**Elektronski Fakultet**

**Univerzitet u Nišu**

****

**Izbor atributa**

**- Seminarski rad -**

**Predmet:**

**Prikupljanje i predobrana podataka**

**Student: Mentor:**

**Dimitrije Petrović (1682) Prof. dr Aleksandar Stanimirović**

**Niš, decembar 2024. Godine**

**Sadržaj**

[**1.** **Uvod** 2](#_Toc178452541)

[**2.** **Filter metode** 4](#_Toc178452542)

[**2.1 Numerički ulaz, numerički izlaz** 5](#_Toc178452543)

[**2.1.1 Pirsonov koeficijent korelacije** 5](#_Toc178452544)

[**2.1.2 Spirmanova rang korelacija** 6](#_Toc178452545)

[**2.1.3 Prag Varijanse** 7](#_Toc178452546)

[**2.2 Numerički ulaz, kategorički izlaz i Kategoricki ulaz, numerički izlaz** 7](#_Toc178452547)

[**2.2.1 ANOVA** 7](#_Toc178452548)

[**2.2.2 Kendalov koeficijent korelacije rangova** 8](#_Toc178452549)

[**2.3 Kategorički ulaz, kategorički izlaz** 8](#_Toc178452550)

[**2.3.1 Hi-kvadrat test** 9](#_Toc178452551)

[**2.3.2 Zajednička informacija (Mutual Infromation - MI)** 9](#_Toc178452552)

[**3.** **Wrapper metode (metode omotača)** 10](#_Toc178452553)

[**3.1 Pretraga unapred** 11](#_Toc178452554)

[**3.2 Pretraga unazad** 12](#_Toc178452555)

[**3.3 Bidirekciona pretraga** 12](#_Toc178452556)

[**3.4 Kompletna pretraga** 13](#_Toc178452557)

[**4.** **Embedded (ugrađene) metode** 13](#_Toc178452558)

[**4.1 Stabla odlučivanja (Decision Trees)** 14](#_Toc178452559)

[**4.2 Slučajne šume (Random Forest)** 14](#_Toc178452560)

[**4.3 LASSO** 14](#_Toc178452561)

[**5.** **Metaheuristički algoritmi**[2] 15](#_Toc178452562)

[**5.1 Genetski algoritam** 15](#_Toc178452563)

[**5.2 Simulirano kaljenje** 15](#_Toc178452564)

[**6.** **Zaključak** 16](#_Toc178452565)

[**7.** **Literatura** 17](#_Toc178452566)

# **Uvod**

Izbor atributa se bavi odabirom podskupa početnog skupa atributa. Ovde je bitno napomenuti da je krajnji rezultat veći od sume svojih delova – ne biramo samo atribute koji su značajni, već koji će zajedno činiti skup koji ima najbolju moć predviđanja. Ciljevi ove metode mogu biti različiti[1] :

* Uklanjanje šuma, posebno u slučaju da imamo veliki broj atributa
* Redukcija vremena izvršavanja algoritma
* Smanjenje kompleksnosti sistema, tj. lakše razumevanje
* Poboljšanje rezultata

Najčešća podela ovog pristupa je na tri metode: filter, wrapper (vraper) i embeded (ugrađene) metode, pored ovih se takođe nekad javlja i četvrta metoda – metaheuristički algoritmi[2]

Pored toga, još jedna podela je na u zavisnosti od toga da li se koristi ciljni atribut pri izboru atributa – tako da imamo i podelu na nadgledane i nenadgledane metode.

U nastavku će biti obrađene ove četiri metode nad problemima klasifikacije i regresije.

