УНИВЕРЗИТЕТ У НИШУ
ЕЛЕКТРОНСКИ ФАКУЛТЕТ

ШУМ У ПОДАЦИМА

Семинарски рад
Студијски програм: Вештачка интелигенција и машинско учење
Модул:

Студент:

Јована Стојановић, бр. инд. 1902

Ментор:

Проф. др Александар Станимировић

Ниш, септембар 2025. година

САДРЖАЈ

ШУМ У ПОДАЦИМА

САЖЕТАК

Присуство шума у подацима је чест проблем који доводи до негативних последица приликом решавања класификационих и регресионих задатака. Шум може настати у различитим фазама - од прикупљања и складиштења података, преко процеса означавања, до саме обраде. Последице укључују смањење тачности модела, повећање времена обраде и деградацију интерпретабилности добијених резултата. Перформансе модела изграђених у таквим условима зависе како од квалитета тренинг скупа, тако и од робусности модела на шум.

У овом раду разматрају се различити облици шума, попут насумичних грешака, погрешно означених класа и оштећених артибута, као и њихово идентификовање. Представљене су технике за ублажавање утицаја шума - од основних метода изглађивања (*binning*, регресија, анализа издвојених вредности) до напредних филтера као што су *Ensemble Filter*, *Cross-Validated Committees Filter* и *Iterative-Partitioning Filter*. На крају, дају се препоруке за избор одговарајуће технике у зависности од природе и обима шума у подацима.

Поред тога, анализирају се и технике за уклањање дупликата, као што су блокирање и кластерисање, као и методе за смањење димензионалности попут PCA, LDA и random forest селекције карактеристика.

Кључне речи: квалитет података; шум у подацима; смањење шума; обрада дупликата; смањење димензија

DEALING WITH NOISY DATA

ABSTRACT

The presence of noise in data is a common issue that can negatively impact the performance of classification and regression tasks. Noise may appear during different stages of the data lifecycle—from collection and storage to labeling and preprocessing—leading to reduced accuracy, longer processing times, and degraded interpretability of results. The performance of models trained under such conditions strongly depends on both the quality of the training data and the robustness of the learning algorithms against noise.

This paper discusses various types of noise, such as random errors, mislabeled classes, and corrupted attributes, along with strategies for their identification. It further presents approaches to mitigating noise effects, ranging from basic smoothing techniques (binning, regression, outlier analysis) to advanced filtering methods such as Ensemble Filter, Cross-Validated Committees Filter, and Iterative-Partitioning Filter. Finally, recommendations are provided on how to select appropriate techniques depending on the characteristics and intensity of noise in the dataset.

In addition, the study considers techniques for duplicate removal, such as blocking and clustering, as well as dimensionality reduction methods including PCA, LDA, and random forest feature selection.

Keywords: data quality; data noise; noise reduction; duplicate handling; dimensionality reduction

1. УВОД

Савремени процеси прикупљања и анализе података суочавају се са сталним изазовом присуства шума у подацима. Шум представља насумичне грешке, погрешно класификоване или непотпуне информације које умањују квалитет скупа података и отежавају доношење тачних закључака. Готово сви скупови података из реалног света садрже одређену количину шума, било да он настаје услед техничких грешака (пад система, квар сензора, грешке при уносу), људског фактора (погрешно означени подаци, скраћенице, жаргон) или сложености самих процеса мерења. Због тога се проблематика шума налази у центру пажње *data science*-а, статистичке анализе и машинског учења.

Материјал који је испитиван у овом раду обухвата различите типове шумних података, са посебним фокусом на шум класе (погрешне ознаке инстанци) и шум атрибута (нетачне или непотпуне вредности). Анализа је обухватила и начине на које шум утиче на класификаторе, доводи до губитка интерпретабилности модела и отежава препознавање образаца. Област испитивања обухвата технике за детекцију, категоризацију и елиминацију шума, са циљем да се побољша квалитет података и смањи ризик од погрешних закључака у аналитичким процесима.

Циљ рада је да се кроз теоријско разматрање и практичне примере прикаже значај препознавања и обраде шума. Фокус је на томе да се покаже како правилно уклањање или ублажавање шума унапређује квалитет анализе, повећава тачност модела и смањује трошкове складиштења и обраде података. Посебна пажња посвећена је методама које се користе у пракси: *smoothing* техникама (*binning*, регресија, анализе издвојених вредности), као и напредним филтрима као што су *Ensemble Filter*, *Cross-Validated Committees Filter* и *Iterative-Partitioning Filter*. Наведене методе представљају основне приступе у борби против шума и представљају полазиште за даља истраживања у овој области.

Поред обраде шума, посебан значај имају и поступци уклањања дупликата и смањења димензионалности, јер директно утичу на квалитет и употребљивост података. Уклањање дупликата подразумева идентификацију и елиминацију сувишних или поновљених записа, при чему се примењују различите технике као што су детекција потпуних и приближних дупликата, блокирање, кластерисање и алгоритамски приступи. Смањење димензионалности има за циљ редукцију броја карактеристика уз задржавање најважнијих информација, чиме се смањује сложеност модела и побољшава интерпретабилност резултата. Најчешће коришћене методе у ове сврхе су анализа главних компоненти (PCA), линеарна дискриминантна анализа (LDA), кернел PCA, квадратна дискриминантна анализа (QDA), избор карактеристика кроз филтер, омотач и уграђене методе, као и поступци попут назадне елиминације, напредне селекције и random forest-а.

2. ШТА ЈЕ ШУМ У ПОДАЦИМА?

2.1. Опште

Шум[1] је насумична грешка или варијанса у мереној променљивој. Односно, подаци који садрже шум садрже, осим тачних, и бесмислене или искривљене податке. Скоро сви скупови података у реалном свету садрже одређену количину нежељеног шума, било да се ради о сензорским мерењима, улазима корисника или дигиталној комуникацији. Такви шумни подаци могу утицати на тачност модела и закључака добијених анализом, па је често неопходно њихово чишћење или филтрирање како би се добио квалитетнији скуп података за даљу обраду.

Овај термин се такође користи као синоним за искварене податке или податке које машине не могу разумети и коректно интерпретирати, попут неструктуираних, недовољно формализованих или непотпуних података. Шум у подацима може се манифестовати у различитим облицима - од неправилних уноса, недоследних формата и граматичких грешака до техничких сметњи насталих током снимања или преноса података.

Илустрација ефекта шума у подацима може бити покушај учествовања у разговору у гужви. Човеков мозак је изузетно способан у филтрирању споредних звукова, па може да се фокусира на један извор гласа. Међутим, ако је окружење превише гласно, постаје тешко или немогуће водити смислену конверзацију и чути саговорника. На исти начин, што се више сувишних или нерелевантних информација додаје скупу података, теже постаје издвајање корисних образаца и доношење поузданих одлука.

