UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET

IZBOR ATRIBUTA

Seminarski rad

Studijski program: Veštačka inteligencija i mašinsko učenje

Student:

Marko Anđelković, br. ind. 1809

Ментор:

prof. dr. Aleksandar Stanimirović

Niš, decembar 2025.

Sadržaj

1. Uvod

Izbor atributa (*Feature Selection, FS*) predstavlja jedan od najvažnijih koraka u pripremi podataka za modele mašinskog učenja. Savremeni skupovi podataka često sadrže veliki broj atributa, među kojima se mogu naći redundantne, neinformativne ili čak štetne karakteristike koje negativno utiču na rad modela. Prevelika dimenzionalnost povećava složenost podataka, usporava proces treniranja i podiže rizik od prenaučenosti (*overfitting*), što direktno umanjuje sposobnost modela da generalizuje na nove podatke.

Cilj selekcije karakteristika jeste da se identifikuje manji, ali informativniji podskup atributa koji najbolje opisuje strukturu podataka i doprinosi prediktivnoj tačnosti modela. Uklanjanjem suvišnih karakteristika postiže se efikasnije treniranje, smanjuje se računarska složenost, poboljšava interpretabilnost modela i umanjuje rizik od prekomernog prilagođavanja.

Zbog svoje sposobnosti da unapredi performanse modela, pojednostavi strukturu podataka i omogući bržu i pouzdaniju analizu, selekcija karakteristika predstavlja nezamenjiv deo savremenog procesa mašinskog učenja.

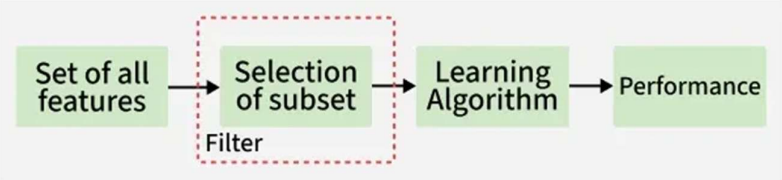
Izbor atributa može se definisati kao proces selekcije optimalnog podskupa karakteristika u skladu sa određenim kriterijumom. Kriterijum određuje način evaluacije podskupova atributa i mora biti usklađen sa ciljevima analize. Optimalan podskup često predstavlja najmanji broj karakteristika koji omogućava najbolju procenu prediktivne tačnosti.

Razlozi za primenu selekcije karakteristika uključuju: uklanjanje nerelevantnih podataka, povećanje tačnosti modela, smanjenje troškova obrade podataka, unapređenje efikasnosti učenja (na primer smanjenjem zahteva za memorijom i računske resurse), smanjivanje složenosti modela i bolje razumevanje podataka.

Proces izbora atributa može se posmatrati iz više perspektiva: kao problem pretrage optimalnog podskupa, kao izbor kriterijuma za procenu značaja atributa ili kao postupak uklanjanja, dodavanja i prilagođavanja karakteristika tokom modelovanja. U literaturi se najčešće izdvajaju tri glavne grupe metoda: **filter**, **wrapper** i **embedded** pristupi, koji se razlikuju prema načinu procene važnosti karakteristika i stepenu zavisnosti od samog modela.

2. Filter metode

Filter metode predstavljaju jednu od najčešće korišćenih grupa tehnika za izbor atributa, zasnovanih na statističkim i informacijskim merama koje procenjuju korisnost atributa nezavisno od modela mašinskog učenja. Ključna ideja ovih metoda jeste da se značaj svakog atributa određuje isključivo na osnovu njegove veze sa ciljnom promenljivom, bez treniranja prediktivnog modela (Slika 1*Slika 1*). Zbog toga su filter pristupi izuzetno brzi, skalabilni i pogodni za analizu skupova podataka sa velikim brojem dimenzija.



Slika 1. Princip rada filter metoda

Najčešće korišćene mere u okviru filter metoda uključuju:

- Prag varijanse (*Variance Threshold*): eliminiše karakteristike sa malom varijansom, jer takvi atributi nose malo informacija i ne doprinose razlikovanju klasa.
- Uzajamna informacija (*Mutual Information*): meri količinu informacije koju atribut pruža o ciljnoj promenljivoj, pri čemu može otkriti i nelinearne zavisnosti između podataka.
- Srednja apsolutna devijacija (*Mean Absolute Difference, MAD*): ocenjuje značaj atributa merenjem prosečnog odstupanja vrednosti između različitih klasa; veća devijacija ukazuje na veću diskriminativnu moć.
- Hi-kvadrat test (*Chi-square test*): statistički test koji procenjuje zavisnost između kategorijalne karakteristike i ciljne promenljive, pogodan za detekciju atributa koji imaju značajan uticaj na klasifikaciju.
- Fišerov rezultat (*Fisher's score*): rangira atribute na osnovu odnosa između interklasne i intraklasne varijabilnosti; veći rezultat znači bolju sposobnost razdvajanja klasa.
- Pirsonova korelacija (*Pearson's correlation*): meri linearnu vezu između atributa i ciljne promenljive; jača korelacija ukazuje na veći uticaj atributa na predikciju.
- ANOVA F-test: Upoređuje srednje vrednosti atributa između različitih klasa i pokazuje koliko atribut doprinosi razdvajanju klasa.
- Odnos disperzije (*Dispersion Ratio*): Odnos varijanse između klasa i unutar klasa; veći odnos znači bolju diskriminativnu sposobnost atributa.

Osnovna prednost filter metoda je njihova računarska efikasnost, koja ih čini pogodnim prilikom obrade velikih skupova podataka. Takođe, zbog nezavisnosti od modela, one su generalno robustne i primenljive u širokom spektru algoritama mašinskog učenja.

Međutim, filter pristupi imaju i ograničenja. Budući da se izbor obavlja bez učešća modela, ove metode ne uzimaju u obzir kompleksne interakcije između atributa niti njihov kombinovani efekat na performanse modela. Moguće je da atributi koji statistički izgledaju relevantno ne doprinose predikciji u praktičnoj primeni ili pak atribut može biti koristan tek u kombinaciji sa drugim, ali će biti odbačen zbog slabe individualne korelacije.

Uprkos ovim ograničenjima, filter metode ostaju dragoceni alat za inicijalnu obradu podataka, brzu eliminaciju nerelevantnih karakteristika i pripremu skupova podataka za sofisticiranije tehnike poput **wrapper** i **embedded** pristupa.

2.1. Prag varijanse (Variance Threshold)

Prag varijanse (*Variance Threshold*) predstavlja jednu od najjednostavnijih filter metoda za izbor atributa, gde se karakteristike eliminišu na osnovu njihove varijanse. Ideja metode zasniva se na tome da atributi sa veoma malom varijansom nose malo informacija, jer se njihove vrednosti gotovo ne menjaju među uzorcima i ne doprinose razlikovanju klasa. Zadržavaju se samo oni atributi čija varijansa prelazi unapred definisani prag, čime se smanjuje dimenzionalnost i eliminišu neinformativne ili redundantne karakteristike.

Ova metoda je posebno korisna kod velikog broja numeričkih atributa i često se primenjuje kao prvi korak pre složenijih tehnika selekcije.

Matematička formulacija

Za dati atribut $X = \{x_1, x_2, \cdots, x_n\}$ varijansa se definiše kao:

$$Var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

Gde je srednja vrednost μ:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Ako važi uslov

$$Var(X) < \Theta,$$

onda se atribut eliminiše; u suprotnom se zadržava.

Prag varijanse Θ je unapred definisan i ako je postavljen 0 onda se uklanjanju samo konstante.

2.2. Uzajamna informacija (Mutal Information)

Uzajamna informacija (*Mutual Information, MI*) je mera zavisnosti između atributa i ciljne promenljive. Za razliku od korelacije, MI može da detektuje nelinearne odnose i ne ograničava se na linearnu povezanost. U kontekstu selekcije karakteristika, MI se koristi da proceni koliko atribut smanjuje neizvesnost o ciljnoj promenljivoj.

Ukoliko atribut ima visoku MI vrednost, znači da njegova raspodela značajno utiče na raspodelu ciljne promenljive → dobar atribut za model.

Ako atribut ima nisku MI vrednost, znači da ne pruža korisnu informaciju → loš ili nerelevantan atribut.

Prednosti MI:

- meri i linearne i nelinearne veze (za razliku od Pearson korelacije),
- radi i sa kategorijskim i numeričkim podacima,
- robustan na transformacije podataka.

Matematička formulacija

Uzajamna informacija (*MI*) između promenljive X i Y definiše se kao:

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}$$

Gde su:

$p(x, y)$ – zajednička verovatnoća X i Y,

$p(x)$ – marginalna verovatnoća X,

$p(y)$ – marginalna verovatnoća Y

2.3. Srednja apsolutna devijacija (Mean Absolute Difference)

Srednja apsolutna devijacija (Mean Absolute Difference, MAD) predstavlja meru rasipanja podataka oko njihove srednje vrednosti. Za razliku od varijanse, koja kvadrira odstupanja i time naglašava ekstremne vrednosti, MAD koristi apsolutna odstupanja, zbog čega je robusnija na prisustvo outliera.

U kontekstu izbora atributa, MAD se koristi kao filter metoda za identifikaciju atributa čije se vrednosti dovoljno razlikuju među instancama. Atributi sa veoma malom prosečnom apsolutnom devijacijom pokazuju malu varijabilnost, što ukazuje da nose malo informacija i da se njihove vrednosti gotovo ne menjaju kroz uzorke, pa se takvi atributi često uklanjaju iz modela. Nasuprot tome, atributi sa višim MAD vrednostima značajnije variraju kroz podatke i imaju veću diskriminativnu moć.

Prednost MAD metode ogleda se u njenoj jednostavnosti i robusnosti, kao i u maloj računskoj složenosti, što je čini pogodnom za brzo rangiranje atributa, naročito kod velikih skupova podataka. Međutim, MAD ne uzima u obzir odnos između atributa i ciljne promenljive, već isključivo meri varijabilnost atributa. Zbog toga može zadržati atribute sa velikom varijabilnošću koji ne doprinose predikciji, kao i propustiti atribute čiji je uticaj zavisan od nelinearnih ili interaktivnih odnosa, što je čini manje efikasnom u složenijim zadacima u poređenju sa metodama poput Mutual Information ili model-baziranih pristupa.

Matematička formulacija

Za dati atribut $X = \{x_1, x_2, \cdots, x_n\}$ srednja apsolutna devijacija se definiše kao:

$$MAD(X) = \frac{1}{n}\sum_{i=1}^{n}|x_i - \mu|$$

Gde je srednja vrednost μ:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

2.4. Hi-kvadrat test (Chi-square test)

Hi-kvadrat test (Chi-square, χ^2) predstavlja statističku meru za procenu zavisnosti između kategorijskog atributa i ciljne promenljive. Osnovna ideja testa zasniva se na poređenju očekivanih frekvencija, koje se dobijaju pod pretpostavkom nezavisnosti, sa posmatranim frekvencijama u tabeli učestanosti. Veće odstupanje između ovih vrednosti ukazuje na jaču povezanost atributa i ciljne promenljive. U kontekstu izbora atributa, hi-kvadrat test se koristi kao filter metoda za rangiranje kategorijskih atributa prema njihovoj informativnosti.

Atributi koji ostvaruju visoku vrednost χ^2 statistike najverovatnije imaju značajnu zavisnost od ciljne promenljive i nose korisne informacije za model, dok niske χ^2 vrednosti ukazuju na slab ili zanemarljiv uticaj atributa, zbog čega takvi atributi mogu biti eliminisani. Metoda je posebno pogodna za velike skupove podataka sa diskretnim atributima, jer je jednostavna i računski efikasna, ali nije primenljiva na numeričke podatke bez prethodne diskretizacije, niti u slučajevima kada su očekivane frekvencije veoma male.

