УНИВЕРЗИТЕТ У БЕОГРАДУ

ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

ЗАВРШНИ (МАСТЕР) РАД

Утицај метода импутације података на тачност предвиђања

Ментор:

др Братислав Петровић

Студент:

Михаило Ступар 2015/3503

Београд

Новембар, 2017.

Комисија која је прегледала рад

кандидата СТУПАР (ЗОРАН) МИХАИЛА

под насловом УТИЦАЈ МЕТОДА ИМПУТАЦИЈЕ ПОДАТАКА

НА ТАЧНОСТ ПРЕДВИЂАЊА и одобрила одбрану

Ментор: др Братислав Петровић, редовни професор

_____

Члан: др Ана Поледица, доцент

_____

Члан: др Борис Делибашић, редовни професор

_____

АПСТРАКТ

Овај рад се бави утицајем различитих метода импутације података на тачност предвиђања коришћењем регресије. Такође, у раду је презентована упоредна анализа три методе импутације података (импутација линеарном регресијом, стохастичком линеарном регресијом и шумама стабала одлучивања). Додатно је предложена нова метода импутације заснована на кластеровању и анализиране су њене перформансе. У првом делу рада описана је линеарна регресија као и методе импутације које се користе у експерименту. Током експеримента, скуп података са различитим процентима недостајућих вредности је попуњен сваком од набројаних метода, и притом су упоређивани резултати саме импутације. Затим је описана нова, предложена метода импутације која се састоји од две фазе. Прва фаза, фаза припреме, је кластеризација података, док друга фаза представља импутацију података на нивоу кластера. Овом приликом, у првој фази се користи кластеровање К-средњих вредности (које подржава недостајуће вредности), док се у другој фази користи метода импутације која је показала најбоље резултате у првом експерименту. На крају рада је дат упоредни приказ свих резултата добијених у раду; резултата из првог експеримента као и резултата добијених приликом импутације предложеном методом.

ABSTRACT

This paper focuses on the influence of different data imputation methods on the correctness of prediction using regression. Also, three data imputation methods are compared and the results of analysis are presented (linear regression imputation, stochastic linear regression imputation and imputation using random decision forests). Additionally, a new imputation method is proposed and its efficiency analyzed. The first part of the paper contains the linear regression description, as well as the description of three imputation methods. During the experiment, a data set with different ratios of missing values is populated by using each of the listed methods while the imputation results are analyzed. Afterwards, a new proposed method is explained, which consists of two phases. The first phase, which is the preparation phase, is clusterization, while the second phase represents data imputation at the cluster level. In the first phase, k-means clustering is used (which supports missing values) and in the second phase the data are imputed using the best method from the first part of the paper. At the end, the comparative view of all results from this paper is presented; the results from the first experiment as well as the result obtained from the proposed method.

CURRICULUM VITAE

Михаило Ступар

e-mail: stupar.mih@gmail.com

Лични подаци:

Датум рођења: 23.01.1992.

Адреса: Камчатска 17, 11000 Београд

Телефон: 063/7107564

Образовање:

- Универзитет у Београду, Факултет организационих наука, одсек информациони системи и технологије, 2011–2015, просечна оцена 9.33
- Гимназија "Пета београдска гимназија", Београд, природно-математички смер, 2007–2011

Радно искуство:

- Software Developer, msg global solutions SEE, од 07.09.2015.

Објављени радови:

- Mihailo Stupar, Pavle Milošević, Bratislav Petrović. (2017). *A Fuzzy Logic-Based System for Enhancing Scrum Method*. Management: Journal of Sustainable Business and Management Solutions in Emerging Economies

Додатне квалификације:

- Java

Садржај

1. Увод

Машинско учење, односно предвиђање коришћењем техника машинског учења, је данас све популарније. То се може приписати доступношћу података (захваљујући интернету), али и олакшаном креирању веома комплексних модела предвиђања (брзина процесора, величина меморија).

Како се наведене технике углавном ослањају на историјске податке приликом тренирања модела машинског учења, квалитет предвиђања директно зависи од квалитета скупа података. Под квалитетом података се подразумева више фактора: количина података, начин на који подаци описују одређену појаву, комплетност података и многи други.

Овај рад се бави скуповима података који нису комплетни, односно утицајем некомплетног скупа података на тачност предвиђања. Такође, бави се и методама којима се проблеми некомплетног скупа могу превазићи. Додатно је предложена нова метода којом се такође може превазићи поменути проблем.

Уколико је скуп података којим се тренира модел предвиђања некомплетан (са недостајућим вредностима) могу се применити две основне технике: обрисати обсервацију у којима недостају вредности или заменити недостајућу вредност и користити посматрану обсервацију.

Приликом брисања некомплетних обсервација долази до значајног губљења информација, па ова метода није препоручљива [29]. Као алтернатива брисању, неопходно је непознату вредност заменити највероватнијом [27]. У последњих двадесет пет година овакве технике (методе) импутације су доживеле револуцију [23]. У литератури се поред метода импутације података користе још и термини замене или уметања података.

Једна од најлакших метода је импутација једне вредности. Овом методом непозната вредност одређене колоне се мења са на пример средњом вредношћу посматране колоне [21]. Поред ње доста је заступљена и импутација регресионим методама [20]. Такође, све популарније су методе

вишеструке импутације. Уместо да се непозната замени једном вредношћу, прави се скуп потенцијалних вредности којима би могла да се замени непозната [20]. Касније, скуп потенцијалних вредности се анализира и одређује се једна вредност којом се мења непозната [24]. Вишеструка импутација се показала као средство којим могу да се реше проблеми услед импутације једном вредношћу [19].

Неке од претходно описаних метода ће се користити у раду приликом експеримената. Над скупом података са непостојећим вредностима биће извршене три методе импутације: импутација линеарном регресијом, стохастичком линеарном регресијом и шумом стабала одлучивања. Приликом сваке од метода рачунаће се грешка саме импутације, али и могућност предвиђања након импутације. На основу дефинисаних параметара одредиће се метода импутације која даје најбоље резултате.

Затим ће бити предложена нова метода импутације која се састоји од припреме података и саме импутације претходно пронађеном најбољом методом. На крају ће бити представљена упоредна анализа предложене методе и претходно означене најбоље методе.

2. Проблем редвиђања - регресиона анализа

Како ће се у раду показати утицај метода импутације на тачност предвиђања, неопходно је да се најпре дефинише сам проблем предвиђања. Техника предвиђања која ће се користити у раду је линеарна регресија и она је описана у одељцима који следе.

2.1. Линеарна регресија

Регресиона анализа концептуално представља једноставан метод проналажења функционалних зависности између променљивих [4]. Та зависност је приказана у облику формуле у којој 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 $x_1, x_2, \dots, x_n$ линеарну регресију можемо представити једначином.

$$y = f(x_1, x_2, \dots, x_p) + \varepsilon \tag{2.1}$$

Ознаком ε се означава грешка апроксимације. Уколико са $\hat{y}$ обележимо апроксимирану вредност (2.2), једначина (2.1) постаје:

$$\hat{y} = f(x_1, x_2, \dots, x_p) \tag{2.2}$$

$$y = \hat{y} + \varepsilon \tag{2.3}$$

Из једначине (2.3) је јасно да грешка ε представља разлику између очекиване и апроксимиране вредности, и пожељно је да та разлика буде што ближа нули[1].

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

$$y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon \tag{2.4}$$

или векторски:

$$\boldsymbol{Y} = \boldsymbol{\beta}\boldsymbol{X}^T + \varepsilon \tag{2.5}$$

$$\boldsymbol{X} = [\boldsymbol{X}_0\ \boldsymbol{X}_1 \dots \boldsymbol{X}_p],\ \boldsymbol{\beta} = [\boldsymbol{\beta}_0\ \boldsymbol{\beta}_1 \dots \boldsymbol{\beta}_p],\ \boldsymbol{X}_0 = \mathbf{1} \tag{2.6}$$

Проналажењем параметара вектора $\boldsymbol{\beta}$, таквих да је вредност ε минимална, одређује се зависност између $\boldsymbol{X}$ и $\boldsymbol{Y}$. Величина вектора $\boldsymbol{\beta}$ и $\boldsymbol{X}$ је $(p+1)$ где p представља број независних промељивих. Разлог за додавање вредности β_0 у вектор $\boldsymbol{\beta}$ је једноставан. Како параметри тог вектора

[1]Заправо уколико је апроксимациона грешка једнака нули (или веома блиска нули), може доћи до претренираности алгоритма што је свакако непожељан ефекат. Другим речима, алгоритам би одлично радио са тренинг подацима али показао би веома лоше резултате на тестном скупу података. Због зога је битно да се након тренирања модела, само модел истестира са до сад невиђеним подацима, и да се тако израчуната грешка узме у обзир.

одређују апрксимирајућу функцију, она би без вредности β_0 засигурно пролазила кроз координатни почетак. Да би вредност β_0 увек била присутна, додата је вредност X_0 у вектор $\boldsymbol{X}$, и као што је приказано у једначини (3.4), она увек има исту вредност.

2.2. Примена линеарне регресије

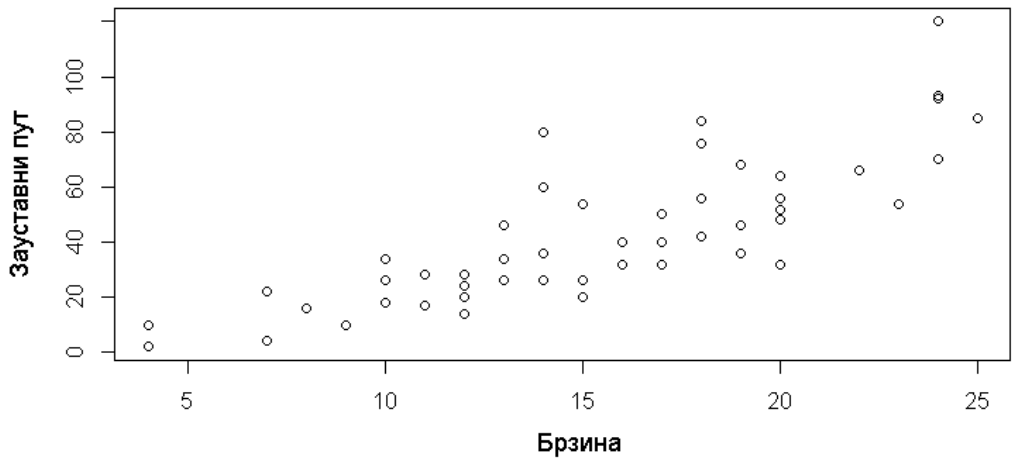
Линеарну регресију из претходног поглавља применићемо на следећем скупу података (табела 1) [22].

Табела 1 Аутомобили – скуп података

Редни број	Брзина (mph)	Зауставни пут (стопе)
1	4	2
2	7	4
3	10	18
4	14	36
⋮	⋮	⋮
49	24	70
50	15	85

Приказани скуп података се састоји од две променљиве, једне независне (брзина аутомобила) и једне зависне (зауставни пут). Укупно постоји 50 обсервација које говоре о зависности брзине аутомобила пре кочења и укупног зауставног пута [20]. Задатак линеарне регресије је да предвиди вредности зауставног пута у односу на брзину аутомобила.

За боље сагледавање скупа података, дат је визуелни приказ на слици 1:



Слика 1 Аутомобили - визуелизавија скупа података

На основу слике 1 јасно је уочљива линеарна зависност између зависне и независне променљиве. Кад се каже да постоји линеарна зависност мисли се на могућност провлачења праве кроз скуп података тако да су јој све тачке веома близу.[2]

Специјалан случај једначине (2.4.) који одговара овом скупу података гласи:

$$y = \beta_0 x_0 + \beta_1 x_1 + \varepsilon\,, \qquad x_0 = 1 \tag{2.7}$$

У једанчини (2.7.), променљива y представља зауставни пут, променљива x_1 брзину, док је вредност β_0 место где функција пресеца x-осу. Наведене променљиве ће имати вредности за које је грешка ε минимална. У конкретном случају, функција линеарне регресије има облик:

$$y = -17.5791 + 3.9324\, x_1 + \varepsilon \tag{2.8}$$

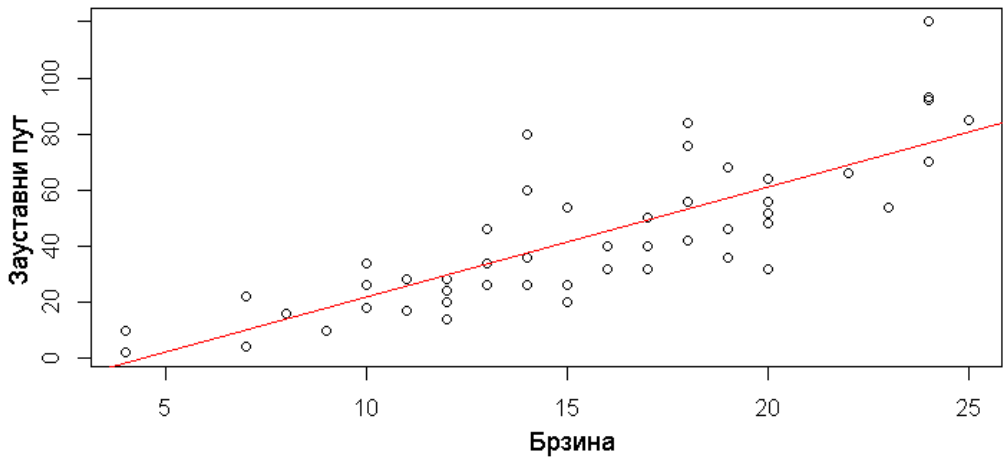
Или на конкретном примеру, процењена дужина зауставног пута за брзину од 10 mph би износила 21.7449 стопа. Поступак је дат у једначинама (2.9) и (2.10)

$$y = -17.5791 + 3.9324 \cdot 10 \tag{2.9}$$

$$y = 21.7449 \tag{2.10}$$

[2]Разлика између измерене зависне вредности и вредности на правој је мала за све обсервације.

Уколико би се права функције (2.8) представила графички, изгледала би као на слици 2.



Слика 2 Аутомобили - визуелизација праве добијене линеарном регресијом

У пракси, неопходно је постојање два скупа података (тренинг и тест скуп) како би се одредила ефикасност или моћ предвиђања модела линеарне регресије. Тренинг скуп служи за креирање модела, док тест скуп података служи за евалуацију тачности предвиђања. Како је ово само пример, скуп описан табелом 1 се користи у оба случаја. Најпре је креиран модел предвиђања (једначина 2.8), а затим је тестиран истим подацима. Том приликом је израчунат корен средње квадратне грешке и резултат је приказан у (2.11):[3]

$$RMSE = 15.06886 \tag{2.11}$$

Корен средње квадратне вредности говори да модел у просеку погреши за око 15 стопа приликом предвиђања дужине заустављног пута за одређену брзину аутомобила.

3. Недостајуће вредности

3.1. Механизми недостајућих вредности

[3] RMSE – (енг. Root Mean Squared Error), корен средње квадратне грешке

Технике машинског учења као што 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 забележених података и недостајућих вредности[8]. Механизми могу бити:

MCAR – Missing Completely At Random – Недостајуће вредности су присутне без икакве законитости. Уколико скуп података има две 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 [11].

MAR – MissingAtRandom – 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 од 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 је њихова оцена са теста ирелевантна и не садржи се у скупу података.

MNAR – MissingNotAtRandom – 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 описан у (6.1. Експериментални подаци) 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 иницијалне вредности (касније ће се те почетне вредности користити за евалуацију технике импутације података).

3.2. Технике импутације 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 колону непопуњену), овом једноставном техником остало би 90 редова (инстанци) као улаз за алгоритам машинског учења.

Нека се одређен скуп података 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 елиминисати 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 [3].

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

3.2.1. Импутација средње вредности

Импутација средње вредности је вероватно најједноставнија метода. Она подразумева замену недостајуће вредности за сваку променљиву (колону) са средњом вредности познатих обсервација у посматраној 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 [26]. 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) унутар једне колоне, логично је да смањујемо корелацију између колона. На пример, скуп података садржи две колоне (висина и тежина), и 5 обсервација. (Табела 2)

Табела 2 Уметање средње вредности

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

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

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

3.2.2. 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 занимљива у истраживањима које 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 пристрасност модела, и такође измењује средњу вредност и варијансу (по променљивој) [13]. Ни ова техника неће бити коришћена у експериментима у овом раду, јер се скуп података за тренинг састоји углавном од нумеричких типова, и притом се пацијенти не посматрају кроз време.

