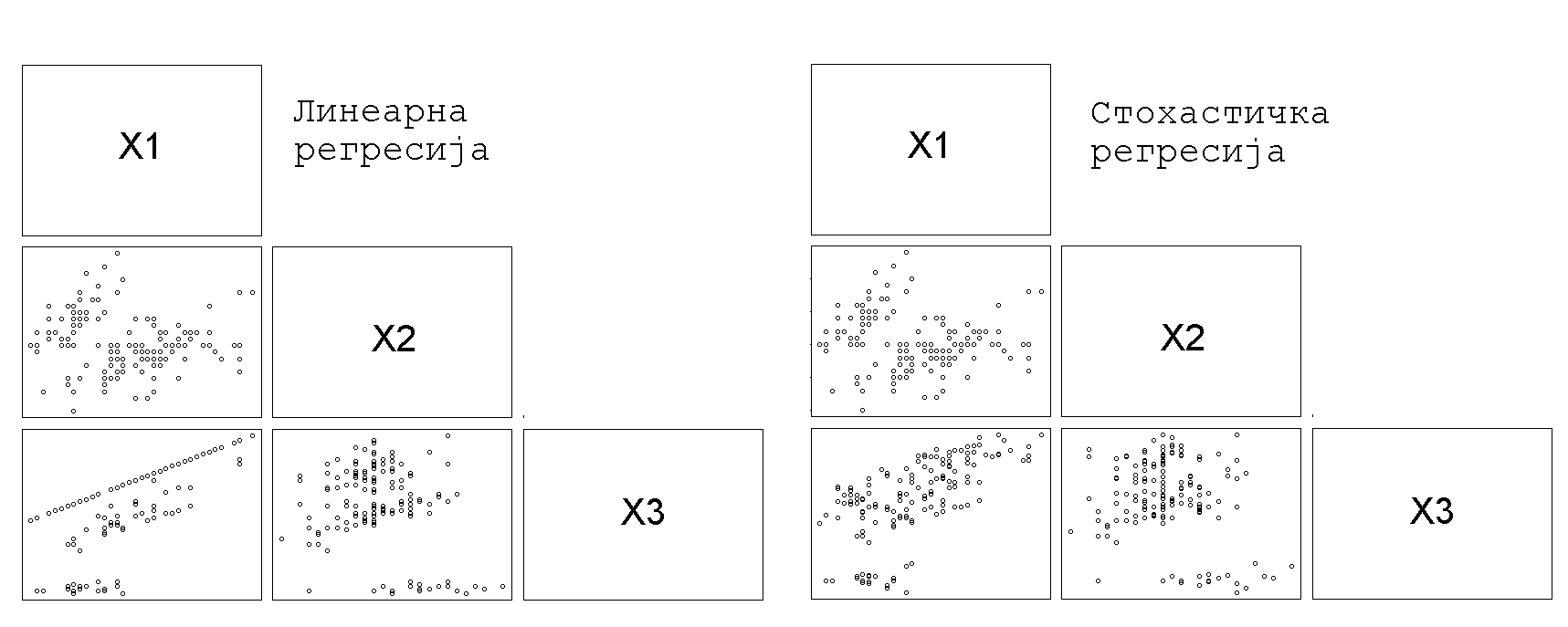
|  |  |
| --- | --- |
|  | (4.6) |

У једначини (4.6), узима вредности од 1 до где је број обсервација у скупу података. Битно је напоменути да је вредност другачија за сваку инстанцу (ред) у скупу. Због те особине није могуће драстично повећати корелацију код попуњеног скупа.

Случајна променљива може да подлеже било којој расподели. Најчешће се користи нормална расподела са очекиваном вредношћу једнаком нули, и стандардном девијацијом једнаком грешци варијанси приликом регресије. [10]

|  |  |
| --- | --- |
|  | (4.7) |

На слици (4) визуелно су приказане обсервације линеарне и стохастичке регресије. Јасно се види да се јака линеарна зависност код линеарне регресије изгубила код стохастичке регресије, мада и даље постоји висок ниво корелације.



Слика 4 Упоредна анлиза импутације линеарном и стохастичком регресијом

Као и техника уметања линеарнoм регресијом, и ова техника ће се користити даље у раду.

### 4.2.5 Уметање коришћењем (шума) стабала одлучивања

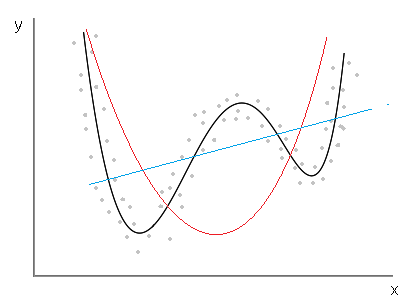
Проблеми са којима се свакодневно сусрећемо често нису линеарни, па коришћење линеарне регресије не даје увек најбоље резултате. Технике 4.2.3 и 4.2.4. се могу унапредити уколико линеарну регресију заменимо са нелинеарним моделом.

Креирање нелинеарне функције није увек тривијалан задатак. Понекад се зависна променљива једино може описати веома комплексном функцијом независне променљиве. Слика 5 показује упоредну анализу различитих регресионих функција на датом скупу података. Подаци се састоје од две променљиве , где је зависна променљива а независна. Плавом линијом је представљена линеарна функција (4.8), црвеном линијом квадратна функција (4.9), а црном линијом функција вишег реда (4.10).

|  |  |
| --- | --- |
|  | (4.8) |

|  |  |
| --- | --- |
|  | (4.9) |

|  |  |
| --- | --- |
|  | (4.10) |



Слика 5 Упоредни приказ регресионих функција

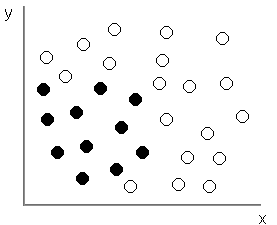
Очигледно је да график функције (4.10) најбоље одговара датом скупу података. Међутим, вероватно је потребно пуно покушаја тренирања са различитим типовима функција да би се добио задовољавајући резултат, односно да регресиона крива постане добар предвиђач.

Другим речима, решавање нелинеарног проблема регресионом кривом може бити веома тежак задатак, а у неким случајевима и немогућ. Због тога ће се у овом раду користити алгоритам тренирања стаблом (стаблима) одлучивања, који подржава и решава нелинеарне проблеме.

