**UNIVERZITET U NIŠU**

**ELEKTRONSKI FAKULTET**

**SEMINARSKI RAD**

**Transformacija podataka**

|  |  |
| --- | --- |
| **Student:** | **Profesor:** |
| Stefan Jovanović 1959 | Prof. dr Aleksandar Stanimirović |

Sadržaj

[1. Uvod 1](#_Toc1)

[2. Motivacija i identifikacija problema 1](#_Toc2)

[3. Normalizacija i standardizacija 3](#_Toc3)

[3.1. Definicije i matematičke formulacije 3](#_Toc4)

[3.1.1. Min-max normalizacija 3](#_Toc5)

[3.1.2. Z-score standardizacija 4](#_Toc6)

[3.2. Kada koristiti koju metodu 4](#_Toc7)

[3.3. Prednosti i mane 5](#_Toc8)

[3.4. Uticaj na algoritme mašinskog učenja 5](#_Toc9)

[4. Skaliranje 6](#_Toc10)

[4.1. RobustScaler 6](#_Toc11)

[4.2. MaxAbsScaler 7](#_Toc12)

[4.3. Kada je skaliranje važno 7](#_Toc13)

[4.4. Razlike u odnosu na normalizaciju i standardizaciju 8](#_Toc14)

[5. Promena raspodele podataka (transforms) 8](#_Toc15)

[5.1. Log transformacija 8](#_Toc16)

[5.2. Box–Cox transformacija 9](#_Toc17)

[5.3. Yeo-Johnson transformacija 9](#_Toc18)

[5.4. Kada i zašto se koristi promena raspodele 10](#_Toc19)

[5.5. Primeri korišćenja u praksi 10](#_Toc20)

[6. Enkodiranje kategoričkih promenljivih 11](#_Toc21)

[6.1. One-hot encoding 11](#_Toc22)

[6.2. Label encoding 11](#_Toc23)

[6.3. Ordinal encoding 12](#_Toc24)

[6.4. Frequency encoding 12](#_Toc25)

[6.5 Kada koristiti koju metodu 12](#_Toc26)

[7. Diskretizacija (binning) 13](#_Toc27)

[7.1 Vrste diskretizacije 13](#_Toc28)

[7.1.1 Uniformna (jednaka širina) diskretizacija 13](#_Toc29)

[7.1.2 Kvantilna (jednaka frekvencija) diskretizacija 14](#_Toc30)

[7.1.3 K-means binning 14](#_Toc31)

[7.2 Primene i prednosti diskretizacije 14](#_Toc32)

[7.3 Ograničenja i oprez pri upotrebi 15](#_Toc33)

[8. Detekcija i obrada outliera 15](#_Toc34)

[8.1 Statističke metode za detekciju outliera 16](#_Toc35)

[8.1.1 Z-score metoda 16](#_Toc36)

[8.1.2 Interkvartilni raspon (IQR) 16](#_Toc37)

[8.2 Vizuelne metode 16](#_Toc38)

[8.3 Tehnike za rukovanje outlierima 17](#_Toc39)

[9. Interakcije između tehnika i preporuke za primenu 17](#_Toc40)

[9.1 Redosled transformacija 18](#_Toc41)

[9.2 Kompatibilnost sa algoritmima 18](#_Toc42)

[9.3 Preporučene prakse: Scikit-learn pipeline 19](#_Toc43)

[9.4 Evaluacija uticaja transformacija 19](#_Toc44)

[11. Zaključak 20](#_Toc45)

[11. Literatura 21](#_Toc46)

# ****1. Uvod****

U savremenoj analitici podataka i mašinskom učenju, kvalitet sirovih podataka retko zadovoljava zahteve koji omogućavaju direktnu primenu algoritama. Većina algoritama pretpostavlja određene karakteristike podataka, kao što su homogena skala atributa, odsustvo ekstremnih vrednosti, numerički formati, kao i distribucije koje pogoduju modeliranju. U praksi, međutim, podaci su često heterogeni, sa vrednostima koje se nalaze u različitim opsezima, uključuju kategorijske promenljive ili pokazuju izražene devijacije i izuzetke (outliere). Na primer, podatak o prihodima može biti izražen u hiljadama, dok broj dece ima ograničen raspon, što bez skaliranja vodi ka lošijim rezultatima algoritama poput **KNN**. Zbog toga je transformacija podataka ključna faza u procesu pripreme podataka za analizu i modelovanje.

Transformacije podataka obuhvataju niz tehnika koje omogućavaju bolje ponašanje algoritama, poboljšavajući efikasnost, preciznost i robusnost modela. Ove tehnike uključuju **normalizaciju i skaliranje**, kojima se obezbeđuje komparabilnost vrednosti različitih atributa (Géron, 2019); **promenu raspodele** podataka radi uklanjanja asimetrije i omogućavanja linearnog ponašanja modela (Zheng & Casari, 2018); **enkodiranje kategoričkih atributa**, kako bi se simboličke vrednosti prevele u numerički oblik prihvatljiv za algoritme (Han, Pei, & Kamber, 2022); **diskretizaciju (binning)**, koja se koristi za uprošćavanje kontinuiranih atributa; kao i **detekciju i obradu outliera**, čime se eliminišu vrednosti koje mogu izobličiti model (García, Luengo, & Herrera, 2015).

Efikasnost navedenih transformacija direktno zavisi od konteksta primene. Na primer, algoritmi kao što su **K‑najbližih suseda (KNN)** i **podrška vektorima (SVM)** posebno su osetljivi na skalu i outliere (James, Witten, Hastie, & Tibshirani, 2021). S druge strane, modeli kao što su **Random Forest** ili **Naive Bayes** pokazuju veću otpornost na nestrukturirane i neskalirane podatke (Pyle, 1999). Izbor odgovarajuće transformacije, stoga, zavisi ne samo od statističkih osobina podataka, već i od prirode modela koji se primenjuje (Kotsiantis, Kanellopoulos, & Pintelas, 2006).

Cilj ovog rada jeste da sistematski predstavi i uporedi najčešće tehnike transformacije podataka, identifikujući njihove prednosti, ograničenja i uslove primene. U praktičnom delu rada, biće demonstrirana primena ovih tehnika na konkretnom skupu podataka korišćenjem biblioteke scikit-learn (Scikit-learn, n.d.), kako bi se analizirali efekti transformacija na modelovanje i performanse algoritama.

# ****2. Motivacija i identifikacija problema****

U savremenim analitičkim sistemima i mašinskom učenju, uspeh modela zavisi ne samo od njegove arhitekture ili hiperparametara, već u velikoj meri i od kvaliteta ulaznih podataka. Bez adekvatne pripreme podataka, uključujući odgovarajuće transformacije, čak i najsloženiji modeli mogu proizvesti neprecizne ili nepouzdane rezultate. Ova činjenica postavlja transformaciju podataka kao ključnu fazu u svakom ozbiljnom analitičkom ili prediktivnom sistemu (Han, Pei, & Kamber, 2022; Géron, 2019).

U realnim skupovima podataka, atributi često dolaze u različitim numeričkim opsezima, oblicima distribucije i tipovima vrednosti. Na primer, atribut „plata“ može imati vrednosti izražene u hiljadama, dok „broj dece“ varira u jednostavnim jednoznamenkastim celobrojnim vrednostima. Ovakve razlike u skali značajno utiču na performanse algoritama koji se oslanjaju na metrike udaljenosti, poput **K‑najbližih suseda (KNN)** ili **linearnih regresionih modela** (James, Witten, Hastie, & Tibshirani, 2021). Bez odgovarajuće normalizacije ili standardizacije, model može dati prekomernu težinu atributima sa većim numeričkim rasponima.

