UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET

TRANSFORMACIJA PODATAKA

- SEMINARSKI RAD -

Predmet:

Prikupljanje i predobrada podataka za mašinsko učenje

Mentor: prof. dr Aleksandar Stanimirović

Kandidat: Matija Špeletić (1672)

Niš, 2024.

Sadržaj

1. Uvod

Kvalitet ulaznih podataka značajno utiče na performanse modela mašinskog učenja. Neobrađeni izvorni (*raw*) podaci su često prilično raznolikih tipova i šumoviti, što ih čini težim za korišćenje, pa je samim tim neophodan niz odgovarajućih transformacija i obrađivanja kako bi se oni mogli koristiti za dalje korake u procesu mašinskog učenja. Značajan deo procesa predobrade podataka (preprocesiranja) jesu transformacije nad podacima.

Transformacija podataka podrazumeva različite izmene nad samim podacima kako bi rezultujući podaci bili pogodni za dalji rad i obučavanje različitih modela. Transformacija podataka podrazumeva promenu formata, strukture ili vrednosti unutar skupa podataka, kako bi on postao „čistiji“ i jednostavniji za rad. Ove transformacije obuhvataju različite tehnike, kao što su: skaliranje, normalizacija, enkodiranje, diskretizacija, rad sa ekstremnim vrednostima (engl. *outliers*).

U ovom seminarskom radu će biti detaljno objašnjene različite tehnike u oblasti transformacije podataka, uz diskusiju njihovih prednosti i mana i analizu situacija kada ih treba (ili ne treba) koristiti.

2. Identifikacija problema

U procesu mašinskog učenja, prvi izazov na koji nailazimo jeste, upravo, rad sa podacima. Naime, podaci koji se koriste u mašinskom učenju su često raznovrsni i dolaze iz različitih izvora, kao što su senzori, ljudski unos, mašine, izveštaji itd. što ih čini podložnim raznim nepravilnostima, koje se mogu javiti kao posledica greške prilikom njihovog prikupljanja, kao što su npr. nedostajuće vrednosti, besmisleno velike ili male (ekstremne) vrednosti. Ili prisustvo šuma. Pored toga, podaci koji se prikupljaju mogu biti različitih tipova (tekst, numeričke vrednosti, različite oznake, audio podaci, slike…) i samim tim zahtevaju određeni vid konverzije ili kodiranja, kako bi se mogli upotrebiti za obučavanje modela. Sve ovo predstavljaju izazovi kojima se bavi preprocesiranje podataka.

Jedna od značajnih faza u okviru preprocesiranja podataka, jeste njihova transformacija. Transformacija podataka obuhvata zadatke kao što su: kodiranje podataka, kojim se kategoričkim atributima, vrši adekvatno dodeljivanje numeričkih vrednosti, kako bi se mogli koristiti za dalje korake, skaliranje podataka, kojim se može promeniti opseg vrednosti koje podaci zauzimaju, diskretizacija kontinualnih vrednosti, rad sa ekstremnim vrednostima, normalizacija vrednosti, promena same strukture podataka, agregacija, generalizacija itd.

Naravno, prilikom korišćenja bilo koje tehnike transformacije podataka, uvek treba obratiti pažnju da ne dođe do narušavanja samog značenja određenih vrednosti, koje mogu nositi visok značaj za konkretne podatke. Pored ovoga, uvek je bitno izvršiti analizu dostupnih tehnika za konkretnu situaciju i razmotriti koju od njih je najbolje upotrebiti.

3. Pregled tehnika transformacije podataka

3.1. Skaliranje podataka (*feature scaling*)

Skaliranje podataka ili skaliranje atributa (kolone) predstavlja korak u postupku transformacije podataka u kome se vrednosti odgovarajućeg atributa transformišu tako da se uklope u određeni opseg. Osnovni cilj skaliranja podataka jeste da se izjednači uticaj svih atributa u procesu obučavanja modela. Pojedini algoritmi mašinskog učenja mogu biti veoma osetljivi na veličine (opsege vrednosti) pojedinačnih atributa. Ovde npr. spada većina algoritama koja se zasniva na rastojanju (KNN, *clustering* algoritmi), ali i optimizacioni algoritmi kao što je opadajući gradijent (engl. *gradient descent*). Na ovaj način sprečavamo da pojedini atributi dominiraju u odnose na druge, zbog razlika u njihovoj veličini.

Treba imati u vidu da skaliranje vrednosti ne znači eliminaciju ekstremnih vrednosti. Ekstremne vrednosti ostaju prisutne i nakon skaliranja. Otpornost tehnike skaliranja na prisustvo ekstremnih vrednosti se ogleda u tome koliko ove ekstremne vrednosti utiči na parametre koji se koriste prilikom određivanja novog opsega. Tehnika skaliranja koja je otporna na prisustvo ekstremnih vrednosti će najveći deo vrednosti svesti na sličan opseg, dok će tehnika koja nije otporna na njih rezultovati u svođenju najvećeg dela vrednosti na opseg veoma blizak nuli.

Osnovne tehnike skaliranja obuhvataju: min-max skaliranje (često se naziva i normalizacija), *Z-score* skaliranje (standardizacija), max-abs skaliranje i *robust* skaliranje. [1]

3.1.1. Min-Max skaliranje

Min-max skaliranje, koje se često naziva i normalizacija, predstavlja postupak kojim se vrednosti svode na zadati opseg, najčešće [0, 1]. Za kolonu u datasetu *x*, nove vrednosti dobijene min-max skaliranjem (u slučaju svođenja na opseg [0, 1]) se dobijaju po formuli:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

gde je *x'* nova vrednost. Ukoliko vršimo skaliranje na opseg [*a*, *b*], formula je:

$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$$

Min-max skaliranje je pogodno u situacijama kada se radi sa algoritmima koji su osetljivi na opseg vrednosti (npr. neuronske mreže). Međutim, veoma je važno biti oprezan prilikom korišćenja ove tehnike, jer je ona veoma osetljiva na ekstremne vrednosti, jer i samo jedna ekstremna vrednost može uzrokovati da se skoro svi podaci svedu na vrednosti bliske 0. Zbog ovoga je pre primene min-max, skaliranja neophodno izvršiti izbacivanje (u opštem slučaju obrađivanje) ekstremnih vrednosti ili primeniti skaliranje koje je otpornije na ekstremne vrednosti, ukoliko je neophodno da one ostanu. [2] [3]

3.1.2. *Z-Score* skaliranje (standardizacija)

Standardizacija atributa dovodi srednju vrednost podataka na 0 a standardnu devijaciju na 1. Ova tehnika funkcioniše tako što se, pre svega, odrede srednja vrednost i standardna devijacija. Sledeći korak je da se od svih vrednosti oduzme njihova srednja vrednost (centriranje), a zatim se one podele standardnom devijacijom. Matematički se to može zapisati na sledeći način:

$$x' = \frac{x - \bar{x}}{\sigma}$$

gde je $\bar{x} = \mathrm{mean}(x)$ srednja vrednost, a σ standardna devijacija.

Pogodna je u situacijama kada vrednosti prate normalnu distribuciju (kada *skewness* nije veliki). Standardizacija je otpornija na ekstremne vrednosti u poređenju sa min-max skaliranjem, međutim i dalje može dati loše rezultate u prisustvu velikog broja ekstremnih vrednosti, jer one značajno utiču na srednju vrednost i standardnu devijaciju. Ovu tehniku je pogodno koristiti kada naš skup podataka ima određenu količinu (ne previše) *outlier*-a, koje nije pogodno izbaciti, tj. kada ovi podaci nose informacije od značaja (npr. mali broj ljudi može imati veoma veliki puls, ali to može biti važno u procesu zaključivanja da li osoba ima srčane probleme).

3.1.3. Max-Abs skaliranje

Max-Abs skaliranje vrši skaliranje vrednosti atributa u zavisnosti od maksimalne apsolutne vrednosti. Drugim rečima, vrši se deljenje svih vrednosti maksimalnom apsolutnom vrednošću:

$$x' = \frac{x}{\max(|x|)}$$

Na ovaj način, vrednosti atributa se dovode na opseg koji je približno [-1, 1].

Ova tehnika je veoma slična min-max skaliranju i nije je pogodno koristiti u prisustvu ekstremnih vrednosti, obzirom da će u takvim situacijama najveći deo podataka biti skaliran na opseg blizak nuli.

3.1.4. Skaliranje po medijani i kvantilima (*robust* skaliranje)

Još jedan vid skaliranja, koji je poznat kao *robust* tj. robusno skaliranje, podrazumeva skaliranje u zavisnosti od vrednosti medijane i odgovarajućih kvantila. Ova tehnika obuhvata oduzimanje vrednosti medijane svih primeraka i deljenjem rezultata sa IQR (*Interquartile range*). IQR predstavlja oblast između prvog i trećeg kvartila (detaljnije o IQR u poglavlju 3.5.2):

$$x' = \frac{x - median(x)}{Q_3 - Q_1}$$

gde su Q_3 i Q_1 treći i prvi kvartil, respektivno.

Ova tehnika je najpogodnija u prisustvu ekstremnih vrednosti, jer za razliku od mera kao što su srednja vrednost, standardna devijacija, minimum ili maksimum, na medijanu i kvantile ne utiču ekstremne vrednosti, pa se korišćenje ove tehnike preporučuje u prisustvu *outlier*-a. Naravno, posledica ovakvog skaliranja jeste da je sama oblast na koju su vrednosti skalirane veća u odnosu na prethodno razmatrane načine skaliranja.

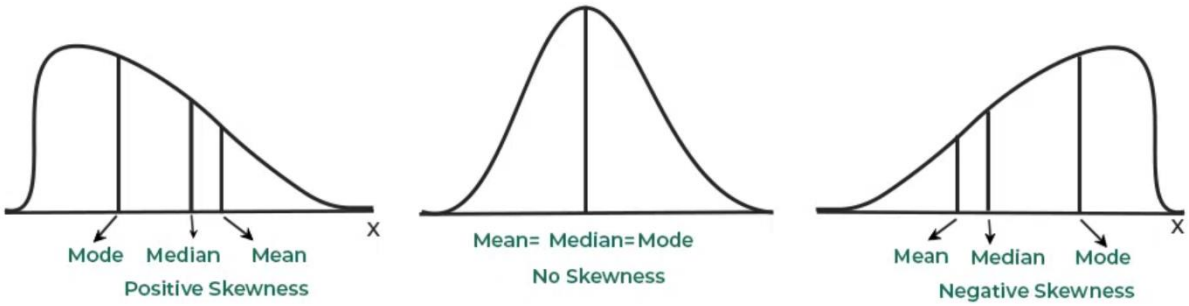
3.1.5. Poređenje različitih tehnika skaliranja

U nastavku je data tabela koja pokazuje prednosti i mane prethodno opisanih tehnika skaliranja podataka.

TEHNIKA	PREDNOSTI	MANE
MIN-MAX SKALIRANJE	Svodi vrednosti na fiksan opseg	U prisustvu *outlier*-a daje neadekvatne rezultate
STANDARDIZACIJA	Otporniji na prisustvo *outlier*-a od Min-Max	Rezultujući opseg vrednosti nije fiksan
MAX-ABS SKALIRANJE	Svodi vrednosti na opseg približno [-1, 1]	U prisustvu *outlier*-a daje neadekvatne rezultate
ROBUST SKALIRANJE	Najotporniji na prisustvo outliera (čak i većeg broja)	Rezultujući opseg nije fiksan i može biti veliki

3.2. Transformacije koje menjaju raspodelu vrednosti

Pored skaliranja, u nekim situacijama je korisno i primeniti nelinearne transformacije koje menjaju raspodelu vrednosti. Cilj ovih tehnika jeste da transformišu podatke tako da raspodela vrednosti što više odgovara normalnoj (Gausovoj) ili uniformnoj (ravnomernoj) raspodeli. Ovakve transformacije je pogodno primenjivati u situacijama kada imamo vrednosti koje prate asimetričnu raspodelu (visok *skewness*, u pozitivnom ili negativnom smeru). Neke od tehnika ovog tipa su: logaritamska transformacija, eksponencijalne transformacije, Box-Cox transformacija, Yeo-Johnson transformacija i kvantilna transformacija. [4] [5]



Slika 1: Ilustracija raspodela sa pozitivnim (levo), bez (sredina) i sa negativnim (desno) skewness-om

3.2.1. Logaritamska transformacija

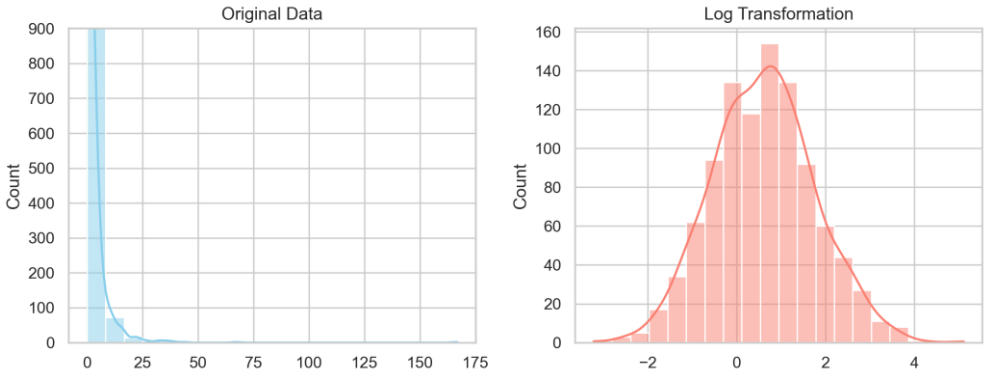
Logaritamska transformacija (*log transform*) može značajno pomoći u situacijama kada su podaci izuzetno asimetrično raspodeljeni i kada postoji veliki broj podataka koji se nalaze daleko od centralnog dela distribucije. Logaritamska transformacija značajnu smanjuje uticaj ekstremnih vrednosti i sužava njihov opseg.