#### 4.2.5.1 Стабло одлучивања

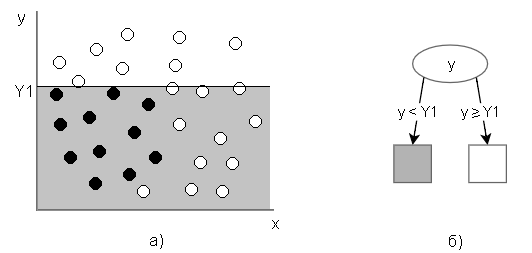
Стабла одлучивања су веома ефектан метод код проблема учења са надгледањем. Основна идеја је поделити скуп података у групе које би требало да буду хомогненије што је могуће више у односу на променљиву по којој се подаци деле.

У случају скупа података где имамо само три променљиве где је зависна (номинална са две класе) променљива, а дељење ће се вршити по променљивима и . Графички приказ овог примера дат је на слици 6. променљива је означена црним и белим круговима.



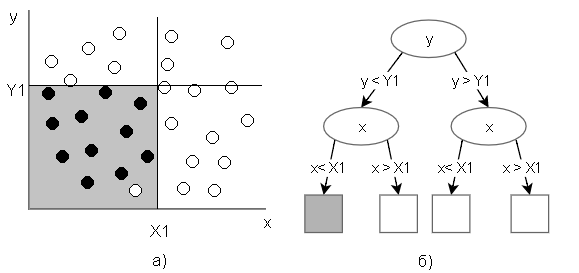
Слика скуп података за тренинг алгоритма стаблом одлучивања

Алгоритам за прављење стабла је једноставан. Најпре се одабере променљива по којој се врши дељење као и вредност која ће поделити ту променљиву. [[4]](#footnote-4) Претпосавка је да је изабрана променљива и да је вредност по којој ће се вршити деоба. У том случају стабло одлучивања и скуп података би изгледао као на слици 7(а). Оно што је битно напоменути је да такво стабло одлучивања има дубину (висину) од једног чвора. Састоји се од једног чвора и два листа. Листови на слици 7(б) имају вредности ``бело`` и ``осенчено``. Другим речима све инстанце које имају вредност (беле или црне) стабло ће их препознати као црне. Очигледно је да је тако мало стабло склоно великој грешци. То се јасно види на слици 7(б), где осенчени део представља предвиђање црног круга у случајевима када је у ствари присутан бео круг.



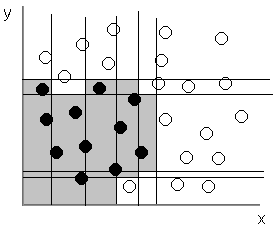
Слика Стабло одлучивања висине 1

Уколико би се креирало стабло са дубином једнаком 2, оно би боље класификовало дати скуп података. На примеру је за други ниво стабла узета променљива са вредношћу . Јасно се види на слици 8 да овакво стабло одлучивања производи знатно мању грешку.



Слика Стабло одлучивања висине 2

Такође, и даље један бео круг припада осенченој области и таква иснтанца би се препознала као класа црне боје. Могуће је стабло још више продубити (повећати му висину). У том случају дошло би то претренираности алгоритма. Резултујући приказ скупа података је приказан на слици 9. Није добро имати ни превише дубоко, ни превише плитко стабло. У првом случају стабло би одлично радило са тренинг подацима али давало би лоше резултате на тестним подацима, док би у другом случају стабло лоше предвиђало и тренинг и тестне податке.

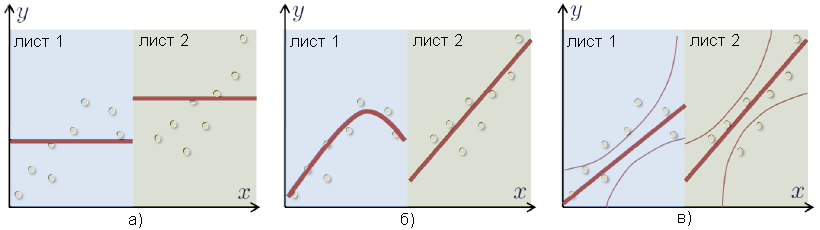


Слика Стабло одлучивања са великом висином

#### 4.2.5.2. Регресионо стабло одлучивања

У претходном поглављу је приказано стабло одлучивања коришћено за класификацију, али такође, могуће је користити стабло и за регресионе проблеме. Метод је веома сличан, само што листови неће предвиђати класу (црно или бело), него ће предвидети одређену вредност.

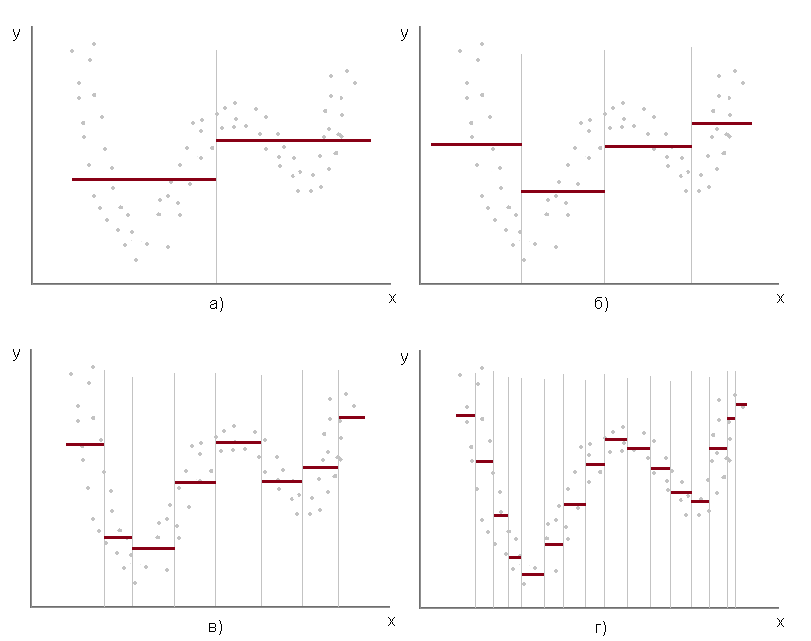
У листовима ће се сад налазити функција , где представља променљиву по којој се тренутно врши деоба, док представља коначну излазну променљиву. Могуће је дефинисати разне функције и оне се могу разликовати између листова унутар једног стабла. Таква функција се назива још и модел предвиђања, и слика 10 приказује примере различитих модела предвиђања. [12]



Слика Примери модела предвиђања

У сва три случаја визуелизованим сликом 10, стабло се састоји од једног чвора и два листа. Самим тим скуп података је подељен на два дела, и сваки лист има своју функцију са скупом података који му одговара. На слици 10а модел предвиђања је константна функција, или другим речима средња вредност одговарајућег подскупа података. Слика 10б има различите моделе предвиђања у листовима: полиномијални и линеарни модел, док слика 10в има линерни модел са укљученом вероватноћом.