Glavno ograničenje hi-kvadrat testa jeste njegova primena isključivo na kategorijske promenljive, pri čemu diskretizacija numeričkih atributa može dovesti do gubitka informacija. Takođe, test je osetljiv na veličinu uzorka, pa u veoma velikim skupovima podataka čak i slabe veze mogu postati statistički značajne. Zbog toga χ^2 test ne meri jačinu veze, već samo postojanje zavisnosti, te se u praksi najčešće koristi kao deo šireg postupka izbora atributa.

Matematička formulacija

Za kategorijski atribut sa tabelom učestanosti koja sadrži očekivanih frekvencija E_{ij} i posmatranih frekvencija O_{ij}, hi-kvadrat statistika definiše se kao:

$$\chi^2 = \sum_i \sum_j \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}}$$

Gde su:

i, j – indeksi kategorije ciljne promenljive i ulaznog atributa

O_{ij} – posmatrana (empirijska) učestalost, broj pojavljivanja kombinacije kategorija

E_{ij} – očekivana učestalost, izračunata pod pretpostavkom da ne postoji zavisnost između atributa i ciljne promenljive

Opšta forma tabela očestanosti pomoću koje se računa $E_{ij} = \frac{R_i C_j}{N}$

	Kategorija Y_1	Kategorija Y_2	...	Kategorija Y_m	Ukupno (X_i)
Kategorija X_1	O_{11}	O_{12}	...	O_{1m}	R_1
Kategorija X_2	O_{21}	O_{22}	...	O_{2m}	R_2
...	...	...	...	...	...
Kategorija X_n	O_{n1}	O_{n2}	...	O_{nm}	R_3
Ukupno (Y_j)	C_1	C_2	...	C_m	N

2.5. Fišerov rezultat (Fisher's score)

Fišerov rezultat (*Fisher's Score*) predstavlja jednu od najpoznatijih filter metoda za izvor atributa, posebno u binarnoj i višeklasnoj klasifikaciji. Osnovna ideja ove metode zasniva se na merenju koliko dobro jedan atribut razdvaja klase - odnosno koliko se njihove srednje vrednosti razlikuju u odnosu na unutarklasnu varijansu. Što je razlika između klasa veća, a varijansa unutar klasa manja, atribut je bolji kandidat za prediktivni model.

U praksi, atributi sa visokim Fišerovim rezultatom pokazuju snažnu diskriminativnu moć: jasno razdvajaju klase jer je njihova raspodela različita između klasa, ali stabilna unutar svake klase.

Nasuprot tome, atributi sa malim Fišerovim rezultatom imaju slične vrednosti za sve klase ili pokazuju veliku varijabilnost unutar klasa, što ih čini lošim prediktorima.

Glavno ograničenje Fisher score-a jeste to što pretpostavlja linearnu razdvojivost i meri značaj atributa pojedinačno, bez uzimanja u obzir međusobnih interakcija. To može dovesti do toga da atributi koji zajedno imaju visok informativni značaj budu zanemareni. Takođe je osetljiv na neravnomernu zastupljenost klasa i na ekstremne vrednosti, što može izmeniti sredine i varijanse i time pogoršati rangiranje.

Matematička formulacija

Za binarnu klasifikaciju, Fišerov rezultat atributa X definisan je kao:

$$F = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

Gde su:

μ_1, μ_2 – srednje vrednosti atributa u klasi 1 i klasi 2

σ_1^2, σ_2^2 – varijanse atributa unutar svake klase

U slučaju višeklasne klasifikacije, Fisherov rezultat se proširuje tako što se računa odnos varijanse između više klasa prema prosečnoj unutar-klasnoj varijansi. Atribut dobija viši Fisherov skor ukoliko dobro razdvaja više različitih klasa, odnosno kada su srednje vrednosti klasa udaljene, a varijansa unutar svake klase mala.

2.6. Pirsonova korelacija (Pearson's correlation)

Pirsonova korelacija predstavlja statističku meru koja opisuje snagu i smer linearnog odnosa između dve numeričke promenljive. Vrednost koeficijenta kreće se od –1 do 1, gde su −1 i 1 indikatori savršene negativne i pozitivne linearne veze, dok vrednost 0 označava odsustvo linearnog odnosa.

U kontekstu izbora atributa, Pirsonova korelacija se koristi da bi se identifikovali oni atributi koji imaju najjaču linearnu vezu sa ciljnom promenljivom, kao i da bi se eliminisali atributi koji pokazuju visoku međusobnu redundantnost.

Glavno ograničenje Pirsonove korelacije jeste to što meri isključivo linearne odnose i ne može otkriti nelinearne zavisnosti, što može dovesti do pogrešnih zaključaka u kompleksnim datasetovima. Osetljiva je na autlajere, jer i jedan ekstremni podatak može značajno promeniti koeficijent i time narušiti procenu veze. Takođe, korelacija ne implicira kauzalnost, pa čak i visoka vrednost koeficijenta ne znači da promenljiva utiče na cilj direktno. U slučajevima kategorijalnih ili jako izobličenih podataka, Pirsonova korelacija postaje nepouzdana.

Matematička formulacija

Pirsonov koeficijent meri stepen linearne korelacije između dve numeričke promenljive X, Y:

$$r_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Gde su:

x_i, y_i – pojedinačni uzorci

$\bar{x}, \bar{y}$ – srednje vrednosti

2.7. ANOVA F-test

ANOVA F-test (*Analysis of Variance*) predstavlja statističku metodu koja ispituje da li se srednje vrednosti numeričke osobine značajno razlikuju između dve ili više klasa ciljne promenljive.

U kontekstu izbora atributa koristi se da proceni koliko dobro određeni atribut razdvaja klase, što je razlika između klasa veća u odnosu na varijaciju unutar klasa, to je F-vrednost veća i atribut informativniji.

Suština ANOVA-e je poređenje između-klasne varijanse i unutar-klasne varijanse, a njihov odnos formira F-statistiku. Visoka F-vrednost ukazuje da se klase jasno razlikuju prema datoj osobini, dok niska vrednost sugeriše da atribut ne poseduje dovoljnu diskriminativnu moć. Metod je efikasan, jednostavan za implementaciju i posebno koristan kod linearnog razdvajanja numeričkih podataka u klasifikacionim zadacima.

ANOVA F-test se najčešće koristi u problemima klasifikacije sa numeričkim prediktorima i kategorijalnim targetom. Pretpostavke metode uključuju normalnu raspodelu unutar klasa i približnu homogenost varijansi, ali test je u praksi prilično robustan i često se koristi i kada ove pretpostavke nisu strogo ispunjene.

Matematička formulacija

F-statistika se definiše kao odnos srednje vrednosti unutar-klasne varijanse i srednje vrednosti između-klasne varijanse:

$$F = \frac{MS_B}{MS_W}$$

Gde su:

MS_B srednja vrednost između-klasne varijanse (Mean Between-class variance)

MS_W srednja vrednost unutar-klasne varijanse (Mean Within-class variance)

Srednje vrednosti se računaju na sledeći način:

$$MS_B = \frac{1}{K-1}\sum_{k=1}^{K} n_k(\bar{X}_k - \bar{X})^2\,;\; MS_w = \frac{1}{K-1}\sum_{k=1}^{K}\sum_{i=1}^{n_k}(X_{ik} - \bar{X}_k)^2\,;$$

Gde su: K – broj klasa (broj različitih vrednosti ciljne promenljive), n_k – broj uzoraka u klasi K,

$\bar{X}_k$ – srednja vrednost klase K, $\bar{X}_k$ – ukupna srednja vrednost svih podataka

2.8. Odnos disperzije (Dispersion Ratio)

Odnos disperzije (*Dispersion Ratio, DR*) predstavlja filter metodu izbora atributa koja meri koliko dobro pojedinačni atribut razdvaja klase ciljne promenljive. Osnovna ideja ove metode zasniva se na poređenju disperzije vrednosti atributa između klasa i disperzije unutar klasa. Atributi koji imaju veliku varijabilnost između klasa, a malu varijabilnost unutar klasa, smatraju se informativnijim za klasifikaciju.

Iako su odnos disperzije i ANOVA F-test zasnovani na istoj ideji poređenja između-klasne i unutar-klasne varijanse, njihova namena se razlikuje. ANOVA F-test predstavlja statistički test kojim se ispituje da li su razlike između klasa statistički značajne, uz korišćenje F-raspodele i p-vrednosti. Nasuprot tome, odnos disperzije se koristi isključivo kao mera za rangiranje atributa u postupku izbora atributa, bez izvođenja statističkog testiranja i statističkog zaključivanja.

Odnos disperzije omogućava jednostavnu i efikasnu procenu diskriminativne moći numeričkih atributa, bez potrebe za treniranjem modela, što ovu metodu čini pogodnom za rad sa velikim skupovima podataka.

Matematička formulacija

Odnos disperzije za atribut X definiše se kao odnos između-klasne varijanse i unutar-klasne varijanse:

$$DR(X) = \frac{\sigma_B^2}{\sigma_W^2}$$

Gde se između-klasna varijansa računa kao varijansa srednjih vrednosti atributa po klasama:

$$\sigma_B^2 = VAR(\bar{X}_1, \bar{X}_2, \bar{X}_3, \dots, \bar{X}_K)$$

Dok se unutar-klasna varijansa računa kao prosečna varijansa atributa unutar pojedinačnih klasa:

$$\sigma_W^2 = \frac{1}{K}\sum_{k=1}^{K} \sigma_k^2$$

Gde su:

K – broj različitih vrednosti ciljne promenljive

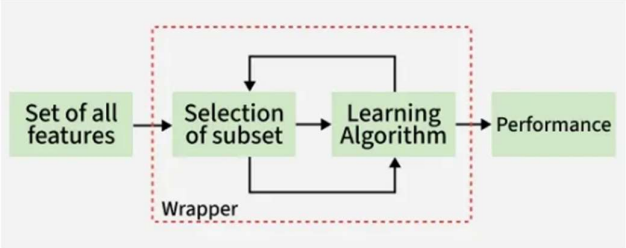
$\bar{X}_K$ – srednja vrednost atributa u klasi K

σ_k^2 – varijansa atributa u klasi K

3. Wrapper metode

Wrapper metode predstavljaju grupu tehnika za izbor atributa koje procenjuju značaj atributa na osnovu performansi konkretnog modela mašinskog učenja. Za razliku od filter metoda, koje posmatraju svaki atribut nezavisno od modela, wrapper metode koriste algoritam učenja kao „crnu kutiju“ i vrednuju različite podskupove atributa treniranjem i evaluacijom modela.

Osnovna ideja wrapper metoda zasniva se na iterativnom pretraživanju prostora mogućih kombinacija atributa, pri čemu se za svaki podskup atributa trenira model i meri njegova tačnost ili neka druga evaluaciona metrika (*Slika 2*). Podskup atributa koji daje najbolje performanse smatra se optimalnim. Zbog ovakvog pristupa, wrapper metode mogu da uzmu u obzir međusobne zavisnosti i interakcije između atributa, što često dovodi do boljih rezultata u poređenju sa filter metodama.