Ova tehnika se primenjuje tako što se nad svim vrednostima primeni logaritamska funkcija, pri čemu se najčešće koristi prirodni logaritam (sa osnovom *e*), a može se koristiti i bilo koja druga vrednost kao osnova:

$$x' = \ln(x) \text{ ili } x' = \log_a(x)$$

Primenjuje se isključivo nad pozitivnim vrednostima, obzirom da logaritamska funkcija nije definisana za 0 i za negativne vrednosti. U situacijama kada u podacima imamo 0 ili negativne vrednosti, pre primene ove transformacija svim vrednostima možemo dodati neku konstantu.

Ova transformacija može dati veoma dobre rezultate u postizanju normalne raspodele. S druge strane, korišćenje logaritamske transformacije može dovesti do gubitka informacija, u situacijama kada su originalne vrednosti važne. [6]



Slika 2: Primer koji ilustruje raspodelu podataka pre i nakon logaritamske transformacije

3.2.2. Eksponencijalna transformacija

Eksponencijalna transformacija podrazumeva stepenovanje svih vrednosti atributa odgovarajućom vrednošću:

$$x' = x^{\lambda}$$

Eksponencijalne transformacije koje se najčešće primenjuju jesu kvadratni koren (λ=1/2), kubni koren (λ=1/3) i recipročna vrednost (λ=-1). Prilikom primene ovih transformacija, treba voditi računa o tome za koje vrednosti je odgovarajuća transformacija definisana, pa je tako npr. kvadratni koren definisan samo za vrednosti veće ili jednake nuli, a recipročna vrednost samo za vrednosti različite od nule.

Kao i logaritamska, ova transformacija u nekim situacijama može dovesti do poboljšanja u raspodeli vrednosti, a isto tako može dovesti i do gubitka informacija. Česta je praksa isprobati više vrednosti λ parametra i odrediti koja vrednost daje najbolji rezultat.

3.2.3. Box-Cox transformacija

Još jedna transformacija koja se često koristi za ispravljanje asimetrične raspodele vrednosti i koja se može posmatrati kao uopštenje logaritamske transformacije i stepenovanja, jeste **Box-Cox transformacija**. Definiše se na sledeći način:

$$x' = \begin{cases} \dfrac{x^{\lambda} - 1}{\lambda}, \lambda \neq 0 \\ \ln(x), \lambda = 0 \end{cases}$$

Ova tehnika je fleksibilnija u odnosu na ostale transformacije, jer dozvoljava podešavanje parametra λ. Za λ=0, transformacija se svodi na logaritamsku. Određivanje parametra λ se može izvršiti empirijski.

Slično logaritamskoj, ova transformacija je definisana samo za pozitivne vrednosti. U slučaju prisustva negativnih vrednosti, može se dodati neka konstanta ili se može upotrebiti *Yeo-Johnson* transformacija koja predstavlja proširenje *Box-Cox* transformacije tako da radi i za negativne vrednosti. Ova transformacija se najčešće primenjuje tako što se isproba za različite vrednosti parametra λ i odredi vrednost koja da je najbolje rezultate.

3.2.4. Yeo-Johnson transformacija

Ova tehnika predstavlja proširenje Box-Cox transformacije tako da radi i za negativne brojeve i nulu. Zbog navedenog, ova transformacija se jednostavnije primenjuje nad različitim tipovima podataka. **Yeo-Johnson transformacija** je data izrazom:

$$x' = \begin{cases} \dfrac{(x+1)^{\lambda}-1}{\lambda}, & \lambda \neq 0, x \geq 0 \\ \ln(x+1), & \lambda = 0, x \geq 0 \\ -\dfrac{(x+1)^{2-\lambda}-1}{2-\lambda}, & \lambda \neq 2, x < 0 \\ -\ln(-x+1), & \lambda = 2, x < 0 \end{cases}$$

Primenjivanje ove transformacije, kao i kod Box-Cox transformacije, podrazumeva isprobavanje za različite vrednosti parametra λ i nalaženje optimalne transformacije.

Ograničenje ove transformacije (a samim tim i prethodno razmatranih) je u tome što one mogu dati optimalne rezultate, samo za određene tipove raspodele i neće uvek dati poželjne rezultate. Pored ovoga, korišćenjem ovih tehnika, moguće je postići isključivo normalnu (Gausovu) raspodelu.

3.2.5. Kvantilna transformacija

Kvantilna (*quantile*) transformacija predstavlja veoma moćnu tehniku za promenu raspodele podataka. Ova tehnika funkcioniše tako što se prvo vrši procena funkcije raspodele (*CDF – cumulative distribution function*) računanjem percentila za sve podatke. Ova funkcija se koristi za mapiranje podataka na uniformnu distribuciju. Zatim, korišćenjem kvantilne funkcije (inverzna funkcija funkcije raspodele) željene distribucije, prethodno dobijene vrednosti se preslikavaju na željenu distribuciju. Matematički izraženo, kvantilna transformacija se može zapisati:

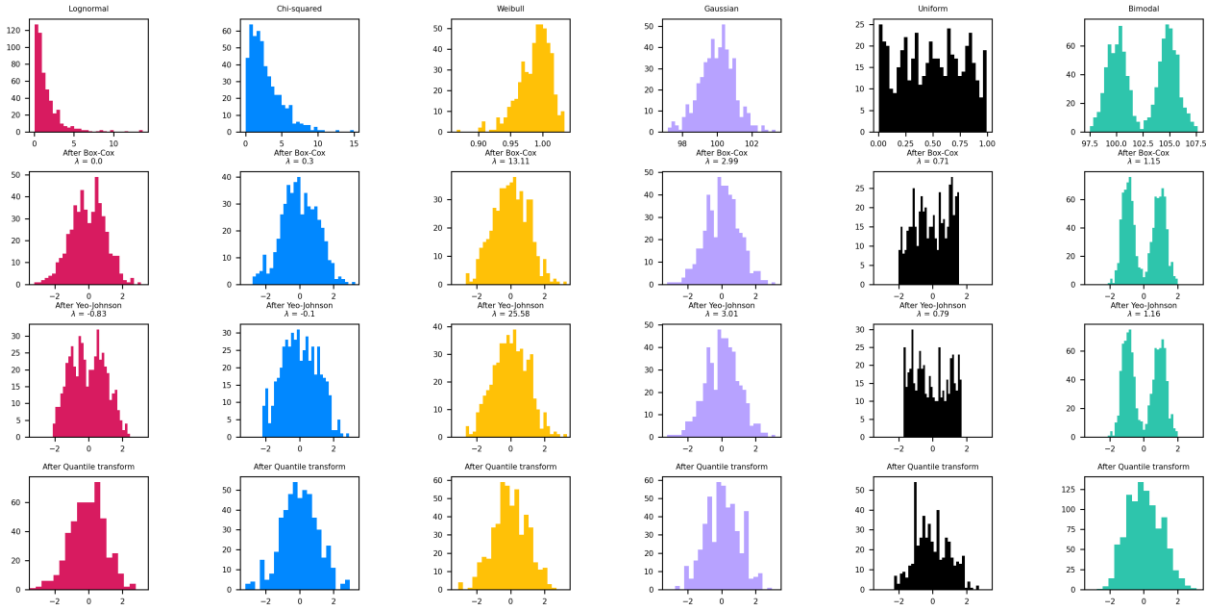
$$x' = G^{-1}(F(x))$$

gde je F – funkcija raspodele promenljive x, a G^{-1} kvantilna funkcija željene distribucije G. Kao željena distribucija, koriste se uniformna ili normalna distribucija.

Kvantilna transformacija predstavlja najmoćniju tehniku za promenu raspodele podataka, jer radi potpuno nezavisno od toga kako su podaci inicijalno raspodeljeni, za razliku od prethodno razmatranih transformacija, koje rade samo sa određenim vrstama inicijalne raspodele. Pored ovoga, kvantilna transformacija ne sadrži parametre, pa nije neophodno tražiti optimalnu transformaciju. Ova transformacija može dati odlične rezultate i otporna je na prisustvo ekstremnih vrednosti i šuma. Osnovni nedostatak ove tehnike je u tome što može dovesti do značajnog gubitka informacija, jer prilikom njene primene dolazi do narušavanja korelacija i rastojanja među atributima, pa je, shodno tome, treba primenjivati sa oprezom. Pored ovoga, kvantilnu transformaciju treba izbegavati prilikom rada sa manjim skupovima podataka, jer njenim korišćenjem može doći do *overfitting*-a. [7] [8]

3.2.6. Poređenje nelinearnih transformacija za promenu raspodele

Sledeći primer (slika 3) ilustruje kako se prethodno razmatrane transformacije (Box-Cox, Yeo-Johnson i kvantila) ponašaju sa različitim tipovima početne raspodele podataka. Cilj je promeniti raspodelu podataka tako da se postigne normalna raspodela. Početne distribucije podataka nad kojima ove metode testiramo obuhvataju: lognormalnu, „hi-kvadrat“, *weibull*, normalnu, uniformnu i bimodalnu (sastoji se od dva „Gausova zvona“).



Slika 3: Poređenje različitih algoritama za promenu raspodele (Box-Cox, Yeo-Johnson, Quantile – dati po redovima) nad različitim tipovima raspodele (log-normalna, chi-squared, weibull, normalna, uniformna, bimodalna – date po kolonama)

Primećujemo da za većinu raspodela, sve tehnike daju relativno dobre rezultate. Međutim, u slučaju uniformne i bimodalne početne raspodele, jedino je kvantilna transformacija uspela da preslika podatke na približno normalnu raspodelu, jer je ovo jedina transformacija koja može da preslika proizvoljnu raspodelu na normalnu. [9] Bitno je imati u vidu da, iako kvantilna transformacija radi uvek, u situaciji kada imamo npr. bimodalnu raspodelu, veoma je verovatno da nije požekjno primeniti ovakvu tehniku, jer takva raspodela podataka sa sobom potencijalno nosi informacije od značaja. U nastavku je data tabela koja prikazuje prednosti i mane korišćenja prethodno opisanih transformacija nad podacima koje menjaju raspodelu vrednosti.

TRANSFORMACIJA	PREDNOSTI	MANE
LOGARITAMSKA	Daje odlične rezultate u slučaju raspodele nalik log-normalnoj	Ne radi sa vrednostima koje su ≤ 0; može dovesti do gubitka informacija
EKSPONENCIJALNA	U određenim situacijama može poboljšati raspodeli	Potrebno je izabrati eksponent; ponekad dovodi do gubitka informacija
BOX-COX	Daje dobre rezultate kod raznih tipova raspodele	Ne radi sa vrednostima koje su ≤ 0, može dovesti do gubitka informacija
YEO-JOHNSON	Daje dobre rezultate kod raznih raspodela, radi i sa pozitivnim i sa negativnim vrednostima	Može dovesti do gubitka informacija
KVANTILNA	Radi uvek i daje dobre rezultate	Često dovodi do većeg gubitka informacija

3.3. Normalizacija

Normalizacija predstavlja proces skaliranja pojedinačnih primeraka (redova) na jediničnu normu (tako da odgovarajuća norma bude 1). Norma vektora predstavlja funkciju koja preslikava vektorski prostor u skup nenegativnih realnih brojeva, tako da definiše rastojanje od koordinatnog početka. [8] Najčešće korišćene norme u postupku normalizacije obuhvataju:

- **L1 normu (Menhetn rastojanje)** – određuje se kao zbir apsolutnih vrednosti komponenata vektora:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

- **L2 normu (Euklidovo rastojanje)** – određuje se kao koren zbira kvadrata komponenata vektora:

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|}$$

- **L∞ normu (max norma)** – određuje se kao maksimum apsolutnih vrednosti komponenata vektora:

$$\|x\|_\infty = \max(|x_1|, \dots, |x_n|)$$

Gde je $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ dat vektor dimenzije *n*.

> ***NAPOMENA**: Bitno je napraviti razliku – skaliranje atributa, o kome je bilo reči u poglavlju 3.1, se odnosi na promenu opsega koji atribut (kolona) zauzima, dok se normalizacija odnosi na promenu norme primeraka (redova).*

Ovaj postupak se najčešće koristi kada je neophodno određivanje sličnosti između proizvoljna dva primerka iz skupa. Određivanje sličnosti između vektora se često koristi prilikom rada sa tekstualnim dokumentima, gde se tekst modeluje vektorom, a sličnost dva teksta rastojanjem između ova dva vektora, što je veoma korisno u zadacima kao što su pretraživanje informacija, klasifikacija i klasterizacija tekstova.

