Узорковање слова

Семинарски рад у оквиру курса
Увод у теорију узорака
Математички факултет, Београд

Лазар Васовић
mi16099@alas.matf.bg.ac.rs

23. септембар 2020.

Сажетак

Размотрена је употреба неколико планова узорковања над скупом података који представља велика слова енглеске абецеде, уз праћење релевантних методолошких правила. Велика пажња посвећена је аспектима теме уско повезаним са информатиком, као што су анализа скупа и утицај узорковања на његову класификацију. Посебан значај дат је *R*-у као моћном вишепарадигматском језику. Предложен је статистички заснован метод за одређивање важности предиктора. Све оцене средње вредности обележја од значаја детаљно су упоређене, не само по прецизности, већ и према особинама плана из кога су настале.

Кључне речи — слова, узорковање, класификација

Садржај

1 Увод

Свако научно истраживање, независно од теме и обима, представља систематско, планско и објективно испитивање неког проблема, према одређеним методолошким правилима, чија је сврха да се пружи поуздан и прецизан одговор на унапред постављено питање. Обично се састоји од серије логички повезаних фаза, које се простиру од уводног одређивања проблема до закључног представљања резултата.[1]

У овом раду спроведено је мини-истраживање над скупом података који представља велика слова енглеске абецеде. Примењена су методолошка правила усвојена на факултетским курсевима Увод у теорију узорака, Статистика и Методологија стручног и научног рада на информатичком смеру Математичког факултета Универзитета у Београду. Једино је скраћена серија фаза, што је последица чињенице да није било прикупљања података, већ је коришћен већ доступан и припремљен скуп ентитета, дакле, у потпуности спремна популација.

Велика пажња посвећена је аспектима теме уско повезаним са информатиком као науком о подацима и њиховој обради, као што су експлоративна анализа скупа и утицај узорковања на његову класификацију. Ово је важно за примене у областима попут истраживања података и машинског учења. Посебан значај дат је и *R*-у као моћном вишепарадигматском програмском језику, при чему је фокус на његовим функционалним, векторским и симболичким концептима, који га издвајају од конкурената у свету савременог статистичког софтвера.

Први циљ спроведеног истраживања био је теоријски, просто стицање увида у преузете податке – шта они и на који начин представљају (начин генерисања, број и врста обележја) – као и сазнање о томе који атрибути су међусобно корелисани, да ли подлежу некој од познатих вероватносних расподела и слично. Паралелно са овим циљем, покушан је проналазак одговора на повезано питање примењене природе – како репрезентативно узорковати, да на основу потпопулације буде могуће направити добар класификациони модел. Идеја је била открити која обележја највише обећавају по питању погађања описаног слова на основну осталих атрибута и то искористити за план узорковања.

2 Скуп података

Скуп података „Letter Recognition“ настао је 1991. године за потребе рада „Letter Recognition Using Holland-Style Adaptive Classifiers" америчких научника информатичара Дејвида Ц. Слејта (енгл. *David J. Slate*) и психолога Питера В. Фреја (енгл. *Peter W. Frey*). Обојица се баве разним проблемима вештачке интелигенције и машинског учења, а у поменутом раду су дискутовали прилагодљиве класификаторе.[2]

Обрађивани подаци јесу они о великим словима енглеске абецеде. Наиме, у питању су црно-беле правоугаоне растерске слике које је потребно препознати као једно од 26 могућих слова. Толики број класа чини овај задатак тешким, за разлику од уобичајеног случаја, када постоје само две односно тек неколико (мали број) категорија.

При генерисању података, употребљено је двадесет различитих фонтова, намерно одабраних тако да обухвате што више различитих стилова и начина писања. Диверзитет и хетерогеност додатно су повећани рандомизованим изобличавањем полазних слика.

Укупно је 20.000 инстанци. Свака од њих је добијена као резултат позива посебно написаног програма за генерисање слике слова. За

одабир параметара попут врсте фонта, типа слова, величине слике и фактора изобличења (искривљења), свеобухватности ради, коришћене су равномерно расподељене случајне променљиве/величине.

Додатни детаљи о коришћеним фонтовима и току генераторског програма могу се видети у поменутом раду. Корисно је засад још напоменути да су излаз програма биле слике просечних димензија 45×45 пиксела, који су искључиво имали вредности „укључено“ и „искључено“ („да“ и „не“, црно и бело, тачно и нетачно, суштина је да су у питању бинарни пискели, са само две вредности), те да су, упркос изобличењима, према процени ауторâ, сва слова са слика махом била препознатљива људима. Пример добијених слова дат је на слици 1.

Слика 1: Пример генерисаних слова

Слике, међутим, нису оно што чини овај скуп података, већ низ нумеричких вредности. Ти бројеви су у даљем процесу прављења података добијени систематским читањем слика пиксел по пиксел, те израчунавањем основних статистичких особина расподеле пискела, о чему ће нешто детаљније бити говорено у наставку текста.

2.1 Опис скупа података

Скуп података „Letter Recognition“, дакле, чини 20.000 слогова (инстанци, ентитета) распоређених у 26 категорија, које представљају одговарајуће велико слово енглеске абецеде које та инстанца описује. Подаци су јавно и бесплатно доступни на интернет страници репозиторијума за машинско учење Универзитета Калифорније у Ервајну.[3]

Формат података је уобичајени вишедимензиони, у ком различита поља у подацима одговарају различитим мерљивим особинама које су тим пољима (атрибутима, димензијама) представљени. Атрибута има 17, од чега је 16 улазних атрибута квантитативног (нумеричког) типа. Посреди су цели бројеви, мада је дискутабилно да ли их је природније тако посматрати или као категорије, пошто узимају коначан број вредности. Преостаје још један излазни атрибут (класа, тип слова) квалитативног (категоричког), односно именског типа. Јасно је, притом, да је и домен последњег атрибута коначан, те је и он дискретан, попут претходно наведених улазних атрибута. Пример је на слици 2.

	slovo	x_kutija	y_kutija	širina	visina	broj_piksela	x_mean	y_mean	x2_var
1	T	2	8	3	5	1	8	13	0
2	I	5	12	3	7	2	10	5	5
3	D	4	11	6	8	6	10	6	2
4	N	7	11	6	6	3	5	9	4
5	G	2	1	3	1	1	8	6	6
6	S	4	11	5	8	3	8	8	6
7	B	4	2	5	4	4	8	7	6
8	A	1	1	3	2	1	8	2	2
9	J	2	2	4	4	2	10	6	2
10	M	11	15	13	9	7	13	2	6

	slovo	y2_var	xy_kor	x2y_mean	xy2_mean	x_ivice	xivy_kor	y_ivice	yivx_kor
1	T	6	6	10	8	0	8	0	8
2	I	4	13	3	9	2	8	4	10
3	D	6	10	3	7	3	7	3	9
4	N	6	4	4	10	6	10	2	8
5	G	6	6	5	9	1	7	5	10
6	S	9	5	6	6	0	8	9	7
7	B	6	7	6	6	2	8	7	10
8	A	2	8	2	8	1	6	2	7
9	J	6	12	4	8	1	6	1	7
10	M	2	12	1	9	8	1	1	8

Слика 2: Глава обрађиваног скупа слова

Оно што овај скуп чини посебно атрактивним јесте чињеница да су сви нумерички атрибути стандардизовани (тј. нормализовани, употреба термина зависи од случаја). Наиме, сваки се налази у целобројном интервалу [0, 15]. Ово је постигнуто линеарним скалирањем, што доприноси компактности података и спречава одређене алгоритме да фаворизују неки атрибут само зато што он има већи распон. Осим тога, на тај начин се олакшава припрема података, односно избегава потреба за претпроцесирањем у контексту скалирања. Ипак, проблем би био покушај опонашања програма за генерисање слова – не би било јасно како трансформисани добијене сирове податке (слике).

У скупу не постоје недостајуће нити бланко вредности, као ни некоректни нити дуплирани подаци, што додатно олакшава припрему података, као и сам рад са њима. Уз то је свака инстанца независна.

У наставку следи опис сваког атрибута, односно, у случају улазних нумеричких вредности, његовог значења пре сабијања у нормализацијом ограничен интервал. Притом се при помињању координата мисли

на уобичајени Декартов координатни систем са почетком у доњем левом углу, у ком x оса расте надесно, док y оса расте нагоре:

1. slovo – тип слова, дискретна именска вредност из домена {A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z}; представља циљни атрибут, то јест, класу (категорију) инстанце којој је придружена,
2. x_kutija – водоравни положај (x координата) центра најмањег правоугаоника који обухвата све „укључене“ пикселе [може се нацртати тако да се сви „укључени“ пиксели налазе у њему],
3. y_kutija – усправни положај (y координата) центра те кутије,
4. širina – ширина (број тј. удео водоравних пиксела) те кутије,
5. visina – висина (број тј. удео усправних пиксела) те кутије,
6. broj_piksela – број тј. удео „укључених“ пиксела на слици,
7. x_mean – средња вредност (математичко очекивање) водоравног положаја (x координате) „укључених“ пиксела у односу на центар кутије подељен ширином кутије (негативна вредност за нпр. налево померено „L“),
8. y_mean – средња вредност (математичко очекивање) усправног положаја (y координате) „укључених“ пиксела у односу на центар кутије подељен висином кутије (негативна вредност за нпр. надоле померено „L“),
9. x2_var – средња вредност квадратне водоравне удаљености (средњеквадратно одступање) „укључених“ пиксела од очекивања из седмог атрибута (дисперзија/варијанса, већа код нпр. хоризонтално раширених „W“ и „M“),
10. y2_var – средња вредност квадратне усправне удаљености (средњеквадратно одступање) „укључених“ пиксела од очекивања из осмог атрибута (дисперзија/варијанса, већа код нпр. вертикално раширених „E“ и „K“),
11. xy_kor – средња вредност производа водоравног и усправног одступања „укључених“ пиксела од очекивања из седмог и осмог атрибута (корелација, позитивна за линије облика $y = x$, негативна за $y = -x$),
12. x2y_mean – средња вредност производа квадратног водоравног и усправног одступања „укључених“ пиксела од очекивања из седмог и осмог атрибута (корелација хоризонталне дисперзије и вертикалног положаја),
13. xy2_mean – средња вредност производа квадратног усправног и водоравног одступања „укључених“ пиксела од очекивања из седмог и осмог атрибута (корелација вертикалне дисперзије и хоризонталног положаја),
14. x_ivice – средња вредност броја ивица („укључен“ пиксел одмах након [десно од] „искљученог“ или лева ивица/граница/крај слике) при читању (скенирању) слике слева надесно (разликовање нпр. „W“ или „M“ и „I“ или „L“),
15. xivy_kor – збир усправних положаја ивица из претходног атрибута (корелација броја вертикалних ивица са хоризонталним положајем, веће за нпр. „Y“),

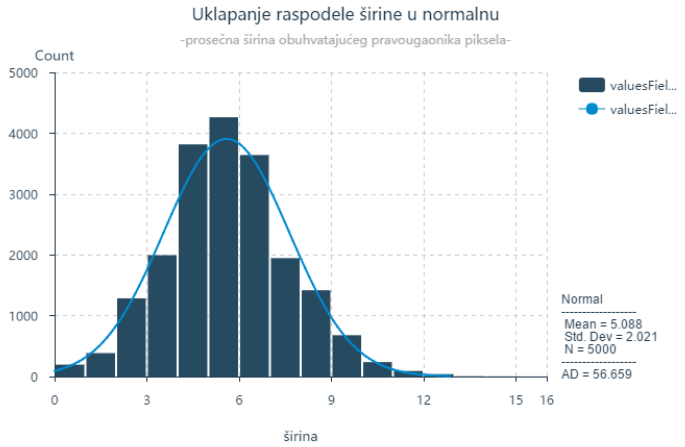
16. y_ivice – средња вредност броја ивица („укључен“ пиксел одмах након [изнад] „искљученог“ или ивица/граница/крај слике) при читању (скенирању) слике од доле нагоре (разликовање нпр. „E“ или „B“ и „I“ или „L“),
17. yivx_kor – збир водоравних положаја ивица из претходног атрибута (корелација броја хориз. ивица са верт. положајем).

2.2 Додатне визуелизације

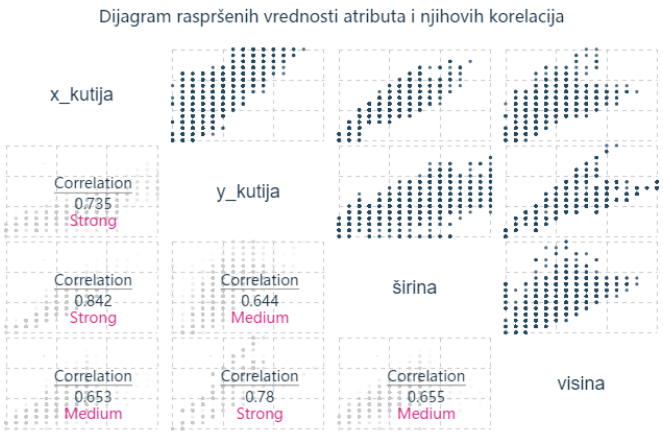
За почетак, неопходно је стећи неки утисак о конкретним подацима, као и могућностима за манипулацију њима. Почетни део експлоративне анализе је, једноставности ради, реализован помоћу статистичког пакета *IBM*-овог *SPSS Modeler*-а.[4] Исто је могло бити урађено помоћу *R*-а, али би захтевало доста кодирања само за увод у причу, тако да се ипак прибегло релативно аутоматизованом приступу. У оквиру овога се само једним кликом могла видети расподела атрибута, број екстрема и аутлајера, као и потврдити већ позната чињеница да су сви подаци исправни и без недостајућих вредности.

У картици за приказ напреднијих визуелизација, дошло се до још занимљивијих резултата. Наиме, најзначајнија је била визуелизација атрибута, што независно (уз општију аутоматског погађања расподеле), то у паровима (ради разматрања зависности), као и Пирсонова матрица корелација. Помоћу ње је утврђено да су атрибути махом независни у пару, те да највећа корелација (0,62–0,85) постоји између атрибута који представљају димензије, што је и очекивано – јасно је да ће нпр. ширина и висина кутије бити у тесној вези са x и y координатом њеног центра и слично. Ово је омогућило да се у даљој обради покуша са редукцијом, што ће касније и бити дискутовано.

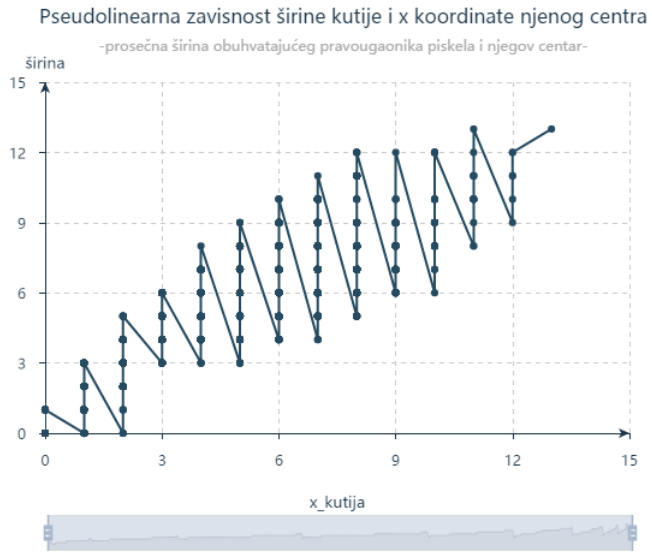
На слици 3 дат је пример уклапања ширине кутије у нормалну расподелу. На наредним сликама 4 и 5 дати су дијаграми корелације атрибута са највећом сличношћу, где се уочава њихова средња до јака псеудолинеарна (псеудо због дискретних скокова) зависност.