Slika 2. Princip rada wrapper metoda

Najčešće korišćene wrapper metode uključuju:

- Sekvencijalni selekcija unapred (*Sequential Forward Selection, SFS*): proces započinje praznim skupom atributa, a zatim se iterativno dodaje atribut koji najviše poboljšava performanse modela.
- Sekvencijalna selekcija unazad (*Sequential Backward Elimination, SBE*): započinje se sa kompletnim skupom atributa, nakon čega se u svakom koraku uklanja atribut čije izbacivanje najmanje utiče na performanse modela.
- Rekurzivna eliminacija atributa (*Recursive Feature Elimination, RFE*): metoda koja koristi važnost atributa iz kako bi iterativno uklanjala najmanje važne atribute.

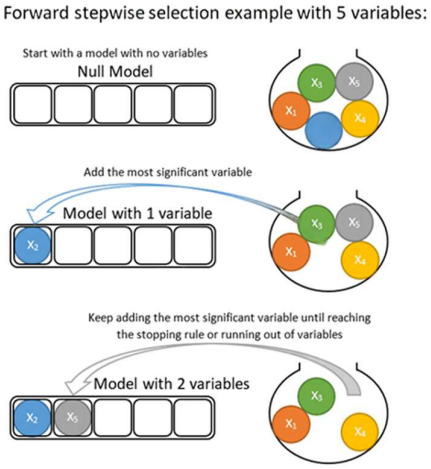
Glavna prednost wrapper metoda ogleda se u njihovoj sposobnosti da pronađu podskup atributa koji je direktno optimizovan za dati model, što često rezultuje većom tačnošću predikcije.

Međutim, ovaj pristup je računski zahtevan, posebno kod velikih skupova podataka sa velikim brojem atributa, jer zahteva višestruko treniranje modela. Takođe, wrapper metode su podložne preprilagođavanju (*overfitting*), naročito kada se koriste sa složenim modelima i malim skupovima podataka.

Uprkos ovim ograničenjima, wrapper metode predstavljaju moćan alat za izbor atributa, naročito kada je cilj postizanje maksimalnih performansi modela i kada su raspoloživi dovoljni računski resursi. U praksi se često kombinuju sa filter metodama, koje služe za preliminarno smanjenje dimenzionalnosti, nakon čega se wrapper pristup koristi za finu selekciju atributa.

3.1. Sekvencijalna selekcija unapred (SFS)

Sekvencijalna unapred selekcija (*Sequential Forward Selection, SFS*) je wrapper metoda izbora atributa koja započinje bez izabranih atributa i iterativno dodaje one koji u datom koraku najviše poboljšavaju performanse modela, sve dok se ne dostigne željeni broj atributa ili dok dodatno poboljšanje više nije moguće (*Slika 3*).



Slika 3. Princip rada sekvencijalne selekcije unapred (SFS)

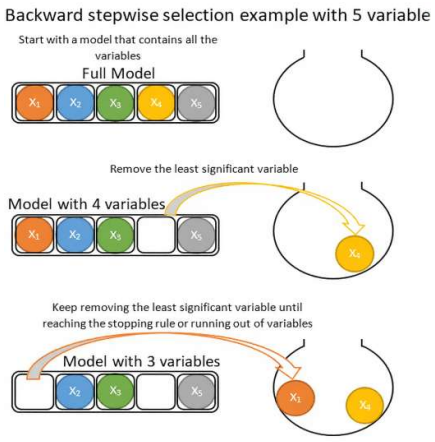
Pseudokod

Ovim pseudokodom opisan je algoritam sekvencijalne unapred selekcije (*SFS*), koji započinje praznim skupom izabranih atributa S. U svakom koraku funkcija ***FindNext(F)*** bira najbolji atribut iz preostalog skupa F, koji se zatim dodaje u S i uklanja iz F. Proces se ponavlja sve dok se ne ispuni kriterijum zaustavljanja ili dok se ne iscrpe svi atributi.

```
function SFG(F – full set, U - measure)
        initialize: S = {} # S stores the selected features
        repeat
                f = FINDNEXT(F)
                S = S ∪ {f}
                F = F – {f}
        until S satisfies U or F = {}
        return S
end function
```

3.2. Sekvencijalna selekcija unazad (SBS)

Sekvencijalna selekcija unazad (*Sequential Backward Selection, SBS*) je wrapper metoda izbora atributa koja započinje sa kompletnim skupom atributa i iterativno uklanja atribute čijim izostavljanjem dolazi do najmanjeg pogoršanja performansi izabranog modela (*Slika 4*). Cilj je dobijanje manjeg, ali informativnog skupa atributa koji zadržava dobar kvalitet predikcije.



Slika 4. Princip rada sekvencijalne selekcije unazad (SBS)

Pseudokod

Ovim pseudokodom opisana je sekvencijalna selekcija unazad (SBS), u kojoj se polazi od punog skupa atributa F i u svakom koraku se uklanja jedan atribut koji je najmanje značajan prema izabranoj meri kvaliteta U. Funkcija ***GETNEXT(F)*** bira atribut čije uklanjanje najmanje utiče na vrednost izabrane mere kvaliteta U. Uklonjeni atributi se čuvaju u skupu S, a postupak se ponavlja sve dok uklanjanje atributa ne naruši zadati kriterijum ili dok skup F ne postane prazan ili dok se ne ispuni kriterijum zaustavljanja. Na kraju se vraća preostali skup atributa koji zadovoljava kriterijum kvaliteta.

```
function SBG(F – full set, U - measure)
        initialize: S = {} # S holds the removed features
        repeat
                f = GETNEXT(F)
                F = F – {f}
                S = S ∪ {f}
        until S does not satisfy U or F = {}
        return F ∪ {f}
end function
```

3.3. Rekurzivna eliminacija atributa (RFE)

Rekurzivna eliminacija atributa *(Recursive Feature Elimination, RFE*) predstavlja wrapper metodu izbora atributa koja koristi informacije o značajnosti atributa dobijene iz treniranog modela. Postupak započinje sa kompletnim skupom atributa, nakon čega se u svakom koraku uklanja atribut sa najmanjom važnošću, određenom na osnovu koeficijenata modela ili drugih mera značajnosti. Proces se ponavlja sve dok se ne dobije unapred definisan broj atributa ili dok se ne ispuni kriterijum zaustavljanja, čime se obezbeđuje izbor kompaktnog i relevantnog skupa atributa (*Slika 5*).

Slika 5. Princip rada rekurzivne eliminacije atributa (RFE)

Razlika između SBS i RFE

Razlika između **SBS** i **RFE** ogleda se u kriterijumu eliminacije atributa: **SBS** u svakom koraku uklanja atribut čijim izostavljanjem dolazi do najmanjeg pogoršanja performansi modela, dok **RFE** uklanja atribute na osnovu njihove značajnosti dobijene iz treniranog modela (npr. koeficijenata ili važnosti atributa). Zbog toga je SBS precizniji u sagledavanju međuzavisnosti atributa, ali računarski zahtevniji, dok je RFE efikasniji i brži, ali zavisi od izbora modela i načina procene značajnosti atributa.

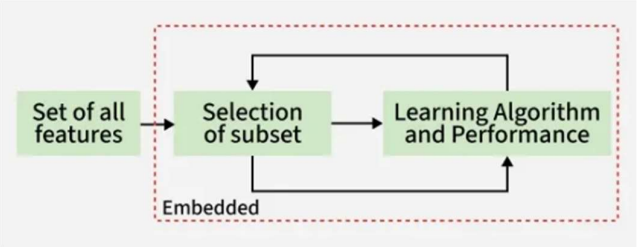
Pseudokod

Ovaj pseudokod prikazuje osnovni tok rekurzivne eliminacije atributa (RFE), pri čemu se u svakom koraku trenira model nad trenutnim skupom atributa F i izračunava njihova značajnost. Funkcija ***GETNEXT(F)*** identifikuje atribut sa najmanjom važnošću, koji se zatim uklanja iz skupa F i dodaje u skup eliminisanih atributa S. Postupak se ponavlja iterativno sve dok se ne ispuni kriterijum zaustavljanja, čime se dobija redukovan skup atributa zasnovan na proceni značajnosti odabranog modela.

```
function RFE(F – full set, U - measure)
        initialize: S = {} # S holds the removed features
        repeat
                f = GETNEXT(F)
                F = F – {f}
                S = S ∪ {f}
        until S does not satisfy U or F = {}
        return F ∪ {f}
end function
```

4. Embedded metode

Embedded metode izbora atributa predstavljaju pristup u kome se selekcija atributa sprovodi tokom samog procesa treniranja modela, bez potrebe za posebnim korakom evaluacije ili pretrage različitih podskupova atributa (*Slika 6*). Ove metode koriste unutrašnje mehanizme modela, poput regularizacije ili strukture stabla odlučivanja, kako bi automatski dodelile veći značaj relevantnim, a potisnule manje važne atribute. Na taj način embedded metode ostvaruju dobar kompromis između kvaliteta izabranih atributa i računarske efikasnosti, jer istovremeno uzimaju u obzir međuzavisnosti atributa i smanjuju složenost modela.



Slika 6 Princip rada embedded metoda

Najčešće korišćene embedded metode uključuju:

- Lasso (L1 regularizacija): metoda koja tokom treniranja modela primenjuje L1 penalizaciju, čime se koeficijenti manje značajnih atributa svode na nulu, što omogućava direktnu selekciju atributa.
- Ridge (L2 regularizacija): koristi L2 penalizaciju kako bi smanjila vrednosti koeficijenata atributa, čije se smanjuje njihov uticaj na model, ali bez potpunog eliminisanja atributa.
- Elastic Net: kombinuje L1 i L2 regularizaciju, čime se istovremeno ostvaruje selekcija atributa i stabilnost modela u prisustvu korelisanih atributa.
- Decision Trees i Random Forest: modeli zasnovani na stablima odlučivanja koji procenjuju značajnost atributa na osnovu smanjenja nečistoće, pri čemu se manje značajni atributi prirodno zanemaruju tokom izgradnje modela.
- Gradient Boosting: ansambl metoda koja gradi model iterativno, pri čemu se značajnost atributa određuje na osnovu njihovog doprinosa smanjenju greške predikcije.

Embedded metode omogućavaju istovremeno treniranje modela i izbor atributa, čime se postiže dobra ravnoteža između kvaliteta selekcije i računarske efikasnosti. Pošto je selekcija integrisana u sam algoritam učenja, ove metode prirodno uzimaju u obzir međuzavisnosti između atributa i često rezultuju kompaktnijim i stabilnijim modelima u poređenju sa filter metodama.

Glavni nedostatak embedded metoda je njihova zavisnost od izabranog modela i njegovih pretpostavki, zbog čega rezultati selekcije mogu varirati u zavisnosti od primenjenog algoritma. Takođe, embedded metode često zahtevaju pažljivo podešavanje hiperparametara (npr. koeficijenata regularizacije), a dobijeni skup atributa može biti manje interpretabilan u složenijim modelima.

Uprkos određenim ograničenjima, embedded metode predstavljaju efikasan pristup izboru atributa, jer integrišu selekciju direktno u proces treniranja modela. Zbog povoljnog odnosa između performansi i računarske složenosti, često se koriste samostalno ili u kombinaciji sa filter metodama za inicijalno smanjenje dimenzionalnosti.

4.1. Lasso

Lasso (*Least Absolute Shrinkage and Selection Operator*) predstavlja embedded metodu izbora atributa koja koristi L1 regularizaciju tokom treniranja modela, sa ciljem istovremenog smanjenja složenosti modela i selekcije relevantnih atributa. Za razliku od klasične linearne regresije, Lasso uvodi penalizaciju apsolutnih vrednosti koeficijenata, čime se smanjuje uticaj manje značajnih atributa.