Slično tome, većina algoritama ne može direktno da obradi kategoričke atribute izražene u tekstualnom obliku, kao što su "pol", "država" ili "kategorija proizvoda". Njihova transformacija u numerički format nije trivijalna, jer neadekvatna enkodiranja mogu uvesti lažne odnose ili hijerarhije između kategorija (Zheng & Casari, 2018). Na primer, korišćenje ordinalnog enkodiranja za nominalne promenljive može implicirati nepostojeći redosled koji model kasnije koristi kao osnovu za donošenje odluka.

Još jedan čest problem jesu distribucije atributa koje odstupaju od normalnosti. Mnogi statistički i mašinski modeli pretpostavljaju da su podaci normalno distribuirani ili barem simetrični. U stvarnosti, podaci su često **asimetrični**, **log-normalni**, **eksponencijalni**, ili sadrže **outliere**. Takva ponašanja mogu ozbiljno narušiti rezultate modela, pogotovo u slučajevima kada je model osetljiv na varijanse i ekstreme (García, Luengo, & Herrera, 2015; Kotsiantis, Kanellopoulos, & Pintelas, 2006). U takvim slučajevima, transformacije raspodele (npr. log-transformacija, Box-Cox ili Yeo-Johnson transformacija) postaju neophodne kako bi se obezbedila robusnost i stabilnost algoritama.

Problemi sa kvalitetom podataka takođe uključuju **prisustvo ekstremnih vrednosti (outliera)** koje mogu imati nesrazmeran uticaj na parametarske modele poput linearne regresije, kao i na procene udaljenosti u klasterskim i klasifikacionim algoritmima. Na primer, čak i jedan ekstremno veliki iznos u atributu poput „prihod“ može pomeriti srednju vrednost i standardnu devijaciju celog skupa, što dovodi do pogrešnih zaključaka i modelskih pristrasnosti (Pyle, 1999).

Motivacija za sprovođenje transformacija nad podacima, stoga, nije samo tehničke prirode, već predstavlja metodološki imperativ. Bez njih, podaci ne odražavaju pravilno odnose u posmatranoj domeni, čime se ugrožava čitav proces zaključivanja. Kako su algoritmi sve sofisticiraniji i zahtevniji u pogledu strukture podataka, tako raste i značaj precizno izvedenih transformacija (Géron, 2019).

Osim toga, efikasna priprema podataka smanjuje potrebu za kompleksnim modelima, jer omogućava jednostavnijim algoritmima da postignu visoke performanse. Na taj način, transformacije doprinose i interpretabilnosti modela, što je izuzetno važno u domenu odlučivanja, zdravstva, finansija i drugih visokorizičnih oblasti (James et al., 2021; Han et al., 2022).

Na osnovu ovih uvida, jasno je da transformacija podataka nije fakultativni korak, već temeljna komponenta svakog ozbiljnog analitičkog procesa. U narednim poglavljima biće detaljno analizirane pojedinačne tehnike transformacije, sa identifikacijom konkretnih problema koje rešavaju, uslova primene, kao i njihovih prednosti i ograničenja.

# ****3. Normalizacija i standardizacija****

Transformacija numeričkih podataka kroz normalizaciju i standardizaciju predstavlja jedan od najosnovnijih, ali i najvažnijih koraka u pripremi podataka za algoritme mašinskog učenja. Ove transformacije imaju za cilj da obezbede komparabilnost različitih atributa, eliminišu pristrasnosti uzrokovane skalom, i stabilizuju numeričke procedure u algoritamskim implementacijama (Géron, 2019; James et al., 2021).

## ****3.1. Definicije i matematičke formulacije****

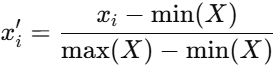
Pre nego što se numerički podaci proslede algoritmima mašinskog učenja, često je neophodno da se njihova skala dovede na standardizovan ili kontrolisan nivo. Ova klasa transformacija, poznata kao **skaliranje**, ima za cilj da izbegne situacije u kojima pojedini atributi, samo zbog većih numeričkih vrednosti, dominiraju nad drugima tokom treniranja modela.

Skaliranje je naročito važno kod algoritama koji se oslanjaju na udaljenosti (npr. KNN), pravce u vektorskom prostoru (npr. SVM), ili dimenzionalnu strukturu podataka (npr. PCA). Bez odgovarajuće transformacije, model može postati osetljiv, pristrasan ili numerički nestabilan.

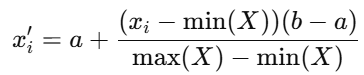
Postoji više pristupa skaliranju, od kojih su najpoznatiji **min-max normalizacija**, **Z-score standardizacija**, **robustno skaliranje** i **maksimalno apsolutno skaliranje**. Svaka od ovih tehnika ima specifične pretpostavke, prednosti i ograničenja, a njihova primenljivost zavisi od distribucije podataka i izbora algoritma.

### ****3.1.1. Min-max normalizacija****

Min-max normalizacija (poznata i kao **reskaliranje**) transformiše podatke tako da svi vrednosti atributa budu projektovane u zadati opseg, najčešće [0,1]. Za datu vrednost xi u vektoru X={x1,x2,…,xn}, transformisana vrednost računa se po formuli:



Alternativno, može se skalirati na proizvoljan opseg [a,b]:



Ova tehnika zadržava relativne odnose među vrednostima, ali je **osetljiva na outliere**, budući da ekstremne vrednosti direktno određuju interval skaliranja (Han et al., 2022).

### ****3.1.2. Z-score standardizacija****

Z-score standardizacija pretvara podatke tako da imaju **srednju vrednost 0** i **standardnu devijaciju 1**. Za datu vrednost xi​, standardizovana vrednost zi računa se kao:



gde su:

* μ - aritmetička sredina skupa X,
* σ - standardna devijacija.

Z-score standardizacija ne ograničava opseg vrednosti, ali omogućava **simetrično tretiranje podataka u odnosu na sredinu**, što je naročito korisno kod algoritama koji pretpostavljaju standardnu normalnu raspodelu (García et al., 2015).

## ****3.2. Kada koristiti koju metodu****

Izbor između min-max normalizacije i Z-score standardizacije zavisi od prirode podataka i zahteva konkretnog algoritma. **Min-max normalizacija** se preporučuje u situacijama kada je važno očuvati opseg vrednosti, naročito kod algoritama koji očekuju ulazne podatke unutar ograničenog raspona. Tipičan primer je primena u **neuronskim mrežama sa sigmoidnim aktivacionim funkcijama**, gde vrednosti koje ulaze u funkciju treba da budu u intervalu između nule i jedan. Ova metoda je naročito pogodna kada su podaci **već dobro kondicionirani**, odnosno kada ne sadrže ekstremne vrednosti koje bi mogle da naruše skaliranje.

S druge strane, **Z-score standardizacija** se pokazuje kao robusnija opcija u slučajevima kada su podaci **raspršeni** i uključuju **outliere**, jer koristi srednju vrednost i standardnu devijaciju za transformaciju, čime omogućava podatke sa različitim raspodelama da budu izraženi u istom statističkom okviru. Ova metoda se posebno preporučuje kod algoritama koji eksplicitno ili implicitno pretpostavljaju **normalnu raspodelu podataka**, kao što su **linearna regresija**, **analiza glavnih komponenti (PCA)** i **podrška vektorima (SVM)**.