На скупу података описаном у 4.2.5 и приказано сликом 5, константан модел предвиђања је приказан на слици 11.



Слика 11 Стабло одлучивања за континуалне податке - константан модел предвиђања

Слика 11а представља модел предвиђања за стабло дубине 1. На графику је оса подељена једном вертикалном линијом, где подскуп података са леве стране те линије припада првом листу, а подскуп података са десне стране припада другом листу. На 11б, исти модел предвиђања је приказан али са стаблом дубине 2, што уствари значи 4 листа. Слика 11в приказује 8 листова или стабло дубине 3, док последња слика 11г визуелизује стабло дубине 4 (16 листова). Сва стабла су у примерима у овом су бинарна (један чвор има тачно двоје деце), али постоје и другачија стабла, -арна, где један чвпр може имати деце.

У примерима изнад је показано како нелинеаран проблем уведен у 4.2.5 може да се решава стаблом одлучивања. И овде важи правило да што је дубље стабло, то је већа вероватноћа да дође до претеренираности.

#### 4.2.5.3 Шуме стабала одлучивања

Уколико је одређивање променљиве по којој се дели јасно дефинисано формулом, онда би само конструисање стабла као и касније предвиђање веома зависило од квалитета скупа података. Како би се такво понашање избегло, уводи се појам случајне променљиве . То значи да кад покренемо алгоритам за прављење стабла пута, где је један од параметара конструисања случајна променљива , добићемо различитих стабала.

Тако конструисана разлилита стабла постају део шуме, и касније се шума користи за предвиђање уместо појединачног стабла. Случајна променљива уводи смањењу корелацију између стабала унутар шуме, што касније значи, повећану генерализацију приликом предиђања. [13]

Замислимо да се шума одлучивања састоји од стабала, и да је потребно да предвидимо резултујућу вредност за одређену до сад непознату обсервацију. Излаз такве шуме ће се састојати појединачних предвиђених вредности (за свако стабло по једна вредност), а коначно предвиђена је просечна вредност појединачних.[[5]](#footnote-5)

Уметање података коришћењем шуме стабала одлучивања је трећа метода која ће бити коришћена у експерименту. Као што је напоменуто у 2.1, експериментални скуп података садржи 10 променљивих нумеричком типа и једну номиналног типа. Попуњаваће се свака колона посебно, коришћењем осталлих колона. На пример, уколико попуњавамо колону , она постаје зависна (предикциона) променљива, а остале вредности се користе као независне. Затим, променљива постаје зависна и тако даље док се не попуне вредности у свим колонама.

# 5. Експеримент

У овом делу ће теорисјки описане методе импутације бити примењене на одређеном скупу података. Најпре ће бити презентован скуп података који се користи у експерименту, а затим ће бити објашњен начин којим ће се обристи одређени проценти вредности из скупа.

Главни део експеримента предтавља примену метода импутације на претходно генерисане скупове података. Након упоредне нализе резултата импутације биће описана метода кластеризације која подржава недостајуће вредности. Тако кластеризовани (али непопуњени) скупови података ће поново бити попуњени једном од три одабране методе, и коначни резултати ће бити приказани као део упоредне анализе свих метода.

## 5.1. Конструисање тренинг скупа

Подаци описани у 2.1. ће служити као основ за креирање скупова података са недостајућим вредностима. Скуп података је иницијално комплетан, али ће коришћењем функције из програмског пакета *R* одређен проценат бити обрисан. Изузетно је битно да се подаци бришу на случајан начин јер се само тако може добити скуп коме недостају подаци по MCAR механизму. Код функције за брисање одређеног процента података из скупа дат у листингу 1. [14]

|  |  |
| --- | --- |
| 1 | # **x** улазни скуп података. |
| 2 | # **noNA** проценат недостајућих вредности у улазном скупу |
| 3 | # (матрици)**x.** |
| 4 | # Подразумевана вредност за **noNA** износи 10%. |
| 5 | prodNA <- function(x, noNA = 0.1){ |
| 6 | n <- nrow(x) |
| 7 | p <- ncol(x) |
| 8 | NAloc <- rep(FALSE, n\*p) |
| 9 | NAloc[sample(n\*p, floor(n\*p\*noNA))] <- TRUE |
| 10 | x[matrix(NAloc, nrow = n, ncol = p)] <- NA |
| 11 | return(x) |
| 12 | } |