3.2.3. Импутација података коришћењем линеарне регресије

Ова техника захтева одређени ниво корелације између променљивих. Идеја је једноставна; у скупу података са три независне променљиве (X_1, X_2, X_3) неке вредности недостају. Потребно је попунити све вредности у нпр. X_1 користећи вредности из X_2 и X_3 као параметре линеарне регресије. Оно што је компликовано у овом примеру је то што приликом импутације података у X_1, очекивано је да се деси да неке од

вредности из X_2,X_3 такође нису познате. Због тога је потребно направити 3 једначине линеарне регресије како би се адекватно попуниле вредности у колони X_1.

$$X_1 = \beta_0 + \beta_2 X_2 + \beta_3 X_3 \tag{3.1}$$

$$X_1 = \beta_0 + \beta_3 X_3 \tag{3.2}$$

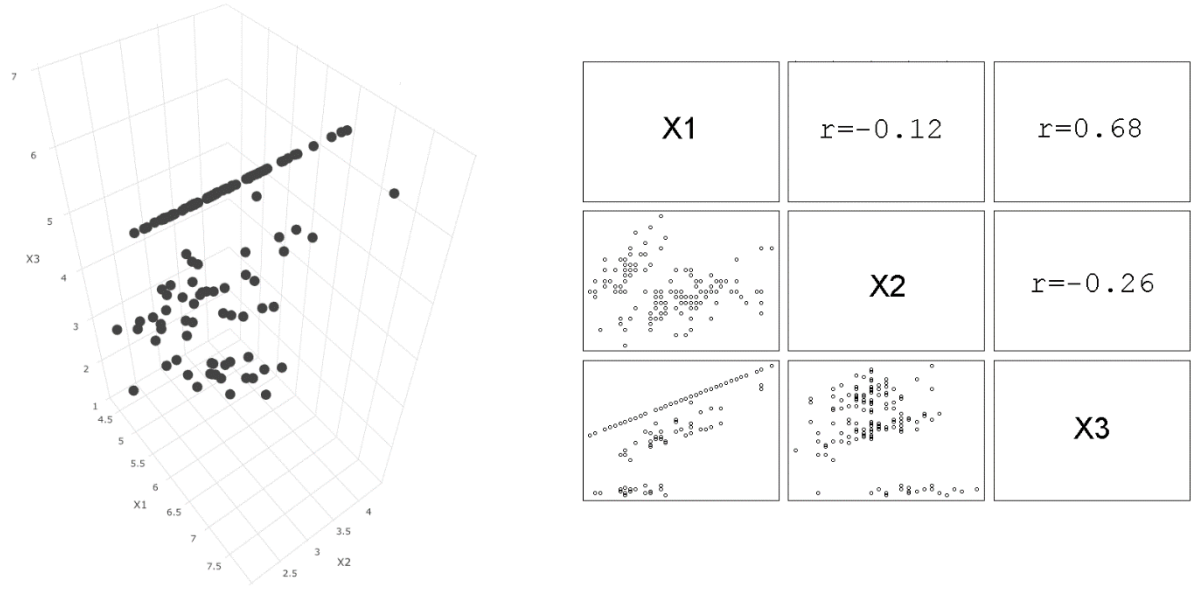
$$X_1 = \beta_0 + \beta_2 X_2 \tag{3.3}$$

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 X_1, такође недостаје и променљива X_2. Другим речима, подаци недостају по MAR механизму. У том случају, за одређивање вредности X_1 користила би се једначина (3.3) и то специјалан облике те једначине:

$$X_1 = X_3 \tag{3.4}$$

Овакав начин импутације значајно би повећао вредност корелације између X_1и X_3. На слици 3 је приказана тако добијена корелација као и просторни приказ једног скупа података.



Слика 3 Корелација након уметања података линеарном регресијом

Импутација 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 вредности као на слици 1. Иако се подаци за експеримент састоје од укупно 11 променљивих ($X_1, \dots, X_{10}, Y$) што може да проузрокује велики број једначина линеарне регресије, ова техника ће се користити у даљем раду.

3.2.4 Импутација података стохастичком регресијом

У претходном примеру су приказани недостаци импутације линеарном регресијом. Као покушај превазилажења тих недостатака, понекад се користи модификована верзија уметања података која се назива и уметање података стохастичком регресијом [2].

Идеја је поприлично интуитивна; у једначину линеарне регресије додати још један параметар (z) који ће на случајан начин да промени резултујућу вредност. Самим тим једначина (3.4) би постала:

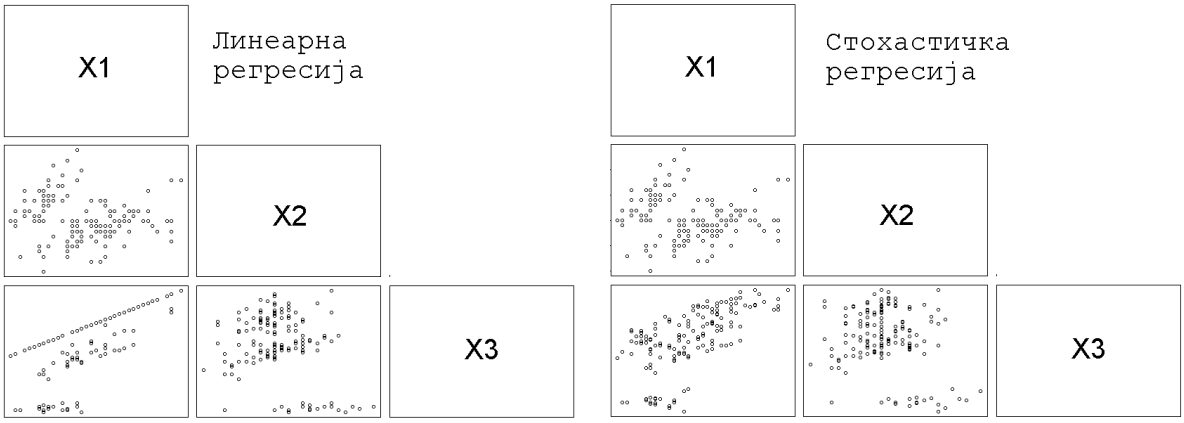
$$X_1 = X_3 + z \tag{3.5}$$

Битно је напоменути да је вредност z другачија за сваку обсервацију (ред) у скупу података. Због те особине није могуће драстично повећати корелацију код попуњеног скупа.

Случајна променљива z може да подлеже било којој расподели. Најчешће се користи нормална расподела са очекиваном вредношћу једнаком нули, и стандардном девијацијом једнаком грешци варијанси приликом регресије [4].

$$z \sim N(0, \sigma^2) \tag{3.6}$$

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 анализа импутације линеарном и стохастичком регресијом

Као и техника импутације линеарном регресијом, и ова техника ће се користити даље у раду.

3.2.5 Импутација коришћењем (шума) стабала одлучивања

Проблеми са којима се свакодневно сусрећемо често нису линеарни, па коришћење линеарне регресије не даје увек најбоље резултате. Технике 3.2.3 и 3.2.4. се могу унапредити уколико линеарни модел заменимо са нелинеарним моделом.

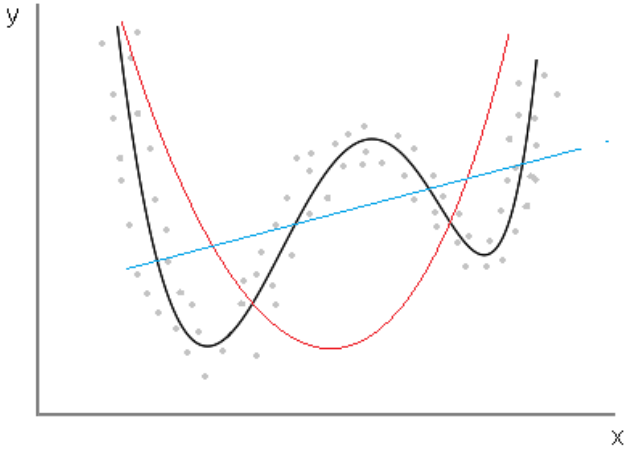
Креирање нелинеарне функције није увек тривијалан задатак. Понекад се зависна променљива једино може описати веома комплексном функцијом независне променљиве. Слика 5 показује упоредну анализу различитих

регресионих функција на датом скупу података. Подаци се састоје од две променљиве (x, y), где је y зависна променљива а x независна. Плавом линијом је представљена линеарна функција (3.7), црвеном линијом квадратна функција (3.9), а црном линијом функција вишег реда (3.9).

$$y = \beta_0 + \beta_1 x \tag{3.7}$$

$$y = \beta_0 + \beta_1 x^2 \tag{3.8}$$

$$y = \beta_0 + \beta_1 x^3 + \beta_2 x^2 + \beta_3 e^x \tag{3.9}$$



Слика 5 Упоредни приказ регресионих функција

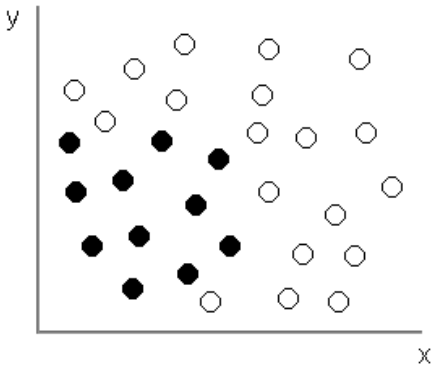
Очигледно је да график функције (3.9) најбоље одговара датом скупу података. Међутим, вероватно је потребно пуно покушаја тренирања са различитим типовима функција да би се добио задовољавајући резултат, односно да регресиона крива постане добар предвиђач.

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.

3.2.5.1 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 хомогеније што је могуће више у односу на променљиву по којој се подаци деле.

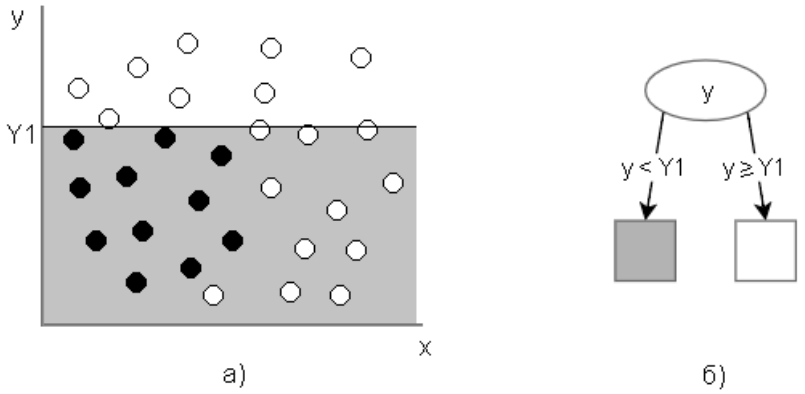
Пример скупа података садржи само три променљиве (x, y, z) где је z зависна (номинална са две класе) променљива, док су x и y две нумеричке независне променљиве. Графички приказ овог примера дат је на слици 6. z променљива је означена црним и белим круговима.



Слика 6 скуп података за тренинг алгоритма стаблом одлучивања

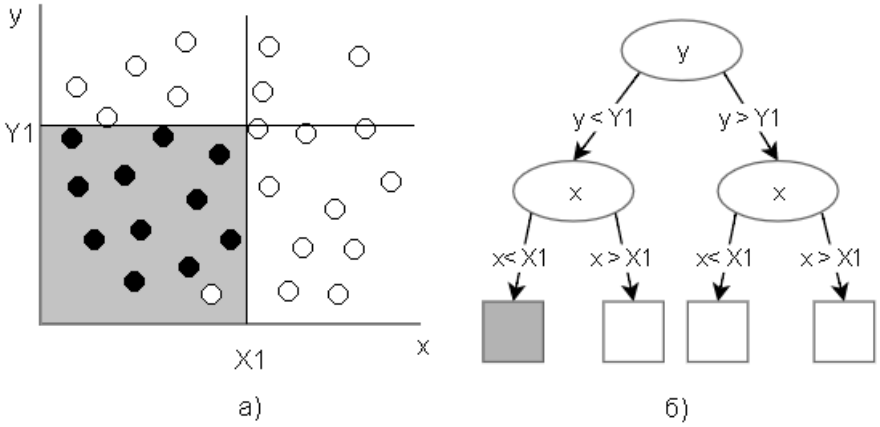
Алгоритам за прављење стабла је једноставан. Најпре се одабере променљива по којој се врши дељење као и вредност која ће поделити ту променљиву.[4] Претпоставка је да је изабрана променљива y и да је вредност Y_1 по којој ће се вршити деоба. У том случају стабло одлучивања и скуп података би изгледао 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 $y < Y_1$ (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 5(б), где осенчени део представља предвиђање црног круга у случајевима када је у ствари присутан бео круг.

[4]Најчешће се користи алгоритам ID3 заједно са алгоритмом за рачунање ентропије C4.5 За више информација[11].



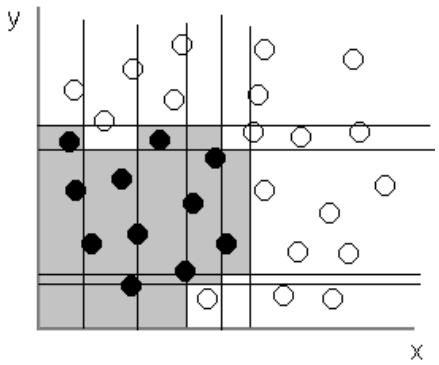
Слика 7 Стабло одлучивања висине 1

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 x са вредношћу X_1. Јасно се види на слици 8 да овакво стабло одлучивања производи знатно мању грешку.



Слика 8 Стабло одлучивања висине 2

Такође, и даље један бео круг припада осенченој области и таква инстанца 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.

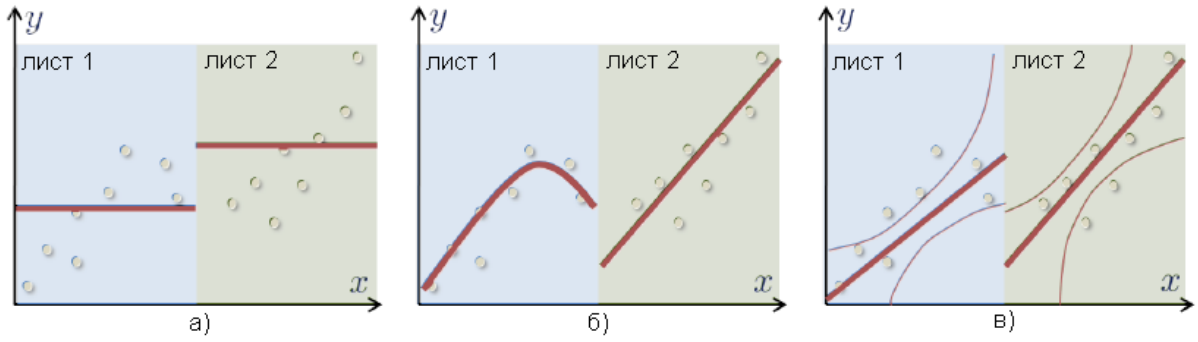


Слика 9 Стабло одлучивања са великом висином

3.2.5.2. Регресионо стабло одлучивања

У претходном поглављу је приказано стабло одлучивања коришћено за класификацију, али такође је могуће 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 $y = f(x)$, где x представља променљиву по којој се тренутно врши деоба, док y представља коначну излазну променљиву. Могуће је дефинисати разне функције $f(x)$ 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 [1].