Слика 3: Уклапање расподеле ширине пиксела у нормалну



Слика 4: Приказ парова атрибута са највећом корелацијом



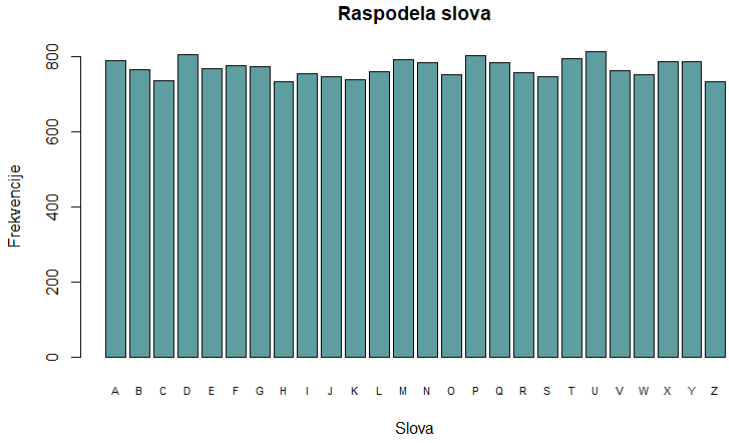
Слика 5: Приказ пара атрибута са највећом корелацијом

На слици 6 приказан је тракасти дијаграм апсолутних фреквенција циљног атрибута, као начин његове визуелизације и уверавања да су класе релативно равномерно распоређене. Ова слика је добијена помоћу приложеног R скрипта 1, уз наведене функционалне концепте.

```
# ucitavanje podataka
slova <- read.csv('slova.csv')
cilj <- slova$slovo
```

```r

# brojanje svakog pojedinacnog slova;
# funkcionalni koncepti: sapply umesto petlje
# i anonimna funkcija bez return naredbe
freq <- sapply(LETTERS,
               function(x) sum(x == cilj))

# iscrtavanje trakastog dijagrama
barplot(freq,
        main = 'Raspodela slova',
        xlab = 'Slova',
        ylab = 'Frekvencije',
        cex.names = .7,
        col = 'cadetblue')
```

Скрипт 1: barplot.r – исцртавање тракастог дијаграма



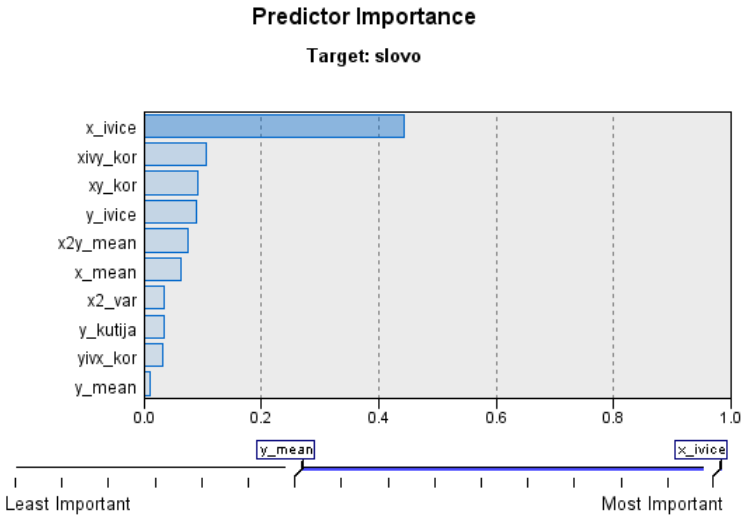
Слика 6: Тракасти дијаграм категорије

На крају, одрађена је још једна аутоматска експлорација помоћу *SPSS*-а, овог пута са циљем откривања атрибута који су у високој корелацији са циљним. Како класа није нумеричка, ово није могло бити виђено приликом формирања Пирсонових коефицијената. Резултат овога је важност предиктора, што је списак атрибута уређен по моћи да се на основу њихове вредности одреди категорија ентитета. Приказ је дат на слици 7. Како је могуће приметити, као убедљиво најкориснији улазни параметар издваја се четрнаести – хоризонтална средња вредност броја ивица слова – тако да је он обележје од значаја.

Једноставности ради, и ова анализа је, попут претходних у *SPSS*-у, аутоматизована. Представљена је као алгоритам црне кутије, без задирања у конкретну имплементацију, већ само преко резултата. Детаљи би превазишли теорију узорака, па је ипак фокус на простом добијању информацијама који ће бити значајне при даљем раду са базом.

2.3 Класификација

Укратко, класификација је проблем препознавања врсте објекта; у конкретном случају – препознавање које слово представљају неки



Слика 7: Важност предиктора

подаци. Шире гледано, она је пример надгледаног машинског учења, што значи да је заједно са скупом улазних података прослеђен и жељени излаз (класа) за сваку инстанцу. Алгоритми класификације приликом учења модела (укалупљивања података у модел) знају која инстанца представља које слово. На основу виђених података покушава се формирање представе о узрочно-последичним односима у обрађеном скупу података, као и уопштавање закључака на невиђене податке, са циљем да излазни модел са великом прецизношћу разликује слова према њиховим особинама. Крајњи резултат тестира се на контролним подацима – оним који нису учествовали у тренирању.

Како је најављено да ће планови узорковања бити коришћени за класификацију скупа, ваља проценити предиктивну моћ модела направљеног над целом популацијом. То је и урађено скриптом 2. Притом је као алгоритам за прављење модела коришћен метод потпорних вектора (енгл. *support-vector machine*, SVM). Овај метод заснован је на статистичкој теорији учења и на идеји векторских простора. Модел је формула на основу које се израчунава класа. Алгоритам налази раздвајајућу хиперраван која раздваја категорије унутар векторског простора података, при чему максимизује размак између хиперравни и најближих јој инстанци које раздваја. Један је од најкомплекснијих метода надгледаног машинског учења, али је зато врло успешан. На обрађеном скупу података постигнута је прецизност од 96,24 %.

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# vertikalna podela skupa na dva dela;
# simbolicki koncept: minus nije oduzimanje
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)
```

```
10
11 # fiksiranje generatora pseudoslucajnosti
12 set.seed(0)
13
14 # pravljenje SVM modela
15 model <- svm(x, y)
16 summary(model)
17
18 # provera kvaliteta modela
19 pred <- fitted(model)
20 prec <- mean(y == pred) # 0.9624
```

Скрипт 2: klasif.r – модел над целом популацијом

Наравно, ово је најпростији случај, без поделе на тренинг и тест скуп, тако да је у питању својеврсна горња граница прецизности. У наставку ће се подела вршити изабраним плановима узорковања, а циљ ће бити задржати што већу прецизност упркос мањем улазу.

3 Узорковање

Узорковање је извлачење подскупа популације. Тај подскуп, очекивано, садржи извесне ентитете који потичу из популације, на бази чијег проучавања се изводе закључци о читавој популацији. Узорковање је важно, чак и када је цела популација (цензус) доступна.[5] Постоји већи број предности делимичног испитивања – мања цена, већа брзина и, можда најзначајније, контрола тачности (нпр. у машинском учењу је узорак тренинг скуп, док је остатак популације тест скуп).

Посматрано кроз статистичку терминологију, сваки скуп конкретних вредности обележја које описују једно слово назива се јединица посматрања (често је то и јединица узорковања), док су сва слова заједно управо популација. Величина популације је 20.000, док је оквир за одабир узорка произвољног обима база дата у формату запетом раздвојених вредности (једна *CSV* датотека). Ознака јединице је индекс ентитета у скупу, пошто он заправо и није неуређени скуп, већ уређени низ. О природи 17 обележја већ је детаљно дискутовано.

3.1 Прост случајан узорак

Прост случајан узорак је једноставан план узорковања у коме је јединица посматрања једнака јединици узорковања.[6] Из популације се сукцесивно узима ентитет по ентитет, све док се не достигне жељени обим узорка. Све инстанце имају једнаку вероватноћу да буду изабране у неком кораку, при чему понављања могу или не морају бити дозвољена. Најчешће се узоркује преко ознака, у конкретном случају – индекси слова. Уобичајене ознаке су N за величину популације, а n за обим узорка. Случајан избор реализује се помоћу генератора случајности, што је овде R-ов програмски генератор псеудослучајних бројева. Приликом закључивања се користи приступ заснован на методу одабира узорка – популација је фиксирана (детерминистичка, све се тачно налази у датотеци), само вредности њених обележја нису позната. Једина случајност, дакле, лежи у одабиру узорка.

Како је при уводним разматрањима утврђено да хоризонтална средња вредност броја ивица слова (четрнаести атрибут) има највећу предиктивну вредност, може се претпоставити да ће солидан успех донети узорак који има сличне статистичке особине тог обележја.

Претпоставка 1 *Класификациони модел направљен над узорком који чува расподелу четрнаестог атрибута постиже већу прецизност од оног направљеног над узорком који ту расподелу не чува.*

Први покушај је, стога, осигурати се да узорак испуњава изнесену претпоставку. Конкретно се може посматрати средња вредност обележја. Њена непристрасна тачкаста оцена је узорачка средња вредност $\overline{Y}$. Ова оцена није прецизна за узорке малог обима, при чему је тачкаста оцена њене дисперзије $\frac{\overline{S}^2}{n}(1-\frac{n}{N})$ за прост случајан узорак без понављања, а без фактора корекције $\frac{\overline{S}^2}{n}$ за узорак са понављањем. Двострани приближни интервал поверења популацијске средње вредности је $\overline{Y} \pm z_{1-\frac{\alpha}{2}}\sqrt{\hat{\sigma}^2}$, при чему је $\hat{\sigma}^2$ оцена дисперзије.

Идеја је да апсолутна разлика између оцене и праве популацијске вредности буде што мања. Може се показати да је $n_0 = (\frac{\sigma z_{1-\frac{\alpha}{2}}}{\Delta})^2$ добра доња граница обима узорка са понављањем када се ради са популацијском средњом вредношћу, док се $\frac{1}{\frac{1}{n_0}+\frac{1}{N}}$ користи код узорка без понављања. Притом је σ популацијска ст. грешка, α дозвољена грешка прве врсте (поверење $1-\alpha$), Δ дозвољена апсолутна грешка, а z одговарајући квантил стандардне нормалне расподеле.

Узорковање без понављања је одрађено у скрипту 3. Изабрана је прилично конзервативна дозвољена апсолутна грешка, уз висок ниво поверења 95 %. Како би се одредио адекватан обим узорка, спроведено је пилот истраживање над сточланим узорком, што је омогућило процену дисперзије. Испоставља се да је оцена средње вредности обележја од значаја одлична – узорачке вредности врло су блиске популацијским, са апсолутном грешком мањом од највеће дозвољене, а дисперзија оцене је мала. И добијени интервал поверења је, упркос високом задатом степену поверења, прилично узак, при чему је, очекивано, успешно обухватио популацијску средњу вредност. На крају, над овим узорком од 3199 инстанци (свака шеста или седма) направљен је класификациони модел који погађа добрих 87,68 % слова.

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
N <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# dozvoljena apsolutna greska
d <- 0.07

# dozvoljena greska prve vrste
alpha <- 0.05

# odgovarajuci kvantil N(0,1)
z <- qnorm(1 - alpha/2) # 1.9600

# fiksiranje generatora pseudoslucajnosti
set.seed(0)
```

```r

# pilot istrazivanje nad malim uzorkom
n <- 100
indeksi <- sample(N, n)
uzorak <- obelezje[indeksi]
sn2 <- var(uzorak) # 4.8562

# neophodan obim uzorka
n0 <- sn2 * (z/d)^2 # 3807
n1 <- (1/n0 + 1/N)^{-1} # 3198
n <- ceiling(n1) # 3199

# uzorkovanje prema izracunatom
indeksi <- sample(N, n)
uzorak <- obelezje[indeksi]

# ocenjivanje srednje vrednosti
xn <- mean(uzorak) # 3.0078
sn2 <- var(uzorak) # 5.4111
D_xn <- sn2/n * (1 - n/N) # 0.0014
greska <- abs(sr - xn) # 0.0383

# interval poverenja
sirina <- z * sqrt(D_xn) # 0.0739
I_xn <- c(xn - sirina, # 2.93
          xn + sirina) # 3.08
upada <- sr >= I_xn[1] &&
         sr <= I_xn[2] # TRUE

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.8768
```

Скрипт 3: srswor.r – модел над *SRSWOR* узорком

Узорковање са понављањем је одрађено у скрипту 4. Код је сличан као за верзију без понављања. Ипак, овде свако извлачење даје реализацију случајне величине независне од осталих, тако да нема фактора корекције (отклона) у формулама, нити теоријске потребе за проширењем централне граничне теореме зарад решавања проблема коначне популације (суперпопулација и слично). Шире гледано, бутстреп (енгл. *bootstrap*) методе – статистички поступци засновани на узорцима са понављањем – популарне су у машинском учењу. Не само што су у позадини лакше за имплементацију (нема провере јединствености), већ се испоставља да често дају боље резултате од метода без понављања. Над узорком од 3808 инстанци (свака пета или шеста) направљен је класификациони модел који погађа 88,29 % слова. Редуковани узорак од 3445 инстанци (уклањање дупликата) учествовао је у прављењу врло сличног модела са прецизношћу 88,11 %.

```r
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
```

```r
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
N <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# dozvoljena apsolutna greska
d <- 0.07

# dozvoljena greska prve vrste
alpha <- 0.05

# odgovarajuci kvantil N(0,1)
z <- qnorm(1 - alpha/2) # 1.9600

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# pilot istrazivanje nad malim uzorkom;
# OVDE JE REPLACE TRUE, PONAVLJANJE
n <- 100
indeksi <- sample(N, n,
                  replace = T)
uzorak <- obelezje[indeksi]
sn2 <- var(uzorak) # 4.8562

# neophodan obim uzorka;
# KORAK MANJE NEGO SRSWOR
n0 <- sn2 * (z/d)^2 # 3807
n <- ceiling(n0) # 3808

# uzorkovanje prema izracunatom;
# OVDE JE REPLACE TRUE, PONAVLJANJE
indeksi <- sample(N, n,
                  replace = T)
uzorak <- obelezje[indeksi]

# ocenjivanje srednje vrednosti;
# BEZ OTKLONA U DISPERZIJI
xn <- mean(uzorak) # 3.0355
sn2 <- var(uzorak) # 5.4899
D_xn <- sn2/n # 0.0014
greska <- abs(sr - xn) # 0.0106

# interval poverenja
sirina <- z * sqrt(D_xn) # 0.0744
I_xn <- c(xn - sirina, # 2.96
          xn + sirina) # 3.11
upada <- sr >= I_xn[1] &&
         sr <= I_xn[2] # TRUE

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
```

```
pred <- predict(model, x)
prec <- mean(y == pred) # 0.8829

# iskljucivanje ponovljenih entiteta
indeksi <- unique(indeksi)
uzorak <- obelezje[indeksi]

# ocenjivanje srednje vrednosti
n <- length(uzorak) # 3445
xn <- mean(uzorak) # 3.0165
sn2 <- var(uzorak) # 5.4385
D_xn <- sum(sapply(1:(N-1),
            function (k) (k/N)^(n-1)/N)) *
            sn2 # 0.0014
greska <- abs(sr - xn) # 0.0296

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.8811
```

Скрипт 4: srswr.r – модел над *SRSWR* узорком

Засад се чини да предложени план узорковања даје добре резултате по питању прецизности добијеног класификационог модела. Проценат погођених слова свеукупно веома је висок код модела направљених над узорцима који чувају расподелу обележја од значаја. На тренинг скупу је, очекивано, још већи, док је на тест скупу сличан просечној вредности. Прецизност опада повећањем дозвољене апсолутне грешке оцене, али је тада и обим узорка мањи. За проверу изнесене претпоставке, неопходно је још показати како се понашају други модели – они који не чувају расподелу хоризонталног броја ивица.