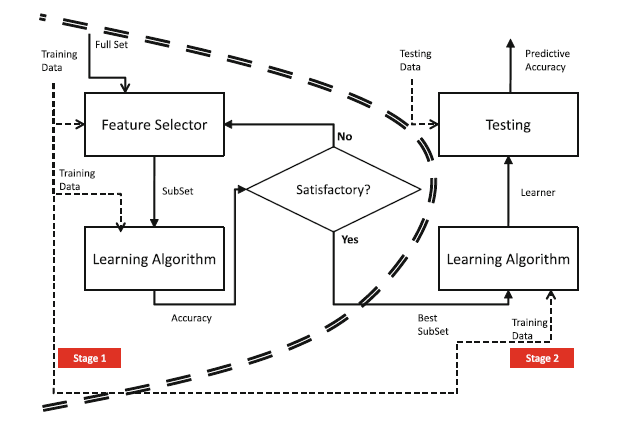
# **Filter metode**

Sam naziv potiče od činjenice da ove metode “filtriraju” suvišne i neželjene atribute pre samog procesa učenja[1] . Ove metode koriste osobine samih podataka, tj. upotrebom tehnika iz statistike kako bi dobili neku metriku (udaljenost, zavisnost, konzistentnost…) između ulaznih atributa i ciljnog atributa (klasa).

Osobine ovih metoda:

* Ne zavise od samog algoritma koji planiramo da upotrebimo za učenje
* Mogu da se koristi za različite modele i različite tehnike
* Nisu vremenski zahtevne
* Mogu da se koristi na velikim skupovima podataka
* Može se desiti da ostave neki atribut koji nije od značaja za naš model
* Mogu da se koriste sa drugim tehnikama za izbor atributa, uglavnom se koristi kao prvi korak u ovom procesu
* Svaki atribut se gleda zasebno
* Uglavnom proverava linearne zavisnosti izmedju ulaznih atributa i ciljnog atributa

Na slici 1. su predstavljeni koraci pri korišćenju ovih metoda. Prvi korak čini odabir atributa na osnovu neke metrike, ne uzimajući u obzir sam algoritam koji planiramo da korisitmo. Drugi korak predstavlja obučavanje i testiranje upotrebom skupa atributa koji smo dobili posle prvog koraka.[1]



*Slika 1. Filter model*[1]

Ove metode možemo podeliti na četiri vrste [3]:

* Numerički ulaz, numerički izlaz
* Numerički ulaz, kategorički izlaz
* Kategorički ulaz, numerički izlaz
* Kategorički ulaz, kategorički izlaz

## **2.1 Numerički ulaz, numerički izlaz**

Ovaj problem se svodi na regresiju, ulazni atributi i ciljni atribut su neprekidni. Ovde se najčešće koristi neka metrika koja izražava korelaciju. Samu korelaciju možemo posmatrati između ulaznog i izlaznog atributa, ili izmedju dva ulazna atributa.

Ako posmatramo ulazne attribute, ako postoji velika korelacija između dva atributa, možemo smatrati da je jedan od njih tj. imamo dupliciranje informacija, zbog čega možemo izbaciti drugi atribut. Ovo može biti problem kod nekih modela kao što su linearna ili logistička regresija, ali kod SVM ovo ne predstavlja veliki problem, moguće je i da sam SVM radi bolje ako ostavimo te atribute; takođe možemo koristiti nelinearna jezgra/kernele.

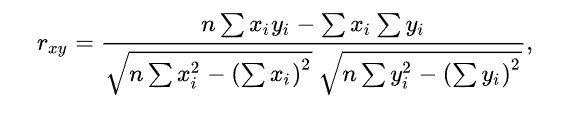
U nastavku će biti obrađene tri metode ovog tipa:

* Pirsonov koeficijent korelacije
* Spirmanova rang korelacija
* Prag varijanse

## **2.1.1 Pirsonov koeficijent korelacije**

Pirsonov koeficijent korelacije nam govori da li postoji neka linearna korelacija između dve neprekidne promenljive, i kreće se od -1 do 1. Vrednosti -1 i 1 znače da linearna jednačina savršeno predstavlja odnos izmedju dve promenljive, gde znak zavisi od nagiba, +1 znači da ako jedna promenljiva raste onda raste i druga, a -1 obrnuto. Vrednost 0 znači da ne postoji linearna zavisnost između promenljivih.[4]

Da bi koristili ovu metriku potrebno je da nemamo nedostajuće podatke, kao i da nemamo autlajere.