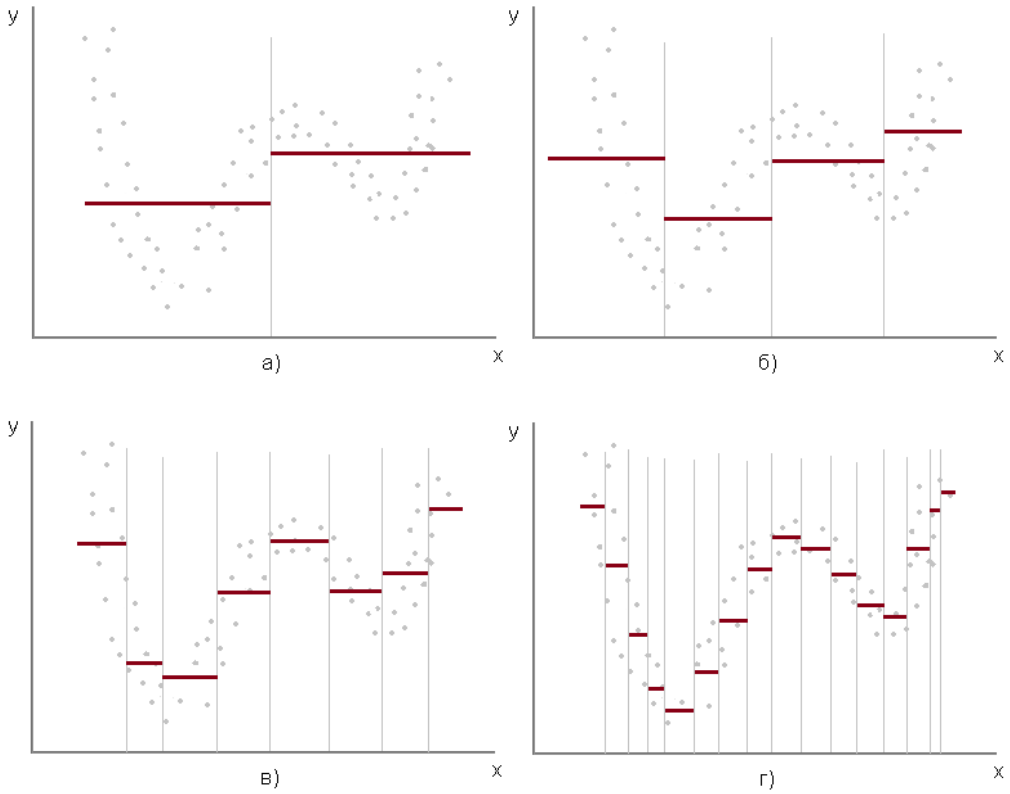


Слика 10 Примери модела предвиђања

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 $y = f(x)$ са скупом података који му одговара. На слици 10а модел предвиђања је константна функција, или другим речима средња вредност одговарајућег подскупа података. Овакав модел предвиђања унутар листа се назива константан модел предвиђања,

и тај термин ће се користити у даљем раду. 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 3.2.5 и приказаним 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 x 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 у ствари значи 4 листа. Слика 11в приказује 8 листова или стабло дубине 3, док последња слика 11г визуелизује стабло дубине 4 (16 листова). Сва стабла у овим примерима су бинарна (један чвор има тачно двоје деце), али постоје и другачија стабла, n-арна, где један чвор може имати n деце.

У примерима изнад је показано како нелинеаран проблем уведен у 3.2.5 може да се решава стаблом одлучивања. И овде важи правило да што је дубље стабло, то је већа вероватноћа да дође до претренираности.

3.2.5.3 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 τ. То значи да кад покренемо алгоритам за прављење стабла n пута, где је један од параметара конструисања случајна променљива τ, добићемо n различитих стабала.

Тако конструисана различита стабла постају део шуме, и касније се шума користи за предвиђање уместо појединачног стабла. Случајна променљива τ уводи смањењу корелацију између стабала унутар шуме, што касније значи повећану генерализацију приликом предвиђања [1].

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

Импутација података коришћењем шуме стабала одлучивања је трећа метода која ће бити коришћена у експерименту. Експериментални скуп овог рада ће садржати 10 променљивих нумеричког типа и једну номиналног типа. Попуњаваће се свака колона посебно, коришћењем осталих колона. На пример, уколико попуњавамо колону X_1, она постаје зависна (предикциона) променљива, а остале вредности се користе као независне. Затим, променљива X_2 постаје зависна и тако даље док се не попуне вредности у свим колонама.

[5] Овакав начин предвиђања вредности y је применљив за константан модел предвиђања који ће бити коришћен у овом раду. Постоје и друге технике одређивања y које узимају у обзир вредности појединачних стабала али оне неће бити даље разматране.

4. Предлог хибридне технике за импутацију

Поред техника импутације наведених у претходном поглављу, у раду ће се подаци попунити и предложеном, хибридном техником. Разлог за називање предложене технике "хибридном" лежи у томе да је она комбинација две до сада познате технике (кластеровање и импутација). Најпре ће се скуп података кластеровати користећи кластеровање к-средњих вредности, а затим ће се једна од дефинисаних техника применити на сваки кластер посебно.

Разлог за овакву припрему је поједностављење проблема приликом импутације. Очекује се да ће подскуп података над којим ће се вршити импутација имати мањи опсег вредности него скуп података пре кластеровања. Додатно, тај подскуп података би требало да се састоји од обсервације сличних једна другој[6], па би сама импутација могла лакше да предвиди недостајуће вредноости.

Такође, варијанса унутар кластера би требало да буде мања, него варијанса целог скупа података. Самим тим, алгоритам импутације би могао брже да конвергира ка решењу, што може да проузрукује већу тачност импутације.

4.1. Кластеровање к-средњих вредности

Основна имплементација алгоритма кластеровања к-средњих вредности не подржава недостајуће вредности у скупу података. Због тога је било неопходно написати код који ће подржати недостајуће вредности. За ту сврху коришћен је програмски језик *Octave*, иначе бесплатна верзија језика веома сличног *Matlabu*. Кластеровање к-средњих вредности ће бити представљена псеудокодом, а део кода ће бити приказан у *Octave*.

```
Улаз:  к (број кластера), X (скуп података)
Излаз: C (скуп кластера)
Метод:
```

[6] Под сличношћу две обсервације x и y подразумева се њихово одстојање $d(x, y)$

```
Случајним избором изабрати к центроида
Креирати празан скуп C на основу са к центроида
Понављај:
  Придружи обсерваније скупа X најближем центроиду скупа C
  Промени положај сваком од к центроида унутар скупа C
Док сваки од к центроида не конвергира
Врати C
```

Листинг 1 Псеудокод к-средњих вредности кластеризације

Пседудокодом из листинга 1, креираће се скуп **C** од **к** кластера који ће садржати инстанце скупа података **X**. Најбитнији део псеудокода је подвучен и садржи рачунање разлике (одстојања) између две тачке у простору.

Једна тачка је дефинисана вектором једне обсервације скупа **X** (један ред унутар скупа података), док је друга дефинисана вектором вредности центроида. Оба вектора имају исту дужину p, па се ова разлика често представља Еуклидским одстојањем:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_p - y_p)^2} \tag{4.1}$$

Две поменуте тачке (вектора), обсервација и центроид су представљене словима x и y респективно. Из једначине (4.1), очигледно је да све вредности оба вектора морају бити присутне како би се израчунала њихова удаљеност. Самим тим, и код овако приказаног кластеровања захтева да све вредности буту присутне.

Уколико би се променила једначина (4.1) тако да може да израчуна одстојање између две тачке (вектора), али са недостајућим вредностима унутар вектора, онда би и сам алгоритам кластеровања подржавао недостајуће вредности скупа **X**.

Листинг 2 садржи псеудокод таквог начина рачунања растојања:

```
Улаз:  x , y (два вектора, x није комплетан)
Излаз: d (одстојање)
Метод:

 ind := indexOfNaN(x)
 val := y(ind)
 x(ind) := val
```

```
ratio := size(y) / (size(y)-size(ind))
d := sqrt(sum((x - y)*ratio)^2)
return d
```

Листинг 2 Псеудокод рачунања растојања између два вектора са недостајућим вредностима

У псеудокоду из листинга 2, улазне параметре представљају два вектора `x` и `y`, где вектор `x` садржи недостајуће вредности. Најпре се те недостајуће вредности попуне вредностима из вектора `y`. То је веома битан корак, јер ће се једино на тај начин разликом између два вектора (`x - y`) креирати нови вектор, који има 0 (нуле) на местима где су биле непостојоће вредности. Затим, за коначно рачунање растојања (дужине) потребно је узети у обзир однос (`ratio`) између броја постојећих и непостојећих вредности. Уколико ниједна вредност вектора `x` и `y` не недостају, `ratio` ће имати вредност 1 (један) и растојање ће бити израчунато претходно описаном Еуклидском једначином.

Међутим, ако постоји бар једна недостајућа вредност у `x`, однос (`ratio`) ће бити већи од 1. У коначној једначини, тај однос има улогу повећања значаја растојања између познатих вредности.

Вредност параметра `ratio` има значај само уколико су вредности свих колона сведени на исту скалу, односно уколико је скуп података нормализован. Може се користити било која техника нормализације и овом приликом ће се све вредности нормализовати тако да одговарају нормалној расподели са очекиваном вредности 0 (нула) и стандардном девијацијом 1 (један).

$$X \sim N(0,1) \tag{4.2}$$

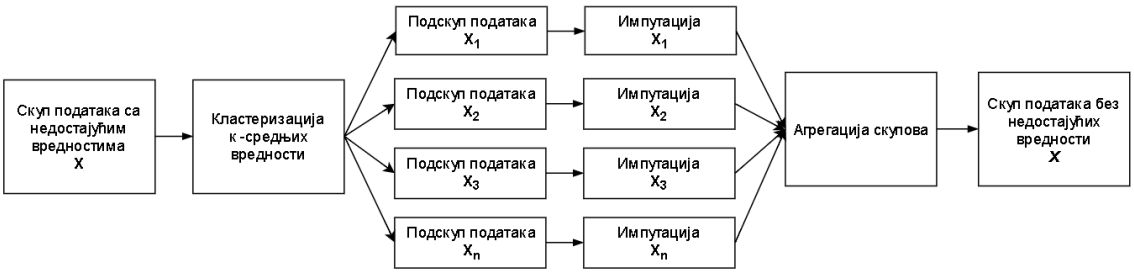
Приказ имплементације псеудокода из листинга 2 у програмском језику *Octave*, је приказан у следећем листингу.

```
1  function d = distance(x, y)
2   size = size(x,2)                % дужина вектора x и y
3   ind = find(isnan(x)) % позиције недостајућих вредности
4   nanSize = sum(isnan(x))  % број недостајућих вредности
5   valSize = sum(!isnan(x))     % број познатих вредности
6   x(ind) = y(ind)          % замена недостајућих познатим
7   d = sqrt(sum(((x-y)*(size/valSize)).^2))
8  end
```

Листинг 3 Имплементација рачунања растојања између два вектора са недостајућим вредностима – Octave

4.2. Импутација на нивоу кластера

Пре него што се прикаже псеудокод предложене методе, неопходно је сагледати цео процес. Приказ процеса се налази на слици 12. Почетни скуп који садржи недостајуће вредности се кластеризује методом к-средњих вредности и добија се n подскупова. Затим се врши импутација на сваки од n подскупова где се они посматрају као комплетан скуп. Након импутације, n попуњених скупова се агрегира у један велики попуњен скуп. На тај начин, иницијални скуп не садржи више недостајуће вредности.



Слика 12 Процес предложене методе за импутацију података

За имплементацију процеса описаног сликом 12 користиће се два програмска језика *Octave* и *R*. Због комплексности писања такве имплементације приказан је само псеудокод, али на веома детаљан начин (листинг 4).

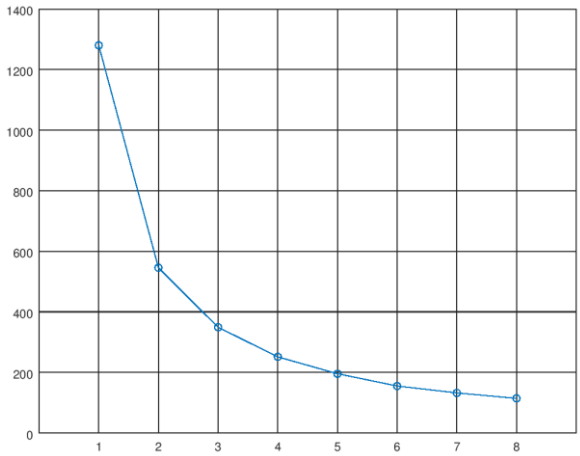
```
Улаз:  X  (скуп података са недостајућим вредностима)
Излаз: X' (попуњени скуп података)

Метод:

 Xnorm := norm(X)              //нормализација вредности
 k := elbow(Xnorm)                          //лакат метода
 clusterIndex := kmeans(Xnorm,k)//индекс кластера (1,к)
 C :{X1,X2,...,Xk} := apply(X, clusterIndex) //кластери
 C':{}  //празан скуп кластера са попуњеним вредностима
 for i = 1 to k:           //за сваки кластер у скупу C
   Xi' := impute(Xi)                          //импутација
   C'(i) := Xi'      //додај резултат импутације у скуп
 endfor
 X' := aggreagate(C':{ X1', X2',..., Xk'}) //агрегација
```

Листинг 4 Псеудо код предложене методе

Сви кораци су до сада описани осим лакат методе. Приликом кластеровања к-средњих вредности није познато који је оптималан број кластера, односно није позната вредност к. Због тога се као корак пред коначно кластеровање извршава кластеровање са различитим вредностима к. Том приликом се за сваку вредност к рачуна просечно одстојање инстанци од центроида унутар кластера. Када се тако израчунато просечно одстојање визуелизује, добије се график сличан графику са слике 13.



Слика 13 Лакат метода

X-оса показује број кластера (односно број к), а у-оса ниво грешке. Визуелно се тражи преломна тачка плаве линије, и она представља оптималан број кластера (број к). У датом примеру то је број 3.

Након одређивања броја кластера, могуће је извршити и само кластеровање. Као резултат добија се низ индекса припадности сваком кластеру. Такав низ се затим користи да се почетни скуп заиста подели на **к** кластера, и затим се врши импутација по кластеру. На крају се агрегишу тако попуњени кластери и добија се попуњен скуп података.

5. 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 3, 4 и 5.

Табела 3 Потпуни (почетни) скуп података

X_1	X_2	X_3	Y
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 3 садржи све вредности и те вредности су референтне вредности за даљу анализу. Табела 4 садржи скуп података без 10% вредности што је добијено вештачким путем.[7] Три колоне (X_1, X_2, X_3) са по седам редова садрже укупно 21 вредност, и укупно су насумично обрисане 2 вредности.

Табела 4 Непопуњен скуп података

X_1	X_2	X_3	Y
4	2	1	7
3		2	4
7	5	6	7
8	5	1	3
4	7		4

[7]Начин брисања одређеног процента вредности је детаљно описан у 6.2. (Конструисање тренинг скупа)

6	3	2	8
2	4	4	1

Табела 5 садржи податке након импутације, и она заједно са табелом 3 представља основ за даљу анализу грешака. Очигледно је да су попуњене две вредности, уместо иницијалне вредности 5 (ред 2, колона X_2), уметнута је вредност 3. Такође, попуњена је вредност 9 (колона 5, колона X_3) уместо почетне вредности 8.

Табела 5 Уметнути подаци

X_1	X_2	X_3	Y
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

Дакле, табела 3 представља оригинални скуп података, табела 4 скуп података са недостајућим вредностима, а табела 5 попуњени скуп података. Ова три термина ће се користити у даљем раду и зато је битно дефинисати их овде.

Грешке које ће се користити за даљу анализу могу се грубо сврстати у две групе: 1) Грешке настале разликом између оригиналног и попуњеног скупа, и 2) Грешке настале методом предвиђања над попуњеним скупом. Следећа два поглавља садрже описе обе групе грешака.

5.1. Грешке настале разликом између оригиналног и попуњеног скупа

Грешке које припадају овој групи посматрају вредности у оригиналном скупу и попуњеном скупу (након импутације). Разлика између тих

вредности представља грешку импутације. Следе описи таквих грешака који ће се корисити у даљој анализи.

5.1.1. 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]

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{5.1}$$

У формули (5.1), x_i означава почетну вредност, $\hat{x}_i$ уметнуту вредност док n означава број уметнутих вредности. У примеру из табела 3,4 и 5, формула (5.1) би изгледала:

$$MSE = \frac{1}{2}((5-3)^2 + (8-9)^2) \tag{5.2}$$

$$MSE = \frac{1}{2}(4+1) \tag{5.3}$$

$$MSE = 2.5 \tag{5.4}$$

У датом примеру средња квадратна грешка износи 2.5.