3.4. Enkodiranje kategoričkih atributa

Enkodiranje podataka je jedan od veoma važnih koraka u postupku preprocesiranja. Odnosi se na konverziju kategoričkih, odnosno tekstualnih podataka u numerički format, kako bi se ovi podaci mogli koristiti kao ulaz algoritama mašinskog učenja, obzirom da većina ovih algoritama očekuje numeričke vrednosti na svom ulazu.

Kategorički podaci su najčešće zadati u *string* formatu i ima konačno mnogo različitih vrednosti. Postoje dve vrste kategoričkih podataka: ordinalni i nominalni.

- **Ordinalni podaci**: Kategorijama definisanim ordinalnim podacima je pridružen neki poredak, odnosno one se mogu rangirati ili sortirati. Primer ordinalnih podataka bi mogao da bude stepen obrazovanja, npr:

  High school < Bachelors degree < Masters degree < PhD

- **Nominalni podaci**: Kategorije koje definišu nominalni podaci ne prate nikakav poredak i ne mogu se međusobno rangirati niti sortirati. Primer nominalnih podataka mogu biti gradovi (npr. grad u kome neko lice stanuje):

  Niš, Beograd, Novi Sad, Kragujevac, Vranje

Izbor načina enkodiranja vrednosti može imati značajan uticaj na performanse modela koje kasnije obučavamo. Zbog ovoga je veoma važno ispravno izabrati način enkodiranja na osnovu tipa i prirode samih podataka.

Neki od najčešće korišćenih metoda za enkodiranje kategoričkih atributa jesu: ordinalno, *label*, *one-hot* (i *dummy*) enkodiranje, a još neke od tehnika koje se nešto ređe upotrebljavaju jesu binarno, *count* i *target* enkodiranje [10] [11].

3.4.1. *Label* enkodiranje

Ova tehnika podrazumeva dodeljivanje jedinstvene celobrojne konstante svakoj kategoriji. Dodeljivanje se vrši nasumično (najčešće po redosledu pojavljivanja kategorija). ***Label* enkodiranje** se može primenjivati nad kategorijama koje nemaju jasan poredak, odnosno nad nominalnim podacima. U nastavku je dat primer koji ilustruje primenu ove tehnike nad kolonom koja označava grad:

Gradovi
Niš
Beograd
Niš
Novi Sad
Beograd

Mapiranje	
Niš	1
Beograd	2
Novi Sad	3

Gradovi (enkodirani)
1
2
1
3
2

Primer 1: Primena label enkodiranja nad kolonom koja sadrži informacije o gradovima

Ovo je jedna od najjednostavnijih tehnika i primenjuje se dosta lako. Nedostatak ove tehnike leži u tome što se ovako dodeljene celobrojne vrednosti mogu pogrešno interpretirati od strane modela koji se obučava. Naime, ovako enkodirane kategorije se mogu interpretirati kao da postoji jasan poredak između njih, iako ga nema, što može dovesti do neispravnih zaključaka.

3.4.2. Ordinalno enkodiranje

Ordinalno enkodiranje se može koristiti kada podaci sami po sebi imaju prirodan poredak. Primenjuje se veoma slično kao i *label* enkodiranje – dodeljivanjem celobrojnih konstanti kategorijama, osim što se kod ove tehnike celobrojne konstante dodeljuju tako da prate poredak klasa. U nastavku sledi primer koji ilustruje korišćenje ove tehnike nad kolonom koja označava veličinu odeće:

Veličine
M
S
L
L
M

Mapiranje	
S	1
M	2
L	3

Veličine (enkodirane)
2
1
3
3
2

Primer 2: Primena ordinalnog enkodiranja nad kolonom koja sadrži informacije o veličini odeće.

Kada koristimo ordinalno enkodiranje, poredak kategorija se zadaje unapred.

3.4.3. *One-Hot* (i *Dummy*) enkodiranje

***One-Hot* enkodiranje** predstavlja jednu od najčešće korišćenih tehnika za enkodiranje kategoričkih atributa, kada je reč o nominalnim podacima. Ova tehnika podrazumeva kreiranje posebnog binarnog atributa za svaku od jedinstvenih kategorija. Ukoliko primerak pripada odgovarajućoj kategoriji, vrednost tog atributa će biti 1, a ostalih kolona 0. Kolona sa N jedinstvenih kategorija se, prilikom *one-hot* enkodiranja, zamenjuje sa N kolona, gde svaka kolona odgovara jednoj kategoriji. U nastavku sledi primer koji ilustruje korišćenje ove tehnike nad kolonom koja označava grad:

Gradovi
Niš
Beograd
Niš
Novi Sad
Beograd

Niš	Beograd	Novi Sad
1	0	0
0	1	0
1	0	0
0	0	1
0	1	0

Primer 3: Primena one-hot enkodiranja nad kolonom koja sadrži podatke o gradovima.

Osnovna prednost ovakve tehnike enkodiranja leži u tome što na ovaj način sve kategorije postaju ravnopravne, za razliku od *label* enkodiranja, kod koga kategorije dobijaju poredak, koji u realnosti ne postoji. Zbog ovoga, korišćenje ove tehnike može značajno doprineti poboljšanju performansi različitih algoritama mašinskog učenja.

S druge strane, veliki nedostatak ove tehnike jeste povećanje dimenzionalnosti podataka. Naime, skupu podataka se dodaje onoliko kolona koliko ima kategorija u kategoričkoj koloni čije vrednosti enkodiramo. Ovo može rezultovati u veoma kompleksnim modelima sa velikim brojem ulaza, za čije obučavanje je potrebno dosta vremena. Pored ovoga, sami podaci postaju retko posednuti, obzirom da će većina *one-hot* enkodiranih kolona imati vrednost 0. Sve ovo, posebno u situacijama kada imamo veliki broj kolona, a mali broj primeraka, može dovesti i do *overfitting*-a. Zbog ovoga, iako ova tehnika često može dati dobre rezultate, treba biti oprezan sa njenim korišćenjem, a u situacijama kada je broj kategorija relativno velik, poželjno je razmotriti i druge tehnike enkodiranja.

Još jedna tehnika koja je gotovo identična kao i *one-hot*, jeste *dummy* enkodiranje. ***Dummy* enkodiranje** funkcioniše na isti način, uz razliku da se jedna kolona enkodira sa svim nulama, pa se zato kolona sa N jedinstvenih kategorija, prilikom *dummy* enkodiranja zamenjuje sa N-1 kolonom (ili drugim rečima, *dummy* eknodiranje kolone se dobija kada se jedna, proizvoljna, kolona se izbacuje nakon *one-hot* enkodiranja). Sva razmatranja koja važe za *one-hot*, važe i za *dummy* enkodiranje.

3.4.4. Binarno enkodiranje

Binarno enkodiranje je tehnika kod koje se svaka kategorija enkodira binarnim vrednostima. Skupu podataka se dodaje onoliko kolona, koliko je binarnih cifara potrebno za enkodiranje svih jedinstvenih kategorija, odnosno, kolona sa N jedinstvenih kategorija se zamenjuje sa $\log_2(N)$ kolona. Na primeru sa kolonom koja sadrži informacije o gradu, primena binarnog enkodiranja bi izgledala na sledeći način:

Gradovi
Niš
Beograd
Niš
Novi Sad
Beograd

Mapiranje	
Niš	00
Beograd	01
Novi Sad	10

Grad_1	Grad_0
0	0
0	1
0	0
1	0
0	1

Primer 4: Primena binarnog enkodiranja nad kolonom koja sadrži podatke o gradovima.

Ova tehnika se može iskoristiti kao zamena za *one-hot* enkodiranje u situacijama kada je broj kategorija veliki, inače se ne koristi previše često. Izuzetak su situacije kada postoje samo dve kategorije, kada se najčešće one kodiraju binarno tako što se jednoj kategoriji dodeli vrednost 1, a drugoj 0 (mada ovo takođe odgovara i tehnikama koje definišu *label* i ordinalno enkodiranje, obzirom da postoje samo dve vrednosti najčešće se koristi termin binarno enkodiranje).

3.4.5. *Count (frequency)* enkodiranje

***Count* enkodiranje** ili enkodiranje frekvencom pojavljivanja, predstavlja tehniku u kojoj se svaka kategorija enkodira brojem njenog pojavljivanja u ovoj koloni. Ova tehnika može biti korisna u situacijama kada u koloni postoji veoma veliki broj jedinstvenih kategorija (npr. više hiljada) i kada prethodno razmatrane tehnike nisu primenljive, a inače se koristi veoma retko. Neki slučajevi u kojima se ova tehnika može iskoristiti bi mogli da budu npr. oznake veb sajtova (veb sajt na kome je korisnik kliknuo na reklamu u marketinškim podacima) ili proizvoda (podaci sa online prodavnica) i sl. U nastavku je dat primer koji ilustruje primenu ove tehnike:

Proizvod
Jabuka
Banana
Banana
Jabuka
Banana

Broj pojavljivanja	
Jabuka	2
Banana	3

Proizvod (enkodiran)
2
3
3
2
3

Primer 5: Primena count enkodiranja nad kolonom koja sadrži proizvode.

Ova tehnika se koristi samo u specijalnim situacijama. Primenjuje se veoma jednostavno i ne utiče na povećanje dimenzionalnosti za veoma veliki broj kategorija. Nedostatak tehnike je u tome što su kolizije veoma česte – različite kategorije se mogu enkodirati istim vrednostima, a pored ovoga, dodavanje novih podataka u trening set nije moguće bez ponovnog kodiranja.

3.4.6. *Target* enkodiranje

Još jedna tehnika enkodiranja koja može biti korisna u situacijama kada u koloni postoji veoma veliki broj kategorija i kada druge tehnike enkodiranja nisu primenljive jeste *target* enkodiranje. Izvan ovakvih situacija, ova tehnika se retko koristi. *Target* enkodiranje u obzir uzima i ciljani atribut čiju vrednost model koji treniramo treba da odredi (*target value*). U zavisnosti od problema koji se rešava, ciljani atribut može biti binarni ili kontinualan.

U slučaju binarnog ciljanog atributa, svaka vrednost se enkodira verovatnoćom da ciljani atribut ima vrednost 1 za datu vrednost atributa, čije se enkodiranje vrši. Kako bi se rešio problem kada veoma mali broj primeraka pripada nekoj kategoriji, ova verovatnoća se „meša“ sa verovatnoćom da ciljani atribut ima vrednost 1, uz odgovarajući težinski faktor. Ovaj princip se može izraziti sledećom formulom:

$$S_i = \lambda(n_i)\frac{n_{iY}}{n_i} + (1 - \lambda(n_i))\frac{n_Y}{n_{TR}}$$

Gde je S_i procena verovatnoće kojom se atribut enkodira, n_{iY} broj primeraka koji pripada kategoriji i za koji ciljani atribut ima vrednost 1, n_i broj primeraka koji pripada kategoriji i, n_Y broj primeraka za koji ciljani atribut ima vrednost 1 i n_{TR} ukupan broj primeraka u skupu podataka za obučavanje. λ je težinski faktor koji se određje na osnovu broja primeraka koji pripada datoj kategoiji.

Kada su u pitanju kontinualni atributi, prethodna formula se može proširiti tako što se umesto verovatnoće računa srednja vrednost, na sledeći način:

$$S_i = \lambda(n_i)\frac{\sum_{k\in L_i} Y_k}{n_i} + (1 - \lambda(n_i))\frac{\sum_{k=1}^{N_{TR}} Y_k}{n_{TR}}$$

Gde je Y_k vrednost ciljanog atributa za k-ti primerak, a L_i skup primeraka koji pripadaju kategoriji i. [12]

Ova tehnika ne dovodi do povećanja dimenzionalnosti, ali može uzrokovati „curenje“ ciljanog fičera, što dovodi do *overfitting*-a modela. Pored ovoga, dodavanje novih podataka trening skupu, bez ponovnog enkodiranja nije moguće. Iz navedenih razloga, ovu tehniku ne treba primenjivati kada je broj kategorija mali i moguće je koristiti ovu tehniku isključivo u slučaju nadgledanog obučavanja (jer je neophodno poznavati vrednost ciljanog atributa).

3.4.8. Poređenje različitih tehnika enkodiranja

U nastavku je data tabela koja prikazuje kada se koja od prethodno opisanih tehnika za enkodiranje treba primenjivati.

ENKODIRANJE	KADA PRIMENJIVATI
LABEL	Kada kategorije ne poseduju poredak (nominalne); često je bolje izbegavati, jer uvodi poredak koji u realnosti ne postoji
ORDINALNO	Kada kategorije poseduju poredak (ordinalne)
ONE-HOT	Kada je broj kategorija relativno mali i one ne poseduju poredak; jednostavno rešenje za nedostajuće vrednosti
BINARNO	Retko se koristi (izuzetak je kada postoje dve kategorije[1]); pogodno kada je broj kategorija veći
COUNT	Kada je broj kategorija veoma velik
TARGET	Kada je broj kategorija veoma velik, ne koristi se toliko često i može dovesti do „curenja“ ciljanog atributa

Treba imati u vidu da je u slučaju velikog broja kategorija često pogodnije primeniti određenu vrstu grupisanja kategorija i svesti vrednosti atributa na relativno mali broj jedinstvenih kategorija, nego primenjivati tehnike enkodiranja koje su specijalizovane za ovakve situacije.