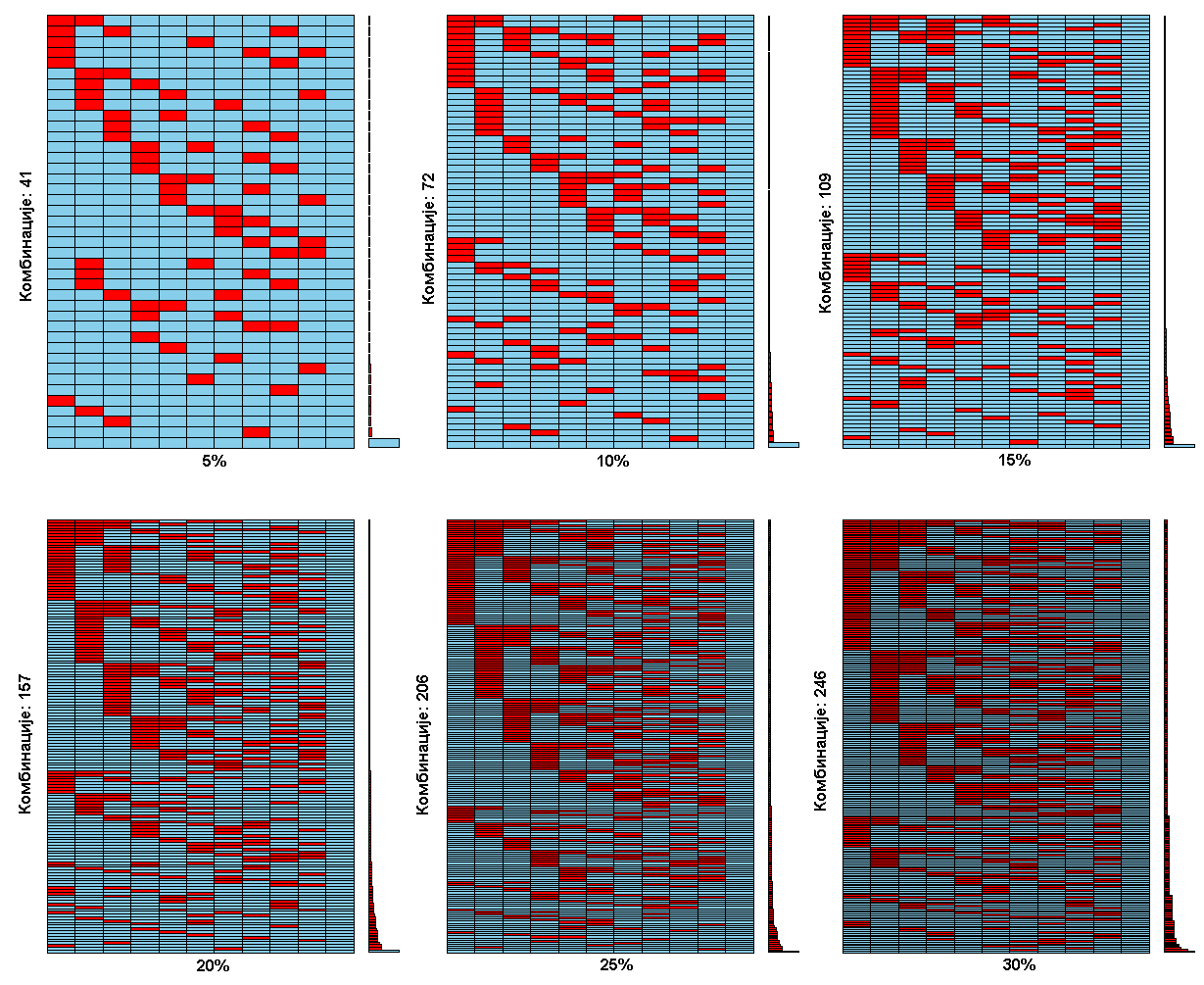
Листинг функција у програмском језику R за генерисање скупа података са недостајућим врендностима

У овом раду ће се за параметар **noNA** користити вредности 5%, 10%, 15% 20%, 25% и 30%. Различите вредности **noNA** ће направити шест различитих скупова за тренинг, које ће се затим попунити техникама описаним у 4.2.3., 4.2.4., 4.2.5. Скрипта написана у језику *R*, којим се од комплетног скупа прави скуп података са недостајућим вредностима се налази у листингу 2.

|  |  |
| --- | --- |
| 1> | XY <- read\_csv("XY.csv") # учитавање са диска |
| 2> | X <- XY[,c(1:10)] # колоне 1-10 су независне променљиве |
| 3> | Y <- XY[,c(11)] # 11-та колона је зависна променљива |
| 4> | X\_05 <- prodNA(X, noNA = 0.05) # брисање 5% подскупа X |
| 5> | XY\_05 <- cbind(X\_05,Y) # враћање зависне променљиве |
| 6> | write**.**csv(XY\_05, file="XY\_05.csv") # уписивање на диск |

Листинг 2 скрипта за брисање 5% података

Листинг 2 садржи пример брисања 5% вредности из екперименталног скупа. За другачије проценте потребно је променити линије 4,5,6, и уместо **05**, ставити жељену вредност. На тај начин је изгенерисано 6 скупова са различитим процентима недостајућих вредности.



Слика Упоредни приказ 6 генерисаних скупова

На слици 12 се налази упоредни приказ свих 6 скупова коришћењем функције из листинга 1[[6]](#footnote-6). Битно је напоменути да сваки од 6 скупова има комплетну последњу колону (ниједна вредност не недостаје). Та колона представља зависну променљиву , и њој не могу недостајати подаци али ће се она свакако користити при импутацији. Такође број комбинација, образаца, се повећава заједно са процентом недостајућих врендости. Очекивано највише комбинација по којима недостају вредности се налази код скупа података коме недостаје 30% података, чак 246 комбинација.

## 5.2. Интерпретација резултата импутације података

У овом поглављу су представљене мере које ће се користити за евалуацију различитих техника импутације података. Скуп података који ће бити коришћен као пример налази се у табелама 4, 5 и 6.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 4 | 2 | 1 | 7 |
| 3 | 5 | 2 | 4 |
| 7 | 5 | 6 | 7 |
| 8 | 5 | 1 | 3 |
| 4 | 7 | 8 | 4 |
| 6 | 3 | 2 | 8 |
| 2 | 4 | 4 | 1 |

Табела 4 Потпуни (почетни) скуп података

Табела 4 садржи све вредности и те вредности су реферетне за даљу анализу. Табела 5 садржи скуп података без 10% вредности што је добијено вештачким путем описаним у 5.1. Три колоне (, , ) са по седам редова садрже укупно 21 вредност, и укупно треба обрисати 2 врдности.[[7]](#footnote-7)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 4 | 2 | 1 | 7 |
| 3 |  | 2 | 4 |
| 7 | 5 | 6 | 7 |
| 8 | 5 | 1 | 3 |
| 4 | 7 |  | 4 |
| 6 | 3 | 2 | 8 |
| 2 | 4 | 4 | 1 |