5.1.2. Корен средње квадратне грешке

У одељку 5.1.1. је уведена средња квадратна грешка. Разлика између уметнуте и почетне вредности се квадрира како би се изгубила важност

[8] MSE – (Mean Squared Error), средња квадратна грешка

знака.[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]

$$RMSE = \sqrt{MSE} \tag{5.5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x_i})} \tag{5.6}$$

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

$$RMSE = \sqrt{2.5} \ , \ RMSE = 1.58 \tag{5.7}$$

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

5.1.3. Просечна релативна грешка

Вредности апсолутне грешке (као и средње квадратне и корена средње квадратне грешке) нису увек најбољи показатељи. На пример, апсолутна грешка вредности $\tilde{x} = 1$ је само 1% од процењене вредности $x = 100$, али чак 50% од процењене вредности $x = 2$ [28]. где у првом случају распон могућих вредности износи 1-100 (нумерички 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 X_1 и X_2 номиналног типа (могуће вредности су 1 и 2) док су колоне Y_1 и Y_2 нумеричког типа (вредности имају распон од 1 до 100).

[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 ($\frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x_i}|$).

[10] RMSE – (Root Mean Squared Error), корен средње квадратне грешке

Табела 6 Вредности пре и након импутације

X_1	X_2	Y_1	Y_2
1	2	45	44
2	2	67	67
2	1	31	32

Колоне са индексом 2 (друга и четврта колона) су попуњене вредности док су колоне са индексом 1 (прва и трећа колона) иницијалне, почетне 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 попуњена вредност за један мања од иницијалне.

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

$$ARE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{|x_i - \hat{x_i}|}{\|x_i\|}} \tag{5.8}$$

$$\|x_i\| = \max(x_i) - \min(x_i) \tag{5.9}$$

Користећи формулу (5.8) и податке из табеле 6, израчуната је просечна релативна грешка. Такође, над истом табелом израчунат је корен средње квадратне грешке и подаци су дати у табели 7.

Табела 7 Поређење средње квадратне грешке и просечне релативне грешке

$RMSE_1$	ARE_1	$RMSE_2$	ARE_2
0.67	0.67	0.67	0.013

Из табеле 7 се јасно види колико бољи показатељ може бити просечна релативна грешка као грешка импутације. Просечна релативна грешка је

[11] ARE – (Average Relative Error), средња релативна грешка

много већа у случају номиналног типа података, што је и очекивано јер је опсег вредности те променљиве веома мали.

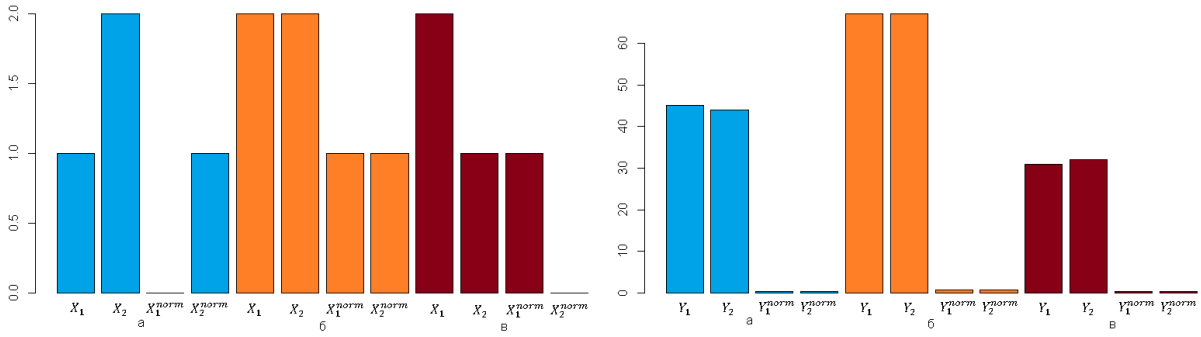
Како би се још боље показао значај погрешно унетих вредности из табеле 7, урађена је нормализација свих вредности једначином (5.10) и резултати су приказани у табели 8.

$$X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{5.10}$$

Табела 8 Упоредни приказ апсолутних и нормализованих вредности

	X_1	X_2	X_1^{norm}	X_2^{norm}	Y_1	Y_2	Y_1^{norm}	Y_2^{norm}
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

Формула (5.10) нормализује све вредности на скуп вредности [0,1]. Очигледно је да су нормализоване вредности номиналног типа (леви део табеле 10) у ствари екстремне вредности посматране скале. Дакле, уколико алгоритам импутације погреши за један (апсолутна мера), по релативној или нормализованој 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 13, где су такође на посебним графицима одвојени различити типови променљивих (лево се налазе графици за номинални, а десно за нумерички тип). Боје на слици 12 одговарају редовима из табеле 8.



Слика 14 Упоредни приказ апсолутних и нормализованих врендости

Трећи и четврти стубић унутар сваке боје говоре о разлици између оригиналне и попуњене вредности. Очигледна је разлика на дијаграмима који представљају номинални тип података, и занемарљива код нумеричког типа.

5.2. Грешке настале методом предвиђања над попуњеним скупом

Четврта грешка којом ће се мерити ефикасност импутације података је корен средње квадратне грешке линеарне регресије. Ова мера ефикасности спада у другу групу и детаљно је описана у следећем одељку.

5.2.1. Корен средње квадратне грешке линеарне регресије

Након што се импутација изврши и добије попуњени скуп података, тај скуп је могуће користити за даље предвиђање. Метода којом ће се предвиђати је линеарна регресија, тј. креираће се модел предвиђања где ће улазни скуп података бити попуњен скуп. Затим ће се одредити тачност таквог предвиђања, односно израчунаће се корен средње квадратне грешке. Таква мера уједно представља и тачност импутације података. Због тога ће се и корен средње квадратне грешке линеарне регресије разматрати приликом мерења ефикасности импутације.

6. Експеримент

У овом одељку ће најпре бити описани подаци који ће се користити у експериментима, а затим начин креирања скупова података за тренинг. Након тога, извршена је импутација података над креираним скуповима и урађена претходно описана анализа грешке. На крају поглавља дат је преглед ефикасности свих метода импутације коришћених у експерименту.

6.1. Експериментални подаци

6.1.1. Опис скупа података

Сви експерименти у раду ће се ослањати на овај скуп података. Подаци се односе на пацијенте који болују од дијабетеса. У табели 9 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).

Табела 9 Тренинг подскуп. Поред демографских података (старост и пол), на пацијентима је вршено 8 мерења који заједно утичу на ниво дијабетеса годину дана касније

	Старост	Пол	ИТМ[12]	КП[13]	Резултати серумских мерења						резултат
Пацијент	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	Y
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

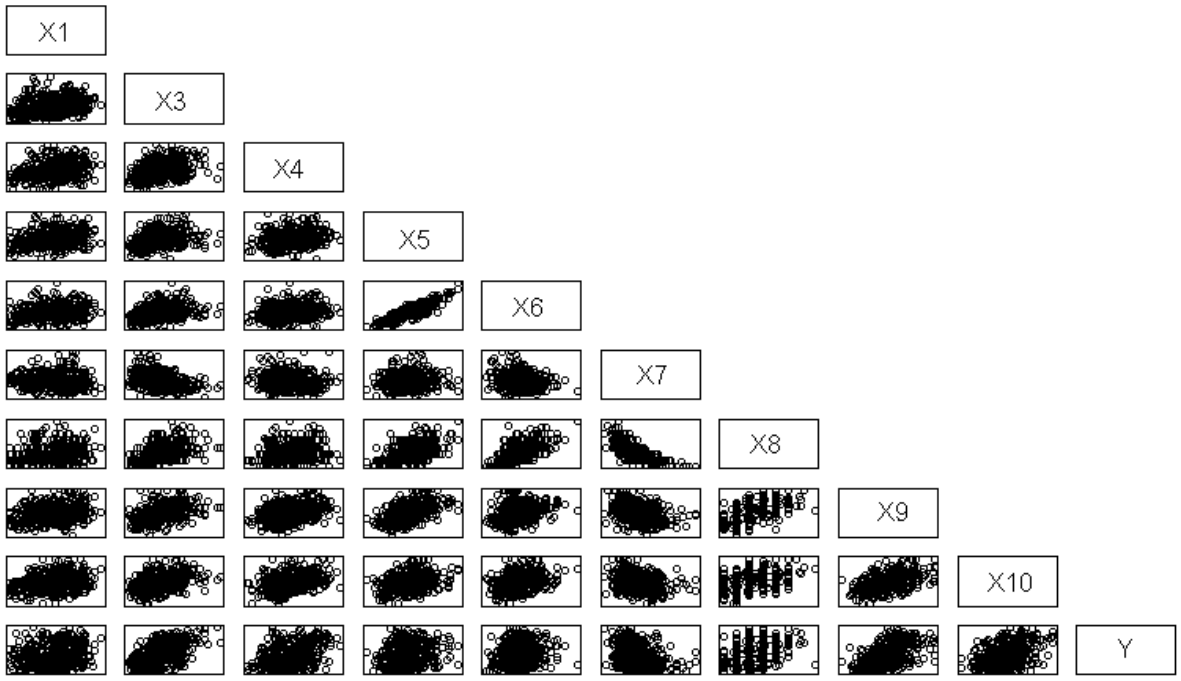
Скуп се састоји од 10 атрибута (варијабли) од који су 8 нумеричког типа ($X_1, X_3 - X_{10}$) и једна номиналног карактера (X_2). Резултат Y представља ниво дијабетеса код људи годину дана након мерења [7]. Такође, резултујућа колона је нумеричког карактера.

6.1.2. Корелациона матрица

[12] ИТМ, индекс телесне масе
[13] КП, крвни притисак

Укључујући резултатску променљиву (колону), скуп података се састоји 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 се испостави да такве зависности (лиенарне) постоје, у даљем раду ће линеарна регресија бити један од заступљенијих алгоритама машинског учења.

Свакако, пре даље анализе, одличан показатељ било које анализе може да донесе визуелизација самог скупа података (Слика 14).



Слика 15 Визуелни приказ скупа података

Са слике 14 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 X_2 није део графичког приказа јер номинални тип података те променљиве није идеалан за визуелизацију.

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

$$m = \frac{n(n-1)}{2} \tag{6.1}$$

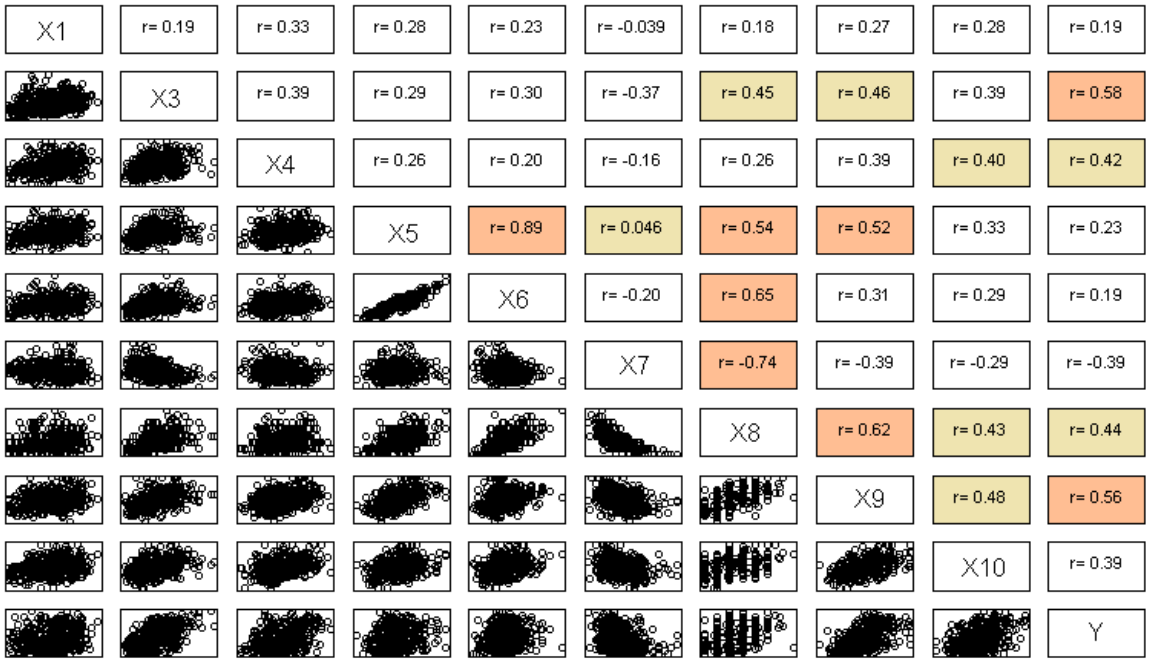
У једначини (6.1), m представља број различитих вредности, док је n укупан број променљивих у скупу података. У овом случају $n = 10$, јер колона пол није узета у обзир.

Корелација између две променљиве (x, y) се рачуна по формули:

$$r = \frac{Cov(x, y)}{s_x s_y} \tag{6.2}$$

$$Cov(x, y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n - 1} \tag{6.3}$$

У једначинама изнад, r представља вредност корелације за променљиве x и y. Додатно коваријанса, $Cov(x, y)$, чија вредност не говори ништа о степену зависности између x и y, се користи приликом рачунања корелације. Са друге стране корелација има вредности у опсегу $[-1,1]$, где екстремне вредности означавају јаку корелацију (јаку зависност), а вредности блиске нули значе да таква зависност уопште не постоји.



Слика 16 Визуелни приказ скупа података са корелационим вредностима

На слици 15 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. Овакав

графички приказ заједно са вредностима корелационе матрице показују да је скуп података подобан за регресиону анализу.

6.2. Конструисање тренинг скупа

Подаци описани у 6.1. 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 је у листингу 4 [10].

```
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  }
```

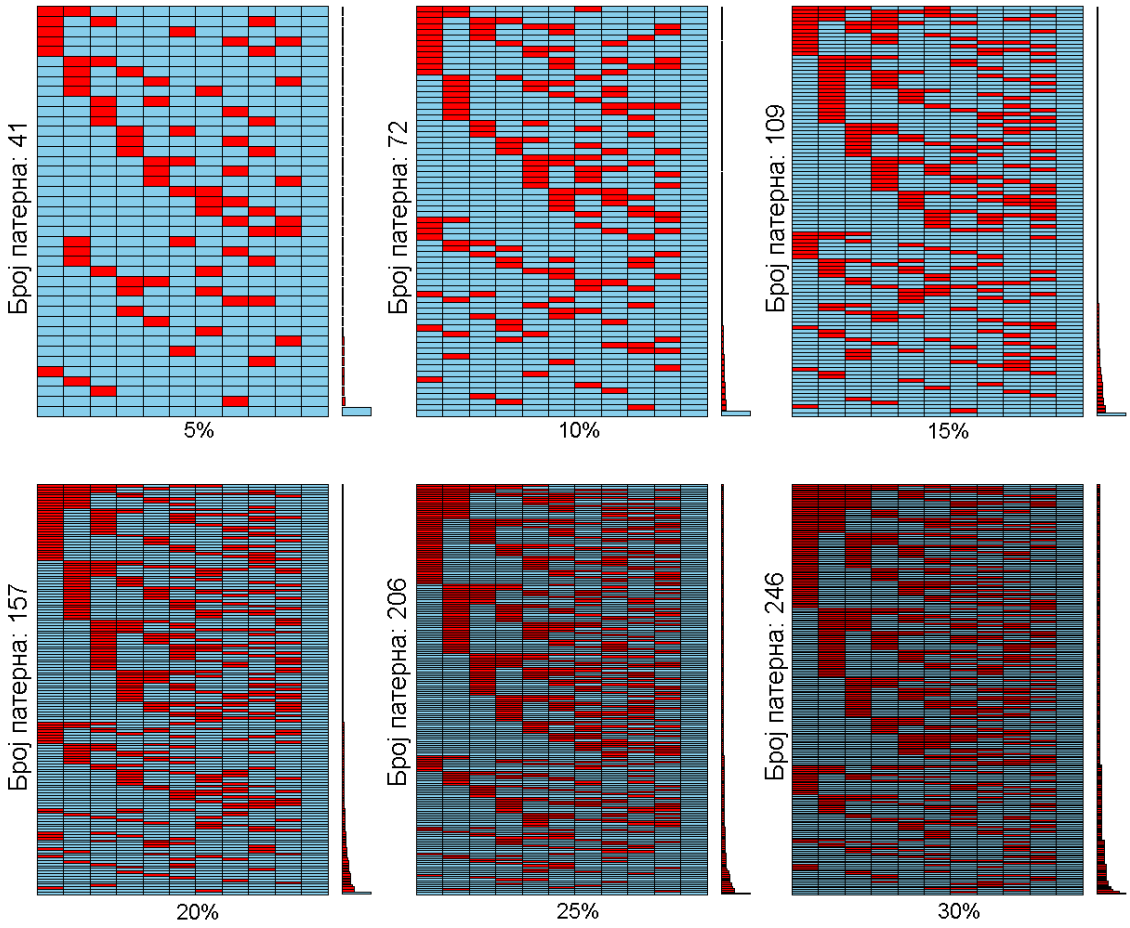
Листинг 5 функција у програмском језику R за генерисање скупа података са недостајућим врендностима

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 3.2.3., 3.2.4., 3.2.5. Скрипта написана у језику *R*, којим се од комплетног скупа прави скуп података са недостајућим вредностима се налази у листингу 5.

```
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 6 скрипта за брисање 5% података

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



Слика 17 Упоредни приказ 6 генерисаних скупова

Слика 16 садржи упоредни приказ свих 6 скупова коришћењем функције из листинга 4[14]. Битно је напоменути да сваки од 6 скупова има комплетну последњу колону (ниједна вредност не недостаје). Та колона представља зависну променљиву y, и њој не могу недостајати подаци али ће се она свакако користити при импутацији. Такође број комбинација (образаца, патерна) се повећава заједно са процентом недостајућих вредности.

Очекивано највише патерна по којима недостају вредности се налази код скупа података коме недостаје 30% података, чак 246 патерна. Другим речима, за импутацију регресионим методама биће потребно направити 246 функција.

[14]За генерисање скупова коришћена је *VIM* библиотека пакета *R* [17].

6.3. Импутација података

6.3.1. Импутација линеарном регресијом

У одељку 3.2.3. је теоријски описан метод импутације линеарном регресијом. Тако описана импутација је имплементирана у софтверском пакету *R*, тачније, имплементација је садржана у библиотеци *mice* [25]. Управо је ова библиотека коришћена за импутацију непостојећих вредности.

У листингу 6 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.