3.2 Узорак са неједнаким вероватноћама

Узорак са неједнаким вероватноћама је још један једноставан план узорковања, с тим што код њега јединица посматрања није нужно једнака јединици узорковања.[7] И у овом случају важе опште тврдње изнесене за прост случајан узорак, са главном разликом да овде свака инстанца има засебну вероватноћу извлачења у неком кораку.

Ред је вратити се на претходну претпоставку. Сада је циљ формирати узорак који не чува расподелу обележја од значаја и упоредити га са оним који чува. За те потребе може се формирати узорак са неједнаким вероватноћама, при чему вероватноћа одабира намерно расте удаљавањем од популацијске средње вредности. На овај начин очекује се веће одступање узорачке средње вредности од популацијске, као и већа узорачка варијанса. Резултат се може оценити неком од познатих непристрасних оцена средње вредности код овог плана узорковања – Хансен-Хурвицовом $\frac{1}{Nn}\sum_{k\in R}\frac{y_k}{\psi_k}$, Хорвиц-Томпсоновом $\frac{1}{N}\sum_{k\in S}\frac{y_k}{\pi_k}$ или Хајековом $\frac{\sum_{k\in S}\frac{y_k}{\pi_k}}{\sum_{k\in S}\frac{1}{\pi_k}}$, при чему су ψ вероватноће извлачења у неком кораку, π вероватноће укључења првог реда, R узорак, а S редуковани узорак без понављања. Познате су и тачкасте

оцене дисперзија претходних оцена, и оне су израчунате у коду, иако, једноставности ради (велике су формуле), нису наведене у тексту.

Овакво узорковање са понављањем одрађено је у скрипту 5. Коришћен је обим узорка израчунат при раду са простим случајним узорком. Вештачки уведене неједнаке вероватноће одабира квадратно расту са удаљавањем обележја од значаја од његове средње вредности, како би се добио узорак који не чува расподелу тог атрибута. Знајући расподелу при одабиру, све три оцене показале су се као одличне при погађању популацијске средње вредности. Веома су прецизне и са малом варијансом, упркос великој узорачкој дисперзији обележја. Ипак, алгоритми класификације не користе податак о вероватноћи одабира, већ само на основу улазних инстанци праве модел, сматрајући да су добили репрезентативан узорак. Тако је на основу узорка од 3808 ентитета добијена прецизност од само 68,785 % (упоредити са претходних 88,29 % за исти обим узорка), док је над редукованим узорком од 2781 инстанце постигнута слична прецизност 68,045 % (претходно 88,11 %, додуше, са мањом редукцијом, због мањег удела дупликата).

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
N <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# ranije odredjen obim uzorka
n <- 3808

# nejednake verovatnoce odabira
psi <- (obelezje - sr)^2
psi <- psi / sum(psi)

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# uzorkovanje prema izracunatom
indeksi <- sample(N, n,
                  prob = psi,
                  replace = T)
uzorak <- obelezje[indeksi]
psi_uz <- psi[indeksi]

# osnovne statistike uzorka
xn <- mean(uzorak) # 5.5087
sn2 <- var(uzorak) # 17.1583

# Hansen-Hurwitzova ocena srednje vrednosti
xn_hh <- 1/N * mean(uzorak/psi_uz) # 2.3304

# ocena disperzije ovakve HH ocene
D_hh_ocena <- 1/(n-1) *
              mean((uzorak/(N*psi_uz) -
```

```
                    xn_hh)^2) # 0.0033

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.68785

# iskljucivanje ponovljenih entiteta
indeksi <- unique(indeksi)
uzorak <- obelezje[indeksi]
psi_uz <- psi[indeksi]

# osnovne statistike uzorka
n <- length(indeksi) # 2781
xn <- mean(uzorak) # 4.5750
sn2 <- var(uzorak) # 14.5171

# verovatnoce ukljucenja prvog reda
pi <- 1 - (1 - psi)^n
pi_uz <- pi[indeksi]

# Horvitz-Thompsonova ocena srednje vrednosti
xn_ht <- 1/N * sum(uzorak/pi_uz) # 3.0334

# ocena disperzije ovakve HT ocene
D_ht_ocena <- sum((1/pi_uz^2 - 1/pi_uz) * uzorak^2)
for (k in 1:n) {
  for (l in 1:n) {
    if (k != l) {
      # verovatnoca ukljucenja drugog reda
      pi_kl <- pi_uz[k] + pi_uz[l] - 1 +
               (1 - psi_uz[k] - psi_uz[l])^n
      D_ht_ocena <- D_ht_ocena +
                    (1/(pi_uz[k]*pi_uz[l]) - 1/pi_kl) *
                    uzorak[k] * uzorak[l]
    }
  }
}
D_ht_ocena <- 1/N^2 * D_ht_ocena # 0.0058

# Hajekova ocena srednje vrednosti
xn_hajek <- sum(uzorak/pi_uz) / sum(1/pi_uz) # 3.0000

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.68045
```

Скрипт 5: upswr.r – модел над неједнаким вероватноћама

Верзија без понављања одрађена је у скрипту 6. Ситуација је слична као код верзије са понављањем, уз разлику да није оцењивана популацијска вредност, пошто прва оцена захтева узорак са понављањем,

док друге две захтевају познате вероватноће укључења, које није лако израчунати. На основу узорка од 3199 инстанци добијена је прецизност од 68,3 % (упоредити са претходних 87,68 % код простог случајног узорка, одабраног тако да чува расподелу обележја од значаја).

```r
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
N <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# ranije odredjen obim uzorka;
# RAZLIKA U ODNOSU NA UPSWR
n <- 3199

# nejednake verovatnoce odabira
psi <- (obelezje - sr)^2
psi <- psi / sum(psi)

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# uzorkovanje prema izracunatom
indeksi <- sample(N, n,
                  prob = psi)
uzorak <- obelezje[indeksi]
psi_uz <- psi[indeksi]

# osnovne statistike uzorka
xn <- mean(uzorak) # 4.5355
sn2 <- var(uzorak) # 14.1192

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.683
```

Скрипт 6: upswor.r – модел над неједнаким вероватноћама

Када се све узме у обзир, могло би се закључити да је предложена хипотеза тачна, те да класификациони модел направљен над узорком који чува расподелу четрнаестог атрибута заиста постиже већу прецизност од оног направљеног над узорком који ту расподелу не чува. Не само то, већ би се могао предложити општи метод за одређивање обележја од значаја за проблем класификације. Наиме, у случају да није познато који је атрибут у најтешњој вези са циљном класом, имало би смисла проћи кроз сваки, при чему би се спровели следећи кораци:

1. извлачење простог случајног узорка одговарајућег обима који чува расподелу текућег обележја и прављење модела,
2. извлачење узорка са неједнаким вероватноћама осмишљеним тако да „кваре“ расподелу текућег обележја и прављење модела,
3. поређење прецизности – разлика указује на предиктивни значај обележја, и то вероватно тако да је већа разлика важнија.

3.3 Количничко и регресионо оцењивање

Количничко оцењивање, поред интересног атрибута Y, узима у обзир вредности помоћних обележја која су висококорелисана са обележјем од значаја, нпр. неког X.[8] На тај начин се боље искоришћавају подаци и добијају оцене са често мањом дисперзијом него када се посматра само интересни атрибут. У најједноставнијем облику, израчуна се однос (количник) узорачких средњих вредности $R = \frac{\overline{Y}}{\overline{X}}$ или тотала, а затим се о непознатим параметрима расподеле Y на целој популацији закључује преко познатих параметара X, нпр. $\hat{\tau}_y^R = \tau_x R$, где је τ_x познати популацијски тотал помоћног обележја, док је $\hat{\tau}_y^R$ количничка оцена тотала обележја од значаја. Осим мање дисперзије, предност овог приступа је и могућност оцењивања када није позната величина популације. Мана је што су оцене благо пристрасне, али су ипак асимптотски непристрасне, те су добре за веће узорке.[9]

Регресионо оцењивање иде корак даље. И овде се предлаже техника којом се побољшава прецизност оцена непознатих популацијских параметара главног обележја Y, коришћењем помоћног обележја X које је у корелацији са Y, уз додатак да је ограничење слободније – прихватљиве су и линеарне везе по правама које не пролазе кроз координатни почетак, док је код количничког оцењивања то неопходно.[10] Главни резултат овог приступа је тачкаста оцена средње вредности $\hat{m}_y^{lr} = \overline{Y} + b(m_x - \overline{X})$, са погодном унапред одабраном константом (алтернативно статистиком) b и другим познатим ознакама.

Овакво оцењивање, међутим, није претерано корисно када је у питању проучавани скуп слова. Иако је обележје од значаја нумеричког типа, категорије нису, тако да нема смисла погађати их на овај начин. С друге стране, сама идеја која стоји иза количничког и регресионог приступа – руковање висококорелисаним паровима атрибута – врло је важна у машиском учењу, па је и размотрена у наставку.

Често је превише захтевно радити са свим обележјима неког скупа података. Тада се прибегава својеврсном вертикалном узорковању – бира се подскуп атрибута уместо подскупа ентитета. Овај процес назива се димензиона редукција,[11] док је њен аспект обрађен у овом раду такозвани одабир карактеристика (енгл. *feature selection*). Једноставна идеја заснива се на искључивању атрибута који су у високој корелацији са неким другим обележјем. Претпоставка је да би се тај атрибут могао израчунати на основу других (идеја еквивалентна оној из количничког и регресионог оцењивања), те да из тог разлога не доприноси варијетету података, што даље повлачи да није користан у прављењу класификационог модела, те да се може избацити.

Претпоставка 2 *Класификациони модел направљен над скупом података из кога су искључена висококорелисана обележја (вертикални узорак) постиже исту прецизност као онај направљен над целим скупом, док искључивање некорелисаних обележја смањује прецизност.*

Ово је проверено у скрипту 7, над целокупним скупом и сва три плана простог случајног узорка. Искључени су други, четврти, пети и шести атрибут, пошто се показало да су у тесној вези (коефицијент корелације већи од 0,75) са првим и трећим обележјем, тј. да се могу преко њих израчунати, те да не доприносе варијабилности. Модел над целим скупом постигао је прецизност 94,89 % (упоредити са 96,24 % пре редукције). Модел над простим случајним узорком без понављања погодио је 85,785 % слова (упоредити са 87,68 %). Модел над простим случајним узорком са понављањем био је успешан у 86,725 % случајева (упоредити са 88,29 % пре редукције), док је над редукованим узорком прецизност била 86,185 % (упоредити са полазних 88,11 %).

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# odredjivanje korelacija
kor <- cor(x)

# izbacivanje visokokorelisanih
viskor <- c(2, 4, 5, 6)
x <- subset(x,
            select = -viskor)

# MODEL NAD CELOKUPNIM SKUPOM PODATAKA;
# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# pravljenje SVM modela
model <- svm(x, y)
summary(model)

# provera kvaliteta modela
pred <- fitted(model)
prec <- mean(y == pred) # 0.9489

# MODEL NAD PSU BEZ PONAVLJANJA;
# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# ranije odredjen obim uzorka
N <- 20000
n <- 3199

# uzorkovanje prema izracunatom
indeksi <- sample(N, n)

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.85785
```

```
50
51 # MODEL NAD PSU SA PONAVLJANJEM;
52 # fiksiranje generatora pseudoslucajnosti
53 set.seed(0)
54
55 # ranije odredjen obim uzorka
56 n <- 3808
57
58 # uzorkovanje prema izracunatom
59 indeksi <- sample(N, n,
60                   replace = T)
61
62 # pravljenje SVM modela
63 model <- svm(x[indeksi,],
64              y[indeksi],
65              fitted = F)
66 summary(model)
67
68 # provera kvaliteta modela
69 pred <- predict(model, x)
70 prec <- mean(y == pred) # 0.86725
71
72 # iskljucivanje ponovljenih entiteta
73 indeksi <- unique(indeksi)
74 n <- length(indeksi) # 3434
75
76 # pravljenje SVM modela
77 model <- svm(x[indeksi,],
78              y[indeksi],
79              fitted = F)
80 summary(model)
81
82 # provera kvaliteta modela
83 pred <- predict(model, x)
84 prec <- mean(y == pred) # 0.86185
```

Скрипт 7: corr.r – модел над редукованим подацима

Свеукупно гледано, прецизност је за све реализоване планове узорковања опала за мање од 2 %, што је углавном занемарљиво мало – уколико је прихватљиво 87 % погодака, прихватљиво је у 85 %. Притом је димензија проблема осетно смањена – 16 улазних атрибута сведено је на 12 (уштеда од једне четвртине) – што двоструко убрзава алгоритам класификације. Остаје испитивање другог дела хипотезе.