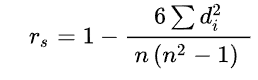


*Slika 2. Pirsonov koeficijent korelacije, formula*[4]

## **2.1.2 Spirmanova rang korelacija**

Koristi se kada se ne vodimo pretpostavkom da varijable imaju linearnu zavisnost. Ovde je takođe bitno napomenuti da veza izmedju varijabli mora biti monotona. Vrednost se takođe kreće od -1 do +1, predznak plus znači da postoji povezanost većih rangova u obe varijable, dok predznak minus znači da postoji relacija između manjih rangova. Vrednost 0 kao i vrednosti oko nule znači da postoji veoma slaba korelacija.[5]

Ova korelacija je za razliku od pirsonovog koeficijenta korelacije otporna na autlajere. Pošto ovaj metod koristi rangove a ne same vrednosti moguć je gubitak informacija o relaciji između samih promenljivih.



*Slika 3. Spirmanova rang korelacija*[6]

Na primeru X = [11,2,31,8,19], Y = [10, 9, 8, 1, 3]

|  |  |
| --- | --- |
| X | Rank |
| 11 (X0) | 3 |
| 2 (X1) | 5 |
| 31 (X2) | 1 |
| 8 (X3) | 4 |
| 19 (X4) | 2 |
| Y | Rank |
| 10 (Y0) | 1 |
| 9 (Y1) | 2 |
| 8 (Y2) | 3 |
| 3 (Y3) | 5 |
| 1 (Y4) | 4 |

di = Rank(Xi) – Rank(Yi) tako da imamo: d0 = 2, d1 = 3, d2 = -2, d3 = -1, d4 = -2

6 \* sum(di^2) = 6\* (4 + 9 + 4 + 1 + 4) = 6 \* 22 = 132

n \* (n^2 - 1) = 5 \* 24 = 120

rs = 1 – 132/120 = 1 – 1.1 = -0.1, što ukazuje na veoma slabu monotonu vezu izmedju promenljivih, tj. možemo zaključiti da ne postoji monotona veza između ove dve promenljive

## **2.1.3 Prag Varijanse**

Ovom metodom odbacujemo sve atribute cija varijansa ne prelazi odredjeni prag – ideja je da ovi atributi nisu od velike koristi kao atributi sa većom varijansom. Prag (graničnu vrednost) sami definišemo.

Ova metode ne uzima u obzir klasu i može imati problema sa nelinearnim relacijama, ali je brza i efikasna. Ovu metodu ne možemo primeniti ako su podaci standardizovani, ako nisu iste veličine - u istim jedinicama.



*Slika4. formula*[7]

U formuli iznad **x** je naš atribut za koji računamo varijansu, **xi** je individualna vrednost atributa, **μ** je srednja vrednost, **n** je broj vrednosti atributa.

## **2.2 Numerički ulaz, kategorički izlaz i Kategoricki ulaz, numerički izlaz**

Kod problema sa numeričkim ulazom, kategoričkim izlazom se radi o klasifikaciji, tako da moramo da uzmemo u obzir i same klase ciljane vrednosti. Slično kao i kod problema sa numeričkim ulazom i numeričkim izlazom, većina metoda ovog tipa se zasnivaju na korelaciji između atributa.

Neke od metoda za ovaj tip problema su ANOVA i Kendalov koeficijent korelacije rangova.

Kategoricki ulaz, numerički izlaz je veoma specifičan primer regresije koji se veoma retko dešava. Ovde takođe mogu da se primene ANOVA I Kendalov koeficijent korelacije rangova.