```
# Улазни параметри:
# XY_05 – скуп података
# method="norm.predict" – означава методу импутације,
#                         тј. линеарну регресију
# m=5 – број функција креираних за сваку променљиву
#
# Повратна вредност:
# XY_05_model који садржи параметре линеарне функције,
# оригинални скуп података, метаподатке...
# Не садржи уметнуте вредности.
XY_05_model <- mice(XY_05, method="norm.predict", m=5)

# Креирање скупа података где су недостајуће вредности
# замењене уметнутим вредностима
XY_05_imp <- complete(XY_05_model)

# Визуелизација резултата (слика 14)
densityplot(XY_05_model)
```

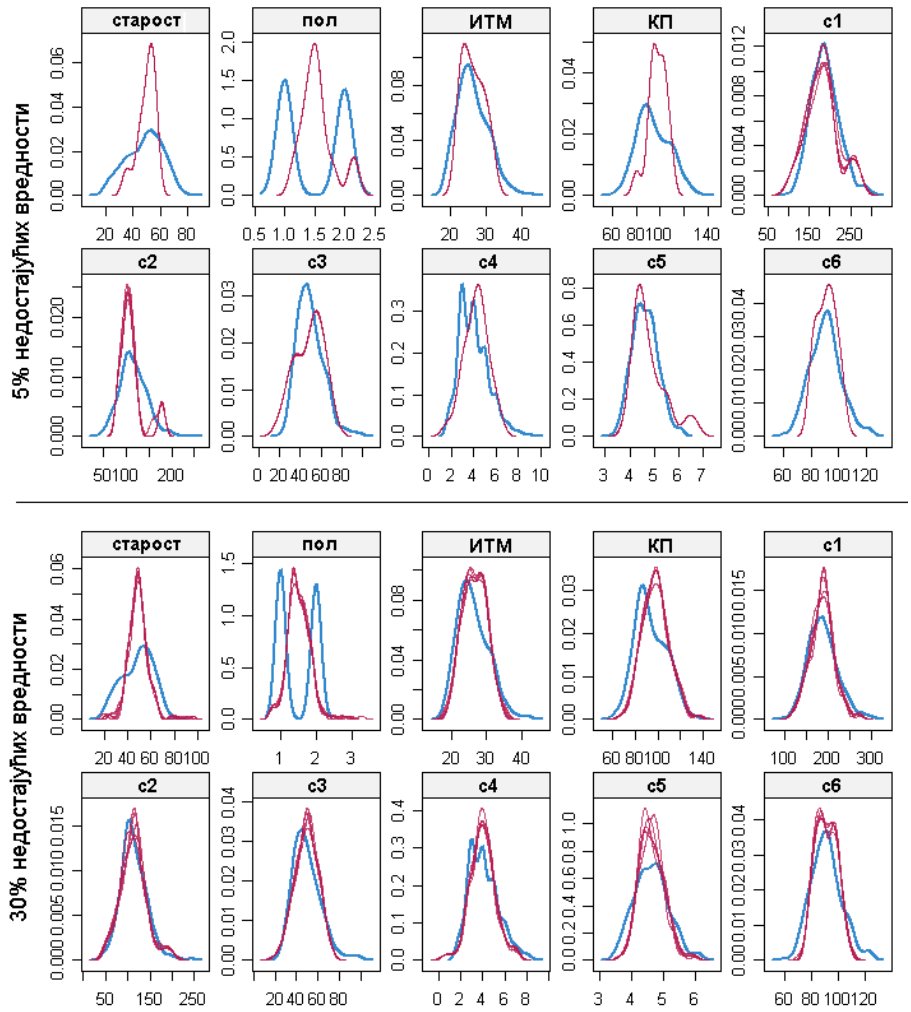
Листинг 7 Импутација линеарном регресијом

Позивањем функције `densityplot` из листинга 6, добија се скуп од 10 дијаграма приказаних на слици 17. Иако скуп података садржи и 11-ту променљиву y, 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 означене расподеле уметнутих вредности. Један дијаграм садржи углавном једну плаву и више црвених линија (највише 5), јер се за једну променљиву конструисало толико фунцкија. Преклапање црвене и плаве линије не значи да је импутација била успешна, али непреклапање сигурно значи да је импутација произвела велику грешку. За право испитивање успешности импутације, неопходно је израчунати грешке описане у одељку 5.1 и 5.2.

Занимљиво је обратити пажњу на дијаграме који одговарају 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 18 Расподеле појављивања измерених и уметнутих вредности

У табели 10 се налази приказ грешака описаних у 5.1 и 5.2.

Табела 10 Приказ грешака насталих приликом импутације линеарном регресијом

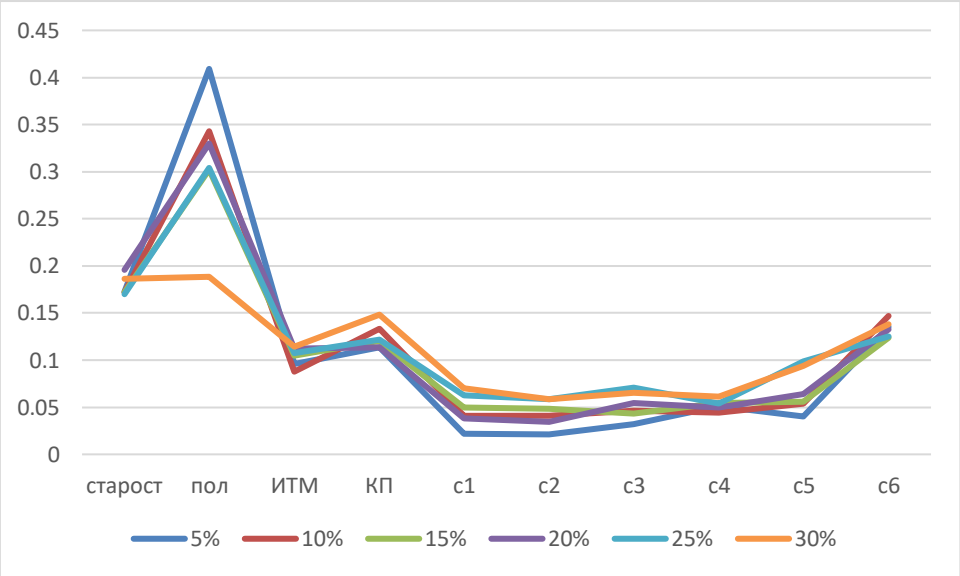
Недостајуће вредности (%)	Средња квадратна грешка	Корен средње квадратне грешке	Просечна релативна грешка	Корен средње квадратне грешке линеарне регресије
5	43.54469	6.59884	0.109273	40.93458
10	98.69796	9.934685	0.111054	40.78019
15	114.4606	10.69863	0.107427	41.76785
20	75.41079	8.683939	0.112437	41.84145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

Поред укупних грешака, у табели 11 су приказане и просечне релативне грешке по променљивој, где се види значај лоше процене линеарне регресије на номиналну променљиву (променљива пол).

Табела 11 Просечна релативна грешка за сваку пеомењљиву- импутација линеарном регресијоом

	Проценат недостајућих вредности у скупу података					
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

Ради лакшег разумевања, табела 11 је графички представљена на слици 18. 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 19 Графички приказ релативне грешке за сваку променљиву - импутација линеарном регресијом

6.3.2. Импутација стохастичком регресијом

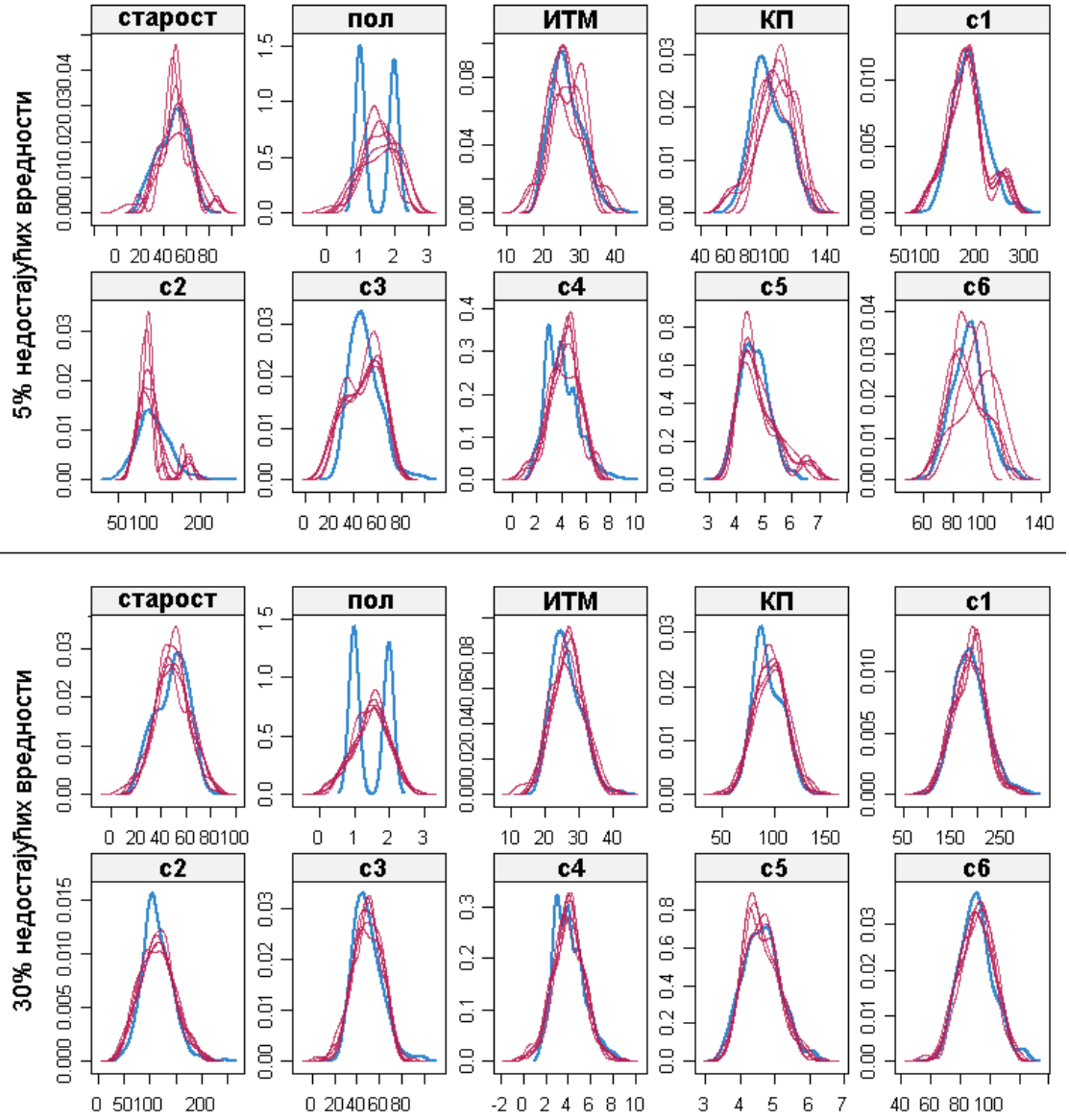
Слично претходном одељку, и импутација стохастичком линеарном регресијом је описана у 3.2.4. И за ову врсту импутације коришћена је имплементација из библиотеке *mice* пакета *R*. Код који се користи за

импутацију је идентичан коду из листинга 6, осим што се уместо методе `"norm.predict"` користи метода `"norm.nob"`. Листинг 7 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 8 пример кода импутације за стохастичку линеарну фрегресију

Као и у претходном одељку, и овде је приказана расподела забележених и уметнутих вредности. Опет плава линија представља расподелу забележених, а црвене линије расподелу уметнутих вредности (слика 19).



Слика 20 Расподела забележених и уметнутих вредности - стохастичка линеарна регресија

У табели 12 су приказане грешке описане у одељку 5.2.

Табела 12 Приказ грешака насталаих импутацијом стохастичком линеарном регресијом

Недостајуће вредности (%)	Средња квадратна грешка	Корен средње квадратне грешке	Просечна релативна грешка	Корен средње квадратне грешке линеарне регресије
5	66.61793	8.161981	0.121119	41.39864
10	142.22068	11.925631	0.129445	41.87477
15	154.98505	12.449299	0.120585	42.27468
20	138.32744	11.761268	0.12555	41.43752
25	194.61844	13.950571	0.123229	40.67999
30	176.85746	13.298777	0.1293	43.11288

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

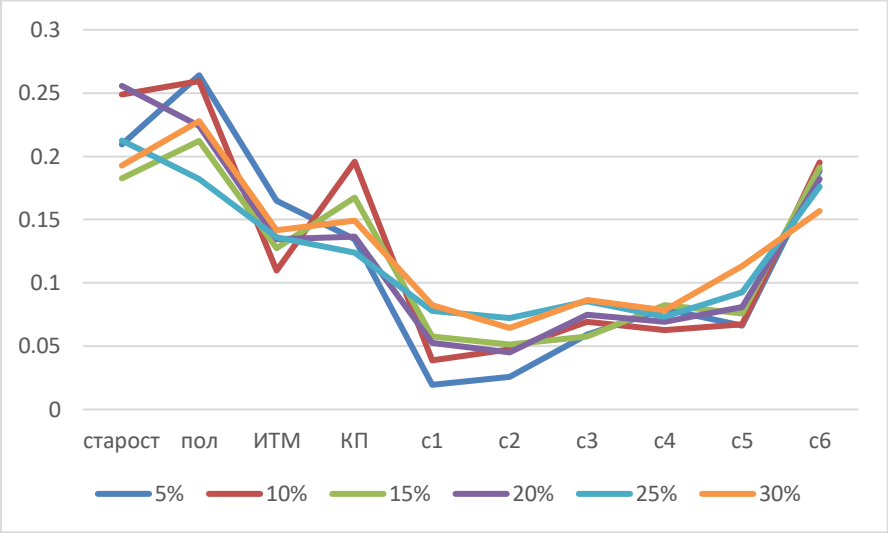
Додатно, табела 13 садржи просечне релативне грешке за сваку променљиву посебно, док су приказане вредности визуелизоване на слици 20.

Табела 13 Просечна релативна грешка за сваку пеоменљиву- импутација стохастичком линеарном регресијоом

	Проценат недостајућих вредности у скупу података					
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	0.1100	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

Опсег вредности просечне релативне грешке по променљивама је мањи код стохастичке линеарне регресије него код линеарне регресије из 6.3. Међутим, уколико се упореде слике 17 и 19, уочљиво је да је расподела релативне грешке за све променљиве приближно једнака. У оба случаја највећа грешка је код променљиве **пол**, док је најмања код променљивих

c1 и **c2** (серумска мерења). Разлог је очигледан; те две променљиве одговарају X_5и X_6, које иницијално имају корелацију $r = 0.89$. (слика 2).



Слика 21 Графички приказ релативне грешке за сваку променљиву - импутација стохастичком линеарном регресијом

6.3.3 Импутација (шумом) стабала одлучивања

Метода описана у 3.2.5. је знатно другачија у односу на претходне две коришћене у експерименту. За импутацију шумом стабала одлучивања ће се уместо библиотеке *mice* користити библиотеке *missForest*, такође као део пакета *R*.

Пример кода којим се врши импутација је дат у листингу 8.