Искључивање нискокорелисаних атрибута, које није могуће исправно израчунати преко осталих (аналогно примени количничке или регресионе оцене на пар обележја са коефицијентом корелације око 0,3 или мањом), урађено је у скрипту 8. Очекивано, губитак у прецизности је велики – око 11 % – што није прихватљива цена редукције.

```
1  # ucitavanje biblioteke i podataka
2  library(e1071)
3  slova <- read.csv('slova.csv')
4
5  # podela skupa na dva dela
6  x <- subset(slova,
7              select = -slovo)
8  y <- as.factor(slova$slovo)
9
10 # odredjivanje korelacija
11 kor <- cor(x)
```

```
12
13 # izbacivanje niskokorelisanih
14 niskor <- c(8, 9, 12, 16)
15 x <- subset(x,
16              select = -niskor)
17
18 # MODEL NAD CELOKUPNIM SKUPOM PODATAKA;
19 # fiksiranje generatora pseudoslucajnosti
20 set.seed(0)
21
22 # pravljenje SVM modela
23 model <- svm(x, y)
24 summary(model)
25
26 # provera kvaliteta modela
27 pred <- fitted(model)
28 prec <- mean(y == pred) # 0.8565
29
30 # MODEL NAD PSU BEZ PONAVLJANJA;
31 # fiksiranje generatora pseudoslucajnosti
32 set.seed(0)
33
34 # ranije odredjen obim uzorka
35 N <- 20000
36 n <- 3199
37
38 # uzorkovanje prema izracunatom
39 indeksi <- sample(N, n)
40
41 # pravljenje SVM modela
42 model <- svm(x[indeksi,],
43              y[indeksi],
44              fitted = F)
45 summary(model)
46
47 # provera kvaliteta modela
48 pred <- predict(model, x)
49 prec <- mean(y == pred) # 0.74635
50
51 # MODEL NAD PSU SA PONAVLJANJEM;
52 # fiksiranje generatora pseudoslucajnosti
53 set.seed(0)
54
55 # ranije odredjen obim uzorka
56 n <- 3808
57
58 # uzorkovanje prema izracunatom
59 indeksi <- sample(N, n,
60                   replace = T)
61
62 # pravljenje SVM modela
63 model <- svm(x[indeksi,],
64              y[indeksi],
65              fitted = F)
66 summary(model)
67
68 # provera kvaliteta modela
69 pred <- predict(model, x)
70 prec <- mean(y == pred) # 0.7544
71
72 # iskljucivanje ponovljenih entiteta
73 indeksi <- unique(indeksi)
```

```r
n <- length(indeksi) # 3434

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.7528
```

Скрипт 8: corrlos.r – модел над редукованим подацима

На основу виђеног, могуће је закључити да је полазна претпоставка тачна – озбиљно класификациони модел направљен над скупом података из кога су искључена висококорелисана обележја (вертикални узорак) постиже сличан успех као онај направљен над целим скупом, док искључивање некорелисаних обележја смањује прецизност – уз напомену да то важи уз одређени степен грешке. Даље истраживање могло би бити усмерено на прецизно оцењивање тог одспутања, што због обима није урађено у овом раду. Грешка од 2 %, примећена на обрађиваном скупу слова, није велика и има смисла да би и на осталим скуповима била подједнако прихватљива и релативно занемарљива.

3.4 Стратификован (раслојен) узорак

Стратификацијом (раслојавањем) популације на основу вредности помоћног обележја за које се сматра да је у вези са атрибутом од значаја добија се скуп међусобно дисјунктних потпопулација (слојева, стратума). Након тога се из сваког слоја могу бирати узорци унапред одређеног обима. Притом су одабири из различитих стратума независни и не нужно начињени према истом плану узорковања. Резултат описаног поступка јесте стратификован (раслојен) узорак.[12]

Под претпоставком да се подузорак из сваког слоја извлачи вероватносним методом, непристрасна тачкаста оцена популацијског тотала једнака је збиру оцена за сваки стратум, са дисперзијом једнаком збиру појединачних дисперзија (последица независности). Закључивање о средњој вредности и интервалима поверења аналогно је, као код других планова, сагласно са везом $m = \frac{\tau}{N}$ и распоном $\overline{Y} \pm z_{1-\frac{\alpha}{2}}\sqrt{\hat{\sigma}^2}$.

Овакав план узорковања, у случају стварне корелације посматраног и помоћног обележја, често смањује вероватноћу одабира нерепрезентативних узорака, што је предност у свакој примени. Оцене непознатих параметара тада су прецизније, а омогућено је и закључивање о појединачним слојевима. Некада је чак и једноставније или јефтиније узорковање по слојевима него на целој популацији одједном. Поред корелације обележја, пожељне особине стратума су релативна хомогеност унутар слојева и релативна хетерогеност између слојева. То се обично подразумева када је подела на стратуме природна.

Поставља се питање како дефинисати слојеве и одредити њихов број, а одговор у суштини зависи од проблема тј. скупа података који се проучава. На примеру обрађиваних слова, како је четрнаести атрибут обележје од значаја и како је он одређен као предиктор високе важности, природно је стратификовати према типу слова које ентитет описује. Тиме је прецизно одређено 26 независних потпопулација.

Претпоставка 3 *Узорак раслојен према типу слова боље оцењује средњу вредност четрнаестог атрибута до простог случајног узорка. Додатно, класификациони модел направљен над тим узорком постиже већу прецизност од оног формираног над простим случајним узорком.*

Други важан проблем јесте одређивање расподеле узорка по појединачним стратумима. Најпростији одговор лежи у пропорционалном распореду – обим подузорка је сразмеран обиму слоја. Нешто сложенији одговори испуњавају додатан захтев за смањеном дисперзијом, а међу њима је Нејманов оптимални распоред, који предлаже обиме $n_h = n\frac{N_h\sigma_h}{\sum_{l\in H} N_l\sigma_l}$ за сваки слој h, где је H скуп свих стратума l. Очигледно, идеја је да више ентитета потиче из већих и/или распршенијих потпопулација, чиме је боље очувана варијабилност полазног скупа.

Напослетку, остаје одабир обима узорка n таквог да минимизује дисперзију оцене, нпр. кад се процењује средња вредност. Наивна идеја је да се користи исти обим као код нпр. простог случајног узорка или ког већ плана који се користи за одабир подузорка унутар стратума. Боља замисао узима у обзир специфичне варијансе сваког појединачног стратума. Тако се као обим узорка може одабрати $n = \upsilon(\frac{z_{1-\frac{\alpha}{2}}}{\Delta})^2$, при чему је оцена $\upsilon = \frac{1}{N}\sum_{h\in H} N_h\sigma_h^2$, док су остале ознаке познате.

Стратификовано узорковање са пропорционалним распоредом код кога се подузорци из сваког стратума извлаче простим случајним узорковањем без понављања одрађено је у скрипту 9. Као и код простог случајног узорка, дозвољена је конзервативна апсолутна грешка, уз висок ниво поверења. Оцена средње вредности обележја од значаја маши популацијску вредност за величину реда 10^{-4}, док јој је дисперзија истог реда, троструко мања него раније. Ширина добијеног интервала поверења је реда 10^{-2}, двоструко мања. На крају, над овим узорком од 3632 инстанце (свака шеста) направљен је класификациони модел који погађа добрих 87,94 % слова, слично као без раслојавања. Подједнако добре оцене добијају се и за дупло мањи узорак, али је за потребе прављења квалитетног класификационог модела узет већи.

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
N <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# raslojavanje prema tipu slova;
# funkcionalni koncept: lapply od niza svih slova
# pravi listu indeksa podataka u kojima tip slova y
# odgovara tekucem slovu k, dakle gde je k == y
strat <- lapply(LETTERS,
                function (k) which(k == y))
obelezje_strat <- lapply(strat,
                         function (k) obelezje[k])
broj_strat <- length(strat) # 26
N_strat <- sapply(strat, length)
```

```r
ok <- sum(N_strat) == N # TRUE

# dozvoljena apsolutna greska
d <- 0.07

# dozvoljena greska prve vrste
alpha <- 0.05

# odgovarajuci kvantil N(0,1)
z <- qnorm(1 - alpha/2) # 1.9600

# tacne statistike po slojevima
sr_strat <- sapply(obelezje_strat, mean)
vr_strat <- sapply(obelezje_strat, var)

# neophodan obim uzorka
ups <- 1/N * sum(N_strat * vr_strat) # 2.3161
n <- ups * (z/d)^2 # 1815
n <- 2*ceiling(n) # 3632

# proporcionalni raspored
n_strat <- round(n*N_strat/N)
ok <- sum(n_strat) == n # FALSE

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# popravka da bi bilo ok
while (!ok) {
  if (sum(n_strat) > n) {
    i <- sample(broj_strat, 1)
    n_strat[i] <- n_strat[i] - 1
  } else {
    i <- sample(broj_strat, 1)
    n_strat[i] <- n_strat[i] + 1
  }
  ok <- sum(n_strat) == n
}

# uzorkovanje prema izracunatom
indeksi <- lapply(1:broj_strat,
           function (i) sample(N_strat[i], n_strat[i]))
ok <- all(sapply(indeksi, length) == n_strat) # TRUE
uzorak <- lapply(1:broj_strat,
          function (i) obelezje_strat[[i]][indeksi[[i]]])
ok <- all(sapply(uzorak, length) == n_strat) # TRUE

# uzoracke vrednosti po stratumima
xn_strat <- sapply(uzorak, mean)
sn2_strat <- sapply(uzorak, var)
D_xn_strat <- sn2_strat/n_strat * (1 - n_strat/N_strat)
greska <- abs(sr_strat - xn_strat)

# ocenjivanje srednje vrednosti
xn <- 1/N * sum(N_strat * xn_strat) # 3.0429
D_xn <- 1/N^2 * sum(N_strat^2 * D_xn_strat) # 0.0005

# interval poverenja
sirina <- z * sqrt(D_xn) # 0.0448
I_xn <- c(xn - sirina, # 3.00
          xn + sirina) # 3.09
upada <- sr >= I_xn[1] &&
```

```
          sr <= I_xn[2] # TRUE

# spojeni uzorak
indeksi <- unlist(sapply(1:broj_strat,
                  function (i) strat[[i]][indeksi[[i]]]))
ok <- length(indeksi) == n # TRUE

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.8794
```

Скрипт 9: strworp.r – модел над проп. страт. узорком без пон.

Стратификовано узорковање са пропорционалним распоредом код кога се подузорци из сваког стратума извлаче простим случајним узорковањем са понављањем одрађено је у скрипту 10. Рађено је како над пуним, тако и над редукованим узорком тј. подузорцима. Резултати погађања су нешто лошији него код верзије без понављања, али су дисперзије оцена и даље боље него код простог случајног узорка.

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
N <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# raslojavanje prema tipu slova
strat <- lapply(LETTERS,
                function (k) which(k == y))
obelezje_strat <- lapply(strat,
                         function (k) obelezje[k])
broj_strat <- length(strat) # 26
N_strat <- sapply(strat, length)

# dozvoljena apsolutna greska
d <- 0.07

# dozvoljena greska prve vrste
alpha <- 0.05

# odgovarajuci kvantil N(0,1)
z <- qnorm(1 - alpha/2) # 1.9600

# tacne statistike po slojevima
sr_strat <- sapply(obelezje_strat, mean)
vr_strat <- sapply(obelezje_strat, var)

```

```r
# neophodan obim uzorka
ups <- 1/N * sum(N_strat * vr_strat) # 2.3161
n <- ups * (z/d)^2 # 1815
n <- 2*ceiling(n) # 3632

# proporcionalni raspored
n_strat <- round(n*N_strat/N)

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# popravka da bi bilo ok
while (sum(n_strat) != n) {
  if (sum(n_strat) > n) {
    i <- sample(broj_strat, 1)
    n_strat[i] <- n_strat[i] - 1
  } else {
    i <- sample(broj_strat, 1)
    n_strat[i] <- n_strat[i] + 1
  }
}

# uzorkovanje prema izracunatom;
# OVDE JE REPLACE TRUE, PONAVLJANJE
indeksi <- lapply(1:broj_strat,
                  function (i) sample(N_strat[i],
                                      n_strat[i],
                                      replace = T))
uzorak <- lapply(1:broj_strat,
          function (i) obelezje_strat[[i]][indeksi[[i]]])

# uzoracke vrednosti po stratumima;
# BEZ OTKLONA U DISPERZIJI
xn_strat <- sapply(uzorak, mean)
sn2_strat <- sapply(uzorak, var)
D_xn_strat <- sn2_strat/n_strat
greska <- abs(sr_strat - xn_strat)

# ocenjivanje srednje vrednosti
xn <- 1/N * sum(N_strat * xn_strat) # 3.0297
D_xn <- 1/N^2 * sum(N_strat^2 * D_xn_strat) # 0.0006

# interval poverenja
sirina <- z * sqrt(D_xn) # 0.0487
I_xn <- c(xn - sirina, # 2.98
          xn + sirina) # 3.08
upada <- sr >= I_xn[1] &&
         sr <= I_xn[2] # TRUE

# spojeni uzorak
indeks <- unlist(sapply(1:broj_strat,
                  function (i) strat[[i]][indeksi[[i]]]))

# pravljenje SVM modela
model <- svm(x[indeks,],
             y[indeks],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.88285
```

```
99
100 # iskljucivanje ponovljenih entiteta
101 indeksi <- sapply(indeksi, unique)
102 uzorak <- lapply(1:broj_strat,
103           function (i) obelezje_strat[[i]][indeksi[[i]]])
104 n_strat <- sapply(indeksi, length)
105 n <- sum(n_strat) # 3333
106
107 # uzoracke vrednosti po stratumima
108 xn_strat <- sapply(uzorak, mean)
109 sn2_strat <- sapply(uzorak, var)
110 D_xn_strat <- sapply(1:broj_strat,
111               function (i) sum(sapply(1:(N_strat[i]-1),
112               function (k) (k/N_strat[i])^(n_strat[i]-1)/
113                            N_strat[i])) * sn2_strat[i])
114 greska <- abs(sr_strat - xn_strat)
115
116 # ocenjivanje srednje vrednosti
117 xn <- 1/N * sum(N_strat * xn_strat) # 3.0311
118 D_xn <- 1/N^2 * sum(N_strat^2 * D_xn_strat) # 0.0006
119
120 # spojeni uzorak
121 indeks <- unlist(sapply(1:broj_strat,
122                  function (i) strat[[i]][indeksi[[i]]]))
123
124 # pravljenje SVM modela
125 model <- svm(x[indeks,],
126              y[indeks],
127              fitted = F)
128 summary(model)
129
130 # provera kvaliteta modela
131 pred <- predict(model, x)
132 prec <- mean(y == pred) # 0.87755
```

Скрипт 10: strwrp.r – модел над проп. страт. узорком са пон.