U praksi, Z-score standardizacija se često koristi kao **podrazumevani pristup**, posebno kada nije unapred poznata raspodela podataka i kada se želi univerzalna primena transformacije nezavisno od algoritma. Min-max normalizacija, iako efikasna, zahteva oprez prilikom rada sa podacima koji sadrže outliere, jer vrednosti van očekivanog raspona mogu znatno narušiti odnos među primerima i destabilizovati proces treniranja modela (Zheng & Casari, 2018).

|  |  |
| --- | --- |
| Metoda | Kada koristiti |
| **Min-max normalizacija** | Kada je važno očuvanje opsega vrednosti (npr. u neuronskim mrežama sa sigmoid aktivacijom), ili kada su ulazni podaci već dobro kondicionirani. Idealna kada **nema outliera**. |
| **Z-score standardizacija** | Kada su podaci **raspršeni** i sadrže outliere, ili kada se koristi algoritam koji pretpostavlja normalnu raspodelu (npr. linearna regresija, PCA, SVM). Pogodna za **univerzalnu primenu**. |

## ****3.3. Prednosti i mane****

**Min-max normalizacija** je popularna zbog svoje jednostavnosti i intuitivnosti. Jedna od njenih ključnih prednosti jeste što održava relativne odnose među vrednostima atributa, omogućavajući da se svi podaci projektuju u fiksni numerički opseg, najčešće od 0 do 1. Ovakav pristup je naročito pogodan za algoritme koji zahtevaju ulaze u strogo ograničenom opsegu, kao što su neuronske mreže sa sigmoidnim ili tanh aktivacionim funkcijama. Međutim, glavna slabost ove metode jeste izrazita osetljivost na ekstremne vrednosti (outliere). Budući da se minimalna i maksimalna vrednost koriste direktno za skaliranje, prisustvo izuzetaka može značajno iskriviti transformaciju preostalih podataka. Dodatni izazov se javlja kada se u test fazi pojave nove vrednosti van opsega definisanog tokom treniranja, što može dovesti do „eksplozije“ rezultujućih transformisanih vrednosti i destabilizacije modela.

S druge strane, **Z-score standardizacija** je robusnija alternativa, naročito u prisustvu outliera. Ona omogućava transformaciju podataka tako da imaju srednju vrednost nula i standardnu devijaciju jedan, čime se vrednosti izražavaju kao broj standardnih devijacija udaljenih od proseka. Ovakva normalizacija je posebno pogodna za algoritme koji pretpostavljaju standardizovane ulaze, kao što su linearna regresija, logistička regresija, SVM i PCA. Ipak, budući da ne ograničava opseg rezultujućih vrednosti, Z-score standardizacija može proizvesti transformisane vrednosti koje su veoma udaljene od nule. Ovo može otežati interpretaciju podataka u nelinearnim modelima, a ponekad zahteva dodatne tehnike skaliranja kako bi se vrednosti zadržale u prihvatljivim granicama.

## ****3.4. Uticaj na algoritme mašinskog učenja****

Transformacije poput normalizacije i standardizacije imaju direktan i često presudan uticaj na performanse mnogih algoritama mašinskog učenja, posebno onih koji se oslanjaju na metričke ili vektorske karakteristike ulaznog prostora.

Algoritam **K‑najbližih suseda (KNN)** temelji se na proračunu udaljenosti između primera, najčešće korišćenjem Euklidske ili Manhattan metrike. U takvom kontekstu, ukoliko se podaci nalaze na različitim skalama, atributi sa većim numeričkim rasponom nerazmjerno utiču na izračunate udaljenosti. To dovodi do situacije u kojoj model favorizuje određene atribute, ne zbog njihove informativne vrednosti, već isključivo zbog njihove numeričke veličine. Stoga je primena normalizacije ili standardizacije kod KNN modela praktično obavezna kako bi se obezbedila uravnotežena procena sličnosti između primera (James et al., 2021).

Kod **algoritama podrške vektorima (SVM)**, transformacija podataka ima sličnu važnost. Ovi algoritmi funkcionišu u višedimenzionalnim vektorskim prostorima i koriste kernela funkcije koje su osetljive na magnitudu vektora ulaznih primera. Bez standardizacije, vektori različitih skala mogu dovesti do nelinearnih granica koje nisu reprezentativne za stvarnu strukturu podataka. Standardizacijom se obezbeđuje stabilniji prostor pretrage i brža konvergencija algoritma, naročito kod optimizacionih metoda koje se koriste u treniranju SVM-a (Géron, 2019).

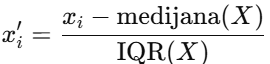
Slično važi i za **analizu glavnih komponenti (PCA)**, koja se oslanja na kovarijacionu matricu kako bi utvrdila pravce najveće varijanse u podacima. Ako ulazni atributi nisu dovedeni na istu skalu, komponente koje sadrže veću numeričku varijansu, nezavisno od stvarne važnosti, dobijaju veći značaj u formiranju novih osa. Time dolazi do iskrivljenja u procesu redukcije dimenzionalnosti. Zbog toga se primena Z-score standardizacije smatra standardnom praksom pre izvođenja PCA transformacije, kako bi se osiguralo ravnomerno učešće svih atributa u analizi (Han, Pei, & Kamber, 2022).

# ****4. Skaliranje****

U savremenim aplikacijama mašinskog učenja, brojčani atributi koji se nalaze u različitim opsezima mogu imati značajan i često negativan uticaj na performanse modela. U cilju obezbeđivanja numeričke stabilnosti i pravilnog ponašanja algoritama, često se primenjuju različite tehnike **skaliranja podataka**, koje za razliku od standardne normalizacije i standardizacije, nude veći stepen robusnosti ili bolje odgovaraju specifičnim tipovima podataka. Skaliranje je naročito važno u prisustvu **asimetričnih distribucija**, **retkih vrednosti**, ili kada su **odstupanja (outlieri)** izražena, što je čest slučaj u realnim datasetovima (García, Luengo, & Herrera, 2015; Géron, 2019).

## ****4.1. RobustScaler****

**RobustScaler** je metoda skaliranja koja koristi **medijanu** i **interkvartilni raspon (IQR)** kao osnovu za transformaciju podataka. Njegova osnovna ideja je da se ukloni uticaj ekstremnih vrednosti, tako što se skala podatka uspostavlja u odnosu na centralni opseg vrednosti. Konkretno, svaka vrednost xi​ transformiše se po formuli:



gde je interkvartilni raspon definisan kao razlika između 75. i 25. percentila. Za razliku od standardizacije, koja koristi srednju vrednost i standardnu devijaciju, ovakav pristup ne podleže deformacijama izazvanim outlierima, jer ni medijana ni kvartili ne zavise od ekstremnih vrednosti.

Ova tehnika se pokazuje naročito korisnom u kontekstu **finansijskih podataka**, **biomedicinskih merenja**, kao i u **industrijskim senzorima**, gde su ekstremne oscilacije u podacima uobičajene. Na primer, ako se u jednoj koloni nalaze vrednosti kao što su: 10, 12, 15, 17, 5000, medijana i interkvartilni raspon će reflektovati pravu strukturu većine podataka, dok bi prosečna vrednost i standardna devijacija bile ozbiljno pomerene.