Шум не само да утиче на тачност анализа већ и непотребно повећава количину података коју је потребно складиштити и обрађивати. Ово може проузроковати веће трошкове, успорити процесирање и довести до неефикасности у свим фазама рада са подацима - од прикупљања, преко трансформације, до визуелизације и тумачења. У контексту *Data Mining*-а (у даљем тексу DM), присуство шума може значајно нарушити резултате анализа, а самим тим и вредност добијених увида.

Статистичке технике, као и савремене методе предобраде података, често се користе за идентификацију и уклањање шума. Историјски подаци, ако су довољно обимни и репрезентативни, могу се искористити за препознавање шаблона понашања података и за одвајање релевантног сигнала од случајног шума. Таква анализа поставља основу за ефикаснији DM и развој робустнијих аналитичких модела.

Алгоритми машинског учења су посебно прилагођени да се носе са одређеним степеном шума. Иако шум може знатно утицати на учинак модела, добро дизајнирани алгоритми могу да апстрахују или игноришу нерелевантне информације и издвоје суштину. Ипак, ако су подаци лошег квалитета или садрже прекомерну количину шума, модел може бити погрешно навођен и довести до нетачних предвиђања или закључака.

Извори шума могу бити различити: падови хардвера, софтверске грешке, погрешни улази од стране корисника, проблеми при мерењима, или неадекватно структурирани подаци. Синтаксне грешке, скраћенице, жаргонски изрази и неформални језик могу пореметити способност машине да исправно обради и разуме текстуалне податке. У сензорским системима, природне осцилације и промене услова околине често доводе до генерисања шумних сигнала. Са друге стране, прикупљање сувише великог и нефилтрираног скупа података може додатно оптеретити аналитички процес и уместо вредности донети конфузију.

Из тог разлога, правилно управљање квалитетом података и адекватна обрада шума представљају кључни корак у свим савременим системима за анализу и доношење одлука, било да се ради о машинском учењу, вештачкој интелигенцији, статистици или бизнис интелигенцији.

2.2. Логичке категорије шумних података

Како су поља *data science* и статистичка анализа веома широка и разнолика, не постоји једна универзално установљена класификација шума у подацима. Међутим, у пракси се често користе неке оквирне категорије које помажу у разумевању узрока појаве шума и његових типова. Ове категорије пружају основ за детаљнију анализу и бољу предобраду података, што је кључно за квалитетне резултате у аналитичким моделима:

- Насумичан шум (*Random noise*)
  Насумичан шум представља додатну информацију која није повезана са очекиваним подацима, али је ипак присутна у мерењима или скупу података. Ова врста шума назива се још и „бели шум“ јер нема специфичан образац и равномерно је распоређена. Скоро свака мерења из реалног света поседују одређену количину насумичног шума услед природних варијација, инструмената за мерење или спољашњих утицаја који нису контролисани;
- Погрешно класификовани подаци (*Misclassified data*)
  Ова категорија обухвата информације које су неисправно означене, етикетиране или сортиране у скупу података. Узроци за ово су најчешће људска грешка приликом уноса или означавања података, као и аутоматске грешке које се дешавају приликом процеса класификације или категоризације. Погрешно класификовани подаци могу значајно нарушити тачност модела и изазвати погрешне закључке;
- Неконтролисане променљиве (*Uncontrolled variables*)
  Овај тип шума настаје када у скупу података постоје фактори који утичу на посматране вредности, али нису узети у обзир током анализе или мерења. Због тога подаци могу изгледати насумично или показивати непредвидиве варијације, иако у стварности постоје узорци или патерни које нису адекватно идентификовани. Неконтролисане променљиве могу замаскирати важне сигнале у подацима и отежати правилно тумачење;
- Сувишни подаци (*Redundant data*)

Сувишни подаци представљају додатне информације које нису релевантне за анализу или истраживање које се спроводи. Превелика количина оваквих података може „заклонити“ и сакрити оне информације које су заправо значајне и потребне. Присуство сувишних података повећава комплексност скупа, успорава обраду и може довести до прецењивања или потцењивања значаја одређених карактеристика.

2.3. Типови шумних података: Шум класе и шум атрибута

Велики број компоненти одређује квалитет скупа података. Међу њима, ознаке класа и вредности атрибута директно утичу на квалитет скупа података за класификацију.

Квалитет ознака класа односи се на то да ли је класа сваког примера исправно додељена и представља кључни фактор за поузданост модела машинског учења. Са друге стране, квалитет атрибута огледа се у њиховој способности да правилно и прецизно карактеришу пример у сврху класификације.

Очигледно је да ако шум утиче на вредности атрибута, ова способност карактеризације, а самим тим и квалитет атрибута, се смањује, што доводи до погрешних закључака и смањене прецизности модела.

На основу ових информација, могу се разликовати две основне врсте шума:

- Шум класе (*Noise in class labels*)
  Шум класе, такође познат као шум ознака, јавља се када је неки пример у скупу података погрешно означен. Ова појава може настати из више разлога, као што су:
  - Субјективност у процесу означавања: различити људи могу класификовати исте примере на различите начине,
  - Грешке приликом уноса података: људски фактор или аутоматски системи могу направити грешке и
  - Неадекватност или непотпуност информација: недовољно тачне или нетачне информације могу довести до погрешних ознака.

  У оквиру шума класе могу се издвојити две подкатегорије:.
  - Противречни примери: у скупу постоје дуплирани записи који имају различите ознаке класа, што отежава правилно учење и
  - Погрешне класификације: примери су означени ознакама класа које се разликују од њихове стварне или исправне класе.
- Шум атрибута (*Noise in attribute values*)
  Ова врста шума односи се на оштећења или грешке у вредностима једног или више атрибута у скупу података. Примери шума атрибута укључују:
  - Нетачне вредности атрибута: нпр. мерења која нису тачна због кварова у уређајима,
  - Недостајуће или непознате вредности: поља која нису попуњена или су означена као непозната и
  - Непотпуне или неодређене вредности: нпр. вредности типа „није важно“ или сличне ознаке које не дају корисне информације.

3. ИДЕНТИФИКОВАЊЕ ШУМА

3.1. Опште

Подаци из реалног света никада нису савршени и често пате од корупција које могу значајно утицати на интерпретацију података, изградњу модела и доношење одлука [2]. У контексту класификације, присуство шума у подацима може негативно утицати на кључне перформансе система, као што су прецизност класификације, време потребно за изградњу модела, величина модела и његова интерпретабилност.