## **2.2.1 ANOVA**

ANOVA (Analysis of variance – analiza varijanse) se koristi da odredimo da li postoji neka razlika između aritmetičkih sredina dve ili više grupe. Kod izbora atributa je možemo koristiti za probleme klasifikacije. Cilj je odrediti koji atributi imaju najviše uticaja u pronalaženju različitih klasa našeg ciljnog kategoričkog atributa. Uslov za ovu analizu je da su atributi nezavisni, kao i da naši ulazni atributi imaju gausovu raspodelu za svaku klasu naseg ciljanog atributa.

Za svaki atribut izračunava se varijansa izmedju srednjih vrednosti tog atributa za svaku od kategorija naseg ciljanog atributa. Nakon toga gledamo da li je varijansa između grupa/kategorija (faktorska varijansa) znatno veće od varijanse unutar svake grupe/kategorije (rezidualna varijansa). Ako je ova tvrdnja tačna onda možemo izabrati ovaj atribut za naš skup atributa od značaja. Ova analiza je jednostavna i brza, može da uoči linearne zavisnosti.

## **2.2.2 Kendalov koeficijent korelacije rangova**

Kao i Spirmanov koeficijent, i ovde se za izračunavanje koeficijenta korelacije koristi rangovi. Ova metoda se može iskoristiti kada imamo ordinalne podatke kod kojih ne postoji linearna relacija, dobro se nosi sa autlajerima ali je njegovo izračunavanje kompleksno pa nije povoljan za velike skupove podataka. Jedna od prednosti ovog koeficijenta u odnosu na Spirmanov je to što se bolje nosi sa slučajevima kada imamo veliki broj istih vrednosti, tj. istih rangova.

Kod ovog koeficijenta se uvode pojmovi saglasnih parova i nesaglasnih parova. Parovi (xi, yi) i (xj, yj) su saglasni ako važi xi>xj i yi>yj, ili xi<xj i yi<yj. Ovaj uslov znači da je red rangova između parova konzistentan, tj saglasan. Parovi su nesaglasni ako važi xi>xj i yi<yj, ili xi<xj i yi>yj. Tada imamo formulu:

S = broj saglasnih parova, N = broj nesaglasnih parova, n = broj vrednosti atributa

Ako ponovo uzmemo primer X = [11,2,31,8,19], Y = [10, 9, 8, 1, 3]

Poređenje parova:

|  |  |
| --- | --- |
| (11, 10) i (2, 9) | Saglasni |
| (11, 10) i (31, 8) | Nesaglasni |
| (11, 10) i (8, 1) | Saglasni |
| (11, 10) i (19, 3) | Nesaglasni |
| (2, 9) i (31, 8) | Nesaglasni |
| (2, 9) i (8, 1) | Nesaglasni |
| (2,9) i (19, 3) | Nesaglasni |
| (31, 10) i (8, 1) | Saglasni |
| (31, 10) i (19, 3) | Saglasni |
| (8, 1) i (19, 3) | Saglasni |

S = 5, N = 5, n = 5, kada ovo ubacimo u formulu imamo

## **2.3 Kategorički ulaz, kategorički izlaz**

Ovo je još jedan primer klasifikacije, ovde će biti prikazan metod Hi-kvadrat test.

## **2.3.1 Hi-kvadrat test**

Ovaj test se koristi da proverimo da li postoji povezanost između dva kategorička atributa. Ovaj test se takođe može primeniti i na numeričke atribute ako izvršimo diskretizaciju. Hi-kvadrat test proverava da li postoji primetna razlika između raspodele nekog (ulaznog) atributa po kategorijama ciljanog atributa – na taj način utvrđujemo da je taj atribut dobar u razlikovanju kategorija ciljanog atributa pa ga možemo iskoristiti za naš model.

Pogodnosti ovog testa su njegova jednostavnost i skaliranje, ali tretira promenljive kao da su zasebne, pored toga je potreban veći broj ulaznih podataka.