Стратификовано узорковање са оптималним Нејмановим распоредом код кога се подузорци из сваког стратума извлаче простим случајним узорковањем без понављања одрађено је у скрипту 11. Резултати класификације слични су као код претходних приступа, с тим што је овде дисперзија оцене још мања, а интервал поверења ужи.

```
1  # ucitavanje biblioteke i podataka
2  library(e1071)
3  slova <- read.csv('slova.csv')
4
5  # podela skupa na dva dela
6  x <- subset(slova,
7              select = -slovo)
8  y <- as.factor(slova$slovo)
9
10 # izdvajanje obelezja od znacaja
11 obelezje <- x$x_ivice
12 N <- length(obelezje) # 20000
13 sr <- mean(obelezje) # 3.0461
14 vr <- var(obelezje) # 5.4407
15
16 # raslojavanje prema tipu slova
17 strat <- lapply(LETTERS,
18                 function (k) which(k == y))
19 obelezje_strat <- lapply(strat,
```

```
                          function (k) obelezje[k])
broj_strat <- length(strat) # 26
N_strat <- sapply(strat, length)

# dozvoljena apsolutna greska
d <- 0.07

# dozvoljena greska prve vrste
alpha <- 0.05

# odgovarajuci kvantil N(0,1)
z <- qnorm(1 - alpha/2) # 1.9600

# tacne statistike po slojevima
sr_strat <- sapply(obelezje_strat, mean)
vr_strat <- sapply(obelezje_strat, var)
sd_strat <- sapply(vr_strat, sqrt)

# neophodan obim uzorka
ups <- 1/N * sum(N_strat * vr_strat) # 2.3161
n <- ups * (z/d)^2 # 1815
n <- 2*ceiling(n) # 3632

# Nejmanov optimalni raspored
n_strat <- round(n*N_strat*sd_strat/
                   sum(N_strat*sd_strat))

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# popravka da bi bilo ok
while (sum(n_strat) != n) {
  if (sum(n_strat) > n) {
    i <- sample(broj_strat, 1)
    n_strat[i] <- n_strat[i] - 1
  } else {
    i <- sample(broj_strat, 1)
    n_strat[i] <- n_strat[i] + 1
  }
}

# uzorkovanje prema izracunatom
indeksi <- lapply(1:broj_strat,
            function (i) sample(N_strat[i], n_strat[i]))
uzorak <- lapply(1:broj_strat,
           function (i) obelezje_strat[[i]][indeksi[[i]]])

# uzoracke vrednosti po stratumima
xn_strat <- sapply(uzorak, mean)
sn2_strat <- sapply(uzorak, var)
D_xn_strat <- sn2_strat/n_strat * (1 - n_strat/N_strat)
greska <- abs(sr_strat - xn_strat)

# ocenjivanje srednje vrednosti
xn <- 1/N * sum(N_strat * xn_strat) # 3.0212
D_xn <- 1/N^2 * sum(N_strat^2 * D_xn_strat) # 0.0005

# interval poverenja
sirina <- z * sqrt(D_xn) # 0.0432
I_xn <- c(xn - sirina, # 2.98
          xn + sirina) # 3.06
upada <- sr >= I_xn[1] &&
```

```
          sr <= I_xn[2] # TRUE

# spojeni uzorak
indeksi <- unlist(sapply(1:broj_strat,
                  function (i) strat[[i]][indeksi[[i]]]))

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.87035
```

Скрипт 11: strworn.r – модел над опт. страт. узорком без пон.

Стратификовано узорковање са оптималним Нејмановим распоредом код кога се подузорци из сваког стратума извлаче простим случајним узорковањем са понављањем одрађено је у скрипту 12. И овде су резултати класификације слични као код претходних приступа, са нижом дисперзијом оцене и ужим интервалом пореверења процене.

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
N <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# raslojavanje prema tipu slova
strat <- lapply(LETTERS,
                function (k) which(k == y))
obelezje_strat <- lapply(strat,
                         function (k) obelezje[k])
broj_strat <- length(strat) # 26
N_strat <- sapply(strat, length)

# dozvoljena apsolutna greska
d <- 0.07

# dozvoljena greska prve vrste
alpha <- 0.05

# odgovarajuci kvantil N(0,1)
z <- qnorm(1 - alpha/2) # 1.9600

# tacne statistike po slojevima
sr_strat <- sapply(obelezje_strat, mean)
vr_strat <- sapply(obelezje_strat, var)
sd_strat <- sapply(vr_strat, sqrt)

# neophodan obim uzorka
```

```r
ups <- 1/N * sum(N_strat * vr_strat) # 2.3161
n <- ups * (z/d)^2 # 1815
n <- 2*ceiling(n) # 3632

# Nejmanov optimalni raspored
n_strat <- round(n*N_strat*sd_strat/
                   sum(N_strat*sd_strat))

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# popravka da bi bilo ok
while (sum(n_strat) != n) {
  if (sum(n_strat) > n) {
    i <- sample(broj_strat, 1)
    n_strat[i] <- n_strat[i] - 1
  } else {
    i <- sample(broj_strat, 1)
    n_strat[i] <- n_strat[i] + 1
  }
}

# uzorkovanje prema izracunatom;
# OVDE JE REPLACE TRUE, PONAVLJANJE
indeksi <- lapply(1:broj_strat,
                  function (i) sample(N_strat[i],
                                      n_strat[i],
                                      replace = T))
uzorak <- lapply(1:broj_strat,
          function (i) obelezje_strat[[i]][indeksi[[i]]])

# uzoracke vrednosti po stratumima;
# BEZ OTKLONA U DISPERZIJI
xn_strat <- sapply(uzorak, mean)
sn2_strat <- sapply(uzorak, var)
D_xn_strat <- sn2_strat/n_strat
greska <- abs(sr_strat - xn_strat)

# ocenjivanje srednje vrednosti
xn <- 1/N * sum(N_strat * xn_strat) # 3.0267
D_xn <- 1/N^2 * sum(N_strat^2 * D_xn_strat) # 0.0006

# interval poverenja
sirina <- z * sqrt(D_xn) # 0.0475
I_xn <- c(xn - sirina, # 2.98
          xn + sirina) # 3.07
upada <- sr >= I_xn[1] &&
         sr <= I_xn[2] # TRUE

# spojeni uzorak
indeks <- unlist(sapply(1:broj_strat,
                  function (i) strat[[i]][indeksi[[i]]]))

# pravljenje SVM modela
model <- svm(x[indeks,],
             y[indeks],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.873
```

```
101
102 # iskljucivanje ponovljenih entiteta
103 indeksi <- sapply(indeksi, unique)
104 uzorak <- lapply(1:broj_strat,
105           function (i) obelezje_strat[[i]][indeksi[[i]]])
106 n_strat <- sapply(indeksi, length)
107 n <- sum(n_strat) # 3305
108
109 # uzoracke vrednosti po stratumima
110 xn_strat <- sapply(uzorak, mean)
111 sn2_strat <- sapply(uzorak, var)
112 D_xn_strat <- sapply(1:broj_strat,
113               function (i) sum(sapply(1:(N_strat[i]-1),
114               function (k) (k/N_strat[i])^(n_strat[i]-1)/
115                            N_strat[i])) * sn2_strat[i])
116 greska <- abs(sr_strat - xn_strat)
117
118 # ocenjivanje srednje vrednosti
119 xn <- 1/N * sum(N_strat * xn_strat) # 3.0313
120 D_xn <- 1/N^2 * sum(N_strat^2 * D_xn_strat) # 0.0006
121
122 # spojeni uzorak
123 indeks <- unlist(sapply(1:broj_strat,
124                  function (i) strat[[i]][indeksi[[i]]]))
125
126 # pravljenje SVM modela
127 model <- svm(x[indeks,],
128              y[indeks],
129              fitted = F)
130 summary(model)
131
132 # provera kvaliteta modela
133 pred <- predict(model, x)
134 prec <- mean(y == pred) # 0.86905
```

Скрипт 12: strwrn.r – модел над опт. страт. узорком са пон.

Судећи по добијеним резултатима, истинит је део хипотезе да раслојен узорак боље оцењује средњу вредност обележја од значаја од простог случајног узорка. Ипак, резултати класификације нису се показали као бољи, већ као једнаки. Ово се може схватити као последица у уводу изнесеног запажања да су јединке скупа релативно униформно расподељене по типу слова које представљају, на основу чега је очекивана сличност стратификованог и простог случајног узорка. Да је ипак неопходно пазити на добар удео категорија у подацима на основу којих се прави класификациони модел,[13] показано је лошом стратификацијом одрађеном у скрипту 13. Распоред је узет из ничим оправдане експоненцијалне распоределе, што је резултовало не само лошијом прецизношћу погодака од 74,095 %, већ и дупло већом дисперзијом оцене средње вредности, те ширим интервалом поверења. Једноставности ради, разматран је само подузорак без понављања.

```
1 # ucitavanje biblioteke i podataka
2 library(e1071)
3 slova <- read.csv('slova.csv')
4
5 # podela skupa na dva dela
6 x <- subset(slova,
7             select = -slovo)
8 y <- as.factor(slova$slovo)
9
```

```
# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
N <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# raslojavanje prema tipu slova
strat <- lapply(LETTERS,
                function (k) which(k == y))
obelezje_strat <- lapply(strat,
                         function (k) obelezje[k])
broj_strat <- length(strat) # 26
N_strat <- sapply(strat, length)

# dozvoljena apsolutna greska
d <- 0.07

# dozvoljena greska prve vrste
alpha <- 0.05

# odgovarajuci kvantil N(0,1)
z <- qnorm(1 - alpha/2) # 1.9600

# tacne statistike po slojevima
sr_strat <- sapply(obelezje_strat, mean)
vr_strat <- sapply(obelezje_strat, var)
sd_strat <- sapply(vr_strat, sqrt)

# neophodan obim uzorka
ups <- 1/N * sum(N_strat * vr_strat) # 2.3161
n <- ups * (z/d)^2 # 1815
n <- 2*ceiling(n) # 3632

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# namerno nepravilan raspored
n_strat <- rexp(broj_strat)
n_strat <- n/sum(n_strat) * n_strat
n_strat <- round(n_strat)

# popravka da bi bilo ok
while (sum(n_strat) != n) {
  if (sum(n_strat) > n) {
    i <- sample(broj_strat, 1)
    n_strat[i] <- n_strat[i] - 1
  } else {
    i <- sample(broj_strat, 1)
    n_strat[i] <- n_strat[i] + 1
  }
}

# uzorkovanje prema izracunatom
indeksi <- lapply(1:broj_strat,
           function (i) sample(N_strat[i], n_strat[i]))
uzorak <- lapply(1:broj_strat,
          function (i) obelezje_strat[[i]][indeksi[[i]]])

# uzoracke vrednosti po stratumima
xn_strat <- sapply(uzorak, mean)
sn2_strat <- sapply(uzorak, var)
D_xn_strat <- sn2_strat/n_strat * (1 - n_strat/N_strat)
```

```r
greska <- abs(sr_strat - xn_strat)

# ocenjivanje srednje vrednosti
xn <- 1/N * sum(N_strat * xn_strat) # 3.0202
D_xn <- 1/N^2 * sum(N_strat^2 * D_xn_strat) # 0.0013

# interval poverenja
sirina <- z * sqrt(D_xn) # 0.0717
I_xn <- c(xn - sirina, # 2.95
          xn + sirina) # 3.09
upada <- sr >= I_xn[1] &&
         sr <= I_xn[2] # TRUE

# spojeni uzorak
indeksi <- unlist(sapply(1:broj_strat,
                  function (i) strat[[i]][indeksi[[i]]]))

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.74095
```

Скрипт 13: strlos.r – модел над лошим страт. узорком

За крај, ваља напоменути да некада није згодно извршити раслојавање пре одабира узорка. Као замена, то се може урадити по узорковању – постстратификација.[14] Идеја је слична, с тим што се јединица класификује у један од постстратума тек по одабиру у узорак. Овде су бројеви n_h заправо случајне величине. Расподела узорка по постстратумима, очекивано, има тенденцију да апроксимира пропорционални распоред код стратификованог узорка. Мада постстратификација нема посебну примену у класификацији, ипак је одрађена у скрипту 14 зарад поређења оцене средње вредности интересног атрибута. Једноставности ради, и овде је разматран само прост случајан подузорак без понављања. Међутим, како је расподела типа слова на популацији и узорку униформна, нису добијена посебна побољшања – дисперзија оцене једнака је дисперзији оцене без постстратификације.

```r
# ucitavanje podataka
slova <- read.csv('slova.csv')

# izdvajanje obelezja od znacaja
obelezje <- slova$x_ivice
N <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# ranije odredjen obim uzorka
n <- 3199

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# uzorkovanje prema izracunatom
indeksi <- sample(N, n)
uzorak <- obelezje[indeksi]
```

```r
tip <- slova$slovo[indeksi]

# raslojavanje prema tipu slova
post <- lapply(LETTERS,
               function (k) which(tip == k))
n_post <- sapply(post, length)
broj_post <- length(post) # 26
ok <- sum(n_post) == n # TRUE

# raspodela populacije po stratumima
N_post <- sapply(LETTERS,
                 function (k) sum(slova$slovo == k))
ok <- sum(N_post) == N # TRUE

# vrednost obelezja na slojevima
obelezje_post <- lapply(post,
                        function (k) obelezje[k])
xn_post <- sapply(obelezje_post, mean)
sn2_post <- sapply(obelezje_post, var)

# ocenjivanje srednje vrednosti
xn <- 1/N * sum(N_post * xn_post) # 2.9877
D_xn <- 1/(n*N) * (1 - n/N) * sum(N_post * sn2_post) +
        1/n^2 * (1 - (n-1)/(N-1)) *
        sum((1 - N_post/N) * sn2_post) # 0.0014