Шум не само да смањује квалитет предвиђања, већ може утицати и на суштинске карактеристике проблема класификације. На пример, шум може створити мале, несистематске кластере инстанци одређене класе у деловима простора који по правилу припадају другој класи. Такође, може уклонити важне инстанце које се налазе у кључним областима конкретне класе или нарушити границе између класа, што доводи до њиховог преклапања.

Ови ефекти значајно кваре знање које се може извући из проблема и деградирају перформансе класификатора изграђених на бази таквих бучних података. У поређењу са класификаторима направљеним од чистих, квалитетних података - који представљају најтачније и најпоузданије имплицитно знање о проблему - модели обучени на шумним подацима често показују смањену тачност и поузданост.

3.2. Стратегије на нивоу машинског учења

Неки од најважнијих приступа за рад са бучним подацима и постизање веће тачности класификације у таквим условима укључују следеће методе:

- Робустни ученици (*Robust learners*)
  Робустни ученици представљају технике и алгоритме машинског учења који су дизајнирани да буду мање подложни негативним утицајима бучних података. Они могу да се носе са одређеним степеном шума без значајног пада перформанси. Пример оваквог робустног алгоритма је C4.5[1], који показује отпорност на мање количине шума у скупу података. Међутим, уколико је ниво шума релативно висок, чак и ови робустни алгоритми могу показати значајно погоршање у тачности и поузданости модела;
- Методе полирања података (*Data polishing methods*)
  Ове методе имају за циљ да исправе или уклоне бучне инстанце пре него што започне процес обучавања модела. Полирање података подразумева детаљну анализу и корекцију нетачних или непоузданих вредности у скупу података, што значајно побољшава квалитет тренирајућег скупа. Међутим, због своје временске и рачунарске захтевности, овај приступ је најпогоднији за мање скупове података. Студије показују да је потпуна или делимична

[1] Алгоритам који користи стратегије орезивања (*pruning*) како би смањио могућност прекомерног прилагођавања (*overfitting*) у тренинг подацима

корекција шума у тренирајућим подацима један од најефикаснијих начина за побољшање перформанси класификационих модела;

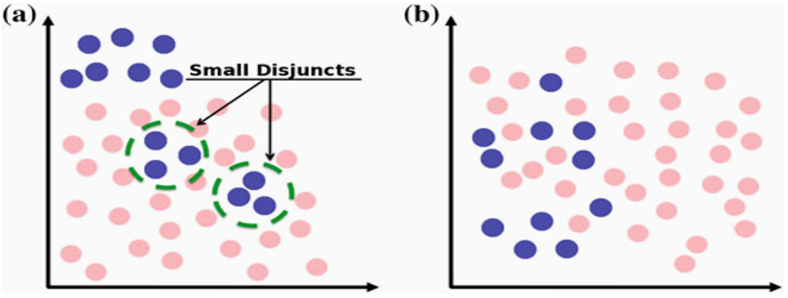
- Филтери за шум (*Noise filters*)
  Филтери за шум се користе за аутоматску идентификацију и елиминацију бучних инстанци из скупа података пре тренирања модела. Ови филтери су посебно корисни када се користе у комбинацији са алгоритмима који су осетљиви на присуство шума. Претходна обрада података уз помоћ ових филтера омогућава смањење утицаја шума и побољшава квалитет и прецизност резултујућих класификатора.

3.3. Структурни изазови у подацима

Сложене и нелинеарне границе између класа представљају значајан изазов у класификацији, јер могу знатно отежати перформансе класификатора. Често је веома тешко направити јасну разлику између природног преклапања класа и присуства шума у подацима, што додатно компликује процес учења модела.

Проблеми који се јављају услед ових услова и који доприносе деградацији перформанси класификатора могу се илустровати кроз следеће феномене:

- Присуство малих дисјунктних група (*Presence of small disjuncts*)
  Овај проблем је представљен на слици 1. *a)* и односи се на ситуацију када мањинска класа није конзистентна и компактна, већ је разложена на бројне мање под-кластере који садрже врло мало примера у сваком од њих. Ове мале групе се налазе у околини инстанци већинске класе, што представља изазов за већину алгоритама за учење. Наиме, такав распоред отежава прецизно откривање и класификацију ових под-концепата, јер модели често не успевају да препознају ове мале и дисјунктне подгрупе;
- Преклапање између класа (*Overlapping between classes*)
  Преклапање класа је илустровано на слици 1. *b)* и односи се на ситуацију када постоје примери из различитих класа са веома сличним или идентичним карактеристикама, посебно у областима око граница одлучивања између класа. Ови преклапајући примери стварају регије класа које нису јасно раздвојене, што чини класификацију тешком и повећава могућност погрешне класификације. У пракси, ова преклапања могу настати услед природних сличности између класа или због ограничења у квалитету и количини података.

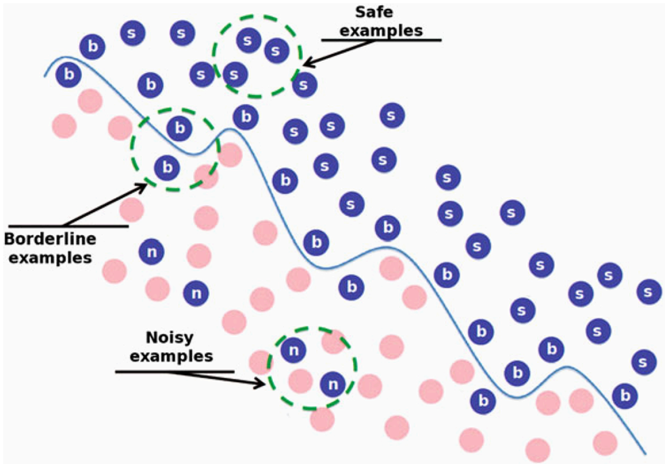


Слика 1. Примери интеракција између класа: a) мали дисјункти и b) преклапање између класа

Блиско повезан са проблемом преклапања класа је и феномен присуства примера који се налазе у области око граница класа, познати као гранични примери (*borderline examples*). Истраживања показују да до грешака у класификацији често долази управо у овим областима, где се класе преклапају или се границе између њих сложено испреплићу. Проналажење ефикасног решења за овај проблем представља значајан изазов у домену машинског учења.