```
1>   # Креирање шуме стабала
2>   model_05 <- missForest(XY_05)
3>
4>   # Приказ уметнутих података
5>   # Налазе се у атрибуту модела: model_05$ximp
6>   view(model_05$ximp)
```

Листинг 9 Импутација шумом стабала одлучивања

Најпре се позивањем функције `missForest` креирају стабла одлучивања која се користе да се попуни скуп података означем као атрибут `ximp` резултујуће променљиве `model_05`.

Тако добијени, резултујући скуп података се користи у даљој анализи грешке. Добијене вредности се налазе у табели 14:

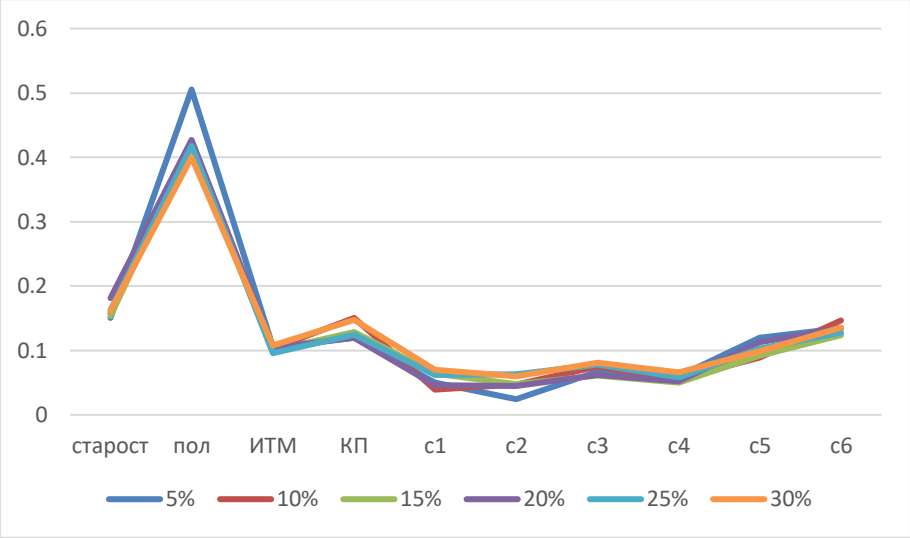
Табела 14 Приказ грешака насталих импутацијом шуме стабала одлучивања

Недостајуће вредности (%)	Средња квадратна грешка	Корен средње квадратне грешке	Просечна релативна грешка	Корен средње квадратне грешке линеарне регресије
5	57.52409	7.584464	0.1336885	41.10916
10	77.39612	8.797506	0.128823	41.47292
15	101.6051	10.07994	0.1239766	42.2715
20	78.26333	8.846656	0.1283581	42.00837
25	112.7208	10.61701	0.1292231	41.17139
30	114.6042	10.70534	0.1328116	43.75881

Такође, као и у претходним примерима, и овде је присутна анализа просечне релативне грешке за сваку променљиву. (табела 15, слика 21)

Табела 15 Просечна релативна грешка за сваку променљиву - импутација шумом стабала одлучивања

	Проценат недостајућих вредности у скупу података					
Променљива	5%	10%	15%	20%	25%	30%
старост	0.1502	0.1579	0.1532	0.1816	0.1624	0.1615
пол	0.5053	0.4247	0.4201	0.4270	0.4177	0.4000
ИТМ	0.1060	0.1026	0.1000	0.1035	0.0961	0.1076
КП	0.1198	0.1508	0.1285	0.1194	0.1250	0.1472
с1	0.0498	0.0391	0.0635	0.0455	0.0616	0.0696
с2	0.0243	0.0470	0.0476	0.0447	0.0625	0.0598
с3	0.0671	0.0749	0.0606	0.0615	0.0780	0.0813
с4	0.0595	0.0561	0.0504	0.0523	0.0582	0.0658
с5	0.1197	0.0886	0.0914	0.1128	0.1032	0.0992
с6	0.1346	0.1461	0.1241	0.1348	0.1271	0.1356



Слика 22 Графички приказ просечне релативне грешке за сваку променљиву - импутација шумом стабала одлучивања

Лако је уочљиво да су разлике у грешкама занемарљиве за различите проценте недостајућих вредности. На пример, променљива **c4** је на скоро истом нивоу погрешно процењена за свих 6 скупова података.

Описана карактериситка може бити веома значајна, јер је приликом екперимента могуће предвидети ниво грешења саме импутације. На пример, скуп података садржи 1000 обсервација са недостајућим вредностима и из њега је могуће издовојити 100 обсервација које имају потпуно комплетну посматрану колону. Затим се из те колоне обрише нпр. 10% вредности и изврши се импутација шумом стабала одлучивања. На крају, добијена релативна грешка за посматрани подскуп може бити примењена и на цео скуп података (који иницијално садржи недостајуће вредности па оваква анализа није могућа).

Такође, висок ниво грешке приликом процене променљиве **пол** је другачији у односу на претходне две методе. Као што је описано у 6.1., **пол** је представљен вредностима 1 и 2. Регресионе методе су недостајуће вредности често замењивале вредностима различитим од очекиваних (нпр. у опсегу од 0 до 3). То овде није случај, јер *missForest* библиотека прави разлику између номиналног и нумеричког типа., међутим, веома је често погрешна вредност предвиђена (уместо 1, предвиђена је вредност 2 и обрнуто), па је зато ниво просечне релативне грешке висок.

6.4. Анализа резултата импутације

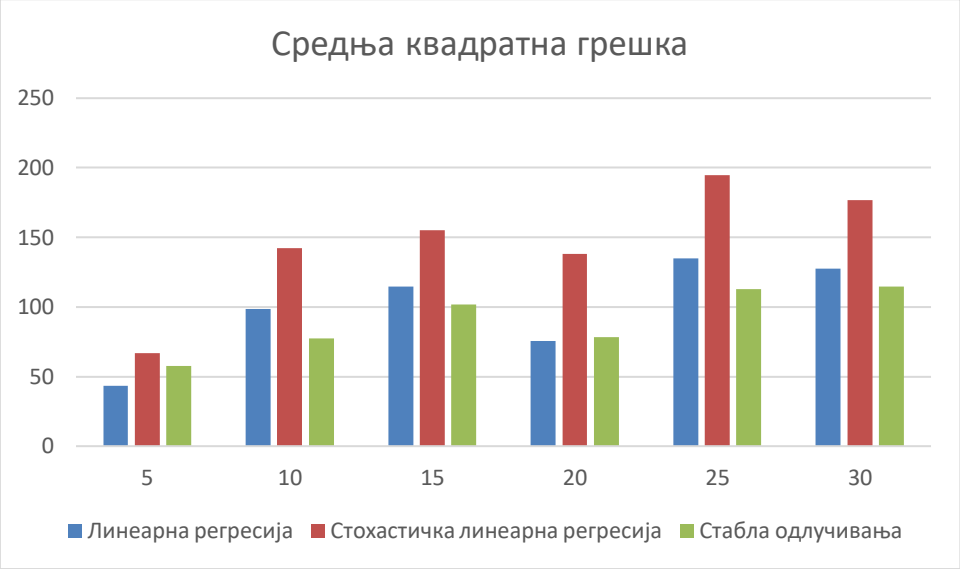
У овом делу ће се упоредно анализирати резултати све три претходно изведене импутације. За сваку врсту грешке ће најпре бити тад табеларан приказ, а затим и график који визуелизује посматрану грешку.

6.4.1. Средња квадратна грешка

Табела 16 Поређење средње квадратне грешке

Проценат недостајућих вредности (%)	Линеарна регресија	Стохастичка линеарна регресија	Стабло одлучивања
5	43.545	66.618	57.524
10	98.698	142.22	77.396
15	114.46	154.99	101.61
20	75.411	138.33	78.263
25	135.01	194.62	112.72
30	127.65	176.86	114.6

У табели 16 и слици 22 се јасно види да је средња квадратна грешка приликом импутације стохасичке линеарне регресије највећа. По овом параметру, најбоља импутација је извршена шумом стабала одлучивања.

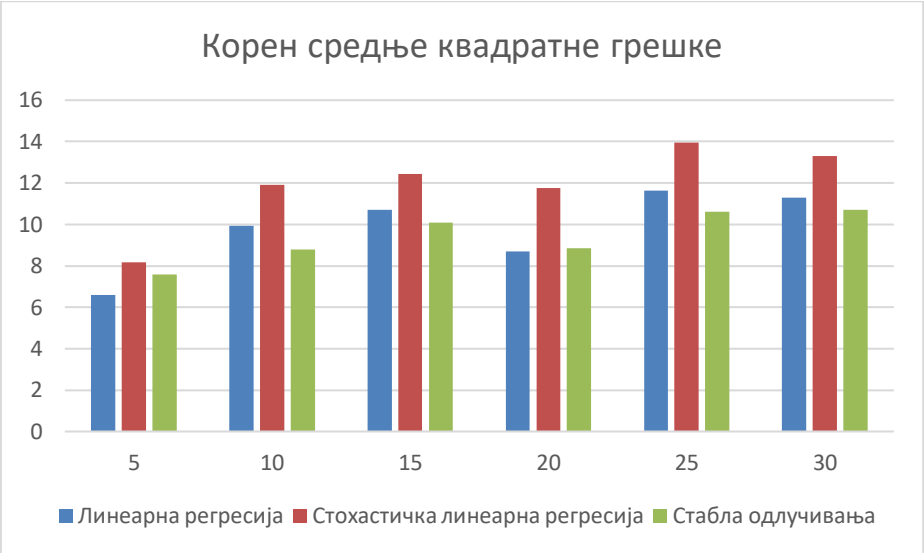


Слика 23 Поређење средње квадратне грешке

6.4.2. Корен средње квадратне грешке

Табела 17 Поређење корена средње квадратне грешке

Проценат недостајућих вредности (%)	Линеарна регресија	Стохастичка линеарна регресија	Стабло одлучивања
5	6.5988	8.162	7.5845
10	9.9347	11.926	8.7975
15	10.699	12.449	10.08
20	8.6839	11.761	8.8467
25	11.62	13.951	10.617
30	11.298	13.299	10.705



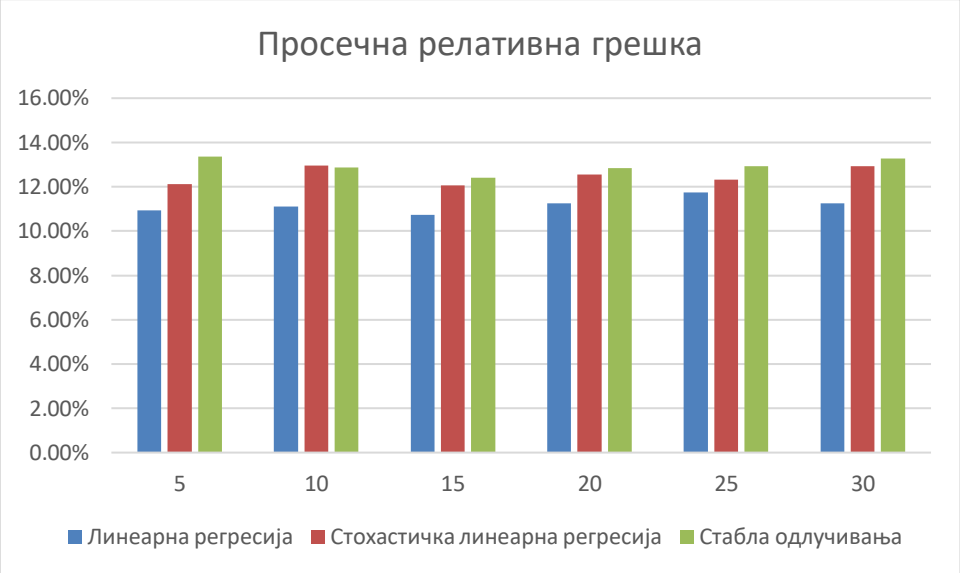
Слика 24 Поређење корена средње квадратне грешке

Исто као и код средње квадратне грешке, њен корен на слици 23 и табели 17 показује да шума стабала одлучивања показује најбоље резултате приликом импутације.

6.4.3. Просечна релативна грешка

Табела 18 Поређење просечне речативне грешке

Проценат недостајућих вредности (%)	Линеарна регресија	Стохастичка линеарна регресија	Стабла одлучивања
5	10.93%	12.11%	13.37%
10	11.11%	12.94%	12.88%
15	10.74%	12.06%	12.40%
20	11.24%	12.56%	12.84%
25	11.73%	12.32%	12.92%
30	11.24%	12.93%	13.28%



Слика 25 Поређење просечне речативне грешке

Просечна релативна грешка је најизраженија код шуме стабала одлучивања. (табела 18, слика 24). Међутим, из претходне анализе (слика 21) се види да је релативна грешка веома изражена за променљиву **пол**. Приликом импутације стаблима одлучивања, грешка за поменуту променљиву је много већа него коришћењем осталих метода. Ипак, и са том екстремном вредношћу, разлика између стубаца ове три методе је веома мала (у просеку 1%).

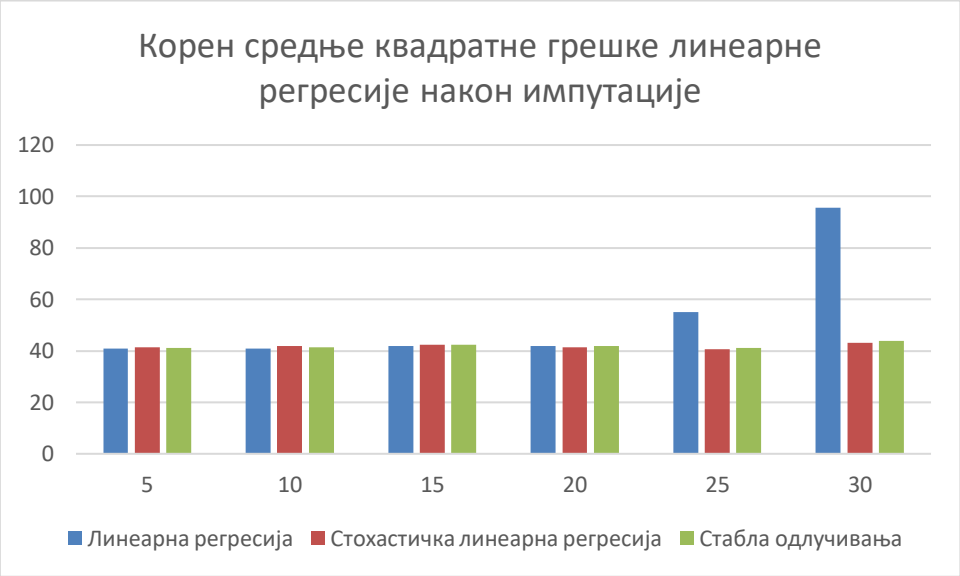
6.4.4. Корен средње квадратне грешке линеарне регресије након импутације

Након импутација, над скуповима података је извршена линеарна регресија и израчунати су корени средње квадратне грешке предвиђања линеарнорм регресијом. Очигледно је да квалитет импутцаије има утицаја на моћ предвиђања, па су упоредни резултати ове мере дати у табели 19 и слици 25.

Табела 19 Поређење корена средње квадратне грешке линеарне регресије након импутације

Проценат недостајућих вредности (%)	Линеарна регресија	Стохастичка линеарна регресија	Стабла одлучивања
5	40.935	41.399	41.109

10	40.78	41.875	41.473
15	41.768	42.275	42.272
20	41.841	41.438	42.008
25	54.972	40.68	41.171
30	95.639	43.113	43.759



Слика 26 Поређење корена средње квадратне грешке линеарне регресије након импутације

Јасно се види да импутација линеарном регресијом над скуповима којима недостаје велики проценат вредности даје веома лоше резултате. Такође, остале две методе дају приближно исте резултате за све скупове података.

6.4.5. Закључак анализе резултата импутације

Шест различитих скупова података са недостајућим подацима су попуњени различитим методама импутације. Свака од метода импутације је показала најбоље резултате по неком од параметара, и због тога није једноставно рангирати поменуте методе по тачности (ефикасноти) саме импутације.

Линеарна регресија се показала као веома лоша метода импутације над скуповима у којима је велики број недостајућих вредности, стохастичка линеарна регресија је имала највећу средњу квадратну грешку (као и њен корен), док је шума стабала одлучивања имала најгоре резултате за просечну релативну грешку, иначе веома доброг показатеља квалитета

импутације. Због значајности тог показатеља, шума стабала одлучивања није изабрана као најбоља метода

1) Стохастичка линеарна регресија
2) Шума стабала одлучивања
3) Линеарна регресија

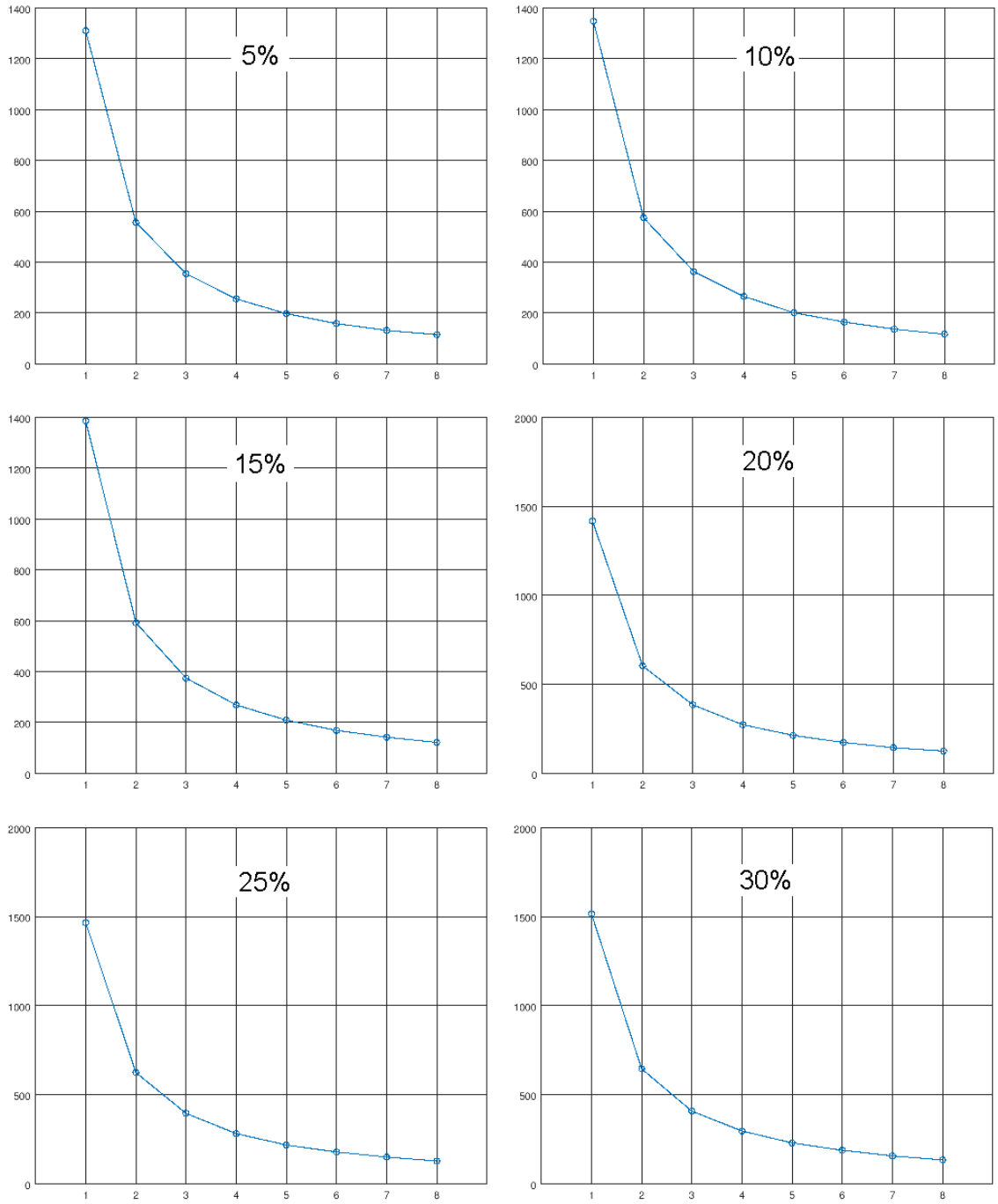
Стохастичка линеарна регресија, као најквалитетнија метода од наведених, ће бити коришћена у даљем раду, као део предложене хибридне методе импутације података.

7. Импутација предложеном хибридном методом

У одељку 4. је теоријски описана предложена метода импутације. У овом делу ће се описана метода применити на експерименталне скупове. Најпре ће се извршити кластеровање скупова података, а затим и метода импутације шумом стабала одлучивања на сваки кластер посебно. На крају ће, такође, бити израчунати сви параметри ефикасности описани у одељцима 5.1 и 5.2.

7.1. Кластеровање

Као корак пред кластеровање потребно је одредити колико кластера ће бити тражено. У ту сврху на слици 27 су приказани резултати лакат методе за свих 6 експерименталних скупова.



Слика 27 Лакат метода над скупом података са недостајућим вредностима

У сваком од 6 случајева, оптималан број кластера је 3. Тако да ће то бити вредност параметра **к** у даљој анализи.

Након кластеровања, сваки од 6 скупова је подељен у 3 групе. Приказ броја обсервација у сваком кластеру је дат у табели 20.

Табела 20 Број обсервација у сваком кластеру

Проценат недостајућих	Редни број кластера		
	I	**II**	**III**

вредности (%)

5	143	154	103
10	195	100	105
15	119	138	143
20	162	94	144
25	144	89	167
30	186	73	141

Из табеле 20 се не могу извући никакви закључци везани за особине обсервација унутар кластера. Она служи само као показатељ да ли су обсервације подједнако заступљене у сваком кластеру. Потенцијално проблематичан скуп података је скуп коме недостаје 30% вредности. Два су могућа разлога за присутност само 73 обсервације унутар једног кластера; дошло је до проблема приликом иницијализације центроида или вредности недостају по таквој расподели да боља кластеризација није могућа.

7.2. Импутација стохастичком линеарном регресијом

Затим је над сваким кластером извршена импутација података стохастичком линеарном регресијом. *R* код за такву импутацију је приказан у листингу 10.