# interval poverenja
alpha <- 0.05
z <- qnorm(1 - alpha/2) # 1.9600
sirina <- z * sqrt(D_xn) # 0.0737
I_xn <- c(xn - sirina, # 2.91
          xn + sirina) # 3.06
upada <- sr >= I_xn[1] &&
         sr <= I_xn[2] # TRUE
```

Скрипт 14: post.r – оцена уз постстратификацију

3.5 Групни (кластер) узорак

Одабир појединачних јединки у узорак из целе популације или стратума често није погодан за велика и сложена истраживања. Ту на сцену ступа (једноетапни) групни узорак, код кога су јединице узорковања тзв. примарне јединице – групе, скупине, серије, кластери ентитета – док су јединице посматрања тзв. секундарне јединице – сами ентитети.[14] Након поделе на дисјунктне групе, одабира се извесни број скупина и све инстанце из њих се укључују, односно не укључују у узорак (ова два одабира, иако супротна, међусобно су еквивалентна).

Овакав план узорковања је значајан када из неког разлога није могуће сагледати читаву популацију, нпр. због њеног великог обима. С друге стране, често групе природно већ постоје, па је лакше узети цео кластер у узорак него извлачити јединке простим случајним узорком. За разлику од раслојеног узорка, пожељне особине кластера су релативна хетерогеност унутар група и релативна хомогеност између група. Очекивано, пошто се целе групе прихватају тј. одбацују, неопходно је да не постоје неке које се значајно разликују од других и чије би искључење резултовало губитком значајне информације о популацији. Такође, већа различитост унутар група чини их репрезентативнијим. Међутим, природне групе имају тенденцију да буду сличног

састава и да се разликују од других група (као стратуми), што условљава смањење тачности оцена непознатих популацијских вредности. Групни узорак је, дакле, јефтинији, но углавном мање прецизан.

Претпоставка 4 *Класификациони модел направљен над групним узорком, таквим да су групе међусобно сличне (а унутар себе различите, као популације у малом), једнако је прецизан као онај направљен над простим случајним узорком. Резултујуће оцене популацијске средње вредности слабије су постојане за сличан број инстанци у узорку.*

Скупине се могу извлачити било којим планом узорковања. Најједноставнији приступ подразумева одабир кластера у виду простог случајног узорка без понављања. Тада се као непристрасна тачкаста оцена популацијске средње вредности интересног обележја Y може искористити $\hat{m}_y^{clu} = \frac{N}{nM}\sum_{l\in S}\tau_l$, при чему је N сада укупан број група, n број група у узорку, M величина популације, а S скуп група. Дисперзија ове оцене једнака је $\frac{N^2}{M^2}\frac{\sigma_\tau^2}{n}(1-\frac{n}{N})$, где је σ_τ^2 варијанса тотала по групама. Могуће је и количнички оцењивати вредности када се величина скупине може схватити као помоћно обележје, и тада је оцена популацијске средње вредности $\hat{m}_y^{Rclu} = \frac{\sum_{l\in S}\tau_l}{\sum_{l\in S}M_l}$, где је M_i величина групе, са дисперзијом попут претходне, с тим што се уместо σ_τ^2 узима $\frac{1}{N-1}\sum_1^N(\tau_i - m_y M_i)^2$, односно процена према узорку. Може се радити и са узорком са неједнаким вероватноћама, сразмерним величини групе, при чему важе већ познате оцене попут Хансен-Хурвицове и Хорвиц-Томпсонове. Међутим, то је једноставно само код извлачења са понављањем, код кога се лако рачунају вероватноће укључења.

Кластер узорак заснован на извлачењу скупина без понављања формиран је у скрипту 15. Групно узорковање у машинском учењу има највише смисла када је доступно више скупова за тренинг, па се уместо спајања свих одаберу само одређени удео њих. Зато је овде улазни скуп подељен на више група, и то редом, по индексима, што је најјефтинија подела која чува варијабилност, у складу са претпоставкама кластер узорка. У наставку је сам скрипт, док су резултати, компактности ради, овога пута прокоментарисани иза кода.

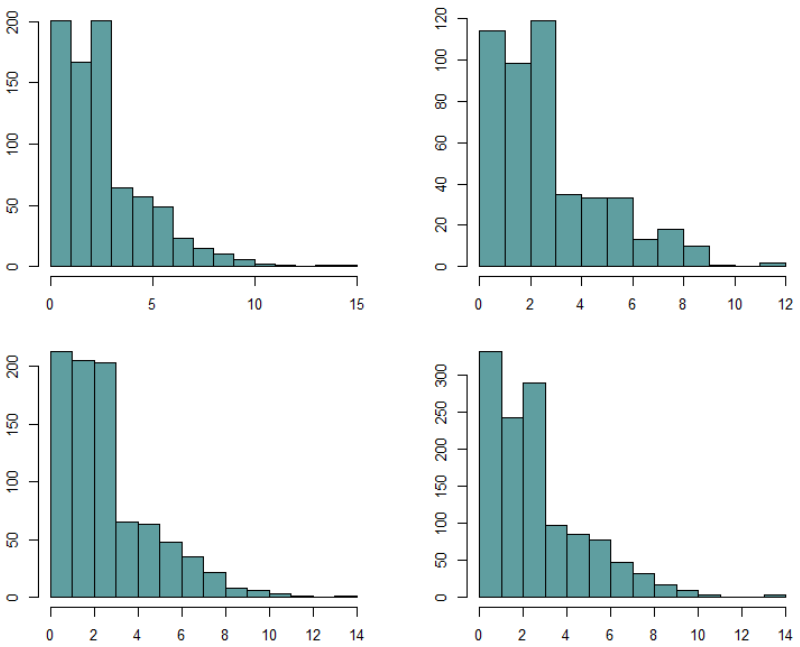
```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
M <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# odredjivanje velicina klastera
M_klast <- c()
while (sum(M_klast) < M) {
  nov <- runif(1, 300, 1300)
```

```r
  nov <- round(nov)
  M_klast <- c(M_klast, nov)
}
N <- length(M_klast) # 23

# popravka da bi bilo ok
while (sum(M_klast) != M) {
  if (sum(M_klast) > M) {
    i <- sample(N, 1)
    M_klast[i] <- M_klast[i] - 1
  } else {
    i <- sample(N, 1)
    M_klast[i] <- M_klast[i] + 1
  }
}

# klasterovanje prema velicini;
# pomoc su kumulativni indeksi
kumul <- c(0, cumsum(M_klast))
klast <- lapply(2:length(kumul),
                function (i) (kumul[i-1]+1):kumul[i])
obelezje_klast <- lapply(klast,
                         function (k) obelezje[k])

# velicina uzorka kako bi broj
# jedinki sto slicniji kao psu
n <- 3199 # zeljen br. jedinki
n <- round(n/(M/N)) # 4 grupe

# uzorkovanje prema izracunatom
ind_klast <- sample(N, n)
indeksi <- lapply(ind_klast,
                  function (i) klast[[i]])
uzorak <- lapply(ind_klast,
                 function (i) obelezje_klast[[i]])

# histogrami uzorkovanih vrednosti
layout(matrix(1:n, n/2, n/2))
lapply(uzorak,
       function (k) hist(k, main = '',
                         xlab = '', ylab = '',
                         col = 'cadetblue'))

# uzoracke vrednosti po skupinama
Mi_klast <- M_klast[ind_klast]
tau_klast <- sapply(uzorak, sum)
sn2_tau <- var(tau_klast) # 885813.6667

# klasicna ocena srednje vrednosti
xn_psu <- N/M * mean(tau_klast) # 2.9388
D_xn_psu <- (N/M)^2 * sn2_tau/n * (1 - n/N) # 0.2419

# kolicnicka ocena srednje vrednosti
xn_kol <- sum(tau_klast)/sum(Mi_klast) # 3.0162
sn2_kol <- 1/(n-1) * sum((tau_klast - xn_kol *
                          Mi_klast)^2) # 1227.1843
D_xn_kol <- (N/M)^2 * sn2_kol/n * (1 - n/N) # 0.0003

# spojeni uzorak
indeksi <- unlist(indeksi)
m <- length(indeksi) # 3389

```

```r
# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.87385
```

Скрипт 15: sscswor.r – модел над групним узорком без пон.

Дакле, прво су одабране величине скупина из одређене равномерне расподеле. Затим су саме групе издвојене редом, према израчунатим величинама. Још је израчунат и обим узорка примарних јединица, такав да број секундарних јединица у узорку одговара раније одређеној вредности. Након тога је извучен узорак. Да је груписање било успешно, сведоче и хистограми по кластерима, приказани на слици 8. Приметно је да су све четири одабране скупине сличне, док је унутар сваке очувана варијабилност. Прецизност класификације је 87,385 %, врло слично као код простог случајног или раслојеног узорка, што је добра вест када се узме у обзир да је овај план јефтинији од претходних, а иде и у прилог изложеној хипотези. Међутим, дисперзија предложене непристране оцене популацијске средње вредности знатно је већа, док је количничка оцена врло прецизна, али пристрасна.



Слика 8: Хистограми одабраних скупина

Кластер узорак заснован на извлачењу скупина са понављањем формиран је у скрипту 16. За разлику од верзије без понављања, где

су групе узорковане простим случајним узорковањем, овде је то учињено узорковањем са неједнаким вероватноћама, сразмерним величини скупине. Прецизност класификације је 89,68 %, практично најбоља досад, док су дисперзије оцена упоредиве са ранијим покушајима, уз напомену да се Хансен-Хурвицова оцена показала као врло прецизна, што не иде у прилог претпоставци, али би се могло објаснити тиме да су групе одлично одабране, што и приказују приложени хистограми.

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
M <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# odredjivanje velicina klastera
M_klast <- c()
while (sum(M_klast) < M) {
  nov <- runif(1, 300, 1300)
  nov <- round(nov)
  M_klast <- c(M_klast, nov)
}
N <- length(M_klast) # 23

# popravka da bi bilo ok
while (sum(M_klast) != M) {
  if (sum(M_klast) > M) {
    i <- sample(N, 1)
    M_klast[i] <- M_klast[i] - 1
  } else {
    i <- sample(N, 1)
    M_klast[i] <- M_klast[i] + 1
  }
}

# klasterovanje prema velicini;
# pomoc su kumulativni indeksi
kumul <- c(0, cumsum(M_klast))
klast <- lapply(2:length(kumul),
                function (i) (kumul[i-1]+1):kumul[i])
obelezje_klast <- lapply(klast,
                         function (k) obelezje[k])

# velicina uzorka kako bi broj
# jedinki sto slicniji kao psu
n <- 3808 # zeljen br. jedinki
n <- round(n/(M/N)) # 4 grupe

# nejednake verovatnoce izvlacenja
# srazmerne velicini skupine
psi <- M_klast/M
```

```
55
56 # uzorkovanje prema izracunatom
57 ind_klast <- sample(N, n,
58                     prob = psi,
59                     replace = T)
60 indeksi <- lapply(ind_klast,
61                   function (i) klast[[i]])
62 uzorak <- lapply(ind_klast,
63                  function (i) obelezje_klast[[i]])
64
65 # uzoracke vrednosti po skupinama
66 Mi_klast <- M_klast[ind_klast]
67 tau_klast <- sapply(uzorak, sum)
68
69 # Hansen-Hurwitzova ocena srednje vrednosti
70 xn_hh <- mean(tau_klast/Mi_klast) # 3.0450
71
72 # ocena disperzije ovakve HH ocene
73 D_xn_hh <- 1/(n-1) *
74            mean((tau_klast/Mi_klast -
75                  xn_hh)^2) # 0.0008
76
77 # verovatnoce ukljucenja prvog reda
78 pi <- 1 - (1 - psi)^n
79 pi_klast <- pi[ind_klast]
80
81 # Horvitz-Thompsonova ocena srednje vrednosti;
82 # nema ponavljanja, pa ni potrebe za redukcijom
83 xn_ht <- 1/M * sum(tau_klast/pi_klast) # 3.2835
84
85 # ocena disperzije ovakve HT ocene
86 psi_klast <- psi[ind_klast]
87 D_xn_ht <- sum((1/pi_klast^2 - 1/pi_klast) * tau_klast^2)
88 for (k in 1:n) {
89   for (l in 1:n) {
90     if (k != l) {
91       # verovatnoca ukljucenja drugog reda
92       pi_kl <- pi_klast[k] + pi_klast[l] - 1 +
93                (1 - psi_klast[k] - psi_klast[l])^n
94       D_xn_ht <- D_xn_ht +
95                (1/(pi_klast[k]*pi_klast[l]) - 1/pi_kl) *
96                tau_klast[k] * tau_klast[l]
97     }
98   }
99 }
100 D_xn_ht <- 1/M^2 * D_xn_ht # 0.0113
101
102 # spojeni uzorak
103 indeksi <- unlist(indeksi)
104 m <- length(indeksi) # 4002
105
106 # pravljenje SVM modela
107 model <- svm(x[indeksi,],
108              y[indeksi],
109              fitted = F)
110 summary(model)
111
112 # provera kvaliteta modela
113 pred <- predict(model, x)
114 prec <- mean(y == pred) # 0.8968
```

Скрипт 16: sscswr.r – модел над групним узорком са пон.

По обичају, на крају је изложено шта се дешава када је лоше примењен текући план узорковања. Тако је кластер узорак – прост случајан без понављања и са неједнаким вероватноћама и понављањем – такав да групе не испуњавају претпоставке метода, формиран у скрипту 17.

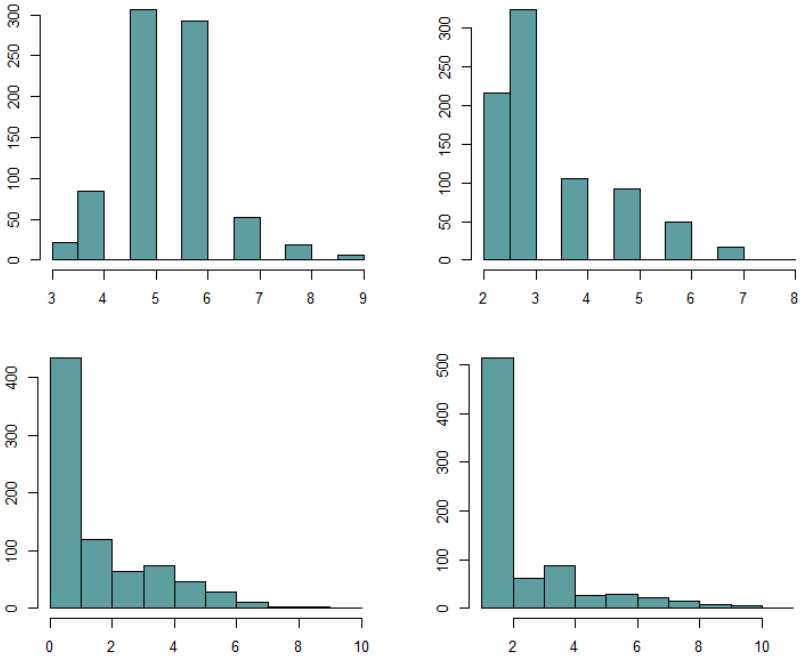
```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
M <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# klasterovanje prema tipu slova
klast <- lapply(LETTERS,
                function (k) which(k == y))
obelezje_klast <- lapply(klast,
                         function (k) obelezje[k])
M_klast <- sapply(klast, length)
N <- length(M_klast) # 26

# IZVLACENJE BEZ PONAVLJANJA;
# velicina uzorka kako bi broj
# jedinki sto slicniji kao psu
n <- 3199 # zeljen br. jedinki
n <- round(n/(M/N)) # 4 grupe

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# uzorkovanje prema izracunatom
ind_klast <- sample(N, n)
indeksi <- lapply(ind_klast,
                  function (i) klast[[i]])
uzorak <- lapply(ind_klast,
                 function (i) obelezje_klast[[i]])

# histogrami uzorkovanih vrednosti
layout(matrix(1:n, n/2, n/2))
lapply(uzorak,
       function (k) hist(k, main = '',
                         xlab = '', ylab = '',
                         col = 'cadetblue'))

# uzoracke vrednosti po skupinama
Mi_klast <- M_klast[ind_klast]
tau_klast <- sapply(uzorak, sum)
sn2_tau <- var(tau_klast) # 1254254

# klasicna ocena srednje vrednosti
xn_psu <- N/M * mean(tau_klast) # 3.523
D_xn_psu <- (N/M)^2 * sn2_tau/n * (1 - n/N) # 0.4484

# kolicnicka ocena srednje vrednosti
xn_kol <- sum(tau_klast)/sum(Mi_klast) # 3.4446
```

```r
sn2_kol <- 1/(n-1) * sum((tau_klast - xn_kol *
                            Mi_klast)^2) # 1251616.4646
D_xn_kol <- (N/M)^2 * sn2_kol/n * (1 - n/N) # 0.4475

# spojeni uzorak
indeksi <- unlist(indeksi)
m <- length(indeksi) # 3147

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.15585

# IZVLACENJE SA PONAVLJANJEM;
# velicina uzorka kako bi broj
# jedinki sto slicniji kao psu
n <- 3808 # zeljen br. jedinki
n <- round(n/(M/N)) # 5 grupa

# nejednake verovatnoce izvlacenja
# srazmerne velicini skupine
psi <- M_klast/M

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# uzorkovanje prema izracunatom
ind_klast <- sample(N, n,
                    prob = psi,
                    replace = T)
indeksi <- lapply(ind_klast,
                  function (i) klast[[i]])
uzorak <- lapply(ind_klast,
                 function (i) obelezje_klast[[i]])

# uzoracke vrednosti po skupinama
Mi_klast <- M_klast[ind_klast]
tau_klast <- sapply(uzorak, sum)

# Hansen-Hurwitzova ocena srednje vrednosti
xn_hh <- mean(tau_klast/Mi_klast) # 2.6710

# ocena disperzije ovakve HH ocene
D_xn_hh <- 1/(n-1) *
  mean((tau_klast/Mi_klast -
          xn_hh)^2) # 0.0799

# spojeni uzorak
indeksi <- unlist(indeksi)
m <- length(indeksi) # 3806

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