Деградација перформанси класификатора у великој мери зависи од количине граничних примера у скупу података. Поред тога, присуство других врста бучних примера који се налазе изван преклапајуће регије такође отежава примену техника као што су методе поновног узорковања (*re-sampling*). Ови проблеми су илустровани на слици 2, где су класификовани следећи типови примера:

- Сигурни примери (*Safe examples*)
  Ови примери се налазе у релативно хомогеним областима унутар класа, где су карактеристике јасно одвојене и представљају типичне представнике своје класе. Они представљају основу на којој класификатори лако граде своје одлуке;
- Гранични примери (*Borderline examples*)
  Гранични примери смештени су у зонама око граница између класа, где се мањинска и већинска класа преклапају или где границе имају сложен и фрактални облик[2]. Ови примери су изазовни за класификаторе јер чак и мала количина шума у вредностима атрибута може померити пример на погрешну страну границе одлучивања, што повећава вероватноћу погрешне класификације;
- Бучни примери (*Noisy examples*)
  Бучни примери су инстанце које су погрешно означене, односно припадају једној класи, али се појављују у сигурним областима друге класе. Ови примери се могу сматрати жртвама шума у ознакама класе, што додатно компликује процес учења и смањује перформансе модела.



Слика 2. Три типа примера: сигурни (означени са s), гранични (означени са b) и бучни (означени са n). Линија представља храницу одлучивања између две класе

[2] Облик који може имати димензију која је између два цела броја

4. ПРОЦЕСИРАЊЕ ШУМНИХ ПОДАТАКА

4.1. Основне технике за рад са шумом у подацима

Елиминација или ублажавање шума представља кључни корак у побољшању квалитета података и повећању поузданости аналитичких модела. Без адекватне обраде шумних података, модели могу дати нетачне предикције, што директно утиче на доношење одлука заснованих на тим анализама.

Постоје два главна приступа за решавање проблема шума у подацима:

- *Smoothing* технике
  Ове методе имају за циљ смањење варијабилности у подацима и ублажавање осцилација, односно "изглађивање" података. Важно је напоменути да *smoothing* технике не уклањају шум у потпуности, већ га редукују тако да не ремети основне трендове и структуру података.
  Smoothing је посебно користан у анализи временских серија, финансијских података, као и у обради сензорских сигнала, где је циљ смањити нагле промене и шум, а истовремено сачувати суштинске обрасце и трендове;
- Алгоритми за филтрирање шума (филтри)
  Овај приступ подразумева напредније методе које активно идентификују и уклањају шумне инстанце из скупа података. Филтри често користе математичке трансформације, статистичке тестове и моделе машинског учења како би открили податке који представљају шум, а затим их елиминисали или кориговали.
  Овакав приступ је неопходан када је присуство шума изражено и када би његов утицај могао озбиљно угрозити тачност и поузданост анализе.

Често, *smoothing* технике омогућавају јасније препознавање образаца у подацима. Посебно су корисне у анализи временских серија, финансијских података и сензорских мерења, где је циљ ублажити нагле промене и осцилације без значајног мењања основне структуре података. Ове технике помажу да се издвоје битни трендови и сигнал, смањујући утицај случајних флуктуација.

Међутим, у случајевима када је шум изражен и може угрозити тачност анализе, неопходно је применити алгоритме за филтрирање шума. Ови алгоритми активно идентификују и уклањају шумне податке, при чему настоје да сачувају суштинску и релевантну информацију која се налази у скупу података.

Smoothing технике се често примењују као први корак пре примене алгоритама за филтрирање шума, али конкретан приступ зависи од специфичног проблема и природе података са којима се ради.

4.2. *Smoothing* технике

Smoothing технике [3] подразумевају *binning* (груписање у корпе), регресију и *outlier analysis* (анализу одступања).

4.2.1. *Binning*

Методе биновања [4] изглађују сортиране вредности података тако што узимају у обзир њихово "суседство", односно околне вредности. Сортиране вредности се распоређују у одређени број "канти" или бинова. Пошто биновања узимају у обзир суседство вредности, оне врше локално изглађивање.

У овој техници:

1. Подаци се прво сортирају;
2. Затим се сортирана листа дели у бинове једнаке дубине;
3. Након тога, могуће је изгладити податке на различите начине: по средњој вредности у бину, по медијани бина, по границама бина итд.

Постоје две методе дељења података у бинове:

- Биновање са једнаком ферквенцијом (*Equal Frequency Binning*): сваки бин има исти број елемената и
- Биновање са једнаком ширином (*Equal Width Binning*): сви бинови имају једнаку ширину, а опсег сваког бина се одређује по формули:

$$[min + \omega],\ [min + 2\omega],\ \ldots,\ [min + n\omega]$$

где је

$$\omega = \frac{max - min}{\text{број бинова}}$$

Изглађивање по средњој вредности бина (*Smoothing by bin means*): свака вредност у бину се замењује просечном (средњом) вредношћу тог бина.

Изглађивање по медијани бина (*Smoothing by bin medians*): Свака вредност у бину се замењује медијаном тог бина.

Изглађивање по границама бина (*Smoothing by boundaries*): Минималне и максималне вредности бина се идентификују као границе бина. Свака вредност у бину се замењује најближом граничном вредношћу.

Пример:

- Сортирани подаци о ценама (у доларима): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
- Дељење на бинове једнаке дубине (по 4 ведности у сваком бину):
  - Бин 1: 4, 8, 9, 15
  - Бин 2: 21, 21, 24, 25
  - Бин 3: 26, 28, 29, 34

  Изглађивање по средњој вредности бина (*Bin Means*):
- Бин 1: 9, 9, 9, 9 (средња вредност: (4+8+9+15)/4 = 9)
- Бин 2: 23, 23, 23, 23 (средња вредност: (21+21+24+25)/4 = 23)
- Бин 3: 29, 29, 29, 29 (средња вредност: (26+28+29+34)/4 = 29)

Слично се може применити изглађивање по медијани бина, где се свака вредност у бину замењује медијаном тог бина.

Изглађивање по границама бина (*Bin Boundaries*):

- Бин 1: 4, 4, 4, 15 (најближе граничне вредности: 4 и 15)
- Бин 2: 21, 21, 25, 25 (најближе граничне вредности: 21 и 25)
- Бин 3: 26, 26, 26, 34 (најближе граничне вредности: 26 и 34)

Овај метод може имати више сврха, као што су смањење грешака у подацима изглађивањем флуктуација, стицање бољег увида у дистрибуцију података, као и претварање континуираних вредности у дискретне интервале [5].

На тај начин, метод помаже у смањењу ефекта грешака и шумова у подацима, а истовремено омогућава лакшу анализу и обраду података који су иначе у облику реалних бројева.

Међутим, ова техника носи и одређене изазове, од којих је најзначајнији избор одговарајуће величине бина за дати скуп података. Превелика величина корпе може довести до губитка важних детаља и прекомерног поједностављења података, док премала величина корпе може задржати превише шума и компликовану структуру. Због тога је неопходно пажљиво прилагодити овај параметар у складу са природом и захтевима конкретне анализе.