Табела 5 Непопуњен скуп података

Табела 6 садржи уметнуте податке коришћењем стохастичке регресије (4.2.4.), и она заједно са табелом 4 представља основ за даљу анализу грешака. Очигледно је да су уметнуте две вредности, уместо иницјалне вредности 5 (ред 2, колона ), уметнута је вредност 3. Такође, попуњена је вредност 9 (колона 5, колона ) уместо почетне вредности 8. Сажет приказ уметнузих вредности је приказан у табели 7.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 4 | 2 | 1 | 7 |
| 3 | 3 | 2 | 4 |
| 7 | 5 | 6 | 7 |
| 8 | 5 | 1 | 3 |
| 4 | 7 | 9 | 4 |
| 6 | 3 | 2 | 8 |
| 2 | 4 | 4 | 1 |

Табела 6 Уметнути подаци

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ред | Колона | Почетна вредност | Уметнута вредност | Разлика |
| 2 |  | 5 | 3 | -2 |
| 5 |  | 8 | 9 | 1 |

Табела 7 Резултат импутације

### 5.2.1. Средња квадратна грешка импутације

У овом случају је могуће упоредити вредности почетног и попуњеног скупа и на основу њих израчунати средњу квадратну грешку импутације. Ова врста грешке се рачуна формулом:[[8]](#footnote-8)

|  |  |
| --- | --- |
|  | (5.1) |

У формули (5.1), означава почетну вредност, уметнуту вредност док означава број уметнутих вредности. У примеру из табеле 7, формула (5.1) би изгледала:

|  |  |
| --- | --- |
|  | (5.2) |

|  |  |
| --- | --- |
|  | (5.3) |

|  |  |
| --- | --- |
|  | (5.4) |

У датом примеру средња квадратна грешка би износила 2.5. Касније ће бити бити дата упоредна анализа свих грешака на датом примеру.

### 5.2.2. Корен средње квадратне грешке

У одељку 5.2.1. је уведена средња квадратна грешка. Разлика између уметнуте и почетне вредности се квадрира како би се изгубила важност знака.[[9]](#footnote-9) Међутим, то проурукује повећању грешке уколико је грешка већа од 1, и смањењу уколико је грешка мања од један. Како би се избегло такво понашање, десни део једначине (5.1) је потребно кореновати.[[10]](#footnote-10)

|  |  |
| --- | --- |
|  | (5.5) |

|  |  |
| --- | --- |
|  | (5.6) |

Другим речима, потребно је пронаћи корен вредности израчунате у 5.4 што износи:

|  |  |
| --- | --- |
|  | (5.7) |

Из једначине (5.7) се види да је корен средње квадратне фрешке 1.58.

### 5.2.3. Просечна релативна грешка

Вредности апсолутне грешке (као и средње квадратне и корена средње квадратне грешке) нису увек најбољи показатељи. На пример, апсолутна грешка вредности је само 1% од процењене вредности , али чак 50% од процењене вредности . [15] где у првом случају распон могућих вредности износи 1-100 (нумверички тип), а у другом случају је од 1-2 (номинални тип).

Другачије речено, за презентацију резултата импутације потребно је узети у обзир и тип као и распон вредности променљиве (колоне) у коју се подаци уносе. На пример, у табели 8 су приказане 4 колоне, где су и номиналног типа (могуће вредности су 1 и 2) и колоне и су нумеричког типа типа (вредности имају распон од 1 до 100).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 | 2 | 45 | 44 |
| 2 | 2 | 67 | 67 |
| 2 | 1 | 31 | 32 |

Табела Вредности пре и након импутације

Колоне са индексом 2 (друга и четврта колона) су уметнуте вредности док су колоне са индексом 1 (прва и трећа колона) иницијалне, праве вредности. У првом реду, алгоритам је унео вреност за један већу од почетне, у другом реду је проценио вредност идентичну почетној, док је у трећем реду уметнута вредност за један мања од иницијалне.

За рачунање просечне релативе грешке користи се једначина (5.8).[[11]](#footnote-11)

|  |  |
| --- | --- |
|  | (5.8) |

|  |  |
| --- | --- |
|  | (5.9) |

Користећи формулу (5.8) и податке из табеле 8, израчуната је просечна релативна грешка. Такође, над истом табелом израчунат је корен средње квадратне грешке и подаци су дати у табели 9.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 0.67 | 0.67 | 0.67 | 0.013 |

Табела Поређење средње квадратне грешке и просечне релативне грешке

Из табеле 9 се јасно види колико бољи показатељ мође бити просечна релативна грешка као грешка импутације. Просечна релативна грешка је много већа у случају номиналног типа података, што је и очекивано јер је опсег вредности те променљиве веома мали.

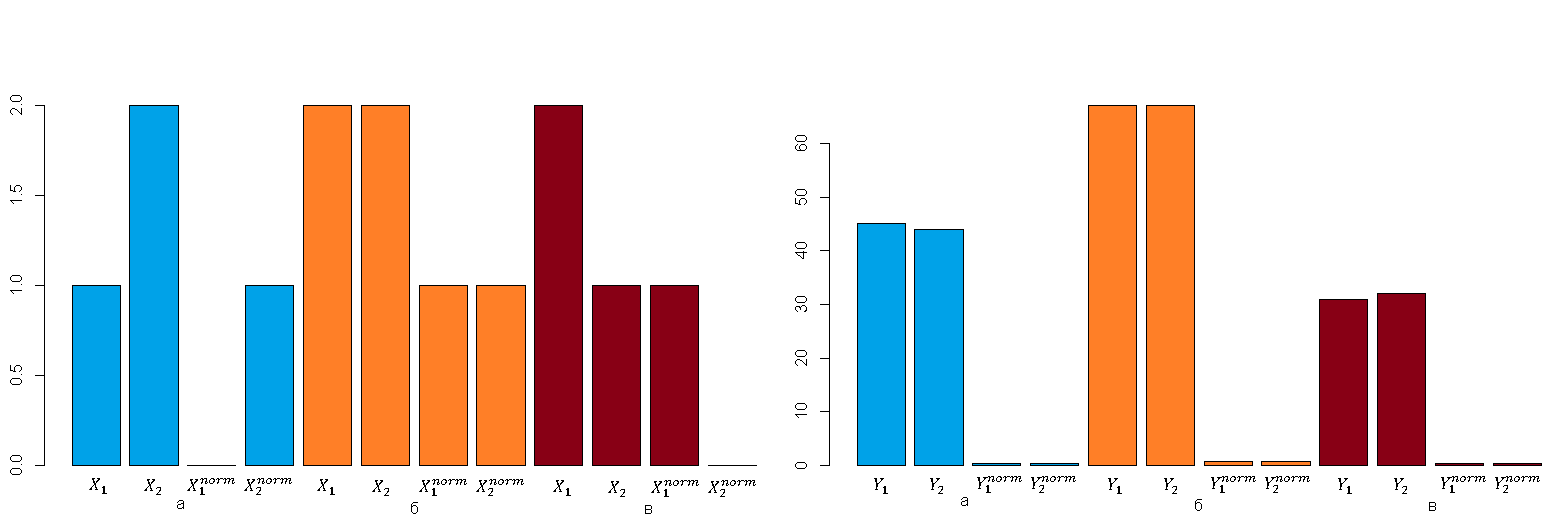
Како би се још боље показао значај погрешно унетих врености из табеле 8, урађена је нормализација свих врености једначином (5.10) и резултати су приказани у табели 10.