## ****4.2. MaxAbsScaler****

**MaxAbsScaler** je metoda koja svaku numeričku vrednost deli sa maksimalnom apsolutnom vrednošću u toj koloni, čime se svi podaci dovode u opseg od –1 do 1. Očuvava znakove, odnose među pozitivnim i negativnim vrednostima, i ne pomera sredinu distribucije. Za razliku od min-max normalizacije, koja koristi globalne minimum i maksimum vrednosti za skaliranje, MaxAbsScaler koristi samo **maksimalnu apsolutnu vrednost**, čime se izbegava uticaj negativnih outliera na donju granicu. Formula transformacije je:



Ova tehnika se u praksi često koristi za **vektorske reprezentacije** koje već imaju centriranu raspodelu, kao što su **TF-IDF vektori u obradi prirodnog jezika**, **sparse matrice**, ili **signali** koji osciluju oko nule. Prednost MaxAbsSkalera je i u tome što ne menja raspodelu podataka, koristi se samo linearno skaliranje, pa je koristan kada model zahteva očuvanje smera vektora, npr. kod kosinusne sličnosti.

## ****4.3. Kada je skaliranje važno****

Skaliranje je naročito važno kada se koriste **algoritmi koji se oslanjaju na metričke odnose**, odnosno udaljenosti i geometrijske operacije. U takve algoritme spadaju:

* **KNN (K-Nearest Neighbors)**: koristi Euklidsku ili drugu metriku za poređenje sličnosti;
* **K-means** i drugi algoritmi klasterizacije: centri klastera računaju se kao srednje vrednosti;
* **SVM (Support Vector Machines)**: nalazi optimalnu marginu između klasa u vektorskom prostoru;
* **PCA (Principal Component Analysis)**: koristi kovarijacionu matricu da bi pronašao pravce maksimalne varijanse;
* **Linearna regresija**: iako nije direktno zavisna od skale, značajno zavisi od ravnoteže među prediktorima.

Bez skaliranja, dimenzije sa većim numeričkim rasponom nerazmerno utiču na rezultate, jer se njihova vrednost percipira kao „važnija“ za model samo zbog veličine brojeva. Skaliranje omogućava **uniformnu interpretaciju svih atributa** i obezbeđuje stabilnost algoritama prilikom učenja i predikcije (James et al., 2021).

## ****4.4. Razlike u odnosu na normalizaciju i standardizaciju****

Za razliku od **min-max normalizacije**, koja projektuje vrednosti u unapred definisan opseg [0,1][0,1][0,1], skaleri kao što su RobustScaler i MaxAbsScaler ne zahtevaju definisani raspon. Njihova snaga leži u **robustnosti i jednostavnosti**, odnosno sposobnosti da funkcionišu bez pretpostavke o distribuciji podataka. Takođe, za razliku od **Z-score standardizacije**, skaleri ne pomeraju centar distribucije ka nuli, već samo utiču na skalu, čime zadržavaju osnovni oblik distribucije (posebno MaxAbsScaler).

Standardizacija koristi parametre koji zavise od celokupne raspodele, uključujući outliere, dok **RobustScaler** koristi parametre koji predstavljaju centralni deo distribucije. Time se sprečava da ekstremne vrednosti imaju značajan uticaj na transformaciju. U mnogim slučajevima, kada podaci nisu standardno distribuirani i kada su prisutni ekstremi, skaliranje se nameće kao **preferirani pristup** (Han et al., 2022; Zheng & Casari, 2018).

# ****5. Promena raspodele podataka (transforms)****

U analizi i modeliranju podataka, često se susrećemo sa numeričkim atributima čija raspodela značajno odstupa od normalne (Gaussove) raspodele. Takve raspodele mogu biti **jako asimetrične (skewed)**, **log-normalne**, **eksponencijalne**, ili opterećene prisustvom **izuzetno velikih ili malih vrednosti (outliera)**. U mnogim slučajevima, naročito kada se koriste modeli koji podrazumevaju linearnost i normalnost greške (npr. linearna regresija, logistička regresija, LDA, PCA), ovi oblici raspodele mogu da kompromituju tačnost i stabilnost modela. Kao rešenje, primenjuju se **transformacije raspodele**, koje imaju za cilj da ublaže asimetriju, stabilizuju varijansu i približe podatke normalnoj raspodeli (García, Luengo, & Herrera, 2015; James et al., 2021).

## ****5.1. Log transformacija****

Jedna od najstarijih i najjednostavnijih transformacija za rešavanje problema **desno asimetričnih raspodela** jeste **log-transformacija**. Ova transformacija se primenjuje na pozitivne numeričke vrednosti i definiše se kao:



U praksi se koristi logaritam baze 10 ili prirodni logaritam (ln). Log-transformacija ima efekat **kompresovanja velikih vrednosti**, čime se smanjuje uticaj outliera i ublažava desna repasta distribucija. Istovremeno, ona širi niže vrednosti, čime raspodela postaje simetričnija. Tipična primena log-transformacije uključuje atribute kao što su prihodi, broj pregleda, brojevi poseta i slične kvantitativne mere koje često imaju "dug rep" ka visokim vrednostima (Géron, 2019).

Na primer, uzmimo dataset sa brojem korisnika po aplikaciji:

{10, 15, 30, 45, 1000}

Bez transformacije, ekstremna vrednost 1000 dominiraće modelom. Nakon primene log-transformacije:

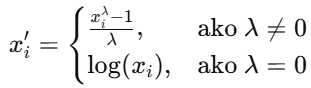
{log(10), log(15), log(30), log(45), log(1000)} ≈ {1.0, 1.18, 1.48, 1.65, 3.0}

Ova transformacija značajno smanjuje disbalans među vrednostima.

Važno je napomenuti da se log-transformacija može primeniti **samo na pozitivne vrednosti**. U slučajevima kada se u podacima nalaze nule ili negativne vrednosti, mora se prvo izvršiti pomeranje (tzv. log(x + c) transformacija), gde je c > 0, što uvodi dodatne izazove u interpretaciji.

## ****5.2. Box–Cox transformacija****

Za razliku od log-transformacije koja koristi fiksnu funkciju, **Box-Cox transformacija** predstavlja **parametarsku familiju transformacija** koja pokušava da pronađe optimalnu funkciju za približavanje normalnosti. Definiše se kao:



gde je λ parametar koji se optimizuje na osnovu podataka (obično tako da se maksimizuje verovatnoća normalnosti).

Box–Cox transformacija ima prednost jer je **fleksibilna i numerički stabilna**, a istovremeno uključuje log-transformaciju kao poseban slučaj. Najveće ograničenje ove metode jeste što se može primeniti **samo na striktno pozitivne podatke**. Zato je u situacijama kada skup podataka sadrži nule ili negativne vrednosti, neophodna prethodna transformacija, što može degradirati prednosti metode (James et al., 2021).

## ****5.3. Yeo-Johnson transformacija****

Kao odgovor na ograničenja Box-Cox transformacije, razvijena je **Yeo-Johnson transformacija**, koja se takođe bazira na parametarskom pristupu, ali za razliku od Box-Cox metode, **može se primeniti i na negativne vrednosti i nule**. Njena definicija zavisi od predznaka ulazne vrednosti i vrednosti parametra λ. Za x ≥ 0, formula je slična Box-Cox transformaciji, dok se za x < 0 koristi poseban oblik koji osigurava kontinuitet i normalizaciju.