3.5. Diskretizacija

Diskretizacija, ili *binning*, predstavlja proces u kome transformišemo kontinualne promenljive u diskretne promenljive tako što formiramo skup uzastopnih intervala, koji se nazivaju *bin*-ovi, koji su takvi da zajedno obuhvataju ceo opseg vrednosti ove promenljive. Diskretizacija se može koristiti da promeni distribuciju podataka sa visokim *skewness*-om ili da minimalizuje uticaj ekstremnih vrednosti i time poboljša performanse pojedinih modela mašinskog učenja.

Diskretizacija smanjuje uticaj ekstremnih vrednosti tako što ih grupiše i smešta u iste binove zajedno sa *inlier* vrednostima date raspodele podataka. Diskretizacija se takođe može iskoristiti da potpuno ujednači raspodelu podataka sa visokim *skewness*-om u slučaju formiranja binova sa (približno) istim brojem primeraka.

Tehnike u okviru diskretizacije se mogu podeliti na nenadgledane i nadgledane. Tehnike nenadgledane diskretizacije ne koriste nikakve dodatne informacije, osim distribucije vrednosti promenljive koju diskretizujemo, za kreiranje binova. S druge strane, nadgledane tehnike diskretizacije koriste vrednost ciljanog atributa za formiranje intervala.

Neke od često korišćenih metoda za diskretizaciju obuhvataju: podela na proizvoljno zadate intervale, podelu na intervale jednake širine, raspoređivanje u intervale jednakih frekvenci, diskretizacija korišćenjem *clustering*-a i diskretizacija korišćenjem stabala odlučivanja. [13] [14]

[1] Kada postoje samo dve kategorije, binarno, label i ordinalno enkodiranje bi dali iste rezultate

3.5.1. Podela na proizvoljno zadate intervale

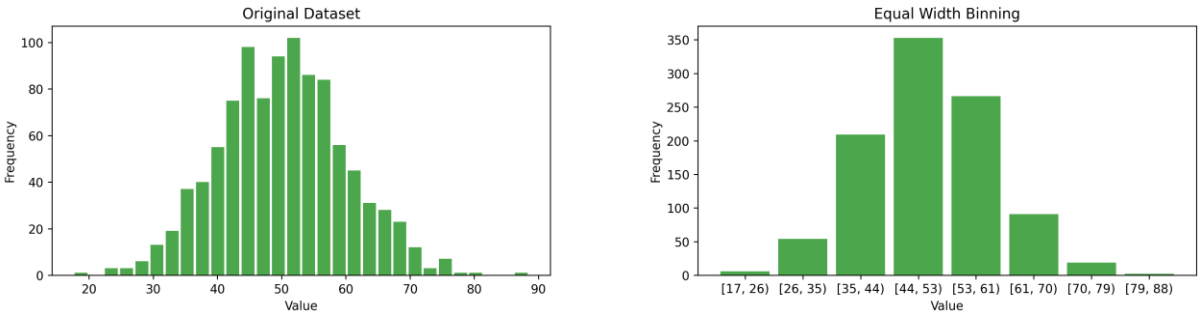
Najjednostavnija tehnika za diskretizaciju kontinualnog atributa jeste podela na proizvoljno zadate intervale. Iako postoje i naprednije tehnike za diskretizaciju, u nekim situacijama je potrebno intervale zadati ručno. Primeri slučajeva u kojima se ova tehnika može upotrebiti mogu biti podaci za koje postoji jasno definisana podela po intervalima (npr. ukoliko umesto broja poena želimo da koristimo ocenu, neophodno je ručno zadati opsege za odgovarajuće ocene) ili u konsultaciji sa domenskim ekspertom.

3.5.2. Raspoređivanje u intervale jednakih širina

Ova tehnika podrazumeva podelu opsega vrednosti kontinualne promenljive na unapred zadat broj intervala jednake širine. Broj intervala se može zadati proizvoljno a širina pojedinačnih intervala se, za kontinualnu promenljivu *X*, određuje na sledeći način:

$$Širina = \frac{\max(X) - \min(X)}{BrojBinova}$$

pri čemu će interval za prvi bin biti $\left[\min(X), \min(X) + Širina\right]$, za sledeći $\left[\min(X) + Širina, \min(X) + 2 \times Širina\right]$, itd. Npr. za opseg vrednosti od 0 do 100 i za 4 bina, intervali će biti: 0-25, 25-50, 50-75 i 75-100.

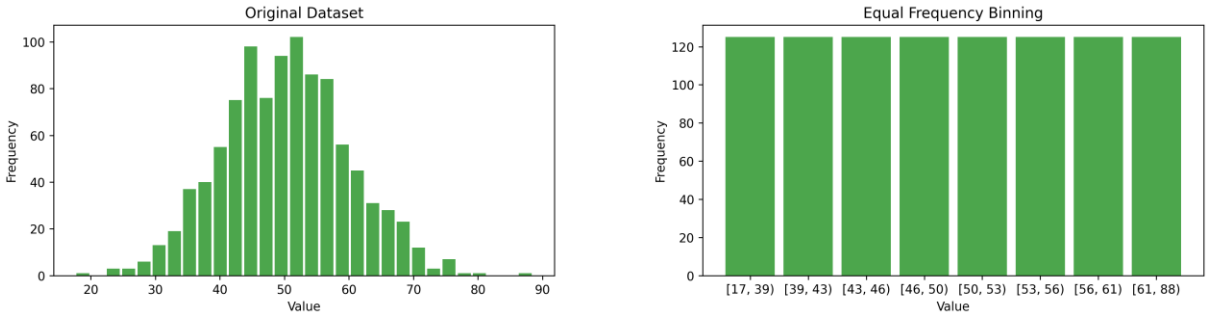


Slika 4: Primena tehnike diskretizacije korišćenjem binova jednakih širina nad podacima koji prate normalnu raspodelu.

Možemo primetiti i na slici 4, da raspodela podataka ostaje ista kao i originalna. Ova tehnika je veoma osetljiva na ekstremne vrednosti i u takvim situacijama rezultuje u neadekvatnoj podeli na intervale, jer će npr. jedna vrednost koja je mnogo veća od ostalih uticati na to da poslednji bin ima samo tu vrednost, prvih nekoliko binova sve ostale vrednosti, dok će ostali binovi između ostati prazni.

3.5.3. Podela na intervale sa jednakom frekvencom (istim brojem primeraka)

Ova tehnika diskretizacije deli opseg kontinualne promenljive na intervale koji su takvi da sadrže isti ili približno isti broj primeraka. Širina intervala se određuje u zavisnosti od kvantila i različiti intervali imaju različite širine. Ova tehnika funkcioniše tako što se kontinualna promenljiva podeli na unapred zadati broj kvantila. Kvantili se određuju tako što se vrednosti promenljive sortiraju, u ovako sortiranom nizu se određuju pozicije odgovarajućih percentila, a zatim se uzimaju vrednosti u nizu na tim pozicijama (drugim rečima, sortirani niz vrednosti se deli na jednake intervale, a granične vrednosti unutar intervala predstavljaju kvantile).

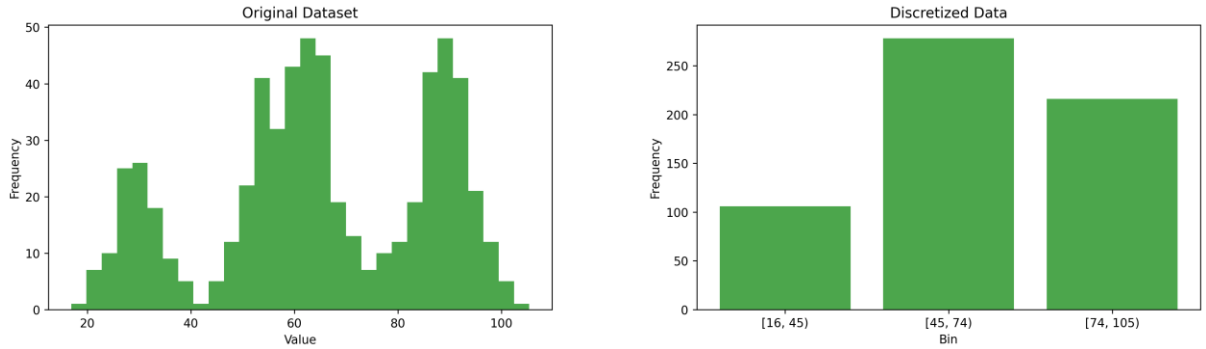


Slika 5: Primena tehnike diskretizacije korišćenjem binova sa istim brojem primeraka nad podacima koji prate normalnu raspodelu.

Možemo primetiti na slici 5 da, nakon diskretizacije korišćenjem binova sa istim brojem primeraka, rezultujuća raspodela postaje uniformna (ravnomerna). Ova tehnika je nešto kompleksnija od prethodne i rezultuje u binovima koji nisu iste širine, ali radi odlično u prisustvu ekstremnih vrednosti i daje dobre rezultate sa podacima koji imaju zakrivljenu raspodelu.

3.5.3. Diskretizacija korišćenjem klasterizacije

Ova metoda podrazumeva korišćenje tehnike klasterizacije (najčešće se koristi *k-means* algoritam, koji će i ovde biti razmatran) za podelu na intervale. Broj klastera se zadaje od strane korisnika. *K-means* klasterizacija se odvija po sledećem principu: U prvom koraku, bira se *k* nasumičnih primeraka i oni se uzimaju za centre *k* klastera, a ostali primerci se dodeljuju najbližem klasteru. U ostalim koracima, koji se obavljaju iterativno, centri klastera se ponovo izračunavaju kao težišne tačke (srednje vrednosti) trenutnih klastera, a zatim se ponovo vrši dodeljivanje primeraka najbližim klasterima. Algoritam se ponavlja sve dok se ne pronađu optimalni centri klastera (dok razlika između prethodnih i novih centara klastera ne postane zanemarljivo mala) ili dok se ne postigne maksimalan broj iteracija.



Slika 6: Primena k-means diskretizacije nad podacima, koji su grupisani u tri regiona veće gustine raspodele. Možemo primetiti da primenom k-means diskretizacije (za k=3) svi podaci iz (približno) iste „grupe“ dobijaju istu vrednost.

Ova tehnika može dati odlične rezultate, jer određuje binove adaptivno u zavisnosti od distribucije i time može detektovati prirodno grupisanje vrednosti kontinualne promenljive, a pored toga je prilično otporna na ekstremne vrednosti. Naravno, u odnosu na ostale tehnike, klasterizacija je dosta zahtevnija u pogledu kompleksnosti i u nekim situacijama, npr. u slučaju normalne distribucije podataka, korišćenje ovakve tehnike nema smisla. Pored ovoga, u slučajevima malih skupova podataka, primena klasterizacije može dovesti do *overfitting*-a.

3.5.4. Diskretizacija korišćenjem stabla odlučivanja

Jedna od naprednijih tehnika diskretizacije je upravo korišćenje stabla odlučivanja za određivanje optimalnih binova i spada u kategoriju tehnika za nadgledanu diskretizaciju. Stablo odlučivanja se konstruiše na osnovu atributa čiju diskretizaciju vršimo i ciljanog atributa. Predikcija stabla odlučivanja podrazumeva dodeljivanje primerka nekom od svojih N listova, pa samim tim, stabla odlučivanja daju

diskretan izlaz. Diskretizacija korišćenjem stabla odlučivanja kreira monotonu vezu između binova i ciljanog atributa. Dubina stabla određuje broj listova.

Glavna prednost ovakve tehnike leži u mogućnosti pronalaženja kompleksnih veza među podacima i optimalne podele na binove, a pored ovoga, formira se i monotona veza između binova i ciljanog atributa. Glavni nedostatak ove tehnike je, kao i kod većine algoritama mašinskog učenja, mogućnost uzrokovanja *overfitting*-a. Pored ovoga, neophodno je i optimizovati hiperparametre stabla koje se koristi za diskretizaciju. [15]

3.5.5. Poređenje različitih tehnika diskretizacije

Tehniku diskretizacije je često potrebno izabrati uz konsultaciju sa domenskim ekspertom, jer primena ovih tehnika gotovo uvek zavisi od konkretnog domena i samih podataka. Diskretizacija je generalno veoma dobra tehnika jer poboljšava interpretabilnost, smanjuje šum i rešava problem *outlier*-a. Neke od karakteristika ovih tehnika koje treba imati u vidu su navedene u sledećoj tabeli.