U kontekstu izbora atributa, Lasso zadržava one atribute čiji koeficijenti imaju značajnu vrednost, dok atribute sa malim doprinosom modelu potiskuje tako što njihove koeficijente svodi na nulu. Na taj način, atributi čiji su koeficijenti jednaki nuli smatraju se neinformativnim i eliminišu se iz modela, dok preostali atributi čine selektovani skup.

Atributi sa velikom apsolutnom vrednošću koeficijenata imaju značajan uticaj na ciljnu promenljivu i zadržavaju se u modelu. Suprotno tome, atributi sa malim ili nultim koeficijentima imaju zanemarljiv doprinos predikciji i bivaju eliminisani. Jačina ovog procesa direktno zavisi od vrednosti parametra regularizacije λ.

Prednost Lasso metode ogleda se u njenoj sposobnosti da izvrši eksplicitnu selekciju atributa, čime se poboljšava interpretabilnost modela i smanjuje rizik od preučenja. Međutim, Lasso može biti osetljiv na korelisane atribute, jer u takvim slučajevima često zadržava samo jedan atribut iz grupe međusobno zavisnih atributa, dok ostale eliminiše.

Matematička formulacija

Lasso metoda određuje koeficijente modela rešavanjem sledećeg optimizacionog problema:

$$\widehat{\boldsymbol{\beta}} = argmin_{\boldsymbol{\beta}}\left(\sum_{i=1}^{n}(y_i - x_i^T\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right)$$

Gde su:

$\widehat{\boldsymbol{\beta}}$ – procenjeni vektor koeficijenata modela nakon treniranja

$\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ – vektor koeficijenata koji određuju uticaj pojedinačnih atributa

n – broj uzoraka

p – broj atributa

y_i – stvarna vrednost ciljne promenljive za i-ti uzorak

$x_i^T\boldsymbol{\beta}$ – predikcija modela za i-ti uzorak

$\lambda \geq 0$ - parametar regularizacije koji kontroliše jačinu penalizacije

4.2. Ridge

Ridge regresija predstavlja embedded metodu izbora atributa koja koristi L2 regularizaciju tokom treniranja modela, sa ciljem istovremenog smanjenja složenosti modela i poboljšanja njegove generalizacije. Za razliku od Lasso metode, Ridge uvodi penalizaciju kvadrata koeficijenata, čime se smanjuju vrednosti koeficijenata, ali se oni u pravilu ne svode na tačno nulu.

U kontekstu izbora atributa, Ridge zadržava sve atribute u modelu, dok se atributi sa manjim doprinosom predikciji potiskuju smanjenjem vrednosti njihovih koeficijenata. Na taj način, iako atributi formalno ostaju u modelu, njihov uticaj može postati zanemarljiv, naročito u slučaju velike vrednosti parametra regularizacije λ.

Atributi sa većim apsolutnim vrednostima koeficijenata imaju značajniji uticaj na ciljnu promenljivu, dok atributi sa veoma malim koeficijentima doprinose predikciji u manjoj meri. Prednost Ridge metode ogleda se u njenoj stabilnosti i dobrom ponašanju u prisustvu međusobno korelisanih atributa, jer ima tendenciju da raspodeli uticaj između takvih atributa. Međutim, pošto ne vrši eksplicitnu eliminaciju atributa, Ridge je manje pogodan kada je cilj dobiti mali i jasno definisan skup atributa.

Matematička formulacija

Ridge metoda određuje koeficijente modela rešavanjem sledećeg optimizacionog problema:

$$\widehat{\boldsymbol{\beta}} = argmin_{\boldsymbol{\beta}}(\sum_{i=1}^{n}(y_i - x_i^T\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}{\beta_j}^2)$$

Gde su:

$\widehat{\boldsymbol{\beta}}$ – procenjeni vektor koeficijenata modela nakon treniranja

$\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_p)$ – vektor koeficijenata koji određuju uticaj pojedinačnih atributa

n – broj uzoraka

p – broj atributa

y_i – stvarna vrednost ciljne promenljive za i-ti uzorak

$x_i^T\boldsymbol{\beta}$ – predikcija modela za i-ti uzorak

$\lambda \geq 0$ - parametar regularizacije koji kontroliše jačinu penalizacije

4.3. Elastic Net

Elastic Net predstavlja embedded metodu izbora atributa koja kombinuje L1 i L2 regularizaciju, sa ciljem objedinjavanja prednosti Lasso i Ridge metoda. Uvođenjem kombinovanog regularizacionog člana, Elastic Net omogućava istovremenu selekciju atributa i stabilizaciju koeficijenata, naročito u prisustvu međusobno korelisanih atributa.

U kontekstu izbora atributa, Elastic Net zadržava atribute sa značajnim koeficijentima, dok atribute sa malim doprinosom predikciji potiskuje ili eliminiše u zavisnosti od odnosa između L1 i L2 penalizacije. Na taj način, metoda može eliminisati nevažne atribute, ali i raspodeliti težine između korelisanih atributa, čime se postiže stabilniji i interpretabilniji model.

Atributi sa većim apsolutnim vrednostima koeficijenata imaju veći uticaj na ciljnu promenljivu, dok atributi sa malim ili nultim koeficijentima doprinose predikciji u manjoj meri. Prednost Elastic Net metode ogleda se u njenoj fleksibilnosti, jer omogućava podešavanje odnosa između selekcije i regularizacije. Međutim, metoda zahteva podešavanje dodatnih hiperparametara, što može povećati složenost procesa treniranja.

Matematička formulacija

$$\widehat{\boldsymbol{\beta}} = argmin_{\boldsymbol{\beta}}(\sum_{i=1}^{n}(y_i - x_i^T\boldsymbol{\beta})^2 + \lambda_1\sum_{j=1}^{p}|\beta_j| + \lambda_2\sum_{j=1}^{p}{\beta_j}^2)$$

Gde su:

$\lambda_1 \geq 0, \lambda_2 \geq 0$ – parametri koji kontrolišu jačinu L1 i L2 penalizacije

4.4. Decision Trees and Random Forest

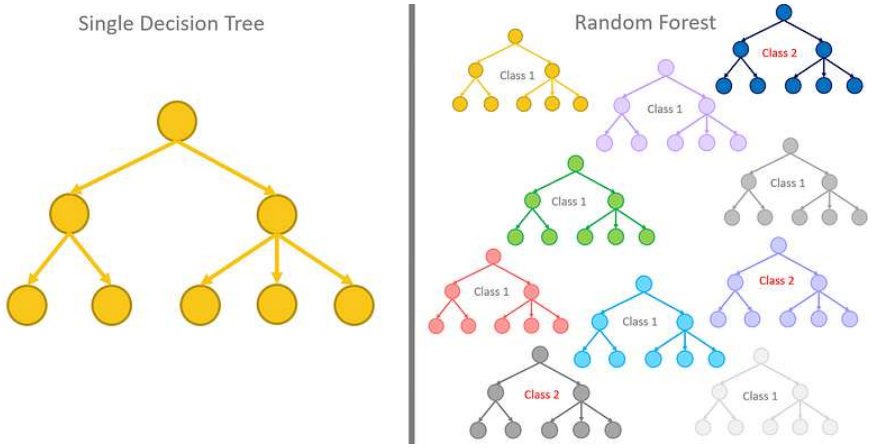
Stabla odlučivanja (*Decision Trees*) i *Random Forest* predstavljaju embedded metode izbora atributa kod kojih se selekcija atributa odvija tokom izgradnje modela. Ovi modeli procenjuju značajnost atributa na osnovu njihovog doprinosa smanjenju nečistoće (npr. Gini indeks ili entropija), pri čemu se atributi koji omogućavaju bolje razdvajanje podataka preferiraju u procesu grananja.

U kontekstu izbora atributa, stabla odlučivanja prirodno favorizuju atribute sa većim informativnim doprinosom, dok se atributi sa malim uticajem ređe koriste ili se uopšte ne pojavljuju u strukturi stabla. Random Forest, kao ansambl više stabala odlučivanja, dodatno stabilizuje procenu značajnosti atributa tako što agregira informacije iz velikog broja stabala, čime se smanjuje varijansa i zavisnost od pojedinačnog stabla.

Atributi sa većom vrednošću mere značajnosti imaju veći uticaj na odluke modela, dok atributi sa malim doprinosom predikciji imaju zanemarljiv uticaj na odluke modela. Prednost ovih metoda ogleda se u njihovoj sposobnosti da modeluju nelinearne odnose i interakcije između atributa, kao i u relativnoj robusnosti na skaliranje podataka. Međutim, procena značajnosti atributa može biti pristrasna u korist atributa sa većim brojem mogućih vrednosti, a pojedinačna stabla odlučivanja mogu biti sklona preučenju (*overfitting*).

Razlika između Decision Trees i Random Forest

Razlika između stabala odlučivanja i Random Forest modela ogleda se u načinu izgradnje i stabilnosti modela (*Slika 7*). Stabla odlučivanja koriste jedno stablo za donošenje odluka, što omogućava visoku interpretabilnost, ali ih čini sklonim preučenju i osetljivim na promene u podacima. Random Forest, kao ansambl metoda, gradi veliki broj stabala na različitim poduzorcima podataka i kombinuje njihove rezultate, čime se smanjuje varijansa modela i dobija stabilnija i pouzdanija procena značajnosti atributa, uz nešto manju interpretabilnost u poređenju sa pojedinačnim stablom.



Slika 7. Razlika između Decision Trees i Random Forest

Matematička formulacija

U stablima odlučivanja, izbor atributa u svakom čvoru zasniva se na maksimalnom smanjenju nečistoće. Za dati atribut i odgovarajuću podelu, smanjenje nečistoće može se izraziti kao:

$$\Delta I = I_{parrent} - \sum_{m=1}^{M} \frac{N_m}{N} I_m$$

Gde su:

$I_{parrent}$ – mera nečistoće roditeljskog čvora (najčešće Gini indeks ili entropija)

I_m – mera nečistoće m-tog deteta

N – broj uzoraka u roditeljskom čvoru

N_m – broj uzoraka u m-tom detetu

M – broj potomaka nakon podele

Mere nečistoće čvora su:

$$Gini = 1 - \sum_{k=1}^{K} p_k^2$$

$$Entropy = -\sum_{k=1}^{K} p_k \log_2 p_k$$

Gde je p_k udeo uzoraka koji pripadaju klasi k

Kod **Random Forest** modela, značajnost atributa dobija se agregacijom smanjenja nečistoće tog atributa preko svih stabala u ansamblu, čime se obezbeđuje stabilnija i pouzdanija procena u odnosu na pojedinačna stabla odlučivanja.

4.5. Gradient Boosting

Gradient Boosting predstavlja embedded metodu učenja zasnovanu na ansamblu, koja gradi model iterativno, dodavanjem novih slabih učenika (najčešće plitkih stabala odlučivanja) tako da svaki sledeći model ispravlja greške prethodnog. Osnovna ideja je da se u svakoj iteraciji minimizuje funkcija gubitka, pri čemu se novi model trenira na osnovu gradijenta greške (reziduala) trenutnog ansambla.

U kontekstu izbora atributa, Gradient Boosting procenjuje značajnost atributa na osnovu toga koliko često i koliko korisno se atribut koristi pri podelama u stablima, odnosno koliki doprinos daje smanjenju greške ili nečistoće tokom izgradnje ansambla. Atributi koji češće dovode do većeg poboljšanja performansi dobijaju veću značajnost, dok atributi sa malim doprinosom imaju nižu vrednost značajnosti.