```
1   # XY_C1     представља први кластер
2   # C1_model модел који садржи попуњене скупове података
3   #           за кластер XY_C1
4   C1_model <- mice(XY_C1, method = "norm.nob", m=5)
5   C2_model <- mice(XY_C2, method = "norm.nob", m=5)
6   C3_model <- mice(XY_C3, method = "norm.nob", m=5)

8   # XY_C1_imp попуњен скуп вредности унутар клатера XY_C1
9   XY_C1_imp <- complete(C1_model)
10  XY_C2_imp <- complete(C2_model)
11  XY_C3_imp <- complete(C3_model)
12
13  # агрегација кластера у коначан попуњен скуп XY_imp
14  XY_imp <- rbind(XY_C1_imp, XY_C2_imp, XY_C3_imp)
```

Листинг 10 Импутација сваког кластера појединачно - R код

Добијен је нови скуп `XY_imp` истих димензија (исти број колона и редова) као и почетни скуп са непостојећим вредностима. На овај начин је импутација предложеном методом комплетна.

7.3. Анализа резултата

Предложена метода се састоји од кластеровања к-средњих вредности и импутације стохастичком линеарном регресијом. Због тога ће поред приказа грешака описаних у 5.1 и 5.2 бити приказана и упоредна анализа импутације предложене методе и посебно импутације стохастичком регресијом.

Табела 21 Анализа грешака импутације предложеном методом

Недостајуће вредности (%)	Средња квадратна грешка	Корен средње квадратне грешке	Просечна релативна грешка	Корен средње квадратне грешке линеарне регресије
5	88.96058	9.431892	0.063785	41.43741
10	108.9428	10.437568	0.070309	41.48645
15	170.1132	13.042745	0.086297	43.1856
20	115.5754	10.750598	0.061253	42.95661
25	173.4557	13.170257	0.075435	41.81272
30	177.5384	13.324353	0.065077	42.89006

Из табеле 21 одмах је уочљива ниска вредност просечних релативних грешака. То је знак да предложена метода предвиђа вредности веома близу очекиваним.

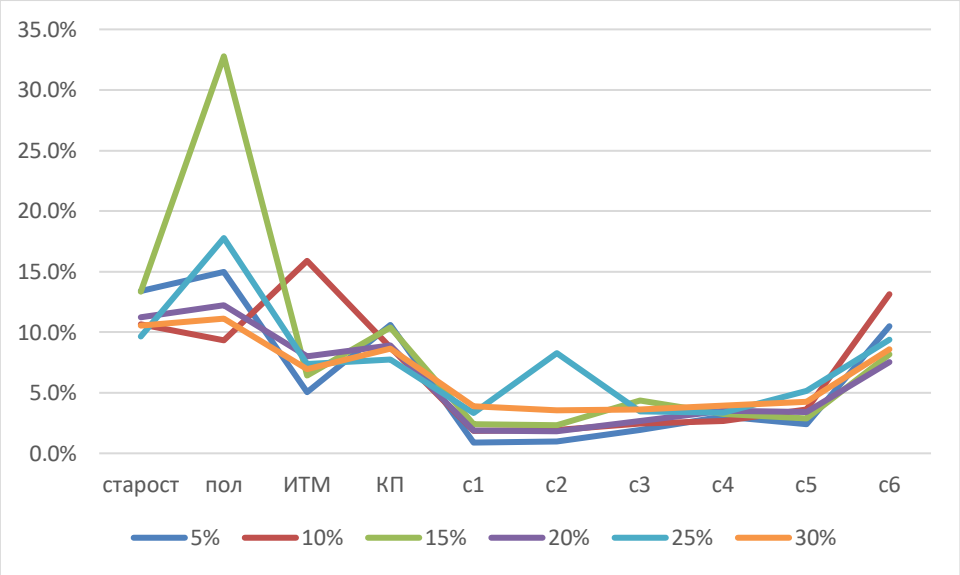
Табела 22 детаљније приказује просечну релативну грешку, по свакој променљивој. Уочљиво је да је вредност грешке веома мала за одређене атрибуте нумеричког типа. Такође, номинална променљива пол тренутно нема високу грешку осим у скупу података коме недостаје 15% вредности.

Табела 22 Просечна релативна грешка импутације предложеном методом

	Проценат недостајућих вредности у скупу података					
Променљива	5%	10%	15%	20%	25%	30%
старост	13.4%	10.6%	13.3%	11.2%	9.7%	10.5%
пол	15.0%	9.3%	32.8%	12.2%	17.8%	11.1%

ИТМ	5.1%	15.9%	6.4%	8.0%	7.4%	6.9%
КП	10.6%	8.8%	10.4%	8.9%	7.7%	8.6%
с1	0.9%	1.9%	2.4%	1.9%	3.3%	3.9%
с2	1.0%	2.0%	2.3%	1.8%	8.3%	3.5%
с3	1.9%	2.5%	4.4%	2.7%	3.4%	3.6%
с4	3.1%	2.7%	3.2%	3.5%	3.3%	4.0%
с5	2.4%	3.6%	2.9%	3.4%	5.1%	4.3%
с6	10.5%	13.1%	8.2%	7.5%	9.4%	8.6%

Слика 28 представља графички приказ табеле 22. Ту се још јасније види моћ предвиђања недостајућих вредности предложеном методом.



Слика 28 Просечна релативна грешка импутације предложеном методом

7.4. Упоредна анализа

У овом делу ће бити представљена упоредна анализа грешака насталих импутацијом предложене методе и стохастичке регресије, методе која је имала најбоље резултате од свих метода који су коришћени на овом скупу података.

Резултати се упоређују са два аспекта. Најпре ће се у поглављима 7.4.1. и 7.4.2. упоредити очекиване и попуњене вредности, и приказаће се тако добијене грешке. Затим ће се у поглављу 7.4.3. упоредити грешке настале услед предвиђања над попуњеним скупом поменутим методама.

7.4.1. Средња квадратна грешка

У овом поглављу ће се приказати средња квадратна грешка и корен средње квадратне грешке израчунате на основу очекиваних и убачених вредности.

Најпре су добијени резултати дати у табелама 23 и 24, а затим су на сликама 29 и 30 приказани односи те две грешке између предложене методе и стохастичке линеарне регресије.

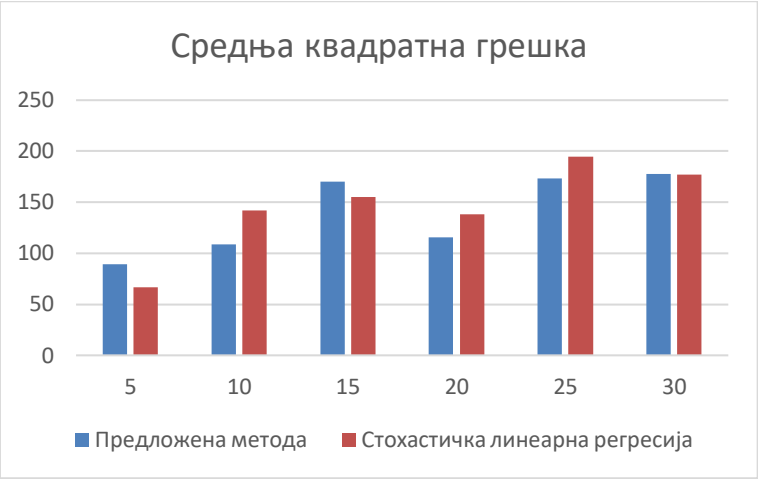
Табела 23 Средња квадратна грешка свих метода коришћених у раду

Проценат недостајућих вредности (%)	Линеарна регресија	Стохастичка линеарна регресија	Стабла одлучивања	Предложена метода
5	43.545	66.618	57.524	88.961
10	98.698	142.22	77.396	108.94
15	114.46	154.99	101.61	170.11
20	75.411	138.33	78.263	115.58
25	135.01	194.62	112.72	173.46
30	127.65	176.86	114.6	177.54

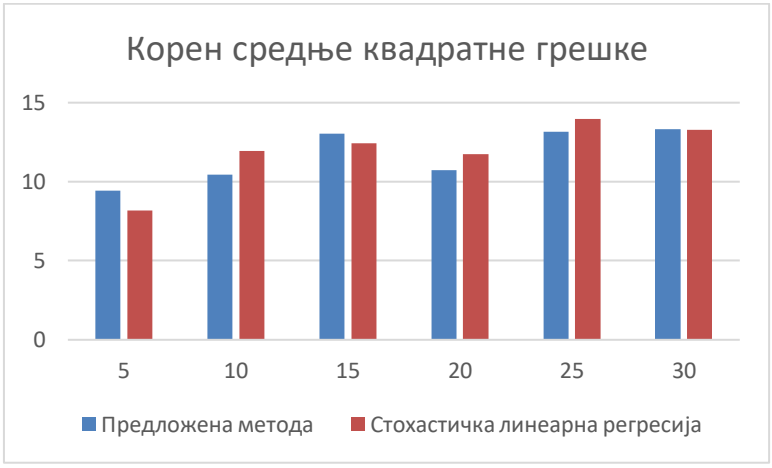
Табела 24 Корен средње квадратне грешке свих метода коришћених у раду

Проценат недостајућих вредности (%)	Линеарна регресија	Стохастичка линеарна регресија	Стабла одлучивања	Предложена метода
5	6.5988	8.162	7.5845	9.4319
10	9.9347	11.926	8.7975	10.438
15	10.699	12.449	10.08	13.043
20	8.6839	11.761	8.8467	10.751
25	11.62	13.951	10.617	13.17
30	11.298	13.299	10.705	13.324

На основу табела 23 и 24 се види да предложена метода у неким ситуацијама остварује боље резултате, а у неким ситуацијама остварује лошије у односу на стохастичку линеарну регресију. То се још јасније може видети на следећим сликама где су приказане вредности грешака искључиво две наведене методе.



Слика 29 Средња квадратна грешка (предложена метода и стохастичка регресија)



Слика 30 Корен средње квадратне грешка (предложена метода и стохастичка регресија)

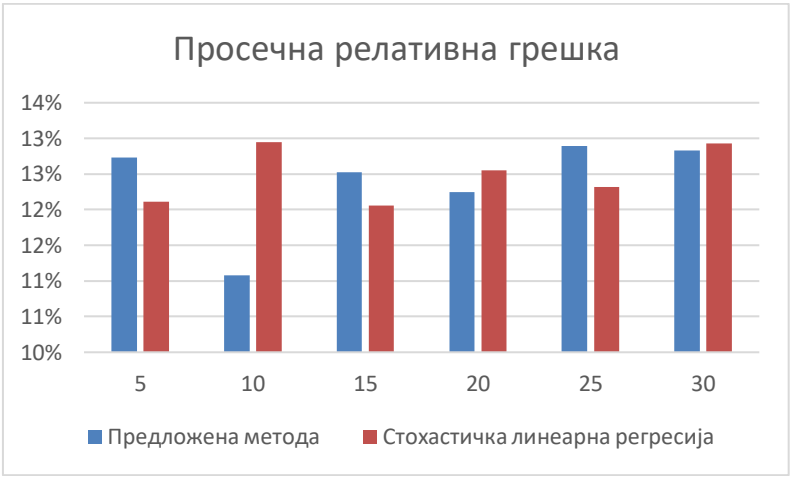
7.4.2. Просечна релативна грешка

У табели 25 су приказани резултати просечних релативних грешака између сваке од извршених метода.

Табела 25 Просечна релативна грешка свих метода коришћених у раду

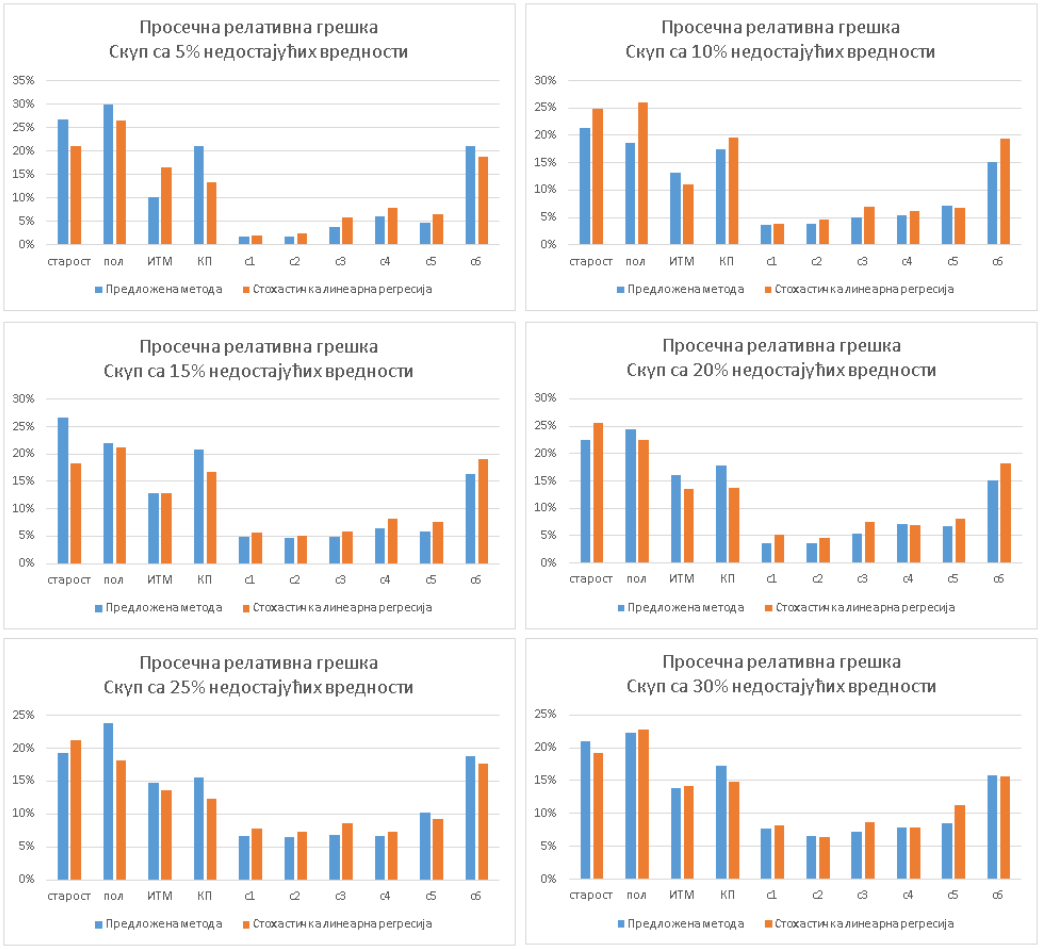
Проценат недостајућих вредности (%)	Линеарна регресија (%)	Стохастичка линеарна регресија (%)	Стабла одлучивања (%)	Предложена метода (%)
5	10.9	12.1	13.4	12.7
10	11.1	12.9	12.9	11.1
15	10.7	12.1	12.4	12.5
20	11.2	12.6	12.8	12.3
25	11.7	12.3	12.9	12.9
30	11.2	12.9	13.3	12.8

Слично као и са средњом квадратном грешком, ни на основу ове мере се не може закључити која је метода боља. Однос грешака између предложене методе и стохастичке линеарне регресије је приказа на слици 31 .



Слика 31 Просечна релативна грешка (предложена метода и стохастичка регресија)

Такође, ову врсту грешке је могуће детаљније аналзирати што је урађено на слици 32 где је представљено 6 графика, сваки за одређен проценат недостајућих вредности.



Слика 32 Просечна релативна грешка по свакој променљивој (предложена метода и стохастичка регресија)

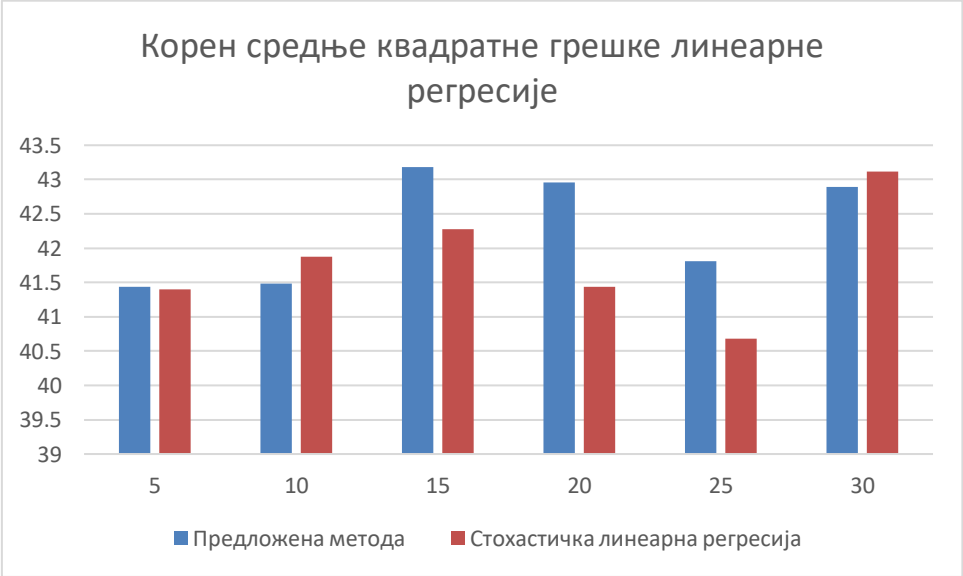
7.4.3. Корен средње квадратне грешке линеарне регресије након импутације

Последња мера којом ће се упоредити предложена метода и стохастичка линеарна регресија је корен средње квадратне грешке настале након предвиђања линеарном регресијом. Дакле, скупови података након импутације (са све четири методе) су представљали скупове података за тренинг линеарном регресијом. Тако креирана четири модела линеарне регресије су тестирана, и израчунат је корен средње квадратне грешке за сваку од њих. Резултати се налазе у табели 26.

Табела 26 Корен средње квадратне грешке линеарне регресије наком импутације свим методама коришћених у раду

Проценат недостајућих вредности (%)	Линеарна регресија	Стохастичка линеарна регресија	Стабла одлучивања	Предложена метода
5	40.93	41.40	41.11	41.44
10	40.78	41.87	41.47	41.49
15	41.77	42.27	42.27	43.19
20	41.84	41.44	42.01	42.96
25	54.97	40.68	41.17	41.81
30	95.64	43.11	43.76	42.89

Такође, слика 33 предтавља упоредну анализу резултата добијених након стохастичке линеарне регресије и предложене методе.



Слика 33 Корен средње квадратне грешке линеарне регресије (предложена метода и стохастичка регресија)