## **2.3.2 Zajednička informacija (Mutual Infromation - MI)**

Ova vrednost pokazuje koliko se neodređenost jednog atributa smanjuje ako poznajemo vrednost drugog atributa – tj. dobitak infromacije za dva atributa ako nam je jedan poznat. Ova vrednost se uglavnom normalizuje tako da uzima opseg od 0 do 1, pri čemu vrednosti blizu jedinice znače da možemo da koristimo jedan od ta dva atributa za naš model (pošto je ova vrednost simetrična, možemo uzeti bilo koji). Zajednička informacija se može koristiti za diskretne i ordinalne vrednosti, ali može da se primeni i za ostale, pogotovo je korisna ako postoji nelinearna veza između atributa.

Ovu vrednost možemo definisati na sledeći način[8]:

Ako imamo dve promenljive **X** i **Y**, posmatramo njihove parove **(X,Y)** gde **X** ima **m** različitih vrednosti (x1, x2, …, xm), a **Y** ima **n** (y1, y2, …, yn)

Tada je zajednička informacija za ove dve promenljive:

Ovo možemo pokazati na primeru x=(−3, −2, −1, 0, 1, 2, 3) i y=(9, 4, 1, 0, 1, 4, 9)[8]

|  |  |  |
| --- | --- | --- |
| xj | P(x=xj) | −P(x=xj)log2 P(x=xj) |
| -3 | 1/7 | 0.4011 |
| -2 | 1/7 | 0.4011 |
| -1 | 1/7 | 0.4011 |
| 0 | 1/7 | 0.4011 |
| 1 | 1/7 | 0.4011 |
| 2 | 1/7 | 0.4011 |
| 3 | 1/7 | 0.4011 |
| H(x) | | 2.8074 |

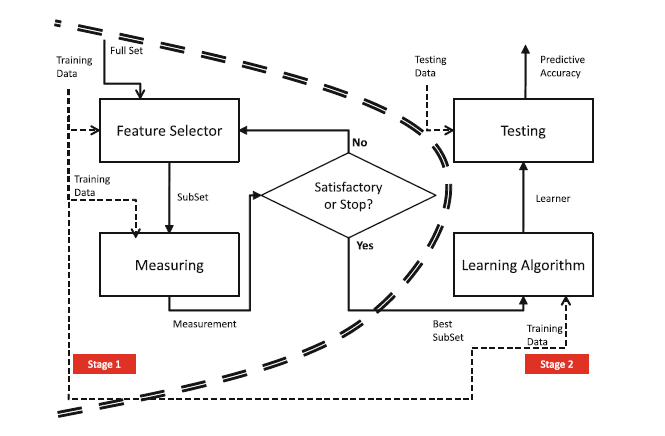
|  |  |  |
| --- | --- | --- |
| yk | P(y=yk) | −P(y=yk)log2 P(y=yk) |
| 9 | 2/7 | 0.5164 |
| 4 | 2/7 | 0.5164 |
| 1 | 2/7 | 0.5164 |
| 0 | 2/7 | 0.5164 |
| H(y) | | 1.9502 |

|  |  |  |  |
| --- | --- | --- | --- |
| xj | yk | P(x=xj, y=yk) | −P(x=xj, y=yk)log2 P(x=xj, y=yk) |
| -3 | 9 | 1/7 | 0.4011 |
| -2 | 4 | 1/7 | 0.4011 |
| -1 | 1 | 1/7 | 0.4011 |
| 0 | 0 | 1/7 | 0.4011 |
| 1 | 1 | 1/7 | 0.4011 |
| 2 | 4 | 1/7 | 0.4011 |
| 3 | 9 | 1/7 | 0.4011 |
| H(x, y) | | | 2.8074 |

I(x, y)=H(x)+H(y)−H(x, y)=1.9502, ako normalizujemo sa log2(min(m, n)) onda imamo I(x, y)=0.9751

# **Wrapper metode (metode omotača)**

Kod ovih metoda za procenu koristimo sam algoritam učenja/klasifikator da bi odlučili šta da radimo sa određenim atributom. Ovde se takođe koristi i tehnika kros validacije da bi izbegli overfitting. Ove metode zavise od samog klasifikatora koji koristimo, tako da mogu u potpunosti da iskoriste njegove osobine. Ako imamo veliki skup podataka ovaj pristup može da bude nepraktičan.