Zahvaljujući toj univerzalnosti, Yeo-Johnson transformacija se često koristi u savremenim sistemima za pripremu podataka, jer eliminiše potrebu za unapred definisanom transformacijom vrednosti, i može se automatski optimizovati za svaki atribut pojedinačno. scikit-learn implementacija klase PowerTransformer omogućava primenu Yeo-Johnson metode uz automatsko određivanje optimalnog λ, što je posebno korisno za **pipelinovane transformacije u mašinskom učenju** (Scikit-learn, n.d.; Han et al., 2022).

## ****5.4. Kada i zašto se koristi promena raspodele****

Transformacije raspodele primenjuju se **kada podaci nisu simetrično distribuirani**, kada imaju **izraženu asimetriju** (npr. desno repaste raspodele), ili kada sadrže **outliere** koji narušavaju stabilnost modela. Njihova uloga je da:

* **Stabilizuju varijansu** atributa;
* **Smanje uticaj ekstremnih vrednosti**;
* **Poboljšaju ponašanje algoritama koji pretpostavljaju normalnost** (npr. linearna regresija, LDA, GMM, PCA);
* **Omoguće efikasnije treniranje modela** koji koriste gradijentne metode.

Na primer, u regresionim modelima, asimetrična raspodela ciljne promenljive može dovesti do nelinearnosti odnosa, što narušava validnost modela. Transformacija ciljne promenljive u takvom slučaju omogućava bolju linearnu aproksimaciju odnosa između prediktora i cilja.

## ****5.5. Primeri korišćenja u praksi****

Transformacije raspodele nalaze široku primenu u različitim oblastima analize podataka.

U analizi bioloških merenja, kao što su enzimske aktivnosti ili broj leukocita, podaci često prate log-normalnu raspodelu, te se log-transformacija koristi kako bi se postigla bolja simetrija i stabilizovala varijansa.

U domenu ekonomskih i finansijskih analiza, transformacije su od ključne važnosti prilikom rada sa atributima poput prihoda, kapitalnih ulaganja ili potrošnje po korisniku, koji tipično imaju desno asimetrične i veoma raspršene raspodele.

Slično tome, u informacionim sistemima za preporuke, kao i u analizi korisničkog ponašanja na platformama za pretragu, broj pregleda ili interakcija sa sadržajem često prati dugorepastu distribuciju, pa su transformacije nužne radi postizanja bolje numeričke stabilnosti i efikasnijeg modeliranja.

U savremenim takmičenjima iz oblasti analitike i mašinskog učenja, poput onih koja se održavaju na platformi Kaggle, primena transformacija poput log-transformacije, **Box-Cox** i **Yeo-Johnson** postala je standardna praksa.

Ove metode se rutinski koriste kao deo procesa finog podešavanja modela u cilju povećanja tačnosti predikcija i poboljšanja generalizacije.

# ****6. Enkodiranje kategoričkih promenljivih****

U savremenim sistemima za analizu i modelovanje podataka, kategoričke promenljive predstavljaju važan deo ulaznog prostora, ali ih većina algoritama mašinskog učenja ne može direktno obraditi, budući da operišu isključivo sa numeričkim podacima. Enkodiranje kategoričkih promenljivih je stoga ključna faza u procesu transformacije podataka, jer omogućava prevod diskretnih, simboličkih vrednosti (npr. „muško/žensko“, „plavo/crveno/zeleno“, „visoko/srednje/nisko“) u numeričke forme koje algoritmi mogu interpretirati. Pravilno odabrana metoda enkodiranja može značajno uticati na performanse modela, dok neadekvatna strategija može uvesti lažne relacije, dovesti do eksplozije dimenzionalnosti, ili degradirati efikasnost učenja (García et al., 2015; Han et al., 2022).

## ****6.1. One-hot encoding****

**One-hot enkodiranje** je najčešće korišćena metoda kada su u pitanju **nominalne promenljive**, odnosno kategorije koje nemaju prirodni redosled. Ova metoda funkcioniše tako što se za svaku kategoriju u atributu kreira poseban binarni indikator (kolona), koji dobija vrednost 1 ako primer pripada toj kategoriji, i 0 u suprotnom. Na primer, atribut „boja“ sa vrednostima {crvena, plava, zelena} bi bio transformisan u tri kolone: „boja\_crvena“, „boja\_plava“ i „boja\_zelena“.

Iako je one-hot enkodiranje jednostavno i lako interpretabilno, ono ima jednu ozbiljnu manu - **brzo povećava dimenzionalnost** ulaznog prostora kada promenljiva sadrži veliki broj različitih vrednosti. Ova pojava, poznata kao **„prokletstvo dimenzionalnosti“ (curse of dimensionality)**, može znatno otežati rad algoritama koji zavise od metričkih svojstava prostora, poput KNN-a i SVM-a. Takođe, povećanjem broja dimenzija raste i potreba za više podataka kako bi se model efikasno trenirao, čime se dodatno povećava kompleksnost zadatka (James et al., 2021).

## ****6.2. Label encoding****

**Label enkodiranje** pretvara svaku kategoriju u jedinstvenu celobrojnu vrednost (npr. „mačka“ → 0, „pas“ → 1, „ptica“ → 2). Ova tehnika se koristi zbog svoje jednostavnosti i numeričke kompaktnosti, ali uvodi **implicitni redosled** među kategorijama, koji u realnosti često ne postoji. Kod modela koji ne koriste strukturu vrednosti atributa, poput stabala odlučivanja (Decision Trees), ova implicitna relacija ne predstavlja problem. Međutim, kod algoritama koji se oslanjaju na metričke odnose ili pretpostavljaju kontinualnost (npr. linearna regresija), label enkodiranje može dovesti do pogrešnih zaključaka, jer model može tretirati „pticu“ (2) kao veću od „mačke“ (0) na način koji nema semantičkog opravdanja (Zheng & Casari, 2018).

## ****6.3. Ordinal encoding****

**Ordinal enkodiranje** je specijalni slučaj label enkodiranja, koji se koristi **samo kada kategorije imaju prirodni redosled**, ali ne nužno konstantne razmake između vrednosti. Na primer, kategorije „nizak“, „srednji“, „visok“ mogu se enkodirati kao 0, 1, 2, pri čemu se očuvava informacija o relativnom položaju. Ova metoda je korisna kod algoritama koji mogu interpretirati redosled, ali treba izbegavati njenu primenu kada ne postoji inherentna ordinalna struktura među vrednostima.

Važno je naglasiti da ordinal encoding takođe ne prenosi informaciju o tome da razmak između kategorija nije numerički precizan - razlika između "nizak" i "srednji" ne mora biti jednaka razlici između "srednji" i "visok", i model to sam po sebi ne zna. Zato se u nekim slučajevima ordinalne promenljive ipak tretiraju kao numeričke sa dodatnim kontrolama, npr. pomoću binovanja ili target-encodinga (Géron, 2019).

## ****6.4. Frequency encoding****

**Frequency encoding** dodeljuje svakoj kategoriji broj koji odgovara frekvenciji njenog pojavljivanja u podacima, tj. verovatnoći pojavljivanja kategorije. Na primer, ako se kategorija „A“ javlja u 30% primera, njoj će biti dodeljena vrednost 0.3. Ova metoda čuva kompaktnost numeričkog formata i može biti korisna za modele koji se dobro nose sa nelinearnim odnosima među atributima, poput slučajnih šuma (Random Forest) ili boosting modela. Međutim, postoji opasnost da se uvede **korelacija između učestalosti i ciljne promenljive**, naročito ako se ova tehnika koristi nekritično ili pre razdvajanja trening i test skupa.