|  |  |
| --- | --- |
|  | (5.10) |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| а | 1 | 2 | 0 | 1 | 45 | 44 | 0.44 | 0.43 |
| б | 2 | 2 | 1 | 1 | 67 | 67 | 0.67 | 0.67 |
| в | 2 | 1 | 1 | 0 | 31 | 32 | 0.30 | 0.31 |

Табела Упоредни приказ апсолутних и нормализованих вредности

Формула (5.10) нормализује све вредности на скуп вредности [0,1]. Очигледно је да су нормализоване ведности номиналног типа (лево део табеле 10) уствари екстремне вредности посматране скале. Дакле, уколико алгоритам импутације погреши за један (апсолутна мера), по релативној или нормалзованој скали, грешка је огромна. Са друге стране, иста апсолутна вредност грешке (један), на нумеричкој скали је занемарљива. Графички приказ табеле 10 је дат на слици 12, где су такође на посебним графицима одвојене различити типови променљивих (лево се налазе графици за номинални, а десно за нумерички тип). Боје на слици 12 одговарају редовима из табеле 10.



Слика Упоредни приказ апсолутних и нормализованих врендости

### 5.2.4. Корен средње квадратне грешке линеарне регресије

Корен средње квадратне грешке је објашњен у одељку 5.2.2., где се грешка објашњавала разликом почетне и уметнуте вредности. Такође, у одељку 3.2. је описана примена линеарне регресије на експериментални скуп и одређене су средња квадратна грешка као и њен корен.

Те две вредности показују колико добро процењује линеарна регресија када су подаци комплетни. Након уметања података и очекује се да се и грешка промени, тако да ће то бити још један од параметара за одређивање ефикасности импутације.

## 5.3. Импутација линеарном регресијом

У одељку 4.2.3. је теоријски описан метод импутације линеарном регресијом. Тако описана импутација је имплементирана у софтверском пакету R, тачније, имплементација је садржана у библиотеци mice. [16] Управо је ова библиотека коришћена за импутацију непостојећих вредности.

У листингу 3 је дат пример кода који се користио за попуњавање једног скупа (са 5% непостојећих вредности). За остале скупове са различитим процентима недостајућих вредности, користила би се иста скрипта само би улазни параметар функције *mice* био другачији.

|  |  |
| --- | --- |
| 1> | # Улазни параметри: |
| 2> | # **XY\_05** – скуп података |
| 3> | # **method= "norm.predict"** – означава методу импутације, |
| 4> | # тј. линеарну регресију |
| 5> | # **m=5** – број функција креираних за сваку променљиву |
| 6> | # |
| 7> | # Повратна вредност: |
| 8> | # **XY\_05\_model** који садржи параметре линеарне функције, |
| 9> | # оригинални скуп података, метаподатке... |
| 10> | # Не садржи уметнуте вредности. |
| 11> | XY\_05\_model <- mice(XY\_05, method="norm.predict", m=5) |
| 12> |  |
| 13> | # Креирање скупа података где су недостајуће вредности |
| 14> | # замењене уметнутим вредностима |
| 15> | XY\_05\_imp <- complete(XY\_05\_model) |
| 16> |  |
| 17> | # Визуелизација резултата (слика 14) |
| 18> | densityplot(XY\_05\_model) |

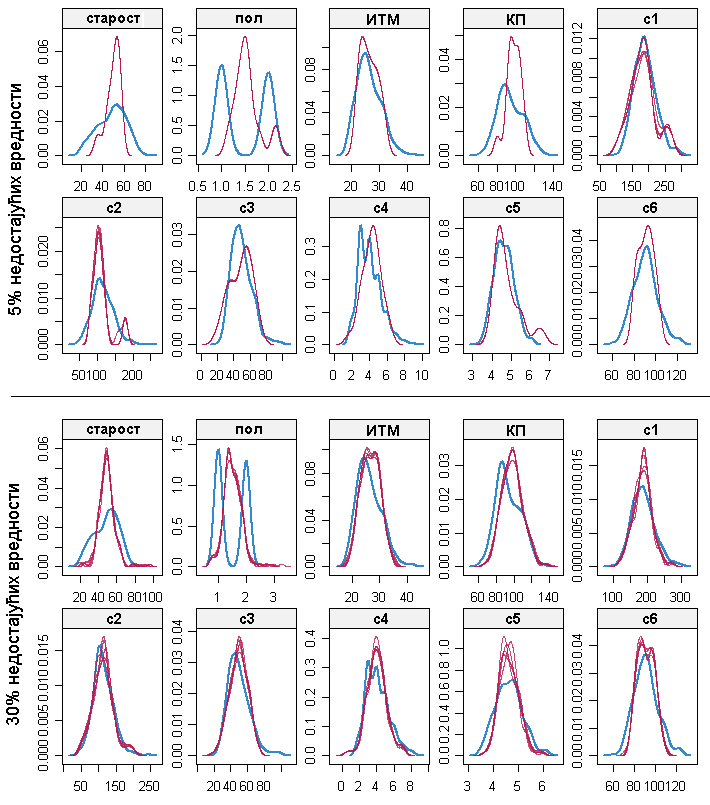
Листинг Импутација линеарном регресијом

Позивањем функције densityplot из листинга 3, добија се скуп од 10 дијаграма приказаних на слици 14. Иако скуп података садржи и једанаесту променљиву , та променљива није приказана јер садржи све вредности, па није било потребе за импутацијом. Плавом линијом на дијаграмима су приказана измерена појављивања, односно њихова расподела, док су црвеним линијама су означене расподеле уметнутих вредности. Један дијграм садржи углавном једну плаву и више црвених линија (највише 5), јер се за једну променљиву конструисало толико фунцкија.