На основу слике 33 види се да је грешка након предвиђања мало мања након импутације стохастичком регресијом. Треба бити опрезан са овом мером, јер приликом импутације се користи и зависна променљива y и самим тим се и повећава корелација између независних и зависне променљиве. Свакако је добро што је ниво грешке на приближно истом нивоу за све посматране скупове. То говори да највероватније није дошло до претренираности алгоритма, односно да подаци нису конвергирали једној вредности.

7.5. Закључак резултата анализе

На основу резултата упоредне анализе из 7.4. може се закључити да је припрема података уведена предложеном методом дала другачије резултате импутације. У неким сличајевима је боља импутација, док је у другим лошија.

Приликом импутације стохастичком линеарном регресијом постојала је само једна непозната; избор вредности које су обрисане случајним избором. Код предложене методе, уведена је још једна непозната; случајан избор почетних позиција центроида. Значајно мања просечна релативна грешка код скупа са 10 процената недостајућих вредности може бити резултат повољног избора почетних позиција центроида.

У сваком случају, кластеровање, односно груписање обсервација по сличности, је у одређеним ситуацијама повећало квалитет импутације. Уколико би се случајан избор центроида, односно крајњи изглед кластера једнозначно одредио, могуће да би и сами резултати импутације били бољи.

8. Закључак

Рад се бавио проблемом недостајућих вредности, методама импутације као и њиховим утицајем на тачност предвиђања. За решавање проблема

недостајућих вредности су коришћене три традиционалне методе импутације:

1) импутација линеарном регресијом,
2) импутација стохастичком линеарном регресијом,
3) имутација шумом стабала одлучивања.

Такође, предложена је нова метода импутације која се састоји од два корака, припреме и саме импутације. Најпре се подаци са недостајућим вредностима кластерују користећи к-средњих вредности кластеровање. Затим се сваки кластер посматра појединачно, и врши се онолико импутација колико има кластера. Том приликом коришћена је импутација стохастичком линеарном регресијом, јер се она показала као најбоља од три изабране традиционалне методе.

Приликом експеримента су на случајан начин обрисани различити проценти вредности из једног скупа података и на тај начин је добијено шест различитих скупова података. Над сваким од шест скупова је извршена импутација сваком од четири наведене методе где се показало да је стохастичка линеарна регресија најбоља метода импутације од набројаних традиционалних метода.

Каснијом анализом се није могло закључити која метода је боља између предложене методе и стохастичке линеарне регресије. Али свакако се може закључити да је корак припреме података у виду кластеровања дао другачије резултате (и боље и лошије).

Такође, тестирана је могућност предвиђања над скупом података након импутације и показало се да је у одређеним случајевима грешка приликом предвиђања јако велика. Таква особина је била изразита над скуповима података коме недостаје преко 20% вредности и импутација је вршена линеарном регресијом.

Предлог за даље истраживање се састоји у даљем унапређењу традиционалних метода импутације. У одређеним ситуацијама предложена метода је показала боље резултате од традиционалне, па је очекивано да разлог за такво побољшање лежи у кластеровању као кораку припреме. Како се кластеровање к-недостајућих вредности ослања на

случајан избор почетних позиција за центроиде, могуће је унапредити алгоритам тако да центроиди са великом вероватноћом заузму положај који одговара датој методи импутације.

Додатно, поред груписања обсервација по сличности, могуће је исто урадити и са променљивима. На тај начин би се додатно смањила комплексност проблема што може да проузрокује повећану тачност.

9. Референце

[1] A. Criminisi, J. Shotton, E. Konukoglu. *Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning*. Microsoft Research technical report TR-2011-114

[2] Amanda N. Baraldi, Craig K. Enders. (2010). *An introduction to modern missing data analyses*, Journal of School Psychology 48 (2010) 5–37

[3] Andrew Gelman, Jeniffer Hill. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press. New York

[4] Alexei Sharow. *Stochastic Model based on Regression*. https://www.ma.utexas.edu/users/davis/375/popecol/lec4/stoch.html. Приступ 08. Новембар. 2017

[5] Badr Hssina, Abdelkarim Merbouha, Hanane Ezzikouri, Mohammed Erritali. *A comparative study of decision tree ID3 and C4.5*. International Journal of Advanced Computer Science and Applications

[6] Bernd Prantner. (2011). *Visualization of imputed values using the R-package VIM*.

[7] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshiran. (2004). *Least Angle Regression*. The Annals of Statistics 2004, Vol. 32, No. 2, 407–499

[8] Craig K. Enders. (2010). *Applied Missing Data Analysis*. The Guilford Press, New York London

[9] Daniel J. Stekhoven, Peter Buhlmann. (2011). *MissForest - nonparametric missing value imputation for mixed-type data*. Oxford Journal's Bioinformatics

[10] D.Stekhoven. *MissForest - nonparametric missing value imputation for mixed-type data*. https://github.com/stekhoven/missForest/blob/master/R/prodNA.R приступио 8. Новембар 2017

[11] Edith D. de Leeuw, Joop Hox, Mark Huisman. (2003). *Prevention And Treatment Of Item Nonresponse*. Journal of Official Statistics, Vol 19, No. 2, 2003, pp. 153-176

[12] Ezekiel, M. (1930). *Methods of Correlation Analysis*. Wiley.

[18] Jun He, Donna McClish. (2015). *The Application of Last Observation Carried Forward in the Persistent Binary Case*. Austin Publishing Group

[19] Kimberly Ault. (2012). *Multiple Imputation for Ordinal Variables: A Comparison of Sudaam Proc Impute and SAS Proc MI*. RTI International, RTP, NC

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

[21] Masayoshi Takahashi, Takayuki Ito. (2012). *Multiple Iimputation Of Turnover In Edinet Data: Toward The Improvement Of Imputation For The Economic Census*. Conference of European Statistians, Oslo, Norway, 24-26 September 2012

[22] McNeil, D. R. (1977). *Interactive Data Analysis*. Wiley.

[23] Paul D. Allison, (2012). *Handling Missing Data by Maximum Likelihood*. SAS Global Forum 2012, Statistical Horizons, Haverford, PA, USA

[24] Samprit Chatterjee, Alli S. Hadi. (2006). *Regression Analysis By Example.* John Wiley & Sons, Inc., Hoboken, New Jersey

[25] Stef van Buuren, Karin Groothuis-Oudshoorn. (2011). *mice: Multivariate Imputation by Chained Equations in R*. Journal of Statistical Software, December 2011, Volume 45, Issue 3.

[26] Stephane Dray, Julie Josse, (2014). *Principal component analysis with missing values*. Springer Science+Business Media Dordrecht 2014

[27] Therese D. Pigott. (2001). *A Review of Methods for Missing Data*. Educational Research and Evaluation 2001, Vol. 7, No. 4, pp. 353-383

[28] X. Rong Li, Zhanlue Zhao. *Relative Error Measures for Evaluation of Estimation Algorithms*. Department of Electrical Engineering, University of New Orleans, New Orleans

[29] Yang C. Yuan. *Multiple Imputation for Missing Data: Concepts and New Development*. SAS Institute Inc., Rockville, MD

[30] Yang Yuan. (2011). *Multiple Imputation Using SAS Software*. Journal of Statistical Software, December 2011, Volume 45, Issue 6.