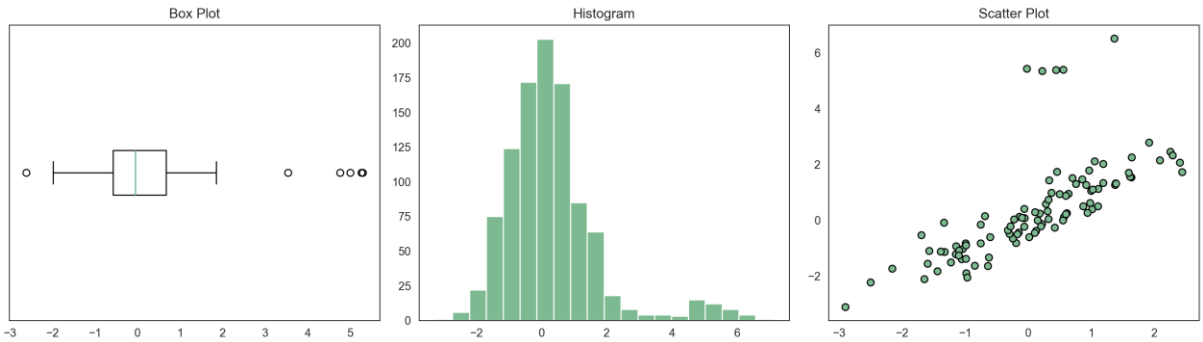
DISKRETIZACIJA	PREDNOSTI	MANE
EQUAL-WIDTH	Rezultujuća raspodela prati originalnu	Radi loše u prisustvu *outlier*-a
EQUAL-FREQUENCY	*Outlier*-i nemaju nikakav uticaj	Rezultujuća raspodela ne prati originalnu
K-MEANS	Radi odlično kada u originalnoj raspodeli postoje jasne tačke nagomilavanja	Ukoliko nema jasno izdvojenih grupa, nema je smisla primenjivati
STABLA ODLUČIVANJA	Može značajno poboljšati rezultate modela	Kompleksnija tehnika, može dovesti do *overfitting*-a

3.6. Rad sa *outlier*-ima (ekstremnim vrednostima)

Outlier ili ekstremna vrednost jeste vrednost koja značajno odstupa od svih ostalih podataka. Statistički parametri kao što su srednja vrednost i varijansa su veoma osetljivi na prisustvo ekstremnih vrednosti. *Outlier*-i takođe mogu loše uticati na performanse modela mašninskog učenja. Iz ovih razloga, često je neophodno koristiti razne tehnike koje služe za uklanjanje ili izmenu vrednosti *outlier*-a.

Postoji puno različitih tehnika koje se mogu koristiti prilikom rada sa *outlier*-ima. Jedna od njih jeste diskretizacija o kojoj je bilo reči u prethodnom poglavlju. Primenom diskretizacije, svi *outlier*-i budu svrstani u neki od binova i time se njihov uticaj ukloni. Još jedna tehnika koja se može koristiti (nakon identifikacije *outlier*-a) jeste rad sa nedostajućim podacima. Ideja kod ovog pristupa je da se umesto njihovog izbacivanja, *outlier*-i tretiraju kao nedostajući podaci i da se izvrši imputacija njihovih vrednosti. Još jedna tehnika koja se može koristiti u radu sa outlier-ima jeste i promena raspodele podataka, kao što je npr. logaritamska transformacija, koja može dati odlične rezultate za jako zakrivljene distribucije. Najjednostavnija tehnika jeste, svakako, izbacivanje primeraka koji sadrže *outlier*-e ili ograničavanje minimalnih i maksimalnih vrednosti.

Postojanje *outlier*-a je najjednostavnije utvrditi korišćenjem tehnika za vizuelizaciju kao što su box-plot dijagrami, histogrami i scatter-plot dijagrami. *Outlier*-i se mogu prepoznati kao tačke koje su značajno udaljene od glavnog dela raspodele podataka na ovim dijagramima (slika 7).



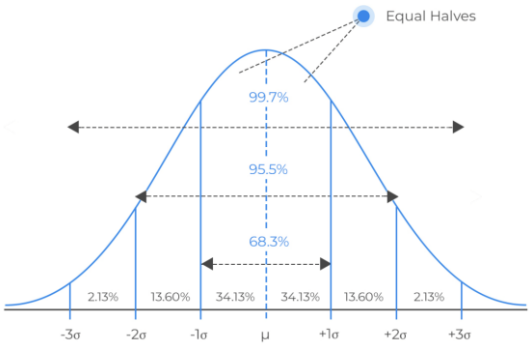
Slika 7: Različite tehnike vizuelizacije koje se najčešće koriste za vizuelnu proveru prisustva outlier-a: box-plot (levo), histogram (sredina), scatter-plot (desno).

Neke od osnovnih tehnika za detekciju *outlier*-a, koje se koriste za detekciju na osnovu pojedinačnih atributa obuhvataju: *Z-score* i modifikovani *Z-score* metod i IQR metod. Od naprednijih tehnika koje vrše automatsku detekciju *outlier*-a na osnovu svih atributa u skupu podataka, tu su uklapanje eliptičnog omotača (*elliptic envelope*), izolacione šume (*isolation forest*) i faktor lokalnih ekstremnih vrednosti (*local outlier factor*) i još mnoge druge. Od tehnika za izmenu vrednosti ili uklanjanje *outlier*-a, tu su jednostavno izbacivanje ekstremnih vrednosti i vinzorizacija (*winsorization*), pri čemu se mogu koristiti i većina tehnika u oblasti rada sa nedostajućim vrednostima. [16]

3.6.1. Detekcija *outlier*-a pomoću *Z-score* metode

Z-score predstavlja način predstavljanja podataka korišćenjem srednje vrednosti i standardne devijacije. Računanje *z-score-a* podrazumeva transformisanje vrednosti tako da njihova srednja vrednost bude 0, a standardna devijacija 1. *Z-score* predstavlja udaljenost podataka od srednje vrednosti u jedinicama standardne devijacije. [17]

Z-score metoda za detekciju *outlier*-a podrazumeva označavanje svih vrednosti čiji je *z-score* po apsolutnoj vrednosti veći od nekog zadatog praga. Za vrednost praga se najčešće uzima 3, obzirom da normalna distribucija podrazumeva da 99.7% podataka upada u opseg od -3σ do +3σ (slika 8).



Slika 8: Grafička ilustracija z-score vrednosti. Z-score se odnosi na broj jedinica standardnih devijacija koliko je neki podatak udaljen od srednje vrednosti.

Ova metoda daje najbolje rezultate kada raspodela što više odgovara normalnoj raspodeli (nizak *skewness*). U slučaju zakrivljenih raspodela, poželjno je razmotriti i druge metode. Pored ovoga, *z-score* metoda se oslanja na srednju vrednost i na standardnu devijaciju, koje, same po sebi, nisu otporne na prisustvo velikog broja *outlier*-a.

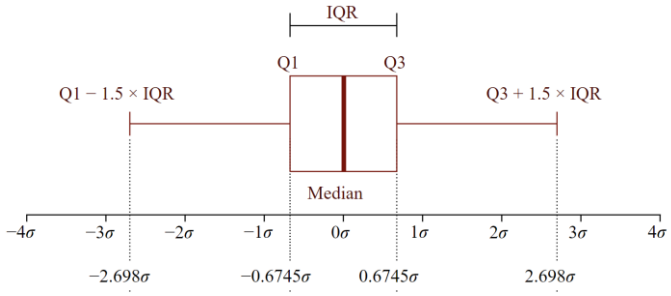
Z-score metoda ima i svoju modifikovanu verziju koja je manje otporna na *outlier*-e, koja umesto srednje vrednosti i standardne devijacije koristi medijanu i MAD (*Median Absolute Deviation* – medijana apsolutnih devijacija svih vrednosti od medijane istih).

3.6.2. Detekcija *outlier*-a pomoću IQR metode

IQR (*Interquartile Range)* se koristi za merenje varijabilnosti unutar podataka, deljenjem podataka na kvartile. Kvartili se izračunavaju sortiranjem podataka u rastući poredak i njihovim deljenjem na četiri jednaka dela. Prvi, drugi i treći kvartil (najčešće se označavaju sa Q1, Q2 i Q3, respektivno) su vrednosti na granicama između ova četiri jednaka dela. Ovi kvartili su 25-ti, 50-ti i 75-ti percentili datih podataka (k-ti percentil je vrednost koja je takva da se tačno k procenata podataka nalazi iza ove vrednosti). IQR predstavlja opseg između prvog i trećeg kvartila i obuhvata 50% podataka. [18]

$$IQR = Q_3 - Q_1$$

Po IQR metodi, sve vrednosti koje se nalaze ispod $Q_1 - 1.5 \times IQR$ ili iznad $Q_3 + 1.5 \times IQR$ se označavju kao *outlier-i* (slika 9). [17]



Slika 9: Grafička reprezentacija inter-kvartilnog opsega (IQR). Možemo primetiti da se na sredini nalazi medijana umesto srednje vrednosti, što čini ovu metodu otpornijom na outlier-e.

IQR metoda je otpornija na ekstremne vrednosti u odnosu na *z-score* metodu. Nije je uvek pogodno koristiti u situacijama kada je distribucija podataka izuzetno asimetrična. Prilikom izbora između *z-score* i IQR metode najbolje je isprobati obe i uporediti rezultate. U slučaju distribucija koje značajno odstupaju od normalne, pogodnije je razmotriti korišćenje drugih tehnika za rukovanje sa *outlier*-ima.

Kao što je već pomenuto, *z-score* i IQR metode za detekciju *outlier*-a rade nad pojedinačnim atributima. Njih je moguće kombinovati tako da rade i sa više atributa, tako što se za svaki atribut izvrši detekcija, a zatim se svi primerci koji sadrže ekstremne vrednosti po N ili više atributa (gde je N prag zadat od korisnika) označe kao *outlier*-i. U mnogim situacijama je umesto ovakvog postupka pogodnije vršiti automatsku detekciju *outlier*-a korišćenjem neke od tehnika o kojima će biti reči u narednim poglavljima, koje su u stanju da „nauče“ kompleksnije veze između atributa i podataka.

3.6.3. Uklapanje eliptičnog omotača (*elliptic envelope*)

Slično prethodno razmatranim tehnikama, jedan od načina za detekciju *outlier*-a podrazumeva pretpostavku da podaci prate normalnu (Gausovu) raspodelu. Na osnovu ove pretpostavke možemo vršiti analizu koji primerci se uklapaju, a koji ne, u „oblik“ koji definiše normalna raspodela. Tehnika uklapanja eliptičnog omotača za detekciju *outlier*-a se upravo oslanja na ovu pretpostavku. Ona podrazumeva uklapanje elipse oko centralnog dela raspodele podataka i označavanje tačaka van ove elipse *outlier*-ima (pri čemu se u procesu uklapanja elipse koristi robusna kovarijansa, koja je otporna na prisustvo *outlier*-a).

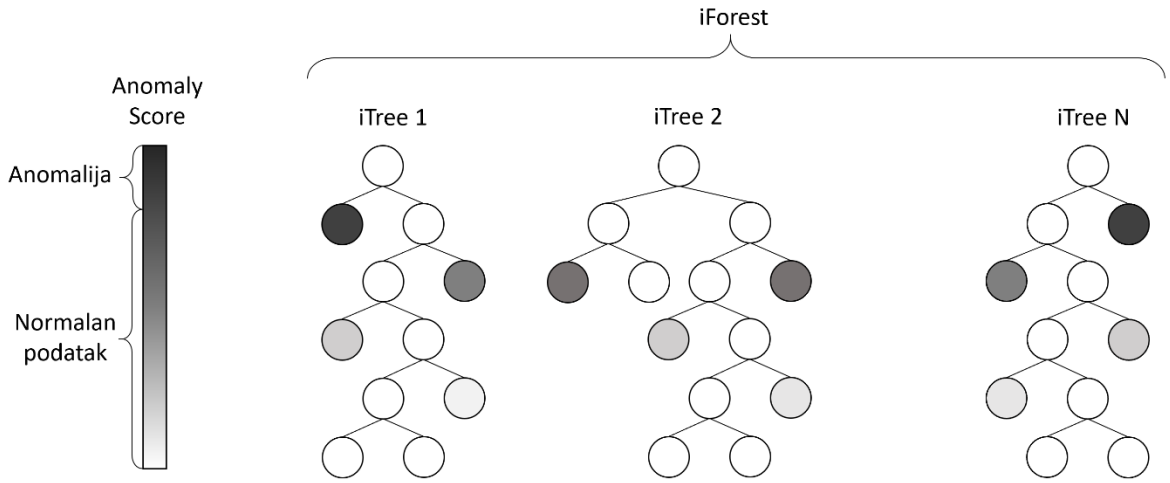
Ova tehnika može dati dobre rezultate u situacijama kada podaci približno prate normalnu raspodelu. Međutim, u situacijama kada podaci ne prate normalnu raspodelu, uklapanje elipse neće biti ispravno, pa će samim tim i detekcija *outlier*-a biti neadekvatna, pa u ovakvim situacijama je pogodnije korišćenje neke od drugih tehnika.