## ****6.5 Kada koristiti koju metodu****

Izbor metode enkodiranja kategoričkih promenljivih zavisi od više faktora: tipa promenljive (nominalna ili ordinalna), broja jedinstvenih kategorija, prirode algoritma, i zahteva u pogledu dimenzionalnosti. One-hot enkodiranje je preporučljivo kada su kategorije nominalne i broj mogućih vrednosti relativno mali, čime se izbegava eksplozija dimenzionalnosti. Label i ordinal enkodiranje su korisni kada postoji prirodni poredak među vrednostima, ili kada se koriste algoritmi koji nisu osetljivi na numeričku interpretaciju vrednosti (npr. stabla odlučivanja). Frequency encoding se preporučuje u slučajevima kada postoji veliki broj kategorija i kada je potrebno očuvati kompaktnost ulaznog prostora, ali se mora primeniti pažljivo kako bi se izbeglo uvodjenje pristrasnosti (Han et al., 2022; James et al., 2021).

# ****7. Diskretizacija (binning)****

U procesu transformacije podataka, jedna od često korišćenih tehnika jeste **diskretizacija** (engl. binning), koja podrazumeva pretvaranje kontinuiranih numeričkih vrednosti u diskretne kategorije ili „kofe“ (binove). Ova tehnika može biti korisna iz više razloga: pojednostavljuje model, povećava interpretabilnost, može poboljšati performanse određenih algoritama, a takođe omogućava kombinovanje numeričkih i simboličkih podataka u istom modelu (Han, Pei, & Kamber, 2022).

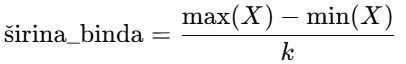
Diskretizacija je naročito korisna u situacijama kada odnosi između atributa i ciljne promenljive nisu linearni, kada su podaci veoma šumoviti, ili kada je potrebno eliminisati sitne numeričke fluktuacije koje nisu informativne za predviđanje. Takođe, u kontekstu algoritama koji rade sa frekvencijama ili entropijama (npr. Naive Bayes ili algoritmi zasnovani na odlučujućim stablima), diskretizacija omogućava stabilnije i brže učenje modela, jer se smanjuje broj različitih vrednosti koje algoritam mora analizirati (James et al., 2021).

## ****7.1 Vrste diskretizacije****

Diskretizacija se može sprovoditi na različite načine, zavisno od strategije podele vrednosti. Dve osnovne grupe metoda su: **nesupervizovane** (ne uzimaju u obzir ciljnu promenljivu) i **supervizovane** (uzimaju u obzir povezanost sa ciljem). U nastavku su predstavljene najčešće korišćene nesupervizovane metode.

### ****7.1.1 Uniformna (jednaka širina) diskretizacija****

Kod uniformne diskretizacije, celokupan opseg vrednosti atributa se deli na kkk intervala (binova) **jednake širine**. Ako su minimalna i maksimalna vrednost atributa označene kao min(X) i max(X), širina jednog binda izračunava se pomoću sledeće formule:



Svaka vrednost xi​ se zatim dodeljuje bindu j, gde je:



**Primer:**

Razmotrimo skup vrednosti:

X=[12,15,21,22,28,34,35,36,38,40]

Opseg podataka je 40−12=28. Ako izaberemo k=3 binda, dobijamo širinu:



Intervali (zaokruženi): [12-21.33), [21.33-30.67), [30.67-40]. Svaka vrednost se zatim smešta u odgovarajući bin.

### ****7.1.2 Kvantilna (jednaka frekvencija) diskretizacija****

Kvantilna diskretizacija funkcioniše tako da svaka grupa (bin) sadrži približno isti broj primera**.** Granice binova određuju se na osnovu percentila. Ova metoda garantuje ravnomernu distribuciju broja instanci po bindeovima, ali može rezultovati intervalima veoma različite širine ako su podaci neravnomerno raspoređeni.

**Primer:**

Za isti skup X = [12,15,21,22,28,34,35,36,38,40], možemo kreirati tri binda tako da svaki sadrži približno 3-4 vrednosti:

Bin 1: [12, 15, 21]

Bin 2: [22, 28, 34]

Bin 3: [35, 36, 38, 40]

Ova metoda osigurava uravnoteženost binova, ali može dovesti do binova sa veoma različitim širina intervala kada su podaci neravnomerno raspoređeni.

Uprkos tome, kvantilna diskretizacija se često koristi jer garantuje raspodelu podataka po grupama jednake veličine, što može biti korisno za algoritme osetljive na broj instanci po klasi (García, Luengo, & Herrera, 2015).

### ****7.1.3 K-means binning****

K-means binning predstavlja napredniji pristup, u kojem se vrednosti atributa grupišu korišćenjem algoritma K-means, čime se formiraju grupe koje minimizuju unutargrupne varijanse. Za razliku od prethodnih metoda koje koriste statičke granice, K-means binning koristi **klasterski pristup** koji se adaptira stvarnoj raspodeli podataka. Ova metoda je pogodna kada podaci imaju višemodalne raspodele ili kada standardne metode ne uspevaju da uhvate relevantne grupe. Ipak, uvođenje nelinearne metodologije kao što je K-means može da poveća složenost sistema i oteža interpretaciju rezultata (Zheng & Casari, 2018).

## ****7.2 Primene i prednosti diskretizacije****

Diskretizacija se koristi u različitim domenima, uključujući medicinu, ekonomiju, psihometriju i segmentaciju korisnika. U kontekstu **eksplorativne analize podataka**, binning može pomoći u identifikaciji obrazaca i trendova, omogućujući lakšu vizuelnu interpretaciju odnosa između promenljivih. Takođe, u **modelima zasnovanim na pravilima i stablima odlučivanja**, diskretizovani podaci dovode do stabilnijih granica podela i manje pretreniranja.

Jedna od ključnih prednosti diskretizacije je **smanjenje osetljivosti na outliere**. Kada se vrednosti grupišu u binove, ekstremne vrednosti se više ne razlikuju od ostalih vrednosti u istom intervalu, što doprinosi robusnosti modela. Pored toga, diskretizacija može doprineti **boljoj interpretabilnosti modela**, jer je lakše razumeti relacije među kategorijama nego među apstraktnim numeričkim vrednostima (Han et al., 2022; Géron, 2019).

## ****7.3 Ograničenja i oprez pri upotrebi****

Glavno ograničenje diskretizacije jeste **gubitak preciznosti**, jer se unutar svakog binda vrednosti više ne razlikuju. Ovo može negativno uticati na algoritme koji zavise od preciznih numeričkih odnosa, kao što su linearna regresija, SVM ili neuronske mreže. Takođe, ako se granice binova biraju bez uvida u relaciju sa ciljnom promenljivom, može doći do **gubitka informacije**, što se negativno odražava na kvalitet modela.

Zato se preporučuje primena supervizovanih tehnika diskretizacije kada god je to moguće — one biraju granice binova tako da se maksimizuje informativnost u pogledu ciljne promenljive. Alternativno, diskretizaciju treba koristiti uz dodatne analize uticaja na performanse modela, posebno kod kompleksnijih struktura podataka (James et al., 2021).