4.2.2. Регресија

Изглађивање података може се такође постићи применом регресије, технике која прилагођава вредности података одређеној функцији како би се моделовао однос између променљивих. Један од најчешће коришћених алгоритама у овом контексту је линеарна регресија [6].

Линеарна регресија описује линеарну везу између независне и зависне променљиве. Независна променљива, која се назива и предикторска или објашњавајућа променљива, представља улазне податке који се не мењају под утицајем других променљивих у моделу. Насупрот томе, зависна променљива варира као функција независне променљиве и користи се за предвиђање исхода или вредности на основу улазних података.

Овај метод изглађивања омогућава идентификовање трендова и облика у подацима, смањујући утицај насумичних осцилација и шума. Линеарна регресија је једноставна за примену и интерпретацију, што је чини погодном за многе практичне примене у статистици и машинском учењу.

Регресиони модел предвиђа вредност зависне променљиве, која представља одзивну или излазну промељиву која се анализира или проучава. Једначине линеарне регресије је:

$$y(x) = p_0 + p_1 \cdot x$$

где су:

- y - излазна променљива. Представља континуирану вредност коју модел предвиђа,
- x - улазна променљива, односно обележје (*feature*),
- p_0 - пресек са y-осом, познат као пристрасност (*bias term*),
- p_1 - регресиони коефицијент или фактор скалирања,
- p_i - тежине, уопштено гледано, су параметри које модел учи током тренинга.

Овај модел елиминише мале варијације у подацима, односно шум, и представља податке кроз најбоље прилагођену праву (*best-fit line*) која минимизује укупну грешку прилагођавања. Тако се добија поједностављена, али репрезентативна слика трендова у скупу података.

Вишеструка линеарна регресија представља проширење основне линеарне регресије, успостављајући везу између више независних променљивих (две или

више) и једне зависне променљиве. Независне променљиве могу бити континуиране, као што су мерења, али и категоричке, као што су класе или групе.

Ова врста регресије омогућава предвиђање будућих вредности зависне променљиве на основу комбинације више фактора, као и процену утицаја сваке независне променљиве на резултат. Захваљујући томе, вишеструка линеарна регресија је моћан алат за анализу комплексних односа у подацима.

Њена формула је:

$$y(x) = p_0 + p_1 x_1 + p_2 x_2 + \ \ldots \ + p_{(n)} x_{(n)}.$$

Вишеструка линеарна регресија се користи када је зависна променљива под утицајем више фактора, односно када на исход утиче комбинација више независних променљивих. Овај модел омогућава детаљнију анализу и боље разумевање како сваки од фактора утиче на зависну променљиву.

Кључне предности регресије укључују једноставну имплементацију, добру интерпретабилност резултата, скалабилност за рад са великим скупом података, као и могућност оптималног подешавања у реалном времену (онлајн подешавања). Међутим, претпоставка линеарног односа између зависних и независних променљивих често може бити превише поједностављена и довести до проблема као што је преприлагођавање (*overfitting*), посебно када је број независних променљивих велик, а подаци садрже шум или комплексне нелинеарности.

4.2.3. *Outlier analysis*

Иако не постоји јединствена, општеприхваћена или строга математичка дефиниција издвојене вредности (*outlier*-а), постоји консензус да се овај термин односи на статистички појам који означава посматрање чија нумеричка вредност знатно одступа од понашања већине података [7]. Издвојене вредности представљају екстремне или необичне вредности које се разликују од осталих података у скупу и могу указивати на грешке у мерењу, варијације у систему или присуство необичних, али значајних феномена.

Оне су најједноставнија и најпознатија врста аномалија у подацима и веома су честе у већини примењених и научних сценарија који укључују прикупљање, обраду и анализу података. Правилно препознавање и третман издвојених вредности је кључан корак за побољшање квалитета података и прецизност добијених модела.

Све методе детекције издвојених вредности анализирају однос између објеката у скупу података, било на глобалном или локалном нивоу, и израчунавају независне оцене издвојености (*outlier scores*) како би донеле закључак да ли је неки објекат издвојена вредност [8]. Ове оцене представљају меру колико одступа појединачна вредност у односу на остатак података.

Метод *Neighborhood Averaging* (NA) је један од често коришћениш техника постпроцесирања ових оцена издвојености. Он примењује *outlier* детектор и затим прилагођава оцене издвојености унутар одређеног окружења (*neighborhood*)

како би се извршило изглађивање података, што може побољшати поузданост детекције и смањити утицај случајних одступања.

Детектори издвојених вредности могу бити глобални, који користе све објекте у скупу података за процену, или локални, који анализирају само ограничени подскуп објеката, као што је *k-Nearest Neighbors* (k-NN), за упоредну анализу и боље прилагођавање контексту података.

Методе које се користе за детекцију одступајућих вредности (*outlier*-а) могу се сврстати у четири главне категорије [9]:

- Приступ заснован на статистици (*Statistical based approach*)
  Овај приступ је погодан за анализу квантитативних реалних вредности, нарочито у једнодимензионалном простору. Типични представници овог приступа су методе као што су *Z-score* и IQR (*Interquartile Range*), које користе основне статистичке мере попут средње вредности, стандардне девијације и квартила за идентификацију екстремних вредности. Међутим, примена овог приступа је ограничена у вишедимензионалним просторима због сложености дефинисања статистичких модела и немогућности да се лако ухвате сложени међузависни односи између више атрибута;
- Приступ заснован на густини (*Density based approach*)
  Карактерише се тиме што не захтева претпоставке о дистрибуцији података, што га чини изузетно ефикасним у детекцији *outlier*-а. Један од најпознатијих представника овог приступа је *Local Outlier Factor* (LOF), који пореди локалну густину сваке тачке са густином њених суседа како би идентификовао аномалије. Ипак, имплементација овог приступа може бити веома сложена, а додатно отежава и то што не постоји могућност ажурирања вредности *outlier score*-а након првобитне анализе;
- Приступ заснован на растојању (*Distance based approach*)
  Добро се скалира у вишедимензионим просторима и релативно је једноставан за примену. Један од практичних примера овог приступа је *Isolation Forest*, који изолује аномалије кроз итеративно насумично раздвајање података, без потребе за прорачуном густине или растојања. Ипак, са повећањем димензионалности долази до пада перформанси услед појаве тзв. "проклетства димензионалности" (*curse of dimensionality*), што може утицати на прецизност уочавања *outlier*-а у сложенијим просторима;
- Приступ заснован на кластеровању (*Clustering based approach*)
  Овај *unsupervised* метод је робустан према различитим типовима података и омогућава инкрементално додавање нових података у постојеће кластере. Не захтева претходно познавање дистрибуције података, али зависи од корисникове иницијалне поделе података у кластере. Такође, овај приступ се често ослања на визуелну идентификацију кластера и нема подршку за накнадно праћење корака (*back-tracking*).