```

```r
# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.15335

# iskljucivanje ponovljenih entiteta
ind_klast <- unique(ind_klast)
indeksi <- lapply(ind_klast,
                  function (i) klast[[i]])
uzorak <- lapply(ind_klast,
                 function (i) obelezje_klast[[i]])
Mi_klast <- M_klast[ind_klast]
tau_klast <- sapply(uzorak, sum)

# verovatnoce ukljucenja prvog reda
pi <- 1 - (1 - psi)^n
pi_klast <- pi[ind_klast]

# Horvitz-Thompsonova ocena srednje vrednosti
xn_ht <- 1/M * sum(tau_klast/pi_klast) # 2.4546

# ocena disperzije ovakve HT ocene
nn <- length(ind_klast)
psi_klast <- psi[ind_klast]
D_xn_ht <- sum((1/pi_klast^2 - 1/pi_klast) * tau_klast^2)
for (k in 1:nn) {
  for (l in 1:nn) {
    if (k != l) {
      # verovatnoca ukljucenja drugog reda
      pi_kl <- pi_klast[k] + pi_klast[l] - 1 +
        (1 - psi_klast[k] - psi_klast[l])^n
      D_xn_ht <- D_xn_ht +
        (1/(pi_klast[k]*pi_klast[l]) - 1/pi_kl) *
        tau_klast[k] * tau_klast[l]
    }
  }
}
D_xn_ht <- 1/M^2 * D_xn_ht # 0.3618

# spojeni uzorak
indeksi <- unlist(indeksi)
m <- length(indeksi) # 3070

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.15335
```

Скрипт 17: sscslos.r – модел над лошим групним узорком

Кластеровање је извршено према типу слова, као да је у питању стратификација. То је резултовало групама које, за разлику од претходног приступа, нису довољно сличне, а међусобно су различите, о чему сведоче хистограми на слици 9. Отуда су и процењене дисперзије свих предложених оцена вишеструко веће. Како је критеријум груписања био тип слова, класификациони модел није могао да научи особине више од четири обухваћене групе, тако да је удео погодатака свега 15-16 %, што управо и јесте блиско очекивању $\frac{4}{26} \approx 0,1538$.



Слика 9: Хистограми одабраних скупина

Сумарно, закључак је да је први део хипотезе тачан – стварно нема негативног утицаја јефтинијег групног узорковања на прецизност класификационог модела. С друге стране, други део претпоставке зависи од квалитета кластеровања. Уколико су скупине довољно репрезентативне, као популације у малом, прецизност оцена је слична или боља него код простог случајног узорка. Постојаност је слабија тек када важи супротно. Ово је у складу са уводним теоријским разматрањем.

3.6 Систематски узорак

Систематски узорак је изразито једноставан план узорковања, једноставнији чак и од простог случајног узорка, који у основи има исту структуру као једноетапни групни узорак.[14] Свака примарна јединица састоји се од секундарних јединица које су на известан систематски начин распоређене широм популације. Најједноставнији случај представља прост систематски узорак, такође назван периодични или механички узорак, код кога је свака скупина фактички низ индекса са кораком. Под претпоставком да важи $N = nK$, величина K назива се корак, односно период(а) узорка. Према систематском плану узорковања, бира се тачно једна примарна јединица, па тако он одговара групном узорку обима један, и то на следећи начин – од првих K секундарних јединица одабере се једна као почетна, а затим свака K-та, чиме се добија жељени узорак величине n или евентуално са неким ентитетом мање, уколико није у питању целобројни умножак.

Највећа предност оваквог плана узорковања јесте његова донекле

банална једноставност, као и ниска цена и економичност. Добра особина је и што постоји мањи укупан број узорака него код простог случајног узорка (тачно K), при чему не постоје преклапања секундарних јединица у узорцима, па је тако лако израчунати нпр. вероватноће укључења сваке инстанце. Генерисање (псеудо)случајних бројева неопходно је само једанпут, за одабир почетне инстанце. Потпуна енумерација ентитета у популацији није потребна ни у једном тренутку. Још неке интуитивне предности систематског узорка јесу равномерна расподела на популацији, као и чињеница да не допушта случајна груписања или пропуштање заступљености неких делова популације.

Као тачкаста оцена популацијске средње вредности може се користити узорачка средња вредност. Њена дисперзија, међутим, одређена са $\frac{1}{n^2 K}\sum_{l=1}^{K}(\tau_l - \frac{\tau_y}{K})^2$, није једноставна за процену на основу узорка, што је главно ограничење у примени систематског узорка. Коришћењем класичне формуле за оцену дисперзије код групног узорка обима један добила би се бескорисна процена нула. Срећом, проблем је могуће превазићи добрим познавањем проучаване популације. Под претпоставком да су ентитети случајно распоређени по популацији, односно да не постоји веза између индекса јединке и вредности обележја од значаја на њој, могуће је применити оцену која важи код простог случајног узорка без понављања. Овако је са проучаваним скупом слова, па се могу очекивати исти резултати као на почетку. С друге стране, када подаци показују монотону или периодичну везу са индексима, онда се систематски узорак може показати као знатно бољи или гори.

Претпоставка 5 *Класификациони модел направљен над систематским узорком скупа слова једнаког је успеха као онај направљен над простим случајним узорком, а исто важи и за прецизност оцена.*

Периодични систематски план узорковања примењен је у скрипту 18. Како је и претпостављено, за исти обим узорка, сви резултати су скоро идентични оним код простог случајног узорка – прецизност класификационог модела, вредност и прецизност оцена. Ово је у складу са запажањем из теоријског увода да је систематски узорак над добро промешаној популацији (у смислу расподеле обележја од интереса) заправо јефтинија верзија простог случајног узорка без понављања.

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
N <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# ranije odredjen obim uzorka
n <- 3199

# odredjivanje koraka uzorkovanja
K <- ceiling(N/n) # 7

# fiksiranje generatora pseudoslucajnosti
```

```r
set.seed(0)

# odabir prve sekundarne jedinice
r <- sample(K, 1) # 6

# uzimanje svih jedinki dok je moguce
indeksi <- seq(r, N, K)
ok <- length(indeksi) == n # FALSE

# popravka da bi bilo ok
if (!ok) {
  kraj <- length(indeksi) # 2857
  dopuna <- cumsum(c(indeksi[kraj] + K - N,
                     rep(K, n - kraj - 1)))
  indeksi <- c(indeksi, dopuna)
  ok <- length(indeksi) == n # TRUE
}

# uzorkovanje prema izracunatom
uzorak <- obelezje[indeksi]

# ocenjivanje srednje vrednosti
xn <- mean(uzorak) # 3.0150
sn2 <- var(uzorak) # 5.7021
D_xn <- sn2/n * (1 - n/N) # 0.0015
greska <- abs(sr - xn) # 0.0311

# interval poverenja
alpha <- 0.05
z <- qnorm(1 - alpha/2) # 1.9600
sirina <- z * sqrt(D_xn) # 0.0758
I_xn <- c(xn - sirina, # 2.94
          xn + sirina) # 3.09
upada <- sr >= I_xn[1] &&
         sr <= I_xn[2] # TRUE

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.87615
```

Скрипт 18: sys.r – модел над систематским узорком

3.7 Вишеетапни узорак

Поред једноетапног групног узорка, постоји и вишеетапна варијанта истог плана узорковања.[15] Наиме, једноетапна верзија, иако веома једноставна, може бити непрактична у случају великих група. Ентитети који сачињавају скупине могу бити толико слични да испитивање свих представља разбацивање ресурса. Кластери су као мале популације, па испитивање сваког члана не повећава значајно репрезентативност узорка, а прави додатан трошак који са собом носи регистровање вредности обележја на секундарним јединицама.

Најпростија верзија овог плана јесте двоетапни узорак, код кога се у првој етапи одабере одређени број примарних јединица, а затим

се из сваке одаберe подузорак секундарних јединица. У обе етапе је произвољан метод одабира, с тим што је у другој то најчешће прост случајан узорак без понављања. Ознаке и оцене остају исте као код једноетапног кластеровања, с тим што се додатно уводи n_l као обим подузорка l-тог кластера, који сада није једнак величини целе скупине. Код оцена је још неопходно проценити тотал за сваки извучени кластер, за шта се користе непристрасне оцене метода примењеног у другој фази. Дисперзије су нешто веће него код једноетапне верзије, с тим што се додатни други сабирак често може одбацити као занемарљиво мали у односу на први, који постоји код обе варијанте.

Претпоставка 6 *Класификациони модел направљен над двоетапним групним узорком једнако је прецизан као онај направљен над једноетапним, када су групе изабране на исти начин. Резултујуће оцене популацијске средње вредности сличне су постојаности.*

Двоетапни узорак заснован на извлачењу скупина без понављања формиран је у скрипту 19. Групе су у првој етапи формиране на исти начин као код једноетапне верзије, док је у другој примењен прост случајан узорак без понављања. Запажања су једнака као код верзије са само једном етапом, што иде у прилог изложеној претпоставци.

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
M <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# odredjivanje velicina klastera
M_klast <- c()
while (sum(M_klast) < M) {
  nov <- runif(1, 300, 1300)
  nov <- round(nov)
  M_klast <- c(M_klast, nov)
}
N <- length(M_klast) # 23

# popravka da bi bilo ok
while (sum(M_klast) != M) {
  if (sum(M_klast) > M) {
    i <- sample(N, 1)
    M_klast[i] <- M_klast[i] - 1
  } else {
    i <- sample(N, 1)
    M_klast[i] <- M_klast[i] + 1
  }
}

# klasterovanje prema velicini;
```

```
# pomoc su kumulativni indeksi
kumul <- c(0, cumsum(M_klast))
klast <- lapply(2:length(kumul),
                function (i) (kumul[i-1]+1):kumul[i])
obelezje_klast <- lapply(klast,
                         function (k) obelezje[k])

# velicina uzorka kako bi broj
# jedinki sto slicniji kao psu
n <- 3199 # zeljen br. jedinki
n <- round(n/(M/N)) # 4 grupe

# PRVA ETAPA DVOETAPNOG UZORKA:
# izvlacenje primarnih jedinica
ind_klast <- sample(N, n)
indeksi <- lapply(ind_klast,
                  function (i) klast[[i]])
uzorak <- lapply(ind_klast,
                 function (i) obelezje_klast[[i]])

# uzoracke velicine skupina
Mi_klast <- M_klast[ind_klast]

# DRUGA ETAPA DVOETAPNOG UZORKA:
# izvlacenje sekundarnih jedinica
odabrani <- lapply(1:n,
            function (i) sample(Mi_klast[i],
                                Mi_klast[i]/2))
indeksi <- lapply(1:n,
           function (i) indeksi[[i]][odabrani[[i]]])
uzorak <- lapply(1:n,
          function (i) uzorak[[i]][odabrani[[i]]])
n_klast <- round(Mi_klast/2)

# ocena prema psu bez ponavljanja
tau_klast <- Mi_klast * sapply(uzorak, mean)
sn2_klast <- sapply(uzorak, var)
sn2_tau <- var(tau_klast) # 838969.9985

# dodatna disperzija zbog druge etape
sn2_dod <- N/M^2 * mean(Mi_klast^2 *
                   sn2_klast/n_klast *
              (1 - n_klast/Mi_klast)) # 0.0003

# klasicna ocena srednje vrednosti
xn_psu <- N/M * mean(tau_klast) # 2.9127
D_xn_psu <- (N/M)^2 * sn2_tau/n * (1 - n/N) +
                                  sn2_dod # 0.2294

# kolicnicka ocena srednje vrednosti
xn_kol <- sum(tau_klast)/sum(Mi_klast) # 2.9127
sn2_kol <- 1/(n-1) * sum((tau_klast - xn_kol *
                          Mi_klast)^2) # 989.5052
D_xn_kol <- (N/M)^2 * sn2_kol/n * (1 - n/N) +
                                  sn2_dod # 0.0005

# spojeni uzorak
indeksi <- unlist(indeksi)
m <- length(indeksi) # 1694

# pravljenje SVM modela
model <- svm(x[indeksi,],
```

```
                y[indeksi],
                fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.81655
```

Скрипт 19: mscswor.r – модел над двоетапним узорком без пон.

Двоетапни узорак заснован на извлачењу скупина са понављањем формиран је у скрипту 20. Као код једноетапне варијанте, за разлику од верзије без понављања, где су групе узорковане простим случајним узорковањем, овде је то у првој етапи учињено узорковањем са неједнаким вероватноћама, сразмерним величини скупине, док је у другој такође примењен прост случајан узорак без понављања. И овде су запажања као код верзије са једном етапом, макар по питању класификације, пошто дисперзије због комплексности нису процењиване.

```
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
M <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# odredjivanje velicina klastera
M_klast <- c()
while (sum(M_klast) < M) {
  nov <- runif(1, 300, 1300)
  nov <- round(nov)
  M_klast <- c(M_klast, nov)
}
N <- length(M_klast) # 23

# popravka da bi bilo ok
while (sum(M_klast) != M) {
  if (sum(M_klast) > M) {
    i <- sample(N, 1)
    M_klast[i] <- M_klast[i] - 1
  } else {
    i <- sample(N, 1)
    M_klast[i] <- M_klast[i] + 1
  }
}

# klasterovanje prema velicini;
# pomoc su kumulativni indeksi
kumul <- c(0, cumsum(M_klast))
klast <- lapply(2:length(kumul),
                function (i) (kumul[i-1]+1):kumul[i])
```

```
obelezje_klast <- lapply(klast,
                         function (k) obelezje[k])

# velicina uzorka kako bi broj
# jedinki sto slicniji kao psu
n <- 3808 # zeljen br. jedinki
n <- round(n/(M/N)) # 4 grupe

# nejednake verovatnoce izvlacenja
# srazmerne velicini skupine
psi <- M_klast/M

# PRVA ETAPA DVOETAPNOG UZORKA:
# izvlacenje primarnih jedinica
ind_klast <- sample(N, n,
                    prob = psi,
                    replace = T)
indeksi <- lapply(ind_klast,
                  function (i) klast[[i]])
uzorak <- lapply(ind_klast,
                 function (i) obelezje_klast[[i]])

# uzoracke velicine skupina
Mi_klast <- M_klast[ind_klast]

# DRUGA ETAPA DVOETAPNOG UZORKA:
# izvlacenje sekundarnih jedinica
odabrani <- lapply(1:n,
            function (i) sample(Mi_klast[i],
                                Mi_klast[i]/2))
indeksi <- lapply(1:n,
           function (i) indeksi[[i]][odabrani[[i]]])
uzorak <- lapply(1:n,
          function (i) uzorak[[i]][odabrani[[i]]])
n_klast <- round(Mi_klast/2)

# ocena prema psu bez ponavljanja
tau_klast <- Mi_klast * sapply(uzorak, mean)

# Hansen-Hurwitzova ocena srednje vrednosti
xn_hh <- mean(tau_klast/Mi_klast) # 3.0238

# verovatnoce ukljucenja prvog reda
pi <- 1 - (1 - psi)^n
pi_klast <- pi[ind_klast]

# Horvitz-Thompsonova ocena srednje vrednosti;
# nema ponavljanja, pa ni potrebe za redukcijom
xn_ht <- 1/M * sum(tau_klast/pi_klast) # 3.2601

# spojeni uzorak
indeksi <- unlist(indeksi)
m <- length(indeksi) # 2000

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
```

```r
prec <- mean(y == pred) # 0.8369
```

Скрипт 20: mscswr.r – модел над двоетапним узорком са пон.