Преклапањем црвене и плаве линије не значи да је импутација била успешна, али непреклапање сигурно значи да ће импутација произвести велику грешку.

Занимљиво је обратити пажњу на дијаграме који одговарју променљивој **пол**. Како је та променљива номиналног типа, за њу би била прикладнија логистичка регресија уместо линеарне. Због тога је велико одступање између измерених и уметнутих расподела.

Такође, ради једноставности слике, приказани су дијаграми добијени импутацијом података у два екстремна случаја, у скупове података са 5% и 30% недостајућих вредности. Види се да су расподеле сличне између истих променљивих у оба случаја, што је добар показатељ. Следи анализа грешке, која ће заиста показати успешност импутације.



Слика 14 Расподеле појављивања измерених и уметнутих вредности

У табели 11 се налази приказ грешака описаних у 5.2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| % | Средња  квадратна  грешка | Корен средње квадратне грешке | Просечна релативна грешка | Корен средње квадратне грешке линеарне регресије |
| 5 | 43.54469 | 6.59884 | 0.109273 | 40.93458 |
| 10 | 98.69796 | 9.934685 | 0.111054 | 40.78019 |
| 15 | 114.4606 | 10.69863 | 0.107427 | 41.76785 |
| 20 | 75.41079 | 8.683939 | 0.112437 | 41.84145 |
| 25 | 135.0136 | 11.61953 | 0.117311 | 54.97207 |
| 30 | 127.6473 | 11.29811 | 0.112443 | 95.63886 |

Табела 11 Приказ грешака насталих приликом импутације линеарном регресијом

Поред укупних грешака, у табели 12 су приказане и просечне релативне грешке по променљивој, где се види значај лоше процене линеарне регресије на номиналну променљиву (променљива пол).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Променљива | 5% | 10% | 15% | 20% | 25% | 30% |
| старост | 0.1725 | 0.1722 | 0.1721 | 0.1955 | 0.1698 | 0.1863 |
| пол | 0.4091 | 0.3429 | 0.3021 | 0.3299 | 0.3040 | 0.1884 |
| ИТМ | 0.0960 | 0.0879 | 0.1044 | 0.1120 | 0.1074 | 0.1142 |
| КП | 0.1134 | 0.1335 | 0.1201 | 0.1138 | 0.1219 | 0.1479 |
| с1 | 0.0220 | 0.0411 | 0.0499 | 0.0384 | 0.0629 | 0.0700 |
| с2 | 0.0211 | 0.0409 | 0.0484 | 0.0344 | 0.0585 | 0.0585 |
| с3 | 0.0322 | 0.0466 | 0.0432 | 0.0546 | 0.0706 | 0.0654 |
| с4 | 0.0510 | 0.0443 | 0.0542 | 0.0493 | 0.0539 | 0.0613 |
| с5 | 0.0399 | 0.0538 | 0.0558 | 0.0637 | 0.0983 | 0.0938 |
| с6 | 0.1351 | 0.1470 | 0.1236 | 0.1322 | 0.1254 | 0.1382 |

Табела 12 Просечна релативна грешка за сваку пеоменљиву- импутација линеарном регресијоом

Ради лакшек приказа, табела 12 је греафички предстваљена на слици 15. Лако је уочљиво да је највећа грешка по променљивој **пол**. Такође, што је већи проценат недостајућих вредности у скупу података, грешка је све већа.

Слика Графички приказ релативне грешке за сваку променљиву - импутација линеарном регресијом

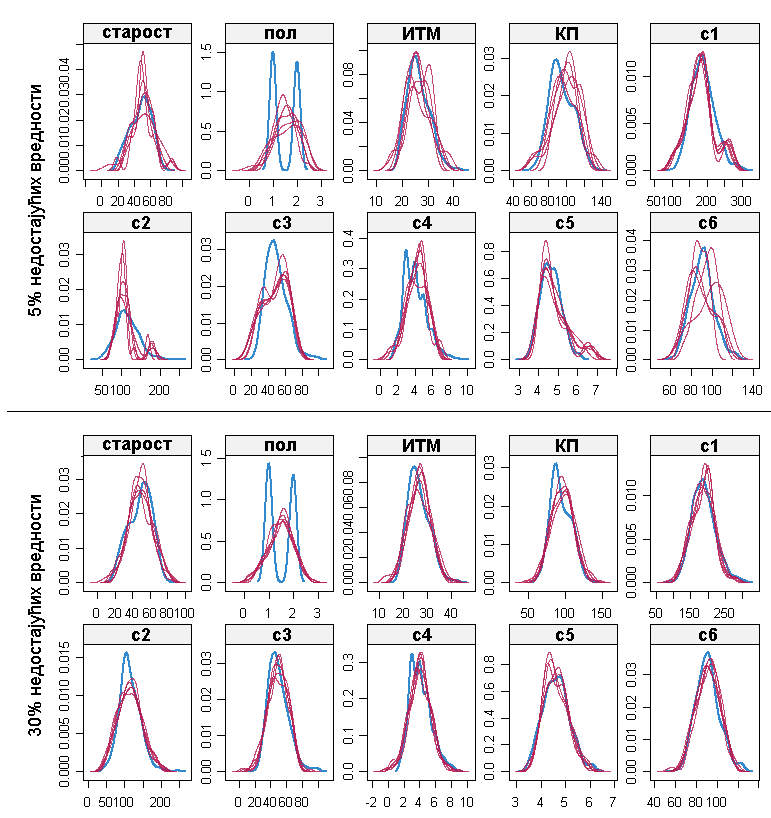
## 5.4. Импутација стохастичком регресијом

Слично претходном одељку, и импутација стохастичком линеарном регресијом је описана у 4.2.4. И за ову врсту импутације коришћена је имплементација из билбиотеке *mice* пакета *R*. Код који се користи за импутацију је идентичан коду из листинга 3, осим што се уместо методе "norm.predict" користити метода "norm.nob". Листинг 4 садржи позив функције којом се извршава импутација стохастичком линеарном регресијом.

|  |  |
| --- | --- |
| 1> | XY\_05\_model <- mice(XY\_05, method="norm**.**nob", m=5) |
| 2> | XY\_05\_imp <- complete(XY\_05\_model) |
| 3> | densityplot(XY\_05\_model) |

Листинг 4 пример кода импутације за стохастичку линеарну фрегресију

Као и у претходном одељку, и овде је приказана расподела забележених и уметнутих вредности. Опет плава линија предстваља расподелу забележених, а црвене линије расподелу уметнутих вредности (слика 16).