*Slika 5. Wrapper model* [1]

Na slici 5. je prikazan wrapper model. Ovde možemo da razlikujemo dva koraka – prvi u kome se bira najbolji podskup atributa koristeći preciznost samog alogirma na skupu podataka za treniranje. Drugi korak predstavlja obučavanje i testiranje upotrebom skupa atributa koji smo dobili posle prvog koraka. [1]

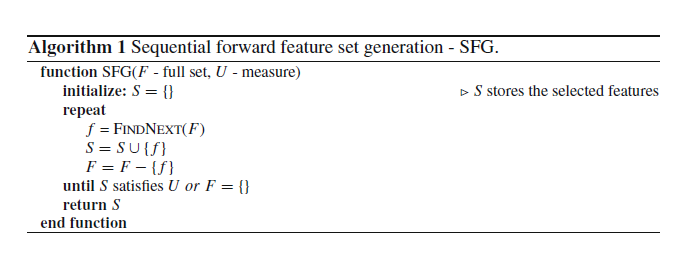
Ovde će biti obrađene sledeće metode:

* (sekvencijalna) pretraga unapred
* (sekvencijalna) pretraga unazad
* Biderekciona pretraga
* Potpuna pretraga

## **3.1 Pretraga unapred**

Kod ove tehnike počinjemo sa praznim skupom ulaznih atributa i u svakom koraku se dodaje jedan novi atribut, izabran na osnovu nekog kriterijuma (npr. preciznost), tj. dodaje se ako poboljšava prediktivnu moć algoritma mašinskog učenja nad našim skupom podataka. Za sam kraj algoritma možemo izabrati određeni broj atributa, tj izbor *K najboljih atributa* ili jednostavno dok algoritam ne prođe kroz ceo skup podataka.

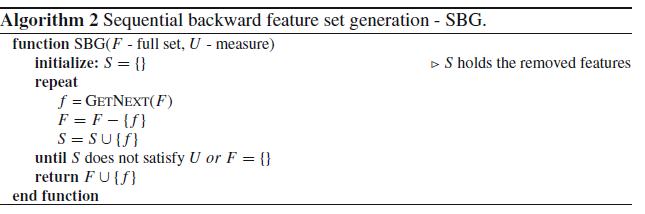
Ovaj algoritam spada u pohlepne algoritme zato što se može zaustaviti pre nego što dođemo do optimalnog skupa atributa.



*Slika 6. Pretraga unapred, pseudokod*[1]

## **3.2 Pretraga unazad**

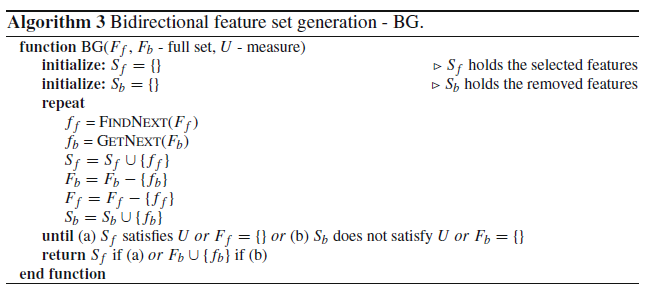
Nasuprot prethodnom pristupu, ovde počinjemo od celog skupa ulaznih atributa i svakim prolazom kroz algoritam nalazimo atribut koji ima najmanji uticaj i izbacujemo ga. Ovaj algoritam takođe spada u pohlepne algoritme.



*Slika 7. Pretraga unazad, pseudokod* [1]

## **3.3 Bidirekciona pretraga**

Ovde se koriste obe prethodne tehnike, tj. krećemo se u oba pravca istovremeno. Ovde se smanjuje mogućnost da naš konačni skup bude neoptimalan.