3.6.4. Detekcija *outlier*-a pomoću *Isolation Forest* metode

Izolaciona šuma (engl. *Isolation Forest*) predstavlja metodu kojom se brzo mogu detektovati ekstremne vrednosti. Tradicionalne metode za detekciju *outlier*-a, poput *z-score* i IQR metoda, pokušavaju da „uklope“ podatke u normalnu distribuciju i vrednosti koje se nalaze van određenog opsega označavaju kao *outlier*-e. Izolacione šume nude drugačiji način za izbacivanje *outlier*-a, uz mogućnost njihove detekcije u prostorima visokih dimenzionalnosti.

Izolacione šume su predložene u radu [19] i zasnivaju se na principu *„few and different“* prilikom detekcije anomalija (*outlier*-a). Drugim rečima, polazi se od pretpostavke da anomalija ima relativno malo i to značajno manje u odnosu na ostatak podataka i da se ove anomalije značajnu razlikuju od ostatka podataka.

Izolacione šume predstavljaju skup (*ensamble*) binarnih stabala, koje rekurzivno određuju particije, tako što nasumično izaberu atribut i nasumično izvrše podelu podataka po ovom atributu dok svi primerci ne budu izolovani (slika 10). Svakom primerku se dodeljuje odgovarajući *anomaly score*, u zavisnosti od toga koliko su brzo bili izolovani. Na ovaj način, *outlier*-ima je potrebno manje particija (kraća putanja u stablu) da budu izolovani. Prethodno opisani proces se ponavlja više puta kako bi se kreirao veći broj stabala, pri čemu svako stablo nezavisno vrši izolaciju anomalija (*outlier*-a). Na kraju se vrši agregacija *anomaly score* vrednosti podataka podataka koje su pojedinačna stabla odredila. Nakon agregacije, nad tako izračunatim *anomaly score* vrednostima se primenjuje odgovarajući prag, kojim se određuje koji od primeraka unutar podataka se mogu označiti kao *outlier*-i, pri čemu korisnik najčešće zadaje procenat primeraka koje je potrebno označiti kao *outlier*-e, a prag se računa automatski. [20]



Slika 10: Grafička reprezentacija izolacione šume. Sastoji se od većeg broja stabala odlučivanja. Što je kraća putanja do nekog podatka u stablu, to je veća šansa da je u pitanju outlier.

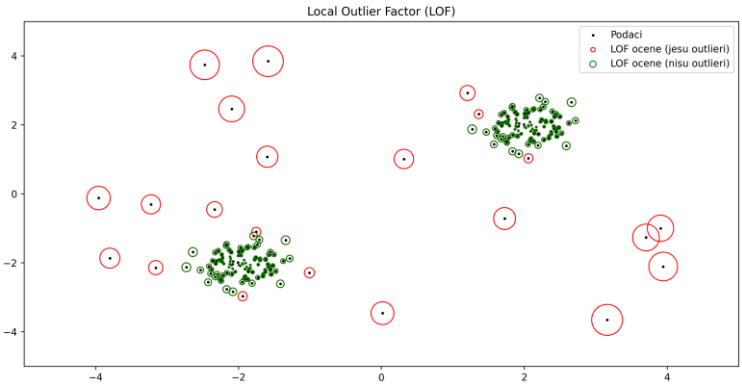
Način funkcionisanja izolacione šume u poređenju sa prethodno razmatranim metodama je fundamentalno različit. Osnovna prednost izolazionih šuma je u tome što se detekcija anomalija ne oslanja na rastojanja niti na gustine distribucije podataka i ne zasniva se na normalnoj distribuciji, što čini ovu metodu pogodnom u raznim situacijama. Ova metoda može biti odlična u radu sa podacima visoke dimenzionalnosti i, kao i većina *ensamble* metoda, veoma je otporna na uticaj *outlier*-a i prisustvo šuma u podacima. Pored ovoga, vremenska kompleksnost ove metoda je linearna, što je takođe čini pogodnom i za velike skupove podataka.

Korišćenje izolacionih šuma može biti odlično u raznim situacijama, ali je potrebno biti oprezan prilikom njihove upotrebe i potrebno je obratiti pažnju na to koje će anomalije zapravo biti

detektovane, obzirom da ova metoda možda nije pogodna za specifične podatke sa kojima se radi. U nekim situacijama (npr. kada je raspodela podataka bliska normalnoj), je možda pogodnije koristiti neku od prethodno razmatranih jednostavnijih tehnika.

3.6.5. Faktor lokalnih ekstremnih vrednosti (*local outlier factor*)

Local outlier factor (LOF) je tehnika za detekciju *outlier*-a, koja podrazumeva računanje *anomaly score* vrednosti za svaki od primeraka u skupu podataka. Ova vrednost se računa u zavisnosti od lokalne devijacije gustine raspodele trenutnog primerka u odnosu na njegove najbliže susede. Drugim rečima, *anomaly score* primerka se računa na osnovu toga koliko je dati primerak izolovan u odnosu na njegove susede, pri čemu se lokalna gustina računa u zavisnosti od rastojanja od njegovih k najbližih suseda. Poređenjem lokalne gustine datog primerka, sa lokalnim gustinama njegovih suseda, identifikuju se primerci čija je lokalna gustina značajno manja u odnosu na njegove susede (dakle primerke koji su izolovani od ostalih). Očekuje se da normalni primerci imaju lokalnu gustinu koja je slična lokalnim gustinama najbližih suseda, dok anomalije u podacima imaju nižu lokalnu gustinu od svojih suseda. Konačna LOF ocena se računa kao odnos prosečne lokalne gustine suseda i lokalne gustine datog primerka [21]. Na slici 11 možemo videti primer primene LOF metode nad skupom podataka sa bimodalnom raspodelom. Primećujemo da je ova metoda uspela da detektuje dve zone gusto raspoređenih podataka, dok su podaci koji se nalaze van ovih zona označeni kao *outlier*-i.



Slika 11: Primena LOF metode nad podacima koji prate bimodalnu raspodelu. Crne tačke označavaju podatak, a prečnik kružića oko ovih tačaka određuje LOF. Zelenom bojom su označeni podaci koji po LOF metodi nisu outlier-i, a crvenom bojom oni koji jesu. Primećujemo da ova metoda radi veoma dobro čak i u ovakvoj situaciji kada podaci ne prate normalnu raspodelu.

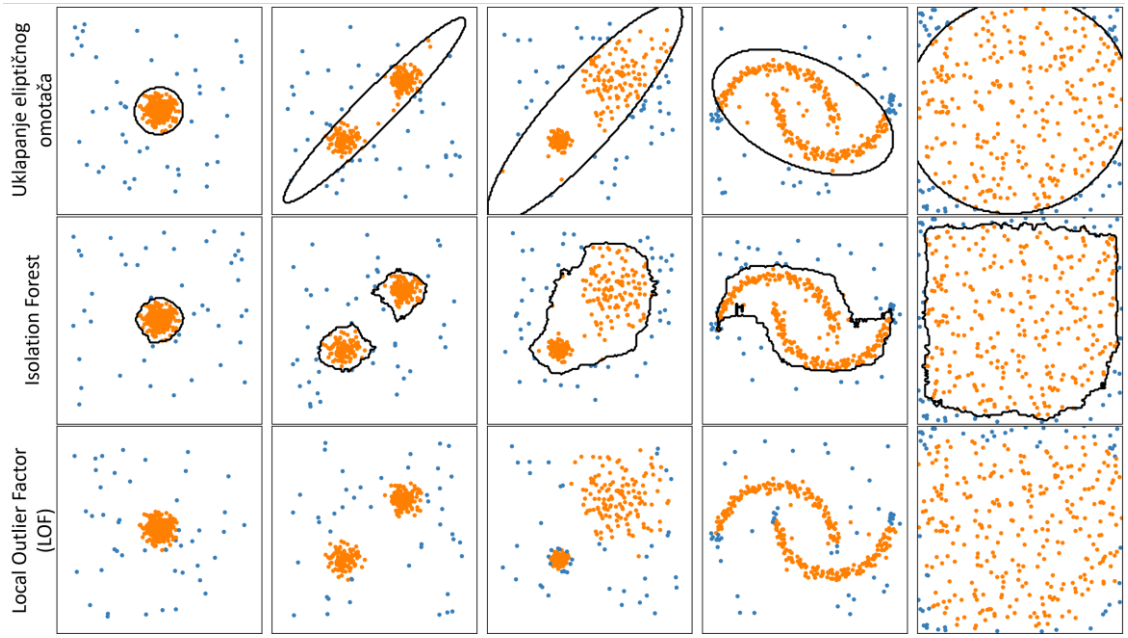
Parametar k se određuje tako da bude veći od minimalnog broja podataka koje klaster (podataka koji nisu *outlier*-i) treba da sadrži ili tako da bude manji od maksimalnog broja bliskih podataka koji mogu biti *outlier*-i. U praksi, ovo nije toliko lako odrediti, pa se najčešće koristi empirijski određena vrednost k=20, koja najčešće daje dobre rezultate. U situacijama kada je procenat *outlier*-a u podacima veći, poželjno je izabrati veću vrednost ovog parametra.

Ova tehnika generalno radi prilično dobro, jer uzima i lokalne i globalne karakteristike skupa podataka i povezanosti između podataka. Daje odlične rezultate čak i u situacijama kada anomalije u skupu podataka imaju različite gustine raspodele, zato što ova tehnika određuje koliko je podatak izolovan u odnosu na svoju okolinu, a ne koliko je izolovan u odnosu na ceo skup podataka. Nedostatak ove tehnike leži u radu sa višedimenzionalnim podacima. Imajući u vidu da se ova tehnika oslanja na metrike rastojanja, nije je pogodno primenjivati nad podacima sa velikim brojem atributa, obzirom da rastojanje postaje manje značajno sa porastom broja dimenzija [22].

3.6.6. Poređenje različitih tehnika za detekciju *outlier*-a

Sledeći primer (slika 12) ilustruje kako se različite tehnike za automatsku detekciju *outlier*-a ponašaju nad različitim raspodelama podataka, pri čemu je svaki skup podataka odabran tako da sadrži 15% *outlier*-a). Sintetički skupovi podataka korišćeni u primeru sa slike 12 obuhvataju sledeće raspodele:

1. **Gausova raspodela** – prvi skup podataka predstavlja jednostavan slučaj gde raspodela podataka prati Gausovu. Ovo predstavlja najjednostavniji slučaj raspodele i očekuje se da svi algoritmi daju dobre rezultate.
2. **Bimodalna Gausova raspodela** – drugi skup podatak se sastoji od dva regiona veće gustine raspodele, pri čemu oba prate Gausovu raspodelu. Ovaj primer bi trebao da ilustruje kako različiti algoritmi razdvajaju regione sa većim lokalnim gustinama raspodele.
3. **Bimodalna Gausova raspodela različitih gustina** – treći skup se, slično drugom, sastoji od dva regiona, pri čemu je lokalna gustina jednog značajno manja od drugog. Cilj ovog primera jeste da pokaže kako se različiti algoritmi ponašaju u situacijama kada gustina raspodele po regionima nije približno ista.
4. **Kompleksna prostorna raspodela** – četvrti skup pokazuje podatke koji prate raspodelu koja se ne može opisati standardnim modelima kao što je Gausov. Cilj ovog primera je da pokaže kako različiti algoritmi uče kompleksne, nelinearne veze među podacima.
5. **Uniformna (nasumična) raspodela** – peti skup podrazumeva podatke koje su ravnomerno raspoređeni u prostoru. Ova je situacija u kojoj ne postoji jasna granica između toga šta jeste, a šta nije *outlier* i služi da prikaže kako se algoritmi ponašaju u ovakvim slučajevima.



Slika 12: Poređenje različitih algoritama za automatsku detekciju outlier-a (uklapanje elipse, izolaciona šuma i LOF – dati po redovima) nad različitim skupovima podataka (normalna; bimodalna; bimodalna kod koje jedan region ima veću a drugi manju gustinu raspodele; „moon" raspodela koja ilustruje kompleksne veze između podataka, uniformna – date po kolonama). Glavna prednost LOF se ogleda u trećem primeru sa različitim gustinama.

Uklapanje eliptičnog omotača podrazumeva da podaci prate normalnu raspodelu i pokušava da uklopi podatke u elipsu. Ovo daje loše rezultate, u situacijama kada podaci ne prate normalnu raspodelu, ili kada postoji više oblasti guste raspodele. Izolaciona šuma i LOF daju ubedljivo bolje rezultate, pogotovu kada postoji kompleksnija veza između podataka i kada postoji više oblasti guste raspodele. Prednost LOF metode se vidi u slučaju trećeg skupa podataka, gde ova metoda uspeva da razdvoji dve oblasti različitih gustina, dok tehnika izolacione šume prepoznaje samo jednu.

Navedeni primeri samo ilustruju kako se ovi algoritmi ponašaju nad dvodimenzionalnim podacima. U praksi, podaci imaju dosta veću dimenzionalnost, pa samim tim i problem detekcije *outlier*-a postaje značajno kompleksniji. Iz ovog razloga, uvek je poželjno što detaljnije analizirati podatke sa kojima se radi, opsege vrednosti, raspodele i isprobati što više različitih metoda, jer kvalitet detekcije *outlier*-a, često mnogo više zavisi od samih podataka, nego od osobina algoritama.