Učenje u Gradient Boosting modelu može se formalno opisati sledećom rekurzivnom relacijom:

$$F_M(x) = F_{M-1}(x) - \nu h_M(x)$$

Gde $h_M(x)$ predstavlja model doda u M-toj iteraciji, a $\nu \in (0,1]$ korak učenja (*learning rate*) koji kontroliše dopsinos svakog novog stabla.

Značajnost atributa u Gradient Boosting modelima određuje se na osnovu njihovog doprinosa smanjenju funkcije gubitka tokom iterativne izgradnje ansambla. Atributi koji se češće koriste pri podelama i koji dovode do većeg smanjenja greške dobijaju veću značajnost, dok atributi sa manjim doprinosom imaju nižu vrednost značajnosti.

Zbog ovih karakteristika, Gradient Boosting se često koristi u praktičnim zadacima kao snažna embedded metoda koja omogućava istovremeno učenje modela i procenu značajnosti atributa.

5. Izazovi i praktični problemi u izboru atributa

Izbor atributa predstavlja ključan korak u procesu izgradnje modela mašinskog učenja, jer direktno utiče na tačnost, generalizaciju i interpretabilnost modela. Iako filter, wrapper i embedded metode predstavljaju efikasnu redukciju dimenzionalnosti, njihova primena u realnim skupovima podataka često je praćena određenim izazovima. Ovi problemi proizilaze iz strukture podataka, međusobnih odnosa između atributa i pretpostavki na kojima se zasnivaju metode selekcije. U ovom poglavlju razmatraju se najčešći praktični problemi koji se javljaju prilikom izbora atributa i koji su značajni za pravilnu interpretaciju rezultata dobijenih u eksperimentalnom delu rada.

Preklapanje i redundantnost atributa

U realnim skupovima podataka često se javlja situacija u kojoj više atributa nosi slične ili identične informacije. Takvi atributi smatraju se redundantnim, jer njihovo istovremeno uključivanje u model ne doprinosi značajnom povećanju informativnosti, već može povećati složenost i smanjiti interpretabilnost modela. Filter metode, koje razmatraju atribute nezavisno jedan od drugog, često nisu u stanju da prepoznaju redundantnost, zbog čega se u praksi preporučuje kombinovanje različitih pristupa selekcije atributa.

Uticaj skaliranja i normalizacije

Različite metode izbora atributa mogu biti osetljive na numerički opseg podataka, naročito one koje se oslanjaju na varijansu, udaljenosti ili regularizaciju. Atributi sa većim numeričkim opsegom mogu imati neproporcionalno veći uticaj na proces selekcije. Zbog toga je u praksi često neophodno primeniti tehnike skaliranja i normalizacije, kao što su standardizacija ili min-max normalizacija. Neadekvatna priprema podataka može dovesti do pogrešnog rangiranja atributa i nestabilnih rezultata selekcije.

Uticaj ekstremnih vrednosti i šuma u podacima

Prisustvo ekstremnih vrednosti i šuma u podacima predstavlja dodatni izazov u izboru atributa. Ekstremne vrednosti (*outliers*) mogu značajno uticati na statističke mere, poput varijanse ili korelacije, što može dovesti do pogrešne procene relevantnosti atributa. Šum u podacima dodatno otežava selekciju, jer može prikriti stvarne obrasce u podacima, zbog čega je važno obratiti pažnju na kvalitet podataka i primeniti odgovarajuće tehnike za ublažavanje njihovog uticaja.

Problem multikolinearnosti

Multikolinearnost se javlja kada između dva ili više atributa postoji jaka linearna zavisnost. Ovaj problem je naročito izražen kod linearnih modela, gde može dovesti do nestabilnih procena parametara i otežane interpretacije modela. U kontekstu izbora atributa, multikolinearnost može uzrokovati da različite metode biraju različite, ali informativno slične atribute. Iako neki embedded pristupi, poput Lasso regularizacije, mogu delimično ublažiti ovaj problem, multikolinearnost i dalje predstavlja značajan izazov.

Stabilnost izbora atributa

Stabilnost izbora atributa odnosi se na konzistentnost rezultata selekcije pri malim promenama u skupu podataka ili korišćenoj metodi. U praksi se često dešava da različite metode selekcije, ili ista metoda primenjena na različitim uzorcima, rezultuju različitim skupovima odabranih atributa. Ova nestabilnost otežava interpretaciju rezultata, zbog čega se često preporučuje upotreba više metoda selekcije i uporedna analiza njihovih rezultata.

Navedeni izazovi uzeti su u obzir prilikom praktične primene metoda izbora atributa, koja je prikazana u narednom poglavlju.

6. Praktična primena izbora atributa

U ovom poglavlju prikazana je praktična primena metoda izbora atributa na realnom skupu podataka Hotel Booking Reservations, preuzetom sa platforme Kaggle. Skup podataka obuhvata veliki broj instanci i heterogene tipove promenljivih, što ga čini pogodnim za analizu problema izbora relevantnih atributa u zadatku klasifikacije. Ciljna promenljiva odnosi se na ishod rezervacije, odnosno na to da li je rezervacija otkazana ili realizovana.

Zbog kompleksnosti skupa podataka i prisustva različitih tipova atributa, bilo je neophodno sprovesti odgovarajuću predobradu kako bi se obezbedili uslovi za pravilnu primenu metoda selekcije. Posebna pažnja posvećena je uklanjanju nerelevantnih i potencijalno problematičnih atributa, kao i pripremi podataka u skladu sa zahtevima pojedinih metoda. Nakon toga, primenjene su filter, wrapper i embedded metode selekcije atributa, a dobijeni rezultati analizirani su i upoređeni kako bi se sagledale njihove prednosti i ograničenja u praktičnoj primeni.

U cilju evaluacije efekata primenjenih metoda izbora atributa, kao osnovni klasifikacioni model korišćena je logistička regresija.

6.1. Opis i analiza skupa podataka

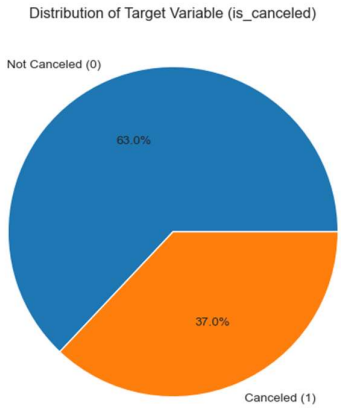
Skup podataka sadrži 119 390 zapisa koji opisuju rezervacije hotela, uključujući informacije o karakteristikama rezervacije, gostima i ishodu rezervacije. Podaci obuhvataju kombinaciju numeričkih i kategorijskih atributa, što omogućava primenu i poređenje različitih metoda selekcije atributa. Ciljna promenljiva u ovom skupu podataka je binarnog tipa i označava da li je rezervacija otkazana, što čini skup pogodnim za analizu značaja atributa u kontekstu klasifikacionih problema.

Na osnovu ciljne promenljive, problem se može formulisati kao binarni problem klasifikacije.

Radi preglednijeg prikaza i lakše interpretacije, atributi su grupisani prema semantičkoj ulozi u procesu rezervacije. Na taj način omogućena je jasnija interpretacija podataka i pravilna primena metoda selekcije atributa.

Grupa atributa	Atributi
Osnovni podaci o tipu hotela i njegovoj lokaciji	hotel, city, country
Atributi vezani za datum i vreme dolaska gosta	arrival_date_year, arrival_date_month, arrival_date_week_number, arrival_date_day_of_month, lead_time
Informacije o dužini boravka gosta	stays_in_weekend_nights, stays_in_week_nights
Podaci o broju i tipu gostiju	adults, children, babies, customer_type, is_repeated_guest
Informacije o dodatnim uslugama i zahtevima	meal, required_car_parking_spaces, total_of_special_requests
Podaci o kanalu i segmentu rezervacije	market_segment, distribution_channel, agent, company
Podaci o prethodnom ponašanju gostiju	previous_cancellations, previous_bookings_not_canceled
Informacije o rezervisanoj i dodeljenoj sobi	reserved_room_type, assigned_room_type
Podaci o ceni i depozitu	adr, deposit_type, booking_changes, days_in_waiting_list
Informacije o ishodu rezervacije	reservation_status, reservation_status_date, is_canceled

Raspodela ciljne promenljive ***is_canceled*** (*Slika 8*) u korišćenom skupu podataka pokazuje da 63% rezervacija nije otkazano (vrednost 0), dok je 37% rezervacija otkazano (vrednost 1). Ovakva raspodela ukazuje na umerenu neravnotežu klasa, pri čemu preovlađuju realizovane rezervacije, ali je i broj otkazanih rezervacija dovoljno značajan da opravda primenu metoda selekcije atributa i modela za klasifikaciju.



Slika 8. Raspodela ciljne promenljive is_canceled

6.2. Predobrada podataka i inženjerstvo atributa

Predobrada podataka predstavlja ključni korak u izgradnji modela mašinskog učenja, jer obezbeđuje da su podaci kvalitetni, relevantni i pogodni za učenje. U ovom procesu se identifikuju i uklanjaju atributi koji uzrokuju curenje podataka, odnosno oni koji sadrže informacije o ciljnoj promenljivoj koje ne bi bile dostupne u realnim uslovima predikcije, kao i identifikacioni atributi koji služe samo za jedinstveno označavanje zapisa i nemaju prediktivnu vrednost. Takođe, uklanjaju se suvišni ili nekorisni atributi koji ne doprinose modelu i mogu povećati šum i složenost. Pored toga, predobrada obuhvata i inženjerstvo atributa, kojim se na osnovu postojećih podataka kreiraju nove, informativnije osobine, čime se poboljšava sposobnost modela da prepozna relevantne obrasce i generalizuje na nove podatke.

Curenje podataka (data leakage)

Atributi ***reservation_status*** i ***reservation_status_date*** direktno otkrivaju ishod rezervacije (Canceled, No-Show, Check-Out), koji je usko povezan sa ciljnom promenljivom ***is_canceled***. Budući da su ove informacije poznate tek nakon završetka procesa rezervacije, njihovo uključivanje u skup podataka dovodi do curenja podataka i veštačkog povećanja performansi modela. Zbog toga je neophodno ukloniti ove atribute pre treniranja modela kako bi se obezbedila realna i pouzdana procena prediktivne moći.

Identifikacioni atributi

Atributi ***agent*** i ***company*** sadrže veliki procenat nedostajućih vrednosti i u najvećoj meri predstavljaju numeričke identifikatore bez stvarnog semantičkog značenja. Kao takvi, oni ne doprinose učenju korisnih obrazaca niti poboljšanju prediktivnih performansi modela, već mogu dovesti do povećanja šuma i složenosti podataka. Zbog navedenih razloga, ovi identifikacioni atributi se uklanjaju iz skupa podataka u fazi predobrade.

Suvišni atributi

Atribut ***city*** ne sadrži nove informacije, budući da je lokacija hotela već implicitno sadržana u atributu ***hotel*** (e.g. "Resort Hotel – Goa"). Usled toga, atribut ***city*** je redundantan i njegovo zadržavanje može dovesti do dupliranja informacija i pojave multikolinearnosti. Iz tog razloga, ovaj atribut se uklanja iz skupa podataka u cilju pojednostavljenja modela.

Nedostajuće vrednosti

Atribut ***country*** sadrži 488 nedostajućih vrednosti, koje su popunjene kategorijom „Unknown", čime je očuvan broj zapisa i izbegnut gubitak podataka. S druge strane, atribut children ima svega 4 zapisa sa nedostajućim vrednostima, koji su uklonjeni iz skupa podataka, budući da u odnosu na više od 100 000 zapisa predstavljaju zanemarljiv deo i ne utiču značajno na ukupnu analizu.