4.3. Филтрирање података

Најчешћи и најновији филтри шума су *Ensemble Filter*, *Cross-Validated Committees Filter* (филтер комитета са унакрсном валидацијом) и *Iterative-Partitioning Filter* [2].

Ови филтри могу значајно побољшати перформансе у детекцији шумова, уједно поједини могу имати високе и рачунарске трошкове у виду ресурса [10].

4.3.1. *Ensemble Filter*

Овај филтер представља добро познат метод у научној литератури, са циљем побољшања квалитета тренинг података као дела процеса предобраде у задацима класификације. Његова основна функција је детекција и елиминација неправилно означених (погрешно класификованих) инстанци, које могу негативно утицати на перформансе модела. Филтер користи скуп алгоритама машинског учења за генерисање више класификатора, при чему се сваки од њих обучава над различитим подскуповима оригиналног тренинг скупа. Добијени класификатори затим делују као "филтери за шум", односно идентификују и избацују сумњиве или неконзистентне инстанце из тренинг скупа, чиме се побољшава тачност и поузданост финалног модела.

Идентификација потенцијално бучних инстанци врши се извођењем Г-FCV (унакрсна валидација са Г поделом, *Gamma-Fold Cross-Validation*) над тренинг подацима, користећи алгоритама класификације, који се називају филтер алгоритми.

Комплетан процес који изводе енсембле филтри:

1. Поделити тренинг скуп података DT на Г подскупова једнаке величине
2. За сваки од μ филтер алгоритама:
   a. За сваки од Г делова, филтер алгоритам се тренира на преосталих Г-1 делова, што резултира са Г различитих класификатора
   b. Ови Г класификатори се затим користе за означавање сваке инстанце у искљученом делу као исправно означене или погрешно означене, упоређујући оригиналну ознаку са ознаком додељеном од стране класификатора.
3. Након овог процеса, свака инстанца у тренинг скупу биће означена од стране сваког филтер алгоритма
4. Додати у DN инстанце означене као бучне у DT, користећи шему гласања, при чему се узима у обзир тачност ознака добијених у претходном кораку од стране μ филтер алгоритама. У овом случају користи се консензусно гласање.
5. Уклонити бучне инстанце из тренинг скупа DT ← DT\DN.

4.3.2. *Cross-Validated Committees Filter*

Cross-Validation Consensus Filter (у даљем тексту CVCF) представља ансамбл метод који се користи као техника претходне обраде у задацима класификације, са циљем идентификације и елиминације погрешно означених

инстанци у тренинг скупу. Овај метод побољшава квалитет података уклањањем шумовитих и непоузданих примера који могу утицати на тачности класификатора.

Углавном се заснива на извођењу Г-FCV, којим се тренинг скуп дели на више подскупова. За сваки од тих подскупова обучава се класификатор, најчешће стабло одлучивања, а добијени класификатори затим служе за процену тачности ознака у оригиналном скупу. Инстанце које нису доследно класификоване у већини интеграција сматрају се потенцијално погрешно означеним и уклањају се из тренинг скупа.

На овај начин, CVCF користи предности ансамбл стратегија и валидирајућих механизама како би побољшао робустност и поузданост модела учења.

Основни кораци ове методе:

1. Поделити тренинг скуп података DT на Г подскупова једнаке величине
2. За сваки од ових Г делова, основни алгоритам за учење тренира се на преосталих Г - 1 делова, што резултира са Г различитих класификатора. Препоручује се коришћење C4.5 као основног алгоритма
3. Ови Г класификатори се затим користе за означавање сваке инстанце у тренинг скупу DT као исправно означене или погрешно означене, упоређујући оригиналну ознаку са ознаком коју додељује класификатор
4. Додати у DN инстанце означене као бучне у DT, користећи шему гласања (већинског гласања), при чему се узима у обзир тачност ознака добијених у претходном кораку од стране Г класификатора
5. Уклонити бучне инстанце из тренинг скупа: DT ← DT\DN.

4.3.3. *Iterative-Partitioning Filter*

Ово је техника претходне обраде података заснована на *Partitioning Filter* методи и користи се за идентификацију и елиминацију погрешно означених инстанци у великим скуповима података. Док већина филтера за шум полази од претпоставке да се скуп података може обрадити у једном пролазу - што је примењиво на мање скупове - ова претпоставка не важи за велике скупове, где је потребно применити стратегије партиционисања и итеративног чишћења података.

Iterative-Partitioning Filter (у даљем тексту IPF) уклања бучне (шумовите) инстанце кроз више узастопних итерација. Процес се понавља све док не буде испуњен дефинисани критеријум заустављања. Конкретно, итеративни поступак престаје када, током *s* узастопних итерација, број идентификованих бучних инстанци у свакој од тих итерација падне испод прага дефинисаног као *p* процената од укупне величине оригиналног тренинг скупа података.

На овај начин, IPF обезбеђује ефикасно чишћење великих и потенцијално шумовитих скупова података, прилагођавајући се специфичностима обраде у окружењима са ограниченим ресурсима и великом количином података.

На почетку постоје:

- DN = ∅ (скуп бучних инстанци)

- DG = ∅ (скуп исправних података)
  Основни кораци сваке итерације:

1. Поделити тренинг скуп DT на Г подскупова једнаке величине, при чему је сваки подскуп довољно мали да га алгоритам може обрадити у једном пролазу.
2. Тренирати основни алгоритам за учење на свком од ових Г подскупова, што резултира са Г различитих класификатора. Препорука је користити C4.5 као основни алгоритам
3. Означити инстанце у тренинг скупу DT као исправно или погрешно означене, упоређујући оригиналну ознаку са ознаком додељеном од стране класификатора
4. Додати у DN инстанце означене као бучне, користећи шум гласања (већинско гласање у IPF методи)
5. Додати у DG проценат у исправно означених података из DT. Овај корак је користан код великих скупова података јер убрзава редукцију података
6. Уклонити бучне инстанце и добре податке из тренинг скупа: DT ← DT \ {DN ∪ DG}.

На крају итерација, пречишћени скуп података састоји се од преосталих инстанци из DT и исправних инстанци из DG, тј. DT ∪ DG. Инстанце означене као бучне морају бити погрешно класификоване и од стране модела тренираног на подскупу у којем се те инстанце налазе. Ниво конзервативности филтера може се подесити променом броја итерација, било у консензусу или већинском режиму гласања.