# ****8. Detekcija i obrada outliera****

U okviru pripreme podataka za analizu i modelovanje, **outlieri**, odnosno **izolovane i ekstremne vrednosti**, predstavljaju jedan od ključnih izazova. Ove vrednosti odstupaju značajno od ostatka skupa podataka i mogu proizvesti **netačne statističke procene**, destabilizovati modele mašinskog učenja, kao i dovesti do **pogrešnih zaključaka** u analizi (García, Luengo, & Herrera, 2015; Han, Pei, & Kamber, 2022). Zbog toga je identifikacija i odgovarajuće rukovanje outlierima od suštinskog značaja za integritet procesa rudarenja podataka i mašinskog učenja.

Važno je naglasiti da outlieri nisu nužno „greške“ u podacima – oni mogu predstavljati **validne ali retke događaje**. Na primer, visoka potrošnja određenog korisnika u bankarskom sistemu može biti i signal anomalije (npr. prevara), ali i indikator korisnika sa velikim kapitalom. Zbog toga detekcija i obrada outliera mora biti pažljivo vođena **kontekstom domene** i ciljevima analize.

## ****8.1 Statističke metode za detekciju outliera****

## ****8.1.1 Z-score metoda****

Z-score metoda detektuje outliere na osnovu odstupanja pojedinačnih vrednosti od aritmetičke sredine izraženog u standardnim devijacijama. Za svaku vrednost xi, Z-score se računa kao:



gde je μ srednja vrednost, a σ standardna devijacija posmatranog atributa.

U praksi se često koristi prag od 3 (odnosno ∣zi∣> 3 za označavanje ekstremnih vrednosti. Ova metoda pretpostavlja da su podaci približno **normalno distribuirani**, zbog čega može biti nepouzdana kod asimetričnih raspodela ili u prisustvu više modova (James, Witten, Hastie, & Tibshirani, 2021).

### ****8.1.2 Interkvartilni raspon (IQR)****

Za razliku od Z-score pristupa, IQR metoda ne zavisi od pretpostavki o raspodeli i stoga je **robustnija**. Izračunava se kao:

IQR = Q3−Q1

gde su Q1 i Q3​ prvi i treći kvartil. Vrednosti koje leže ispod:

**donja granica = Q1−1.5⋅IQR**

ili iznad:

**gornja granica = Q3+1.5⋅IQR**

se klasifikuju kao outlieri. IQR metoda se često primenjuje u domenu biomedicine i ekonomije upravo zbog svoje **otpornosti na asimetrične distribucije** i stabilnosti u prisustvu ekstremnih vrednosti (García et al., 2015).

### ****8.2 Vizuelne metode****

Vizuelizacija igra ključnu ulogu u preliminarnoj identifikaciji outliera. **Boxplot** (dijagram kutije i brkova) je standardna tehnika kojom se prikazuju kvartili, medijana i potencijalni outlieri.

Na ovom grafikonu:

* centralna kutija predstavlja opseg između Q1 i Q3​,
* linija u kutiji označava medijanu,
* "brkovi" se protežu do vrednosti unutar 1.5 IQR,
* tačke izvan brkova predstavljaju potencijalne outliere.

Boxplot omogućava **brzu detekciju ekstremnih vrednosti**, vizuelno upoređivanje više atributa, kao i poređenje raspodela između različitih grupa. Dodatne tehnike uključuju **scatter plot** (kod bivariantnih odnosa) i **histograme** za uvid u učestalost ekstremnih vrednosti (Géron, 2019).

## ****8.3 Tehnike za rukovanje outlierima****

Jednom kada su outlieri identifikovani, postoji više pristupa za njihovo rukovanje, pri čemu je izbor metode uslovljen prirodom podataka, ciljem analize i modelom koji se koristi.

**Ignorisanje outliera** je često prihvatljivo u robustnim modelima (npr. random forest), ili kada se zna da ekstremne vrednosti nisu štetne po strukturu podataka. Međutim, ovaj pristup je rizičan ako se koristi kod modela koji su osetljivi na udaljenosti (npr. KNN, regresija), jer takvi outlieri mogu imati disproporcionalan uticaj.

**Uklanjanje outliera** se primenjuje kada su oni rezultat očiglednih grešaka u unosu, senzorima ili preprocesiranju. Ovaj pristup mora biti pažljivo dokumentovan i primenjen uz domenski uvid. Neselektivno uklanjanje može dovesti do gubitka vrednih informacija.

**Imputacija outliera** podrazumeva zamenu ekstremnih vrednosti procenjenim vrednostima (npr. medijanom, kvartilima, ili predikcijom na osnovu sličnih instanci). Ovaj pristup se koristi kada broj outliera nije zanemarljiv, ali njihovo potpuno uklanjanje nije prihvatljivo.

**Robustne metode** predstavljaju sofisticiraniji pristup, gde se koriste modeli i statistike koje su otporne na uticaj ekstremnih vrednosti. Na primer, **robustna regresija** koristi ponderisane procene koje smanjuju uticaj outliera, dok algoritmi kao što su **Isolation Forest**, **Elliptic Envelope**, ili **Local Outlier Factor (LOF)** u okviru scikit-learn biblioteke, detektuju i modeluju outliere direktno.

Važno je naglasiti da se pristupi ne isključuju – u praksi se često kombinuju, uz procenu njihovog efekta na performanse modela (Zheng & Casari, 2018; Han et al., 2022).

# ****9. Interakcije između tehnika i preporuke za primenu****

Transformacije podataka u procesu pripreme za analizu nisu izolovane operacije koje se primenjuju proizvoljnim redosledom. Naprotiv, njihova **međusobna zavisnost, kompatibilnost sa konkretnim algoritmima**, kao i **redosled primene**, direktno utiču na performanse modela i tačnost zaključaka. Razumevanje međusobnih odnosa između transformacija je ključno za izgradnju robusnog, efikasnog i interpretabilnog sistema mašinskog učenja (Han, Pei, & Kamber, 2022; Géron, 2019).

## ****9.1 Redosled transformacija****

Redosled u kojem se primenjuju različite tehnike transformacije podataka igra ključnu ulogu u očuvanju statističkih svojstava i stabilnosti modela. Na primer, **detekcija i obrada outliera treba da prethode standardizaciji ili skaliranju**. Ako se outlieri ne uklone ili ublaže pre izračunavanja srednje vrednosti i standardne devijacije, one će biti iskrivljene, što vodi do pogrešnih Z-score vrednosti i neadekvatnog skaliranja.

Slično tome, **enkodiranje kategoričkih promenljivih** treba da se izvrši **pre spajanja sa numeričkim atributima** u istu obradu, kako bi se izbeglo uključivanje nestrukturiranih podataka u transformacione operacije namenjene brojevima. Takođe, u slučajevima kada se koristi **dimenzionalna redukcija (npr. PCA)**, ona mora slediti skaliranje i imputaciju, jer se bazira na kovarijacionoj matrici koja je osetljiva na skalu i prazne vrednosti (James, Witten, Hastie, & Tibshirani, 2021).