Nedefinisane vrednosti

Atributi ***distribution_channel*** i ***meal*** sadrže nedefinisane vrednosti označene kao „Undefined", koje su zamenjene odgovarajućim i semantički smislenim kategorijama. Konkretno, vrednost „Undefined" kod atributa ***distribution_channel*** zamenjena je sa „TA/TO" (Travel Agency/Tour Operator), dok je kod atributa ***meal*** zamenjena vrednošću „SC" (Self Catering). Na ovaj način obezbeđena je konzistentnost podataka i poboljšana interpretabilnost atributa.

Razdvajanje atributa

Razdvajanje atributa ***hotel*** na zasebne atribute ***hotel_type*** i ***hotel_city*** izvršeno je kako bi se iz jednog složenog tekstualnog atributa izdvojile semantički različite informacije. Na ovaj način su jasno razdvojene informacije o tipu hotela (npr. Resort Hotel ili City Hotel) i njegovoj lokaciji, čime se povećava interpretabilnost podataka i omogućava preciznija analiza uticaja svake od ovih karakteristika na ciljnu promenljivu.

Inženjerstvo atributa

U okviru inženjerstva atributa kreirani su novi, informativniji atributi koji na sažet i smislen način opisuju ponašanje gostiju i karakteristike rezervacije.

Atribut ***total_nights*** predstavlja ukupnu dužinu boravka i dobijen je sabiranjem broja noćenja tokom radnih dana i vikenda, čime je objedinjena informacija koja je prethodno bila razdvojena u dva atributa.

Na sličan način, atribut ***total_people*** formiran je kao zbir broja odraslih, dece i beba, i opisuje ukupan broj gostiju uključenih u rezervaciju.

Pored toga, uveden je binarni atribut ***is_family***, koji označava da li rezervaciju čini porodica (prisustvo dece ili beba), čime se omogućava modelu da lakše uoči razlike u obrascima otkazivanja između porodičnih i neporodičnih rezervacija.

6.3. Enkodiranje kategorijskih atributa

Pošto algoritmi mašinskog učenja uglavnom zahtevaju numeričke ulaze, kategorijski atributi moraju biti adekvatno transformisani u numerički oblik. U ovom radu primenjene su različite tehnike kodiranja u zavisnosti od prirode atributa i njihove semantičke strukture, čime je obezbeđena optimalna reprezentacija podataka i izbegnuto uvođenje veštačkih relacija među kategorijama.

Najpre su identifikovani svi kategorijski atributi tipa object, među kojima se nalaze atributi vezani za vreme dolaska, tip obroka, segment tržišta, kanal distribucije, tip sobe, vrstu depozita, tip korisnika i karakteristike hotela. Na osnovu njihove interpretacije primenjene su tri različite strategije kodiranja: ordinalno kodiranje, one-hot kodiranje i frekvencijsko kodiranje.

Za atribut ***arrival_date_month*** primenjeno je ordinalno kodiranje, budući da meseci imaju prirodan redosled. Svakom mesecu dodeljena je numerička vrednost od 1 do 12, čime je sačuvana hronološka informacija i omogućeno modelu da prepozna sezonalne obrasce u podacima. Ovakav pristup je opravdan jer redosled kategorija nosi značenje, za razliku od nominalnih atributa kod kojih to nije slučaj.

Većina preostalih kategorijskih atributa, kao što su ***meal***, ***market_segment***, ***distribution_channel***, ***reserved_room_type***, ***assigned_room_type***, ***deposit_type***, ***customer_type***, ***hotel_type*** i ***hotel_city***, kodirana je primenom one-hot kodiranja. Ovom tehnikom svaka kategorija se transformiše u zaseban binarni atribut, čime se izbegava uvođenje lažnog redosleda ili numeričke distance između kategorija. Kako bi se smanjila dimenzionalnost i izbegla multikolinearnost, primenjena je opcija drop-first, kojom se jedna referentna kategorija izostavlja.

Za atribut ***country*** korišćeno je frekvencijsko kodiranje, pri čemu je svaka država zamenjena brojem pojavljivanja u skupu podataka. Ovaj pristup je posebno pogodan za atribute sa velikim brojem kategorija, jer značajno smanjuje dimenzionalnost u poređenju sa one-hot kodiranjem, uz zadržavanje informacije o učestalosti pojavljivanja pojedinih kategorija. Na ovaj način omogućeno je modelu da razlikuje češće i ređe vrednosti bez značajnog povećanja broja atributa.

Izbor strategije enkodiranja vršen je u skladu sa prirodom atributa i zahtevima primenjenih modela, čime je obezbeđena uravnotežena kombinacija interpretabilnosti, informativnosti i numeričke stabilnosti.

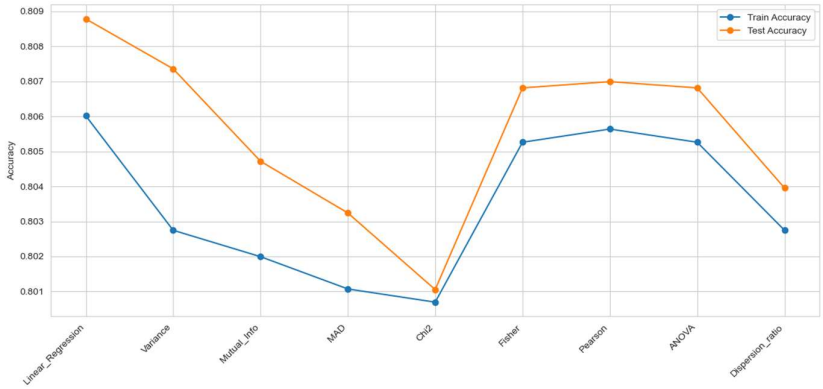
Kombinovanjem različitih tehnika kodiranja, u skladu sa specifičnim karakteristikama pojedinih atributa, obezbeđena je numerički konzistentna, informativna i modelima pogodna reprezentacija podataka. Ovakav pristup predstavlja ključni korak u procesu predobrade, jer direktno utiče na stabilnost, robusnost i performanse primenjenih modela mašinskog učenja. Nakon sprovedene predobrade i dodavanja novih izvedenih atributa, ukupan broj atributa u skupu podataka iznosi 73.

6.4. Primena filter metoda

U ovom poglavlju prikazana je praktična primena filter metoda izbora atributa na analiziranom skupu podataka, sa ciljem procene njihovog uticaja na izbor relevantnih atributa i performanse klasifikacionog modela. Filter metode su primenjene nezavisno jedna od druge, u skladu sa svojim statističkim i informacijskim pretpostavkama, bez direktnog učešća klasifikacionog modela u procesu selekcije. U okviru analize korišćene su sledeće filter metode: prag varijanse (Variance Threshold), uzajamna informacija (Mutual Information), srednja apsolutna devijacija (Mean Absolute Deviation), hi-kvadrat test (Chi-square test), Fišerov rezultat (Fisher's score), Pirsonova korelacija (Pearson's correlation), ANOVA F-test i odnos disperzije (Dispersion Ratio). Ovakav pristup omogućava objektivno poređenje načina na koji različite filter tehnike rangiraju atribute prema njihovoj informativnosti u odnosu na ciljnu promenljivu, kao i sagledavanje razlika u selektivnosti i stepenu redukcije dimenzionalnosti koje pojedine metode ostvaruju.

Poređenje filter metoda

Poređenje performansi modela po metodama izbora atributa pokazuje da se vrednosti tačnosti na test skupu kreću u uskom opsegu oko 80%, nezavisno od primenjene filter metode (*Slika 9*). Nešto više vrednosti tačnosti ostvaruju metode poput Praga Varijanse, Pirsonove korelacije, Fišerovog rezultata i ANOVA F-testa, dok Hi-kvadrat test i srednja apsolutna devijacija (MAD) postižu nešto niže, ali i dalje stabilne rezultate.



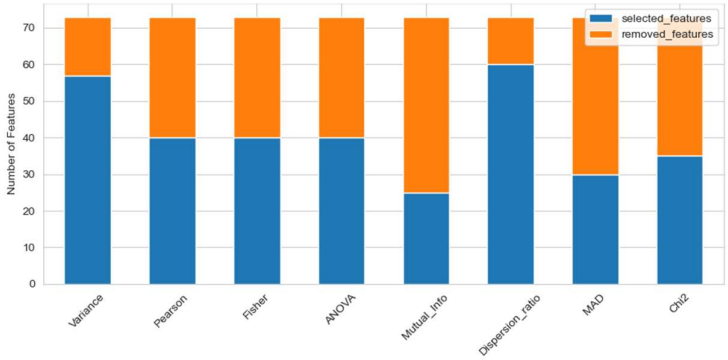
Slika 9. Poređenje tačnosti modela na trening i test skupu za različite filter metode izbora atributa

Tabelarni prikaz (*Tabela 1*) dodatno potvrđuje da razlike između metoda ostaju male, što ukazuje na robusnost modela i stabilnu informativnost odabranih atributa. Upoređivanjem tačnosti na trening i test skupu može se uočiti da razlike između ove dve metrike ostaju male kod svih filter metoda, što ukazuje na odsustvo prenaučenosti i dobru sposobnost generalizacije. Ovakvo ponašanje potvrđuje da izbor filter metode ne uvodi nestabilnost u model, već prvenstveno utiče na izbor i redukciju atributa.

Metoda	Linear Regression	Variance Threshold	Pearson	Fisher	ANOVA	Mutal Info	Dispersion Ratio	MAD	Chi-square
Train_acc	0.806	0.803	0.806	0.805	0.805	0.802	0.803	0.801	0.801
Test_acc	0.809	0.807	0.807	0.807	0.807	0.805	0.804	0.803	0.801

Tabela 1. Poređenje trening i test tačnosti za filter metode izbora atributa.

Analiza broja izabranih i uklonjenih atributa ukazuje na značajne razlike u selektivnosti filter metoda (*Slika 10*). Metode poput odnosa disperzije i praga varijanse zadržavaju veći broj atributa, dok Uzajamna Informacija (Mutual Information) i Hi-kvadrat test pokazuju veću selektivnost i agresivnije uklanjanje atributa. I pored ovih razlika, performanse modela ostaju slične, što ukazuje da je značajan deo informacija sadržan u relativno manjem podskupu atributa.



Slika 10. Broj izabranih i uklonjenih atributa po filter metodi izbora atributa

Na osnovu sprovedenog poređenja može se zaključiti da izbor filter metode ima ograničen uticaj na samu tačnost klasifikacije, ali značajnu ulogu u kontroli dimenzionalnosti i interpretabilnosti modela, čime se filter metode nameću kao efikasan prvi korak u procesu izbora atributa.

6.5. Primena wrapper metoda

U ovom poglavlju prikazana je praktična primena wrapper metoda izbora atributa nad analiziranim skupom podataka. Za razliku od filter metoda, wrapper pristupi procenjuju relevantnost atributa kroz direktnu interakciju sa izabranim klasifikacionim modelom, pri čemu se kvalitet pojedinih podskupova atributa vrednuje na osnovu postignutih performansi modela. Na ovaj način omogućena je detaljnija analiza uticaja izbora atributa na ponašanje modela, ali uz povećanu računarsku složenost.

Svaka wrapper metoda primenjena je nezavisno, a dobijeni rezultati predstavljeni su u zasebnim poglavljima. Na osnovu ostvarenih performansi, broja izabranih atributa i razlika između trening i test skupa, izvršena je analiza prednosti i ograničenja wrapper pristupa u kontekstu praktične primene izbora atributa.