Слика Расподела забележених и уметнутих вредности - стохастичка линеарна регресија

У табели 13 су приказане грешке описане у одељку 5.2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| % | Средња  квадратна  грешка | Корен средње квадратне грешке | Просечна релативна грешка | Корен средње квадратне грешке линеарне регресије |
| 5 | 43.54469 | 6.59884 | 0.121119 | 41.39864 |
| 10 | 98.69796 | 9.934685 | 0.129445 | 41.87477 |
| 15 | 114.4606 | 10.69863 | 0.120585 | 42.27468 |
| 20 | 75.41079 | 8.683939 | 0.12555 | 41.43752 |
| 25 | 135.0136 | 11.61953 | 0.123229 | 40.67999 |
| 30 | 127.6473 | 11.29811 | 0.1293 | 43.11288 |

Табела 13 Приказ грешака насталаих импутацијом стохастичком линеарном регресијом

Занимљиво је приметити да вредности просечне релативне грешке веома слични за све проценте недостајућих вредности. Исто се може рећи и за последњу колону, вредност грешке након извршавања линеарне регресије.

Додатно, табела 14 садржи просечне релативне грешке за сваку променљиву посебну, док су приказане вредности визуелизоване на слици 17.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Променљива | 5% | 10% | 15% | 20% | 25% | 30% |
| старост | 0.2095 | 0.2488 | 0.1825 | 0.2556 | 0.2125 | 0.1926 |
| пол | 0.2640 | 0.2594 | 0.2121 | 0.2241 | 0.1820 | 0.2278 |
| ИТМ | 0.1648 | 0.11 | 0.1276 | 0.1345 | 0.1360 | 0.1414 |
| КП | 0.1343 | 0.1956 | 0.1673 | 0.1367 | 0.1238 | 0.1489 |
| с1 | 0.0195 | 0.0387 | 0.0575 | 0.0524 | 0.0780 | 0.0823 |
| с2 | 0.0255 | 0.0473 | 0.0513 | 0.0451 | 0.0722 | 0.0644 |
| с3 | 0.0589 | 0.0695 | 0.0577 | 0.0746 | 0.0854 | 0.0865 |
| с4 | 0.0792 | 0.0626 | 0.0823 | 0.0693 | 0.0734 | 0.0783 |
| с5 | 0.0662 | 0.0671 | 0.0758 | 0.0808 | 0.0926 | 0.1133 |
| с6 | 0.1888 | 0.1950 | 0.1912 | 0.1820 | 0.1759 | 0.1569 |

Табела 14 Просечна релативна грешка за сваку пеоменљиву- импутација стохастичком линеарном регресијоом

Слика Графички приказ релативне грешке за сваку променљиву - импутација стохастичком линеарном регресијом

[1] Applied Missing Data Analysis, 3

[2] Little, R.J.A. and Rubin, D.B. (1987) Statistical Analysis with Missing Data. John Wiley & Sons, New York

[3] Prevention And Treatment Of Item Nonresponse, 155

[4] Regression Analysis By Example, 1

[5] Least Angle Regression

[6] Data Analysis Using Regression and Multilevel/Hierarchical Models, 531

[7] Principal component analysis with missing values, 659

[8] The Application of Last Observation Carried Forward in the Persistent Binary Case

[9] An introduction to modern missing data analyses, 13

[10] <https://www.ma.utexas.edu/users/davis/375/popecol/lec4/stoch.html>

[11] A comparative study of decision tree ID3 and C4.5

[12] Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning, 50

[13] Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning, 15

[14] <https://github.com/stekhoven/missForest/blob/master/R/prodNA.R>

[15] Relative Error Measures for Evaluation of Estimation Algorithms

[16] mice: Multivariate Imputation by Chained Equations in R

[17] Visualization of imputed values using the R-package VIM

1. ИТМ, индекс телесне масе [↑](#footnote-ref-1)
2. КП, крвни притисак [↑](#footnote-ref-2)
3. Заправо уколико је апроксимациона грешка једнака нули (или веома блиска нули), може се доћи до претренираности алгоритма што је непожељан ефекат. Другим речима, алгоритам би одлично радио са тренинг подацима али показао би веома лоше резултате над тестном скупу података. Претренираност је честа појава код нелинеарних апроксимација, што у овом раду није случај, па је апроксимирајућа грешка блиска нули пожељна. [↑](#footnote-ref-3)
4. Најчешће се користи алгоритам ID3 заједно са алгоритмом за рачунање ентропије C4.5 За више информација[11] [↑](#footnote-ref-4)
5. Овакав начин предвиђања вредности је применљив за константан модел предвиђања који ће бити коришћен у овом раду. Постоје и друге технике одређивања које узимају у обзир вредности појединачних стабала али оне неће бити разматране у овом раду. [↑](#footnote-ref-5)
6. За генерисање скупова коришћена је *VIM* библиотека пакета *R*. [17] [↑](#footnote-ref-6)
7. Листинг 1, линија 9 садржи функцију под називом floor која има дефиницију [↑](#footnote-ref-7)
8. MSE – (Mean Squared Error), средња квадратна грешка [↑](#footnote-ref-8)
9. Приликом квадрирања позитивне и реципрочне негативне вредности добија се исти резултат. Разлог због ког се често користи квадратна, а не апсолутна грешка је могућност израчунавања првог извода у даљој анализи. Апсолутна грешка такође занемурује важност знака ( ). [↑](#footnote-ref-9)
10. RMSE – (Root Mean Squared Error), корен средње квадратне грешке [↑](#footnote-ref-10)
11. ARE – (Average Relative Error), средња релативна грешка [↑](#footnote-ref-11)