Uopšteno, preporučeni redosled transformacija je:

**1. Uklanjanje duplikata i nekonzistentnosti**

**2. Detekcija i obrada outliera**

**3. Imputacija**

**4. Skaliranje / standardizacija**

**5. Enkodiranje**

**6. Selektovanje / redukovanje dimenzija.**

## ****9.2 Kompatibilnost sa algoritmima****

Različiti algoritmi imaju različite zahteve u pogledu forme i raspodele ulaznih podataka. **Algoritmi bazirani na udaljenosti** (npr. KNN, SVM, K-means) zahtevaju skalirane podatke kako bi svi atributi imali jednak uticaj na izračunavanje udaljenosti. **Linearni modeli** (npr. linearna regresija, LDA) mogu biti osetljivi na multikolinearnost i zahtevaju standardizaciju. **Naivni Bayes** dobro funkcioniše sa diskretnim i kategoričkim atributima, dok **stabla odlučivanja** (npr. Decision Trees, Random Forest) ne zahtevaju skaliranje i otporna su na outliere.

Zbog toga je izbor transformacija neraskidivo povezan sa algoritmom koji se koristi. Na primer, za SVM ili PCA preporučuje se prethodna standardizacija Z-score metodom, dok za Random Forest skaliranje nije potrebno, ali imputacija može poboljšati robusnost modela (García, Luengo, & Herrera, 2015; Han et al., 2022).

## ****9.3 Preporučene prakse: Scikit-learn pipeline****

Biblioteka scikit-learn obezbeđuje formalni mehanizam za upravljanje nizom transformacija kroz klase Pipeline i ColumnTransformer. Ovi objekti omogućavaju povezivanje više koraka transformacije (npr. imputacija, skaliranje, enkodiranje) u jedinstven tok, čime se postiže:

* **konsistentnost transformacija** u trening i test fazi,
* **smanjenje grešaka** u pripremi podataka,
* **lakše eksperimentisanje** sa različitim modelima.

Na primer, kombinacija SimpleImputer, StandardScaler, i OneHotEncoder može se primeniti na odgovarajuće kolone pomoću ColumnTransformer, dok se ceo proces, uključujući treniranje modela, obuhvata Pipeline objektom.

from sklearn.pipeline import Pipeline

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.impute import SimpleImputer

from sklearn.ensemble import RandomForestClassifier

Ovakav pristup omogućava potpunu **reproduktivnost i automatizaciju** toka podataka, što je od suštinskog značaja u profesionalnim okruženjima i prilikom produkcionih implementacija (Géron, 2019; Scikit-learn, n.d.).

## ****9.4 Evaluacija uticaja transformacija****

Jedan od važnijih zadataka u obradi podataka je **kvantifikacija efekta transformacija** na performanse modela. Ova evaluacija se može vršiti:

* **Kvalitativno**, kroz vizuelizaciju distribucija pre i posle transformacije (npr. histogrami, boxplotovi, scatter plotovi nakon PCA).
* **Kvantitativno**, putem poređenja metrika modela (npr. tačnost, F1-score, RMSE) sa i bez određene transformacije.

Posebno se preporučuje sprovođenje **eksperimenata sa različitim transformacionim tokovima**, kako bi se empirijski potvrdilo koja kombinacija transformacija daje najbolje rezultate. U scikit-learn okruženju to se može realizovati korišćenjem GridSearchCV u kombinaciji sa Pipeline, što omogućava testiranje više varijanti istovremeno.

Evaluacija transformacija ne služi samo za poboljšanje performansi, već i za **razumevanje koje karakteristike podataka najviše utiču na model**, što doprinosi interpretabilnosti i transparentnosti procesa (Zheng & Casari, 2018).

## ****11. Zaključak****

Transformacija podataka predstavlja neizostavan deo savremene analitike i procesa mašinskog učenja. Kroz ovaj rad pokazano je da kvalitet transformacija direktno utiče na tačnost modela, stabilnost algoritama, robusnost prema ekstremnim vrednostima, kao i na interpretabilnost i prenosivost rešenja. U osnovi svakog uspešnog modela nalazi se **dobro oblikovan proces pripreme podataka**, koji podrazumeva ne samo čišćenje i imputaciju, već i pažljivo osmišljene transformacije numeričkih i kategoričkih atributa.

Jedna od ključnih pouka ovog rada jeste da **ne postoji univerzalna tehnika transformacije** koja je optimalna za sve probleme. Umesto toga, izbor konkretne metode zavisi od:

* raspodele podataka (npr. da li postoji asimetrija),
* prisustva ekstremnih vrednosti (outliera),
* prirode atributa (kategorički, numerički, ordinalni),
* i, što je najvažnije, **zahteva modela koji se koristi**.

Na primer, dok standardizacija (Z-score) daje odlične rezultate kod algoritama osetljivih na veličine (npr. SVM, PCA), robustni skaleri su pogodniji u prisustvu izuzetaka. Slično tome, one-hot enkodiranje je efikasno za niskokardinalne nominalne promenljive, dok se kod visokokardinalnih atributa preporučuju frekventni ili cilj-no-vođeni pristupi. Redosled primene transformacija je takođe od ključne važnosti, jer loš raspored (npr. skaliranje pre detekcije outliera) može dovesti do značajnih iskrivljenja.

U ovom radu su takođe prikazane konkretne preporuke za integraciju transformacija u sistemima mašinskog učenja, posebno kroz scikit-learn okruženje koje omogućava izgradnju konzistentnih i reproduktivnih tokova pomoću Pipeline i ColumnTransformer struktura. Time se obezbeđuje ne samo tehnička ispravnost, već i održivost modela u realnim, produkcionim uslovima.

Kao pravac za dalju analizu i unapređenje, otvara se potreba za razvojem sistema koji mogu **automatski birati optimalne transformacije** u zavisnosti od problema. U tom kontekstu, sve veću pažnju privlače **AutoML sistemi**, koji osim izbora modela i hiperparametara, uključuju i fazu automatske obrade podataka — uključujući selekciju transformacija, enkodera i skalera. Automatizovana evaluacija uticaja različitih transformacija, zasnovana na metrikama učinka modela, može značajno smanjiti potrebu za manuelnim eksperimentisanjem i ubrzati razvoj prototipa.

U zaključku, transformacija podataka nije puka tehnička procedura, već **strategijska komponenta** svakog sistema zasnovanog na podacima. Njeno razumevanje, pravilna primena i evaluacija predstavljaju osnov za izgradnju robusnih, pouzdanih i interpretabilnih modela u svim domenima primene mašinskog učenja.

# 11. Literatura

1. Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O’Reilly Media.
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning: With applications in R (2nd ed.). Springer.
3. Zheng, A., & Casari, A. (2018). Feature engineering for machine learning: Principles and techniques for data scientists. O’Reilly Media.
4. Han, J., Pei, J., & Kamber, M. (2022). Data mining: Concepts and techniques. Elsevier.
5. Kelleher, J. D., Mac Namee, B., & D’Arcy, A. (2020). Fundamentals of machine learning for predictive data analytics: Algorithms, worked examples, and case studies. MIT Press.
6. Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised learning. Journal of Intelligent Information Systems, 27(1), 1–23.
7. García, S., Luengo, J., & Herrera, F. (2015). Data preprocessing in data mining. Springer.
8. Pyle, D. (1999). Data preparation for data mining. Morgan Kaufmann.
9. Scikit-learn developers. (n.d.). Preprocessing data. scikit-learn documentation. Retrieved July 29, 2025, from https://scikit-learn.org/stable/modules/preprocessing.html
10. Towards Data Science. (n.d.). A complete guide to data preprocessing in machine learning. Retrieved July 29, 2025, from https://towardsdatascience.com/