Pre primene wrapper metoda izbora atributa, na inicijalni skup podataka primenjena je metoda **praga varijanse** u cilju eliminacije atributa sa niskom varijabilnošću. Nakon ovog preliminarnog koraka, početni broj atributa smanjen je na **59**, čime je redukovana dimenzionalnost prostora atributa i omogućena efikasnija i računski prihvatljivija primena wrapper metoda, bez značajnog gubitka informacija.

Sekvencijalna selekcija unapred (SFS)

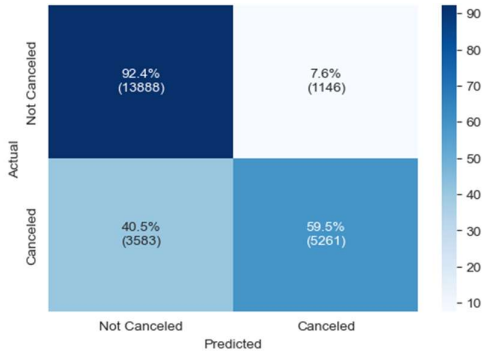
Primenom SFS metode izvršena je dodatna redukcija prethodno definisanog skupa od 59 atributa, pri čemu je izabran podskup od **49** atributa koji ostvaruje optimalne performanse klasifikacionog modela.

Na osnovu prikazanih rezultata (*Tabela 2*) može se uočiti da model, nakon primene SFS metode, postiže ukupnu tačnost od oko **80.2%.** U poređenju sa metodom praga varijanse, nije zabeleženo značajno poboljšanje ukupne tačnosti, ali je ostvaren dodatni stepen redukcije broja atributa. Time je potvrđeno da SFS metoda omogućava efikasniju reprezentaciju podataka uz zadržavanje stabilnih performansi modela.

	preciznost	odziv (recall)	f1-score	broj uzoraka
Not Canceled (0)	0.795	0.924	0.855	15034.000
Canceled (1)	0.821	0.595	0.690	8844.000
Accuracy	0.802	0.802	0.802	0.802
Macro Avg	0.808	0.759	0.772	23878.000
Weighted Avg	0.805	0.802	0.794	23878.000

Tabela 2. Rezultati klasifikacije – Sequential Forward Selection

Konfuziona matrica (*Slika 11*) pokazuje da model zadržava visok odziv za klasu Not Canceled, dok je odziv za klasu Canceled i dalje niži, što je posledica neravnomerne distribucije klasa u skupu podataka.



Slika 11. Matrica konfuzije – Sequential Forward Selection

Sekvencijalna selekcija unazad (SBS)

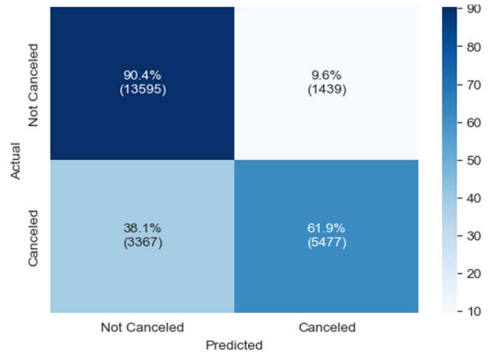
Primenom SBS (Sequential Backward Selection) metode izvršena je dodatna redukcija prethodno definisanog skupa od 59 atributa, pri čemu je iterativnim uklanjanjem najmanje relevantnih atributa izdvojen podskup od svega **7 atributa**, koji ostvaruje stabilne performanse klasifikacionog modela. Na osnovu prikazanih rezultata (*Tabela 3*) može se uočiti da model, nakon primene SBS metode, postiže ukupnu tačnost od približno **79.9%**, što je neznatno niže u poređenju sa SFS metodom, ali uz drastično smanjenje dimenzionalnosti ulaznog prostora.

Dobijeni rezultati ukazuju da SBS metoda omogućava veoma efikasnu redukciju broja atributa uz minimalan gubitak performansi modela. Posebno je značajno to što su trening (**79,9%**) i test (**79,8%**) tačnost gotovo identične, što ukazuje na dobru generalizacionu sposobnost modela i odsustvo izraženog preprilagođavanja. Ovim je potvrđeno da SBS metoda uspešno identifikuje mali, ali informativan skup atributa, čime se postiže povoljan kompromis između složenosti modela i njegove prediktivne moći.

	preciznost	odziv (recall)	f1-score	broj uzoraka
Not Canceled (0)	0.801	0.904	0.850	15034.000
Canceled (1)	0.792	0.619	0.695	8844.000
Accuracy	0.799	0.799	0.799	0.799
Macro Avg	0.797	0.762	0.772	23878.000
Weighted Avg	0.798	0.799	0.792	23878.000

Tabela 3. Rezultati klasifikacije – Sequential Backward Selection

Konfuziona matrica (*Slika 12*) pokazuje da model zadržava visok odziv za klasu Not Canceled, dok je odziv za klasu Canceled i dalje niži, slično kao kod SFS metode.U poređenju sa SFS metodom, SBS ostvaruje nešto bolji odziv za klasu Canceled, što dodatno potvrđuje njegovu sposobnost da zadrži ključne informacije i pored izrazite redukcije broja atributa



Slika 12. Matrica konfuzije – Sequential Backward Selection

Rekurzivna eliminacija atributa (RFE)

Primenom RFE (Recursive Feature Elimination) metode izvršena je selekcija atributa iterativnim uklanjanjem najmanje značajnih atributa na osnovu težina dobijenih iz klasifikacionog modela. Polazeći od inicijalnog skupa od 59 atributa, RFE metodom izdvojen je podskup od **15 atributa** koji obezbeđuje stabilne performanse klasifikacionog modela.

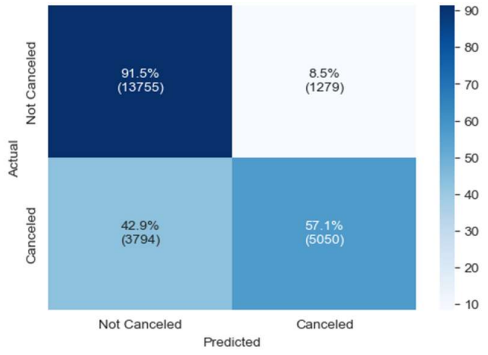
Na osnovu prikazanih rezultata (*Tabela 4*) može se uočiti da model, nakon primene RFE metode, postiže ukupnu tačnost od približno 78,8%, što je nešto niže u poređenju sa SFS i SBS metodama. Ipak, razlika između trening tačnosti (**78,75%**) i test tačnosti (**79,04%**) je minimalna, što ukazuje na dobru generalizacionu sposobnost modela i odsustvo izraženog preprilagođavanja.

Dobijeni rezultati pokazuju da RFE metoda ostvaruje uravnotežen kompromis između smanjenja dimenzionalnosti i očuvanja performansi modela. U poređenju sa SFS metodom, RFE zadržava znatno manji broj atributa, dok u odnosu na SBS metod, RFE obezbeđuje nešto stabilnije metrike za većinsku klasu, ali uz slabiji odziv za klasu Canceled.

	preciznost	odziv (recall)	f1-score	broj uzoraka
Not Canceled (0)	0.784	0.915	0.844	15034.000
Canceled (1)	0.798	0.571	0.666	8844.000
Accuracy	0.788	0.788	0.788	0.788
Macro Avg	0.791	0.743	0.755	23878.000
Weighted Avg	0.789	0.788	0.778	23878.000

Tabela 4. Rezultati klasifikacije – Recursive Feature Elimination

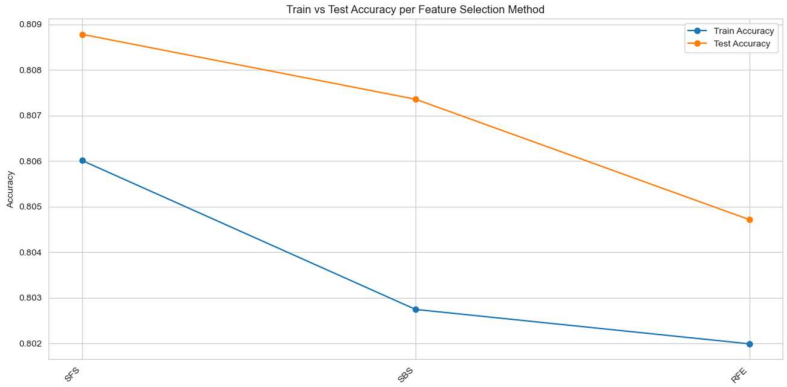
Konfuziona matrica (*Slika 13*) pokazuje da model postiže visok odziv za klasu Not Canceled, dok je odziv za klasu Canceled primetno niži. U poređenju sa SBS metodom, RFE ostvaruje niži odziv za klasu Canceled, ali zadržava stabilniju ukupnu tačnost pri umerenoj redukciji broja atributa.



Slika 13. Matrica konfuzije – Recursive Feature Elimination

Poređenje wrapper metoda

Poređenje performansi modela nakon primene wrapper metoda izbora atributa pokazuje da se vrednosti tačnosti na test skupu kreću u veoma uskom opsegu, između 80,5% i 80,9%, nezavisno od primenjene metode (***Error! Reference source not found.***). Najvišu test tačnost ostvaruje SFS metoda (0,809), dok SBS (0,807) i RFE (0,805) postižu neznatno niže vrednosti. Ovakvi rezultati ukazuju da izbor konkretne wrapper metode ima ograničen uticaj na samu tačnost klasifikacije, ali značajno utiče na dimenzionalnost i složenost modela.



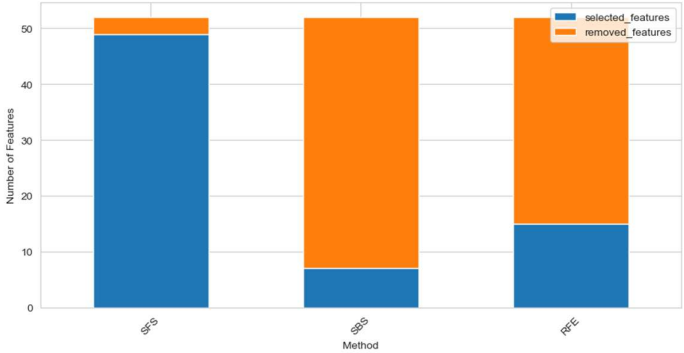
Slika 14. Poređenje tačnosti modela na trening i test skupu za različite wrapper metode izbora atributa

Upoređivanjem trening i test tačnosti može se uočiti da su razlike između ove dve metrike kod svih wrapper metoda minimalne (*Tabela 5*), što ukazuje na odsustvo izraženog preprilagođavanja i dobru generalizacionu sposobnost modela. SFS metoda pokazuje blago više vrednosti obe metrike, dok SBS i RFE ostvaruju nešto niže, ali stabilne rezultate, čime se potvrđuje da wrapper metode ne narušavaju robusnost modela uprkos različitim strategijama selekcije atributa.

Metoda	SFS	SBS	RFE
Train_acc	0.806	0.803	0.802
Test_acc	0.809	0.807	0.805

Tabela 5. Poređenje trening i test tačnosti za wrapper metode izbora atributa

Analiza broja izabranih i uklonjenih atributa (*Slika 15*) ukazuje na značajne razlike u selektivnosti wrapper metoda. SFS metoda zadržava najveći broj atributa (49), što rezultuje najvišom tačnošću, ali i većom složenošću modela. Nasuprot tome, SBS metoda ostvaruje najagresivniju redukciju, zadržavajući svega 7 atributa, uz minimalan pad performansi, dok RFE metoda predstavlja kompromisno rešenje sa 15 izabranih atributa, balansirajući između tačnosti i smanjenja dimenzionalnosti.