U nastavku je data tabela koja prikazuje prednosti i mane različitih tehnika za detekciju *outlier*-a.

TEHNIKA	PREDNOSTI	MANE
Z-SCORE	Odlična u sličaju raspodele koja je bliska normalnoj	Nije pogodna ako postoji veći broj *outlier*-a; ako raspoela ne prati normalnu, daje loše rezultate
IQR	Odlična kada raspodela odgovara normalnoj, čak i u slučaju većeg broja *outlier*-a	Ako raspodela ne prati normalnu daje loše rezultate
ELLIPTIC ENVELOPE	Odlična za višedimenzionalne prostore, kada atributi većinski prate normalnu raspodelu	Ne daje dobre rezultate u kompleksnijim situacijama i kada podaci ne prate normalnu raspodelu
ISOLATION FOREST	Daje veoma dobre rezultate u raznim situacijama kod višedimenzionalnih prostora	
LOF	Daje veoma dobre rezultate u višedimenzionalnim prostorima, u situacijama kada postoje oblasti različite gustine raspodele	Nije pogodna za podatke visoke dimenzionalnosti.

3.6.7. *Winsorization* (vinzorizacija)

Vinzorizacija predstavlja tehniku transformacije podataka, koja radi tako što ograničava ekstremne vrednosti. Vinzorizacija podrazumeva da se vrednost nekog atributa ograniči gornjom i donjom granicom, tako da se svim vrednostima izvan ovog opsega dodeli granična vrednost. Granične vrednosti se biraju na osnovu percentila, tako što se zadaju vrednosti gornjeg i donjeg percentila. Izbor percentila je veoma bitan i može značajno uticati na efektivnost ove metode.

Na primer, u slučaju vinzorizacije sa donjom granicom 5. i gornjom granicom 95. percentila, svim vrednostima koje su iznad 95. percentila se dodeljuje vrednost 95. percentila, a svim vrednostima ispod 5. percentila se dodeljuje vrednost 5. percentila.

Vinzorizacija se jednostavno primenjuje i može biti odlična u smanjivanju uticaja *outlier*-a, bez njihovog izbacivanja. Može se primenjivati ukoliko je to u skladu sa specifičnim podacima (za šta je često potrebno i domensko znanje). Mana ove tehnike je u tome što se ne uzima u obzir veličina ekstremnih vrednosti i što potencijalno izaziva gomilanje podataka na krajevima raspodele. Ova tehnika nije pogodna u situacijama kada vrednosti primeraka koji će biti označeni kao *outlier*-i potencijalno nose informacije od značaja. [14]

3.7. Konstrukcija atributa

Konstrukcija atributa predstavlja proces u kome kreiramo nove atribute (kolone) na osnovu postojećih u cilju izvlačenja relevantnih informacija iz podataka sa kojima radimo, što često dovodi do poboljšanja performansi modela mašinskog učenja. Konstrukcija atributa se može vršiti:

- Na osnovu domenskog znanja – kombinacija atributa po odgovarajućim pravilima ili po uputstvima industrijskih standarda. Na primer, na osnovu visine i težine možemo izračunati BMI, koji potencijalno nosi više informacija od značaja za pojedine domene; ili računanje DTI (*Debt-To-Income ratio*) u slučaju rada sa bankarskim podacima, koji može nositi informacije od značaja za npr. odluku o izdavanju kredita.

- Na osnovu samih podataka – kombinovanje atributa na osnovu analize podataka, kao što je agregacija. Na primer, u slučaju rada sa podacima o nekretninama, možemo sabrati broj obližnjih vrtića, osnovnih i srednjih škola i kreirati atribut koji se odnosi na ukupan broj škola.
- Sintetički – kombinovanje atributa korišćenjem matematičkih funkcija u cilju nalaženja nelinearnih veza ili korišćenje metoda kao što su stabla odlučivanja za kreiranje atributa.

Metodi koji se mogu koristiti prilikom kreiranja novih atributa u najvećoj meri zavise od domena i podataka sa kojima radimo. Neke od češće korišćenih tehnika obuhvataju: kombinovanje atributa korišćenjem statističkih ili matematičkih operacija, polinomne kombinacije (polinomna ekspanzija), stabla odlučivanja za kreiranje novih atributa. [23] [24] [14]

3.7.1. Kombinovanje atributa korišćenjem statističkih ili matematičkih operacija

Jedan način za kreiranje novih atributa jeste primena raznih statističkih ili matematičkih operacija nad postojećim atributima. Jedna od operacija koja se često može koristiti u ovu svrhu jeste sabiranje. Na primer, možemo izračunati ukupan dug korisnika sabiranjem pojedinačnih dugovanja (agregacija).

Još neke informacije od značaja se mogu izvući iz podataka korišćenjem statističkih operacija kao što su maksimum ili minimum (npr. maksimalan dug korisnika, ili minimalno vreme provedeno na veb stranici), srednja vrednost (npr. prosečno vreme provedeno na veb stranici) itd.

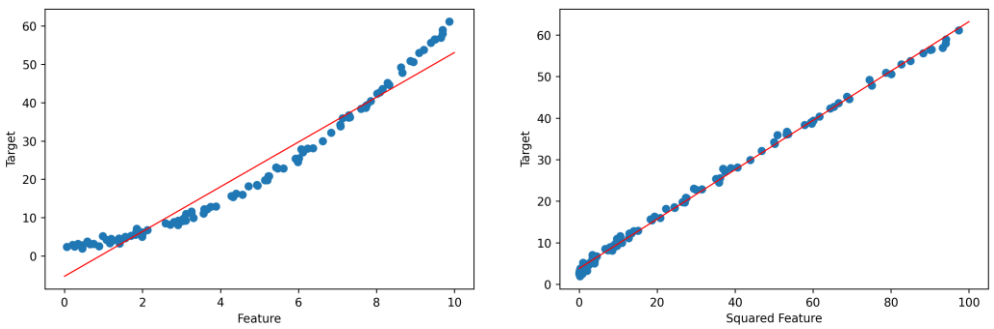
Oduzimanje i deljenje takođe mogu biti korisne u nekim situacijama prilikom konstrukcije atributa. Na primer, za računanje DIB odnosa je potrebno podeliti ukupan dug sa ukupnim prihodima, a za računanje raspoloživog prihoda, možemo oduzeti dugovanje od prihoda.

Primena ove tehnike zavisi od podataka i od konkretnog domena problema i ovakve tehnike se primenjuju shodno tome. U situacijama kada je primena ovakve tehnike moguća, ona može smanjiti broj ulaza modela i dati bolje rezultate prilikom njegovog obučavanja.

3.7.2. Polinomna ekspanzija

Pored jednostavnih matematičkih i statističkih operacija, o kojima je bilo reči u prethodnom poglavlju, za kreiranje novih atributa se mogu koristiti i polinomne kombinacije atributa sa samim sobom ili sa drugim atributima u svrhu izvlačenja informacija od značaja. Na primer, u situacijama kada ciljani atribut prati kvadratnu zavisnost sa nekim drugim atributom, može biti korisno da kreiramo polinom drugog stepena na osnovu ovog atributa, čime bismo potencijalno omogućili korišćenje linearnih modela.

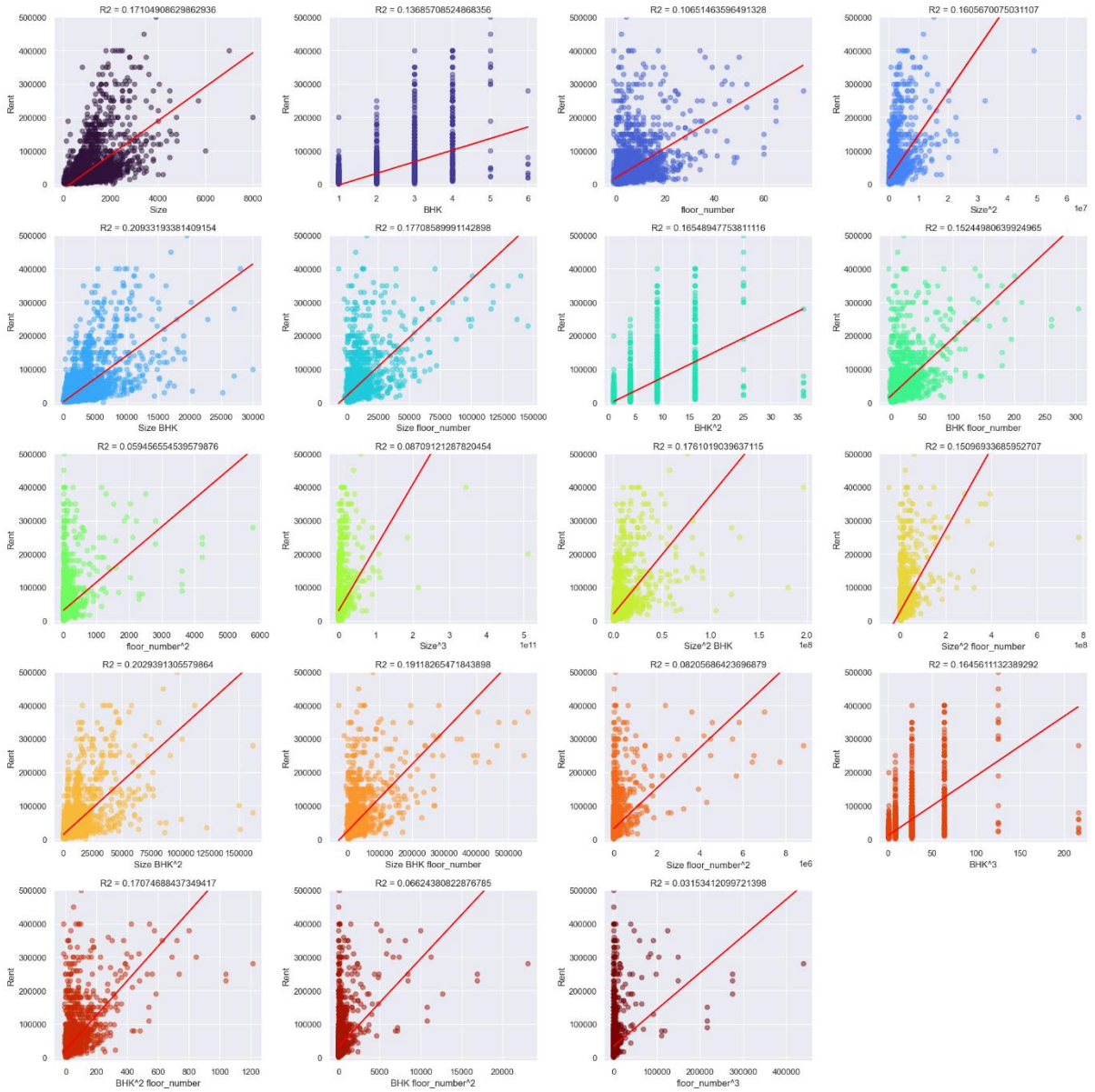
Sledeći primer ilustruje situaciju u kojoj postoji kvadratna veza između (nekog) atributa i ciljanog atributa. Ukoliko pokušamo da treniramo linearan model nad ovakvim podacima rezultati će biti loši, ali zato ako primenimo kvadratnu operaciju nad odgovarajućim atributom, možemo primetiti da je linearna veza uspostavljena.



Slika 13: Primer koji ilustruje poboljšanje koje se može postići prilikom primene linearne regresije uz kvadriranje odgovarajućeg atributa.

Na sličan način, mogu se kreirati proizvoljne polinomne kombinacije bilo kojih atributa i analizirati zavisnost novo-dobijenih atributa i ciljanog atributa. Princip koji se najčešće koristi jeste da se izabere proizvoljan podskup atributa i da se izvrši analiza zavisnosti svih mogućih polinomnih kombinacija ovih atributa zadatog stepena i ciljanog atributa.

Ideja kod ove tehnike jeste da polinomna kombinacija jedne ili više promenljivih može rezultovati u novom atributu koji nosi više informacija od značaja, što može doprineti procesu obučavanja modela mašinskog učenja. Ova tehnika je najviše primenljiva kod linearnih modela. Naravno, ovo neće uvek dovesti do dobrih rezultata u situacijama kada polinomna zavisnost ne postoji. Takođe, prilikom izbora atributa i stepena polinoma za koje se vrši analiza zavisnosti, nije poželjno birati velike vrednosti stepena ili veliki broj atributa, jer ovo veoma lako može rezultovati u generisanju prevelikog broja novih atributa. Primer primene ove tehnike nad nekim skupom podataka je prikazan na slici 14. Dodatno, za svaki novi atribut je i primenjen postupak linearne regresije, kako bi se utvrdilo da li bi potencijalno povoljno uticao na performanse linearnog modela.



Slika 14: Primer konstrukcije svih mogućih polinomnih kombinacija stepena 3 nad 3 atributa. Za svaki novi atribut je izvršena linearna regresija (predstavljeno crvenom linijom).