И сада је крају изложено шта се дешава када је лоше примењен текући план узорковања. Тако је двоетапни узорак – прост случајан без понављања и са неједнаким вероватноћама и понављањем у првој, а прост случајан без понављања у другој етапи – такав да групе не испуњавају претпоставке метода, формиран у скрипту 21. Кластери су формирани на исти начин као раније – према типу слова. Још једном, утисци су непромењени у односу на варијанту са једном етапом.

```r
# ucitavanje biblioteke i podataka
library(e1071)
slova <- read.csv('slova.csv')

# podela skupa na dva dela
x <- subset(slova,
            select = -slovo)
y <- as.factor(slova$slovo)

# izdvajanje obelezja od znacaja
obelezje <- x$x_ivice
M <- length(obelezje) # 20000
sr <- mean(obelezje) # 3.0461
vr <- var(obelezje) # 5.4407

# klasterovanje prema tipu slova
klast <- lapply(LETTERS,
                function (k) which(k == y))
obelezje_klast <- lapply(klast,
                         function (k) obelezje[k])
M_klast <- sapply(klast, length)
N <- length(M_klast) # 26

# IZVLACENJE BEZ PONAVLJANJA;
# velicina uzorka kako bi broj
# jedinki sto slicniji kao psu
n <- 3199 # zeljen br. jedinki
n <- round(n/(M/N)) # 4 grupe

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# PRVA ETAPA DVOETAPNOG UZORKA:
# izvlacenje primarnih jedinica
ind_klast <- sample(N, n)
indeksi <- lapply(ind_klast,
                  function (i) klast[[i]])
uzorak <- lapply(ind_klast,
                 function (i) obelezje_klast[[i]])

# uzoracke velicine skupina
Mi_klast <- M_klast[ind_klast]

# DRUGA ETAPA DVOETAPNOG UZORKA:
# izvlacenje sekundarnih jedinica
odabrani <- lapply(1:n,
            function (i) sample(Mi_klast[i],
                                Mi_klast[i]/2))
indeksi <- lapply(1:n,
           function (i) indeksi[[i]][odabrani[[i]]])
```

```r
uzorak <- lapply(1:n,
          function (i) uzorak[[i]][odabrani[[i]]])
n_klast <- round(Mi_klast/2)

# ocena prema psu bez ponavljanja
tau_klast <- Mi_klast * sapply(uzorak, mean)
sn2_klast <- sapply(uzorak, var)
sn2_tau <- var(tau_klast) # 1185951.4721

# dodatna disperzija zbog druge etape
sn2_dod <- N/M^2 * mean(Mi_klast^2 *
                        sn2_klast/n_klast *
                        (1 - n_klast/Mi_klast)) # 0.0001

# klasicna ocena srednje vrednosti
xn_psu <- N/M * mean(tau_klast) # 3.5131
D_xn_psu <- (N/M)^2 * sn2_tau/n * (1 - n/N) +
                                  sn2_dod # 0.4241

# kolicnicka ocena srednje vrednosti
xn_kol <- sum(tau_klast)/sum(Mi_klast) # 3.4349
sn2_kol <- 1/(n-1) * sum((tau_klast - xn_kol *
                          Mi_klast)^2) # 1180892.9221
D_xn_kol <- (N/M)^2 * sn2_kol/n * (1 - n/N) +
                                  sn2_dod # 0.4223

# spojeni uzorak
indeksi <- unlist(indeksi)
m <- length(indeksi) # 1572

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.1546

# IZVLACENJE SA PONAVLJANJEM;
# velicina uzorka kako bi broj
# jedinki sto slicniji kao psu
n <- 3808 # zeljen br. jedinki
n <- round(n/(M/N)) # 5 grupa

# nejednake verovatnoce izvlacenja
# srazmerne velicini skupine
psi <- M_klast/M

# fiksiranje generatora pseudoslucajnosti
set.seed(0)

# PRVA ETAPA DVOETAPNOG UZORKA:
# izvlacenje primarnih jedinica
ind_klast <- sample(N, n,
                    prob = psi,
                    replace = T)
indeksi <- lapply(ind_klast,
                  function (i) klast[[i]])
uzorak <- lapply(ind_klast,
                 function (i) obelezje_klast[[i]])
```

```
113
114 # uzoracke velicine skupina
115 Mi_klast <- M_klast[ind_klast]
116
117 # DRUGA ETAPA DVOETAPNOG UZORKA:
118 # izvlacenje sekundarnih jedinica
119 odabrani <- lapply(1:n,
120             function (i) sample(Mi_klast[i],
121                                 Mi_klast[i]/2))
122 indeksi <- lapply(1:n,
123            function (i) indeksi[[i]][odabrani[[i]]])
124 uzorak <- lapply(1:n,
125           function (i) uzorak[[i]][odabrani[[i]]])
126 n_klast <- round(Mi_klast/2)
127
128 # ocena prema psu bez ponavljanja
129 tau_klast <- Mi_klast * sapply(uzorak, mean)
130
131 # Hansen-Hurwitzova ocena srednje vrednosti
132 xn_hh <- mean(tau_klast/Mi_klast) # 2.6848
133
134 # spojeni uzorak
135 indeksi <- unlist(indeksi)
136 m <- length(indeksi) # 1902
137
138 # pravljenje SVM modela
139 model <- svm(x[indeksi,],
140              y[indeksi],
141              fitted = F)
142 summary(model)
143
144 # provera kvaliteta modela
145 pred <- predict(model, x)
146 prec <- mean(y == pred) # 0.1531
147
148 # PRVA ETAPA DVOETAPNOG UZORKA:
149 # iskljucivanje ponovljenih entiteta
150 ind_klast <- unique(ind_klast)
151 indeksi <- lapply(ind_klast,
152                   function (i) klast[[i]])
153 uzorak <- lapply(ind_klast,
154                  function (i) obelezje_klast[[i]])
155 Mi_klast <- M_klast[ind_klast]
156 nn <- length(ind_klast) # 4
157
158 # fiksiranje generatora pseudoslucajnosti
159 set.seed(0)
160
161 # DRUGA ETAPA DVOETAPNOG UZORKA:
162 # izvlacenje sekundarnih jedinica
163 odabrani <- lapply(1:nn,
164             function (i) sample(Mi_klast[i],
165                                 Mi_klast[i]/2))
166 indeksi <- lapply(1:nn,
167            function (i) indeksi[[i]][odabrani[[i]]])
168 uzorak <- lapply(1:nn,
169           function (i) uzorak[[i]][odabrani[[i]]])
170 n_klast <- round(Mi_klast/2)
171
172 # ocena prema psu bez ponavljanja
173 tau_klast <- Mi_klast * sapply(uzorak, mean)
174
```

```r
# verovatnoce ukljucenja prvog reda
pi <- 1 - (1 - psi)^n
pi_klast <- pi[ind_klast]

# Horvitz-Thompsonova ocena srednje vrednosti
xn_ht <- 1/M * sum(tau_klast/pi_klast) # 2.4717

# spojeni uzorak
indeksi <- unlist(indeksi)
m <- length(indeksi) # 1534

# pravljenje SVM modela
model <- svm(x[indeksi,],
             y[indeksi],
             fitted = F)
summary(model)

# provera kvaliteta modela
pred <- predict(model, x)
prec <- mean(y == pred) # 0.15285
```

Скрипт 21: mscslos.r – модел над лошим двоетапним узорком

4 Закључак

Узорковање је важан поступак у раду са подацима, поготову оним великог обима. Над скупом ентитета који представљају велика слова енглеске абецеде примењено је неколико планова узорковања, са посебном пажњом усмереном на познати проблем класификације.

Након детаљног описа података, прво су примењене методе простог случајног узорка и узорка са неједнаким вероватноћама извлачења. Том приликом је потврђена претпоставка да су бољи модели направљени над подацима код којих је очувана расподела обележја високе погађачке моћи. Ово је послужило за формулацију новог статистички заснованог начина одређивања важности предиктора.

Затим је испитана и потврђена претпоставка да модели направљени над скупом података из кога су искључена висококорелисана обележја (вертикални узорак) постижу сличан успех као они направљени над целим скупом, док искључивање некорелисаних обележја смањује прецизност, уз напомену да то важи уз одређени степен грешке.

Показана је важност раслојавања тј. обраћања пажње на добар удео категорија у подацима на основу којих се прави класификациони модел, као и чињеница да је стратификовани узорак углавном бољи за процену популацијских вредности од простог случајног узорка.

Испитано је и да ли јефтинији план узорковања нужно даје лошије резултате и потврђено је да није тако – једноетапни групни узорак код кога су групе добро одабране (унутар себе различите, као популација у малом, а међусобно сличне) дао је не само добре резултате класификације, већ и најразличитије оцене високе прецизности. Подједнако добро понашање показао је и двоетапни узорак са истим скупинама и дупло мањим обимом, као и периодични систематски узорак.

Када је у питању оцењивање средње вредности обележја од значаја (четрнаестог атрибута), уз закључивање засновано на методу одабира узорка, најбољу оцену у смислу прецизности/постојаности дало је количничко оцењивање код једноетапног групног узорка – процењена варијанса 0,0003. Иако прецизна, ова оцена је пристрасна. Код истог

плана узорковања, дисперзија је процењена на високих 0,2419 када се користи непристрасна оцена. Сви прости случајни узорци (без понављања, са понављањем, редуковани са понављањем) дали су оцену чија је дисперзија процењена као 0,0014. Неједнаке вероватноће са понављањем условиле су Хансен-Хурвицову оцену дисперзије 0,0033 и Хорвиц-Томпсонову 0,0058 за редуковани узорак. Раслојено узорковање по типу слова са Нејмановим оптималним распоредом и простим случајним узорком без понављања као позадинским приступом резултовало је дисперзијом оцене од нешто испод 0,0005. Како је ово најмања варијанса код једне тачне (непристрасне и постојане) оцене, она је најбоља у средњеквадратном смислу. Постстратификација простог случајног узорка није донела побољшања због релативно униформне расподеле типа слова. Код једноетапног групног узорка са понављањем и неједнаким вероватноћама избора сразмерним величини кластера дисперзија Хансен-Хурвицове оцене процењена је као добрих 0,0008, а Хорвиц-Томпсонове као нешто лошијих 0,0113. Варијансе код двоетапног кластер узорка са истим скупинама биле су врло сличне. Систематски узорак био је еквивалентан простом случајном узорку без понављања, са процењеном дисперзијом оцене око 0,0015.

Даље истраживање могло би да се фокусира на додатну и ригорознију проверу хипотеза изнесених у раду, нпр. на већем броју различитих база података или пак другим обележјима скупа слова. Осим тога, не би било лоше испитати успешност нешто сложенијих планова узорковања, као што је двофазни план. У том случају би у првој фази било могуће сакупити додатне значајне информације о популацији, које би олакшале другу, главну фазу, слично пилот истраживању које је спроведено за одређивање обима простог случајног узорка.

Литература

[1] Ленка Главаш. Увод у теорију узорака – презентација 1. Математички факултет, Универзитет у Београду, доступно на интернет страници: http://www.matf.bg.ac.rs/p/files/45-prez1_.pdf.

[2] Peter W. Frey and David J. Slate. Letter Recognition Using Holland-Style Adaptive Classifiers. Machine Learning, 6. Boston, Kluwer Academic Publishers, 1991, стр. 161-182, доступно на: http://www.cs.uu.nl/docs/vakken/mpr/Frey-Slate.pdf.

[3] David J. Slate. Letter Recognition Data Set. UCI Machine Learning Repository. Irvine, Center for Machine Learning and Intelligent Systems, Bren School of Information and Computer Science, University of California, доступно на интернет страници: http://archive.ics.uci.edu/ml/datasets/Letter+Recognition.

[4] Документација за IBM SPSS Modeler и коришћене R библиотеке.

[5] S. L. Lohr. *Sampling: Design and Analysis*. Advanced (Cengage Learning). Cengage Learning, 2009. доступно на адреси: https://drive.uqu.edu.sa/_/maatia/files/Sampling.pdf.

[6] Ленка Главаш. Увод у теорију узорака – презентација 2. Математички факултет, Универзитет у Београду, доступно на интернет страници: http://www.matf.bg.ac.rs/p/files/45-prez2_.pdf.

[7] Ленка Главаш. Увод у теорију узорака – презентација 4. Математички факултет, Универзитет у Београду, доступно на интернет страници: http://www.matf.bg.ac.rs/p/files/45-prez4_.pdf.

[8] Ленка Главаш. Увод у теорију узорака – презентација 5. Математички факултет, Универзитет у Београду, доступно на интернет страници: http://www.matf.bg.ac.rs/p/files/45-prez5_.pdf.

[9] William G. Cochran. *Sampling Techniques, 3rd Edition.* John Wiley, 1977. доступно на адреси: https://renasf.fiocruz.br/sites/renasf.fiocruz.br/files/artigos/COCHRAN%2C%20W.%20Sampling%20techniques_compressed.pdf.

[10] Ленка Главаш. Увод у теорију узорака – презентација 6. Математички факултет, Универзитет у Београду, доступно на интернет страници: http://www.matf.bg.ac.rs/p/files/45-prez6_.pdf.

[11] Ненад Митић. Истраживање података 2 – редукција. Математички факултет, Универзитет у Београду, доступно на: http://poincare.matf.bg.ac.rs/~nenad/ip2/redukcija_podataka.pdf.

[12] Ленка Главаш. Увод у теорију узорака – презентација 7. Математички факултет, Универзитет у Београду, доступно на интернет страници: http://www.matf.bg.ac.rs/p/files/45-prez7_.pdf.

[13] Jad Ramadan, Mark Culp, Ken Ryan, and Bojan Cukic. Multi-Stage Stratified Sampling for the Design of Large Scale Biometric Systems. CiTER, West Virginia University, линк: https://www.nist.gov/system/files/documents/2020/09/15/12_wednesday_cukic.pdf.

[14] Ленка Главаш. Увод у теорију узорака – презентација 9. Математички факултет, Универзитет у Београду, доступно на интернет страници: http://www.matf.bg.ac.rs/p/files/45-prez9_v1.pdf.

[15] Ленка Главаш. Увод у теорију узорака – презентација 10. Математички факултет, Универзитет у Београду, доступно на интернет страници: http://www.matf.bg.ac.rs/p/files/45-prez10_.pdf.