Slika 15. Broj izabranih i uklonjenih atributa po wrapper metodi izbora atributa

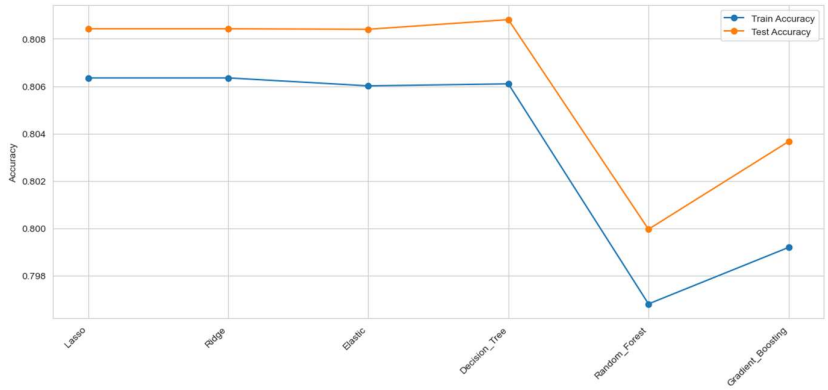
Na osnovu sprovedenog poređenja može se zaključiti da wrapper metode imaju ograničen uticaj na apsolutne vrednosti tačnosti, ali značajno utiču na složenost, interpretabilnost i efikasnost modela. SFS metoda je pogodna u situacijama gde je cilj maksimizacija performansi, SBS metoda je naročito korisna kada je prioritet snažna redukcija atributa uz očuvanje stabilnosti, dok RFE predstavlja uravnotežen pristup koji kombinuje umerenu redukciju i stabilne performanse. Time se wrapper metode potvrđuju kao ključan korak u procesu optimizacije modela i izbora atributa.

6.6. Primena embedded metoda

U ovom poglavlju prikazana je praktična primena embedded metoda izbora atributa na analiziranom skupu podataka, sa ciljem procene njihovog uticaja na izbor relevantnih atributa i performanse klasifikacionog modela. Embedded metode integrišu proces selekcije atributa direktno u fazu treniranja modela, koristeći unutrašnje mehanizme algoritama učenja, kao što su regularizacija i struktura stabla odlučivanja. U okviru analize korišćene su sledeće embedded metode: Lasso (L1 regularizacija), Ridge regresija (L2 regularizacija), Elastic Net, stabla odlučivanja (Decision Trees), Random Forest i Gradient Boosting. Ovakav pristup omogućava istovremenu optimizaciju performansi modela i redukciju dimenzionalnosti, kao i sagledavanje razlika u načinu na koji različiti embedded pristupi procenjuju značajnost atributa u kontekstu klasifikacionog zadatka.

Poređenje embedded metoda

Poređenje performansi klasifikacionog modela nakon primene embedded metoda izbora atributa pokazuje da se vrednosti tačnosti na test skupu kreću u relativno uskom opsegu, između 80,0% i 80,9%, nezavisno od primenjene metode (*Slika 16*). Najvišu test tačnost ostvaruje Decision Tree (0,809), dok Lasso, Ridge regresija i Elastic Net postižu gotovo identične vrednosti test tačnosti (0,808), što ukazuje na sličan efekat regularizacije ovih metoda u kontekstu analiziranog skupa podataka. Gradient Boosting i Random Forest ostvaruju nešto niže vrednosti test tačnosti, ali i dalje zadržavaju stabilne performanse.



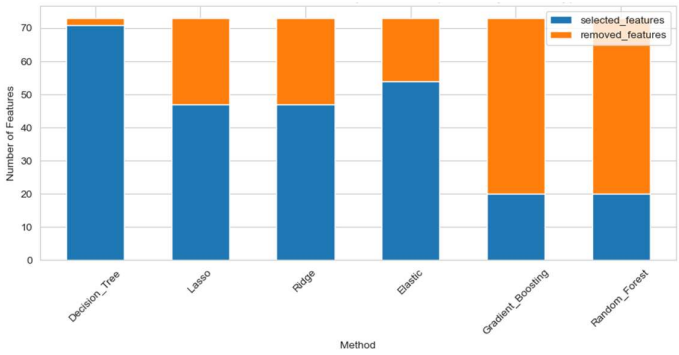
Slika 16. Poređenje tačnosti modela na trening i test skupu za različite embedded metode izbora atributa

Upoređivanjem trening i test tačnosti može se uočiti da su razlike između ove dve metrike kod većine embedded metoda minimalne (*Tabela 6*), što ukazuje na dobru generalizacionu sposobnost modela i odsustvo izraženog preprilagođavanja. Kod metoda zasnovanih na regularizaciji (Lasso, Ridge i Elastic Net) uočava se gotovo identično ponašanje na trening i test skupu, dok metode zasnovane na ansamblima stabala pokazuju nešto izraženije razlike, ali bez značajnog negativnog uticaja na stabilnost modela.

Metoda	Decision Tree	Lasso	Ridge	Elastic Net	Gradient Boosting	Random Forest
Train_acc	0.806	0.806	0.806	0.806	0.799	0.797
Test_acc	0.809	0.808	0.808	0.808	0.804	0.800

Tabela 6. Poređenje trening i test tačnosti za embedded metode izbora atributa

Analiza broja izabranih i uklonjenih atributa (*Slika 17*) ukazuje na jasne razlike u selektivnosti embedded metoda. Decision Tree metoda zadržava najveći broj atributa, što doprinosi višoj tačnosti, ali i većoj složenosti modela. Nasuprot tome, metode zasnovane na regularizaciji, posebno Lasso, ostvaruju umereniju redukciju atributa, dok Gradient Boosting i Random Forest vrše znatno agresivniju selekciju, zadržavajući manji broj atributa uz prihvatljiv pad performansi. Ovi rezultati ukazuju na izražen kompromis između tačnosti i redukcije dimenzionalnosti u okviru embedded pristupa.



Slika 17. Broj izabranih i uklonjenih atributa po embedded metodi izbora atributa

Na osnovu sprovedenog poređenja može se zaključiti da embedded metode, slično filter i wrapper metodama, imaju ograničen uticaj na apsolutne vrednosti tačnosti, ali značajno utiču na složenost i interpretabilnost modela. Metode zasnovane na regularizaciji pokazuju stabilne i konzistentne rezultate, dok ansambl metode omogućavaju snažniju redukciju atributa uz blago smanjenje tačnosti. Time se embedded metode potvrđuju kao efikasan kompromis između performansi modela i kontrole dimenzionalnosti ulaznog prostora.

7. Zaključak

U ovom seminarskom radu analiziran je problem izbora atributa kao jedan od ključnih koraka u procesu izgradnje modela mašinskog učenja. Kroz teorijski i praktični deo rada prikazane su i upoređene tri osnovne grupe metoda za selekciju atributa: filter, wrapper i embedded pristupi. Poseban akcenat stavljen je na razumevanje principa rada svake metode, njihovih prednosti i ograničenja, kao i na njihovu primenu u realnom scenariju klasifikacije.

Praktična analiza sprovedena je nad skupom podataka Hotel Booking Reservations, koji zbog svoje veličine, heterogenosti atributa i realnog poslovnog konteksta predstavlja pogodan primer za ispitivanje efekata izbora atributa. Nakon detaljne predobrade podataka, uklanjanja izvora curenja informacija i primene inženjerstva atributa, formiran je kvalitetan i stabilan skup sa 73 atributa, pogodan za dalju analizu. Kao osnovni klasifikacioni model korišćena je logistička regresija, čime je omogućeno jasno sagledavanje uticaja izbora atributa na performanse modela.

Rezultati filter metoda pokazali su da različite statističke i informacione mere dovode do sličnih vrednosti tačnosti modela, uz značajne razlike u broju zadržanih atributa. Ovakvi nalazi potvrđuju da filter metode imaju ograničen uticaj na samu tačnost klasifikacije, ali značajnu ulogu u kontroli dimenzionalnosti i pojednostavljenju modela, zbog čega su pogodne kao inicijalni korak u procesu selekcije atributa.

Wrapper metode su omogućile dublju analizu međuzavisnosti atributa i njihovu direktnu optimizaciju u odnosu na performanse modela. Posebno se ističe metoda sekvencijalne selekcije unazad (SBS), koja je uspela da drastično smanji broj atributa uz minimalan gubitak tačnosti, čime je postignut povoljan kompromis između složenosti modela i njegove prediktivne moći. Ovi rezultati potvrđuju da wrapper metode mogu značajno doprineti redukciji dimenzionalnosti, ali uz povećanu računarsku cenu.

Embedded metode, poput Lasso, Ridge, Elastic Net i ansambl metoda zasnovanih na stablima odlučivanja, pokazale su se kao efikasan kompromis između filter i wrapper pristupa. Integracijom selekcije atributa direktno u proces treniranja modela, omogućena je stabilna i efikasna identifikacija relevantnih atributa, uz očuvanje dobre generalizacije modela.

Na osnovu sprovedene analize može se zaključiti da ne postoji univerzalno najbolja metoda izbora atributa, već da izbor odgovarajućeg pristupa zavisi od karakteristika skupa podataka, složenosti problema i ciljeva analize. U praksi, kombinovanje filter metoda za inicijalno smanjenje dimenzionalnosti sa wrapper ili embedded pristupima za finu selekciju predstavlja najefikasniju strategiju. Ovim radom potvrđena je značajna uloga izbora atributa u unapređenju efikasnosti, interpretabilnosti i stabilnosti modela mašinskog učenja.

8. Literatura

[1] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*. Springer, 2015.

[2] *Feature Selection Techniques in Machine Learning,* GeeksforGeeks. *Link: https://www.geeksforgeeks.org/machine-learning/feature-selection-techniques-in-machine-learning/.* Pristupljeno: 15.12.2025.

[3] *Feature selection*, Wikipedia - The Free Encyclopedia. Link: https://en.wikipedia.org/wiki/Feature_selection. *Pristupljeno*: 19.12.2025.

[4] M. A. Haque, *Feature Engineering & Selection for Explainable Models: A Second Course for Data Scientists*. LULU Press, 2022.

[5] *Feature Selection (PDF)* Link:http://eio.usc.es/pub/mjginzo/descargas/leyenda/Desktop/master/Master%201/Analisis%20exploratorio%20de%20datos/seleccion%20de%20variables/Relieve-D48003C0d01.pdf Pristupljeno: 16.12.2025.

[6] S. Buddacharya, *A Comprehensive Guide to Feature Selection in Machine Learning*, Medium. Link:https://medium.com/@sangambuddhacharya/a-comprehensive-guide-to-feature-selection-in-machine-learning-72ff5c0607f1. Pristupljeno: 16.12.2025.

[7] *Variance Threshold*. Link: Variance Threshold - GeeksforGeeks. Pristupljeno: 16.12.2025.

[8] *Chi-Square Test*. Link: Chi-Square Test - GeeksforGeeks. Pristupljeno: 17.12.2025.

[9] *Feature Selection using F-Anova*. Link: Feature Selection using F-Anova - GeeksforGeeks Pristupljeno: 17.12.2025.

[10] *Random Forest Regression in Python*. Link: Random Forest Regression in Python - GeeksforGeeks Pristupljeno: 18.12.2025.

[11] Difference Between Random Forest and Decision Tree. Link: Difference Between Random Forest and Decision Tree - GeeksforGeeks Pristupljeno: 18.12.2025.