5. ОБРАДА ДУПЛИКАТА

5.1. Опште

Брзе методе за упоређивање записа у базама података који су слични или идентични постају све важније како величина база података расте [12]. Мање грешке у посматрању, обради или уносу података могу довести до креирања више неповезаних, скоро дуплираних записа за један реалан ентитет. Поред тога, записи се често састоје од више атрибута или поља; мала грешка или недостајући унос у било ком од тих поља може проузроковати дуплирање.

Проблеми откривања дупликата не скалирају се добро. Број поређења који је потребан расте квадратично са бројем записа, а број могућих подскупова расте експоненцијално. Неповезани дупликати повећавају величину базе података и отежавају њено компресовање у друге формате. Дупликати такође знатно отежавају анализу података, смањују тачност, па чак и онемогућавају многе врсте анализа, јер подаци више не представљају верну слику стварности. Може се на дупликате гледати као на сувишне податке (поглавље 2, потпоглавље 2).

5.2. Технике детекције дупликата

Од изузетног значаја пречишћавање скупа података пре тренинга модела. Постоје различите технике детекције дупликата у зависности од тога какве дупликате желимо да пронађемо.

5.2.1. Детекција потпуних дупликата (*Exact duplicates*)

Ова техника подразумева идентификацију записа који су у потпуности идентични по свим атрибутима. Најчешће се спроводи помоћу функција као што су *drop_duplicates()* у *Python* библиотекама попут *pandas* [13]. Овај приступ је ефикасан када се ради са добро структурираним подацима.

5.2.2. Детекција приближних дупликата (*Near duplicates*)

У многим реалним скуповима података, дупликати нису потпуно идентични већ се разликују у мањим детаљима - нпр. правописним грешкама, различитом форматирању или скраћеницама [14]. За овакве случајеве користе се *fuzzy matching* технике, попут *Levenshtein* растојања, *Jaccard* сличности или *cosine* сличности над текстуалним ентитетима. У *Python*-у, библиотеке као што су *fuzzywuzzy* или *textdistance* омогућавају мерење сличности између записа.

5.2.3. Блокирање (*Blocking*)

Блокирање је техника која смањује број поређења међу записима тако што групише (или „блокира“) записе према одређеним кључевима - нпр. иницијалима, поштанским бројевима или ID-евима [15]. Унутар сваког блока, примењују се технике фази поређења. Ова метода значајно убрзава процес откривања дупликата у великим скуповима података.

5.2.4. Кластеровање

Када се идентификују слични записи, они се могу груписати у кластере. Сваки кластер представља један ентитет, а један репрезентативни запис (нпр. онај са најмање недостајућих вредности или највишим нивоом поверења) се задржава [16]. Кластерисање се може обавити помоћу алгоритама као што су *K-means*, DBSCAN или хијерархијско кластерисање.

5.2.5. Алгоритамске методе на основу правила

У напреднијим приступима, креирају се логичка правила или модели засновани на учењу који аутоматски откривају дупликате [17]. Ови приступи укључују ручно дефинисане услове, као и моделовање засновано на вештачкој интелигенцији (на пример, класификатори који предвиђају да ли су два записа дупликати или не).

5.3. Метрике за мерење дупликата

Да би се проценио утицај дупликата, могу се користити метрике као што су [18]:

- Проценат дупликата: однос броја дупликата и укупног броја записа,
- Степен сличности: просечна/медијална сличност међу сличним записима и
- Утицај на модел: поређење перформанси модела пре и после уклањања дупликата.

6. СМАЊЕЊЕ ДИМЕНЗИОНАЛНОСТИ

6.1. Разумевање смањења димензионалности

Редукција димензија је моћна техника у машинском учењу и анализи података која подразумева трансформацију података са великом димензијом у простор са мањим бројем димензија, при чему се настоји задржати што је могуће више важних информација [19]. Податаци високе димензије односе се на скупове података који имају велики број карактеристика или променљивих, што може представљати значајне изазове за моделе машинског учења. Редукција димензионалности и уклањање шума често иду руку под руку, јер оба процеса имају за циљ побољшање квалитета података које користиш у анализи, учењу модела или визуализацији.

Подаци високе димензије, иако богати информацијама, често садрже сувишне или нерелевантне карактеристике. То може довести до неколико проблема:

- Преклетство димензија: Како број димензија расте, тачке података постају разређене, што отежава моделима машинског учења да пронађу обрасце,
- Прекомерно прилагођавање (*overfitting*): Скупови података са високим бројем димензија могу довести до прекомерног прилагођавања модела, јер модели могу научити шум уместо основних образаца,
- Рачунска сложеност: Више димензија значи веће рачунске трошкове, што успорава обуку и предвиђање и
- Проблеми визуелизације: Визуелизација података са више од три димензије отежава разумевање структуре података.

Редукција димензија решава ове проблеме поједностављавајући податке, а при том задржавајући најважније карактеристике. Ово не само да побољшава перформансе модела, већ и олакшава интерпретацију и визуелизацију података.

6.2. Технике редукције димензија

Технике редукције димензија могу се грубо поделити у две категорије [20]:

- Избор карактеристика (*Feature selection*)
  Овај приступ подразумева задржавање релевантних (оптималних) карактеристика и одбацивање нерелевантних, како би се осигурала висока тачност модела. Најчешће коришћене методе избора карактеристика су: филтер методе (*filter*), омотач методе (*wrapper*) и уграђене методе (*embedded*).
- Екстракција карактеристика (*Feature extraction*)
  Овај процес се назива и пројекција карактеристика, при чему се подаци из вишедимензионалног простора трансформишу у простор мањег броја димензија. Познате методе екстракције карактеристика укључују: анализу главних компоненти (PCA - *Principal Component Analysis*), линеарну

дискриминантну анализу (LDA - *Linear Discriminant Analysis*), кернел PCA (K-PCA - *Kernel Principal Component Analysis*) и квадратну дискриминантну анализу (QDA - *Quadratic Discriminant Analysis*).