3.7.3. Kreiranje atributa korišćenjem stabala odlučivanja

Jedna od metoda koje se mogu koristiti za kreiranje novih atributa kombinovanjem dva ili više atributa je predložena u [25] i uključuje korišćenje stabala odlučivanja. Ova tehnika podrazumeva kreiranje stabla odlučivanja korišćenjem podskupa atributa (najčešće dva ili tri) i korišćenje predikcije ovog stabla odlučivanja kao novi atribut.

Ova tehnika je pogodna za uspostavljanje monotone veze sa ciljanim atributom, koja je često poželjna za obučavanje linearnih modela. Osim uspostavljanja monotone veze, ova tehnika može uspostaviti i kompleksne veze i interakcije između atributa. Nedostatak ove tehnike je pre svega u kompleksnosti, obzirom da je pored obučavanja stabla neophodna i optimizacija njegovih hiperparametara, a pored toga, tu je i mogućnost *overfitting*-a.

3.7.4. Konstrukcija atributa na osnovu podataka u vremenskom formatu

Tip atributa koji se često sreće u skupovima podataka jeste vreme i datum. Primeri ovakvih atributa mogu biti datum rođenja, vreme nekog događaja, datum i vreme uplate itd. Ove podatke nije moguće direktno koristiti kao ulaz modela mašinskog učenja, već je potrebno na neki način izvući informacije od značaja iz ovih podataka. Neke od metoda koje se mogu koristiti prilikom rada sa podacima o datumu i vremenu uključuju:

- Korišćenje pojedinačnih vrednosti – posebno čuvamo dan u mesecu, mesec, godinu, sate, minute i sekunde, ili samo neke od ovih atributa.
- Računanje informacija o danu u nedelju – na osnovu datuma računamo dan u nedelji, pri čemu se kao ulaz za model može koristiti konkretan dan u nedelji ili samo indikator da li je u pitanju vikend ili radni dan.
- Računanje vremenskog intervala – na osnovu dva atributa u vremenskom formatu možemo izračunati broj sekundi, minuta, sati, dana… koliko je prošlo između ova dva trenutka. Može biti korisno za računanje godina starosti ili vremena trajanja nekog procesa.

3.7.5. Konstrukcija atributa na osnovu podataka u tekstualnom formatu

> *NAPOMENA: Iako su kategorički atributi često zadati u tekstualnom formatu, takvi atributi imaju ograničen skup mogućih vrednosti, gde svaka vrednost nosi odgovarajuće značenje. Ovo poglavlje se odnosi na tekstualne podatke date u slobodnom formatu tj. izražene prirodnim jezikom, kao što su npr. opisi.*

Još jedan tip podataka koji se ne može direktno koristiti kao ulaz modela mašinskog učenja jesu tekstualni podaci. Ovakvi podaci najčešće dolaze od samih korisnika i tipični primeri su opisi, recenzije, informacije o nekom događaju, nazivi proizvoda itd. Ovi podaci, za razliku od do sada razmatranih podataka nemaju jasnu strukturu, mogu biti različitih dužina i proizvoljnog sadržaja, što dodatno otežava njihovo korišćenje.

Oblast nauke koja predstavlja presek lingvistike i računarskih nauka i bavi se problemom programiranja računara tako da mogu razumeti ljudski jezik se naziva obrada prirodnih jezika (NLP – *Natural Language Processing*). NLP obuhvata veliki broj tehnika za razumevanje sintakse, semantike i diskusija i predstavlja veoma široku oblast nauke.

Neke od mogućih tehnika za izvlačenje potencijalno korisnih atributa iz (relativno kratkih) delova teksta u prirodnom jeziku obuhvataju:

- Prebrojavanje karaktera, reči i vokabulara – podrazumeva određivanje kompleksnosti teksta na osnovu mera kao što su dužina teksta u karakterima, ukupan broj reči, ukupan broj

jedinstvenih reči, leksički diverzitet (količnik ukupnog broja reči i ukupnog broja različitih reči), prosečna dužina reči (količnik broja karaktera u tekstu i broja reči u tekstu). Ove mere mogu ukazivati na to koliko informacija sam tekst nosi.

- Procenjivanje kompleksnosti brojanjem rečenica u tekstu
- Konstrukcija atributa korišćenjem *bag-of-words* (BoW) tehnike i n-grama – BoW predstavlja pojednostavljenu reprezentaciju teksta, koja obuhvata reči koje se pojavljuju u tekstu i broj pojavljivanja svake od reči u tekstu. Ova tehnika ne uzima u obzir gramatiku i redosled reči u rečenici. U cilju zadržavanje dodatnih informacija, BoW se često koristi zajedno sa n-gramima. N-gram je sekvenca n uzastopnih reči u datom tekstu. Npr. ukoliko bismo primenili tehnike BoW i *bag of n-grams* (za n=2) nad rečenicom „*Dogs like cats, but cats to not like dogs*“, rezultat bi bio:

dogs	*like*	*cats*	*but*	*do*	*not*	*dogs like*	*like cats*	*cats but*	*but do*	*do not*	*like dogs*
2	2	2	1	1	1	1	1	1	1	1	1

- Implementacija TF-IDF (*term frequency – inverse document frequency*) – predstavlja statističku meru koja daje informaciju o tome koliko je reč u dokumentu relevantna u odnosu na kolekciju dokumenata. Neke reči, kao što su *the, a, is…* u engleskom jeziku ili *pa, ovaj, je(jeste)…* u srpskom jeziku se pojavljuju veoma često u raznim tekstovima i ne nose puno informacija o konkretnom tekstu. TF-IDF predstavlja način da se izmeri značaj reči na osnovu broja pojavljivanja ove reči u konkretnom tekstu u odnosu na broj pojavljivanja iste u svim tekstovima. Na ovaj način, česte reči kao što su *pa, ovaj, je* imaju manju težino, a reči koje su specifične za neku temu, npr. *diskretizacija* će imati veću težinu.
- Čišćenje teksta i određivanje korena reči – predstavlja proces koji je poželjno obaviti pre primene neke od ostalih tehnika. Podrazumeva izbacivanje interpunkcije, izjednačavanje veličine slova, izbacivanje stop reči (*the, a, an…*) i izvlačenje korena reči.

Obrada prirodnih jezika predstavlja veoma kompleksnu oblast i navedene tehnike su samo neke od dostupnih.

4. Zaključak

Transformacija podataka je jedan od ubedljivo najznačajnijih koraka u procesu mašinskog učenja, jer skoro svaki sledeći korak zavisi od toga sa kakvim podacima se radi. Podaci sa kojima se radi u praksi su raznovrsni i teško je jasno definisati „najbolji“ način za njihovu transformaciju.

Postoji ogroman broj različitih tehnika i neke od njih će dovesti do značajnog poboljšanja, neke možda uopšte neće uticati na konačan rezultat obučavanja modela, a neke mogu značajno pogoršati rezultate. Sve ovo u ogromnoj meri zavisi od samih podataka sa kojima se radi i od konkretnog domena problema, zbog čega je veoma bitno upoznati se sa podacima sa kojima se radi i potencijalno konsultovati se sa ekspertima u odgovarajućem domenu. Naravno, u procesu izbora tehnika koje će biti korišćene za transformaciju podataka je jako bitno poznavati šta radi i kako funkcioniše koja tehnika i kada je treba, a kada je nije poželjno primenjivati.

Osim navedenog, nikad se ne treba u potpunosti oslanjati na „naprednije“ tehnike, koje najčešće imaju interfejs visokog nivoa, i veći deo posla obavljaju automatski, jer iako one često mogu dati dobre rezultate, postoje i situacije kada njihova primena dovodi do *overfitting*-a ili situacije kada je u potpunosti dovoljno primeniti jednostavniju tehniku.

Transformacija podataka je proces kome treba posvetiti puno vremena i pažnje i uvek je poželjno isprobati više tehnika i uporediti izlaze.

5. Reference

[1] R. RV, G. Lemaitre i T. Unterthiner, „Compare the effect of different scalers on data with outliers,“ Scikit-learn, [Na mreži]. Available: https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html.

[2] „Everything you need to know about Min-Max normalization: A Python tutorial,“ Medium, 28 May 2020. [Na mreži]. Available: https://towardsdatascience.com/everything-you-need-to-know-about-min-max-normalization-in-python-b79592732b79.

[3] „Compare the effect of different scalers on data with outliers,“ scikit-learn, [Na mreži]. Available: https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html.

[4] „Skewness – Measures and Interpretation,“ Geeks for Geeks, [Na mreži]. Available: https://www.geeksforgeeks.org/skewness-measures-and-interpretation/.

[5] A. Banerjee, „Scaling vs Normalization, are they the same?,“ Medium, 15 December 2022. [Na mreži]. Available: https://medium.com/geekculture/scaling-vs-normalization-are-they-the-same-348035afe5ca.

[6] „Log Transformation: Purpose and Interpretation,“ Medium, 29 February 2020. [Na mreži]. Available: https://medium.com/@kyawsawhtoon/log-transformation-purpose-and-interpretation-9444b4b049c9.

[7] „Quantile Transformer,“ scikit-learn, [Na mreži]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html.

[8] „Data Preprocessing,“ scikit-learn, [Na mreži]. Available: https://scikit-learn.org/stable/modules/preprocessing.html.

[9] „Comparing anomaly detection algorithms for outlier detection on toy datasets,“ scikit-learn, [Na mreži]. Available: https://scikit-learn.org/stable/auto_examples/miscellaneous/plot_anomaly_comparison.html.

[10] „Categorical Data Encoding Techniques,“ Medium, 14 March 2023. [Na mreži]. Available: https://medium.com/aiskunks/categorical-data-encoding-techniques-d6296697a40f.

[11] „https://medium.com/anolytics/all-you-need-to-know-about-encoding-techniques-b3a0af68338b,“ Medium, 30 September 2023. [Na mreži]. Available: https://medium.com/anolytics/all-you-need-to-know-about-encoding-techniques-b3a0af68338b.

[12] D. Micci-Barreca, „A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems,“ *ACM SIGKDD Explorations Newsletter ,* 2001.

[13] „Crafting Insights: A Guide to Mastering Data Transformation,“ Medium, 2023 November 2023. [Na mreži]. Available: https://baotramduong.medium.com/data-preprocessing-data-transformation-step-in-203af423ae5a.

[14] S. Galli, Python Feature Engineering, BIRMINGHAM - MUMBAI: Packt, 2020.

[15] „Discretisation Using Decision Trees,“ Medium, 24 December 2018. [Na mreži]. Available: https://towardsdatascience.com/discretisation-using-decision-trees-21910483fa4b.

[16] „Dealing with Outliers in Data Science: Techniques and Best Practices,“ Medium, 20 April 2023. [Na mreži]. Available: https://syedabis98.medium.com/dealing-with-outliers-in-data-science-techniques-and-best-practices-a08172643b7a.

[17] „Removing Outliers. Understanding How and What behind the Magic.,“ Medium, 5 April 2021. [Na mreži]. Available: https://medium.com/analytics-vidhya/removing-outliers-understanding-how-and-what-behind-the-magic-18a78ab480ff.

[18] „Interquartile Range to Detect Outliers in Data,“ geeks for geeks, [Na mreži]. Available: https://www.geeksforgeeks.org/interquartile-range-to-detect-outliers-in-data/.

[19] F. T. Liu, K. M. Ting i Z.-H. Zhou, „Isolation forest,“ u *2008 eighth ieee international conference on data mining*, 2008.

[20] „Unsupervised Outlier Detection with Isolation Forest,“ Medium, 17 March 2022. [Na mreži]. Available: https://medium.com/mlearning-ai/unsupervised-outlier-detection-with-isolation-forest-eab398c593b2.

[21] M. Breunig, H.-P. Kriegel, R. T. Ng i J. Sander, „LOF: Identifying Density-Based Local Outliers,“ u *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000.

[22] K. Beyer, J. Goldstein, R. Ramakrishnan i U. Shaft, „When is “nearest neighbor” meaningful?,“ u *Database Theory—ICDT’99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7*, 1999.

[23] „Data Transformation in Data Mining,“ java T point, [Na mreži]. Available: https://www.javatpoint.com/data-transformation-in-data-mining.

[24] „What is Feature Engineering?,“ geeks for geeks, [Na mreži]. Available: https://www.geeksforgeeks.org/what-is-feature-engineering/.

[25] A. Niculescu-Mizil, C. Perlich, G. Swirszcz, V. Sindhwani, Y. Liu, P. Melville, D. Wang, J. Xiao, J. Hu i M. Singh, „Winning the KDD cup orange challenge with ensemble selection,“ u *KDD-Cup 2009 competition*, 2009, pp. 23-34.

[26] „Crafting Insights: A Guide to Mastering Data Transformation,“ Medium, 9 November 2023. [Na mreži]. Available: https://baotramduong.medium.com/data-preprocessing-data-transformation-step-in-203af423ae5a.

[27] „Categorical Encoding with CatBoost Encoder,“ geeks for geeks, [Na mreži]. Available: https://www.geeksforgeeks.org/categorical-encoding-with-catboost-encoder/.

[28] „How CatBoost encodes categorical variables?,“ Medium, 10 February 2021. [Na mreži]. Available: https://towardsdatascience.com/how-catboost-encodes-categorical-variables-3866fb2ae640.