Најчешће коришћене технике редукције димензија и избора карактеристика у машинском учењу су управо ове:

- Анализа главних компоненти (PCA)
  Principal component analysis (PCA) врши ортогоналне трансформације ради претварања запажања која садрже корелисане карактеристике у скуп линеарно некорелисаних карактеристика. Нове карактеристике називају се „главне компоненте“. Овај статистички метод је кључна техника у анализи података и предикативном моделирању.
- Унос са пропуштеним вредностима (*Missing value ratio*)
  Када скуп података садржи значајан број пропуштених вредности, те варијабле се могу елиминисати јер не пружају релевантне или поуздане информације. Такав процес подразумева дефинисање прага: ако варијабла има више пропуштених вредности од дозвољеног прага, одмах се изоставља. То значи да што је праг строжији, ефикасност метода опада.
- Назадна елиминација карактеристика (*Backward feature elimination*)
  Ова техника се обично користи током развоја линеарног или логистичког регресионог модела. Процес почиње тренирањем модела на свим n варијаблима из скупа података. Након евалуације перформанси модела, карактеристике се по једна уклањају, а модел и даље тренира на n-1 варијабли, и тако н пута. Ипак, коришћењем ове итерације доћи ће се до варијабле чије уклањање утиче најмање на перформансе модела - она се трајно елиминише, а поступак се наставља док се не дође до стања где више није могуће изоставити ниједну варијаблу без значајног утицаја на перформансе.
- Напредна селекција карактеристика (*Forward feature selection*)
  Овај приступ је супротан назадној елиминацији. Уместо да се карактеристике бришу, одређује се најзначајнији скуп карактеристика који доноси надпросечно унапређење модела. Процес почиње са једним карактеристикама, а затим се додаје по једна - свака се самостално евалуира, а она са најбољим резултатом остаје у моделу, и тако се наставља док год постоји побољшање перформанси.
- *Random forest*
  Random forest представља приступ селекције карактеристика које већ у себи имају пакет за процену значајности (*feature importance*), тако да није потребна додатна имплементација. Метод гради више одлучујућих стабала у односу на циљну варијаблу и, на основу статистике, идентификује најважније карактеристике. Пошто random forest тражи нумеричке податке, потребна је “*one-hot*” енкодирање за претварање свих улазних података у нумерички формат.

7. ЗАКЉУЧАК

У раду је обрађена тема шума у подацима, са посебним нагласком на изворе његовог настанка, врсте и последице које изазива на аналитичке процесе. Показано је да шум може значајно деградирати перформансе класификатора, продужити време обраде, повећати потребан меморијски простор и довести до погрешних одлука приликом доношења закључака. Анализирани су главни типови шума - шум класе и шум атрибута - и приказано је како сваки од њих на специфичан начин утиче на квалитет података и поузданост модела.

Избор методе обраде шума мора зависити од природе података и циљ истраживања. За блажи шум и временске серије препоручују се *smoothing* технике, док се у случајевима већих и комплекснијих шумова примењују напредни филтери. Посебно је наглашено да комбинација више метода може дати стабилније и прецизније резултате, јер омогућава баланс између једноставности примене и тачности. Поред тога, значајну улогу имају и поступци уклањања дупликата и редукције димензија, који побољшавају квалитет и ефикасност анализе.

Као предлог за будуће радове корисно би било упоредити ефикасност различитих филтера на реалним скуповима података из различитих домена, као и истражити могућности комбиновања класичних статичких метода са савременим алгоритмима машинског учења у циљу оптимизације процеса обраде шума. На тај начин могуће је даље унапредити поузданост аналитичких система и обезбедити квалитетније доношење одлука заснованих на подацима.

ЛИТЕРАТУРА

[1] Gavin Wright. *What is noisy data?* 12. април 2024. (https://www.techtarget.com/searchbusinessanalytics/definition/noisy-data)

[2] Salvador García, Julián Luengo, Francisco Herrera. *Data Preprocessing in Data Mining*. 30. август 2014.

[3] Jiawei Han, Micheline Kamber, Jian Pei. *DATA MINING: Concepts and Techniques - 3rd Edition*. 9. јун 2011.

[4] Abhishek U. *Binning Method in Data Cleaning*. 13. децембар 2024. (https://medium.com/@abhishekulligadla/binning-method-in-data-cleaning-307faec44f1d)

[5] Mose Kabungo. *Data Binning Explained*. 18. фебруар 2023. (https://medium.com/@mose.kabungo/binning-explained-557aa3cce591)

[6] Vijay Kanade. *What Is Linear Regression? Types, Equation, Examples, and Best Practices for 2022*. 7. април 2022. (https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/)

[7] Aguinaldo Bezerra Batista Júnior, Paulo Sérgio da Motta Pires. *An Approach to Outlier Detection and Smoothing Applied to a Trajectography Radar Data*. 1. септембар 2014.

[8] Jiawei Yang, Susanto Rahardja, Pasi Fränti. *Smoothing Outlier Scores Is All You Need to Improve Outlier Detectors*. 7. децембар 2023.

[9] Dr. Namita Srivastava, Ruchi Trived. *A Comprehensive Study of Outliers*. 3. март 2022.

[10] Luís P. F. Garcia, Ana C. Lorena, Stan Matwin, André C. P. L. F. de CarvalhoBrad Dwyer. *Ensembles of label noise filters: a ranking approach*. 13. јул 2016.

[11] Shivani Gupta, Atul Gupta. *Dealing with Noise Problem in Machine Learning Data-sets: A Systematic Review*. 24. јул 2019.

[12] Yves van Gennip, Blake Hunter, Anna Ma, Daniel Moyer, Ryan de Vera, Andrea L. Bertozzi. *Unsupervised record matching with noisy and incomplete data*. 23. maj 2018.

[13] Soledad Galli, Surya Tripathi, Harry Snart. *How can you handle duplicate data in machine learning data cleaning?* (https://www.linkedin.com/advice/1/how-can-you-handle-duplicate-data-machine-learning)

[14] Barbara Hammer, Elettra Virgili, Federico Bilotta. *Evidence-based literature review: De-duplication a cornerstone for quality*. 20. децембар 2023.

[15] Tolga Urban, Michael J. Franklin, Laurent Amsalegt. *Cost-based Query Scrambling for Initial Delays*. 1. јун 1998.

[16] Bilal Khan, Azhar Rauf, Sajid Shah, Shah Khusro. *Identification and Removal of Duplicated Records*. јануар 2011.

[17] Zeinab Bahmani, Leopoldo Bertossi, Nikolaos Vasiloglou. *ERBlox: Combining Matching Dependencies with Machine Learning for Entity Resolution*. 7. фебруар 2016.

[18] Barbara Hammer, Elettra Virgili, Federico Bilotta. *Evidence-based literature review: De-duplication a cornerstone for quality*. 10. децембар 2023.

[19] Abid Ali Awan. *Understanding Dimensionality Reduction*. 21. јануар 2025. (https://www.datacamp.com/tutorial/understanding-dimensionality-reduction)

[20] Vijay Kanade. *What Is Dimensionality Reduction? Meaning, Techniques, and Examples*. 22. децембар 2022. (https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-dimensionality-reduction/)