*Slika 8. Bidirekciona pretraga, pseudokod* [1]

## **3.4 Kompletna pretraga**

Ovde je cilj pretražiti ceo skup atributa kao i sve moguće kombinacije atributa, tako da garantuje optimalno rešenje. Međutim, kako je jedan od ciljeva izbora atributa da se smanji vreme traženja rešenja, kao i činjenica da je ovaj pristup nepraktičan za velike skupove podataka, ovaj pristup je retko upotrebljiv na realnim skupovima podataka.

# **Embedded (ugrađene) metode**

Kod ovih metoda odabir atributa radi sam algoritam za mašinsko učenje u toku svojeg obučavanja i zavisi od samog algoritma. Ono što je zajedničko za ove algoritme je da gledaju veze između atributa kao i između ciljnog atributa.

Primeri ovih metoda su: decision trees (stabla odlučivanja), random forest (slučajne šume), bagging i boosting algoritmi (XGBoost), tehnike regularizacije (LASSO, Ridge, ElasticNet)

Nadalje će biti predstavljeni:

* Stabla odlučivanja
* Slučajne šume
* LASSO

## **4.1 Stabla odlučivanja (Decision Trees)**

Stabla odlučivanja su modeli koji dele skup podataka u odnosu na neki atribut, tj. u zavisnosti od vrednosti tog atributa. Ovi modeli mogu da služe i za probleme regresije i klasifikacije.

Samo stablo se generiše od korena ka listovima nekim pohlepnim algoritmom – za grananje se gleda samo trenutno stanje, ne gleda se kako će se grananje izvršiti u narednom koraku. Ovaj postupak izbora “najbolje” podele se nastavlja sve dok ne dođemo do našeg kriterijuma zaustavljanja. Čvor odgovara ulaznom atributu, grana predstavlja neku vrednosti tog atributa a listovi su izlazni atributi. Na kraju se dobija potpuno razgranato stablo.

Ovaj algoritam može dovesti do overfitting-a, zbog čega se često radi odsecanje nekih čvorova.

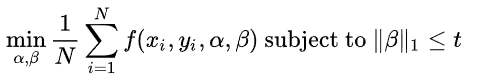
## **4.2 Slučajne šume (Random Forest)**

Ovaj algoritam predstavlja prostu agregaciju stabla odlučivanja. Prosta agregacija ovde znači da obučavamo veći broj modela (stabala odlučivanja), greške ovih modela su nezavisne. Svako stablo se trenira na posebnom skupu atributa, kako bi se postarali da greške budu nezavisne bitno je obučavanje vršiti na različitim podskupovima. Za razliku od stabla odlučivanja ovde se ne radi odsecanje i sam model nije interpretabilan.

## **4.3 LASSO**

LASSO je vrsta linearne regresije koja uz pomoć l1 regularizacije automatski izvršava odabir atributa tako što neke koeficijente stavlja na vrednost 0 – na ovaj način se vrši “proređivanje” (sparse/sparsification) modela i oni atributi koji nemaju vrednost 0 su nam od značaja.

Ovi koeficijenti se dobijaju tako što se kroz optimizaciju trudimo da minimizujemo sumu kvadrata razlike između očekivanih vrednosti i predikcija (RSS – residual sum of squares) na koju dodajemo sumu apsolutnih vrednosti koeficijenata pomnoženu regularizacionim parametrom (izračunamo sumiramo L1 normu za svaki koeficijent i pomnožimo je regularizacionim parametrom). Ovo se takođe može definisati na sledeći način:



*Slika 9. Cilj LASSO regresije*[9]

# **Metaheuristički algoritmi**[2]

Ako posmatramo na izbor atributa kaon a problem optimizacije, tj da optimizujemo našu „pretragu“ za skup atributa, možemo iskoristiti i ove vrste algoritma. Od ovih algoritma možemo da očekujemo malo poboljšanje, i ne možemo očekivati da ćemo dobiti optimalno rešenje [2]. Nadalje će biti objašnjeni genetski algoritam i simulirano kaljenje.

## **5.1 Genetski algoritam**

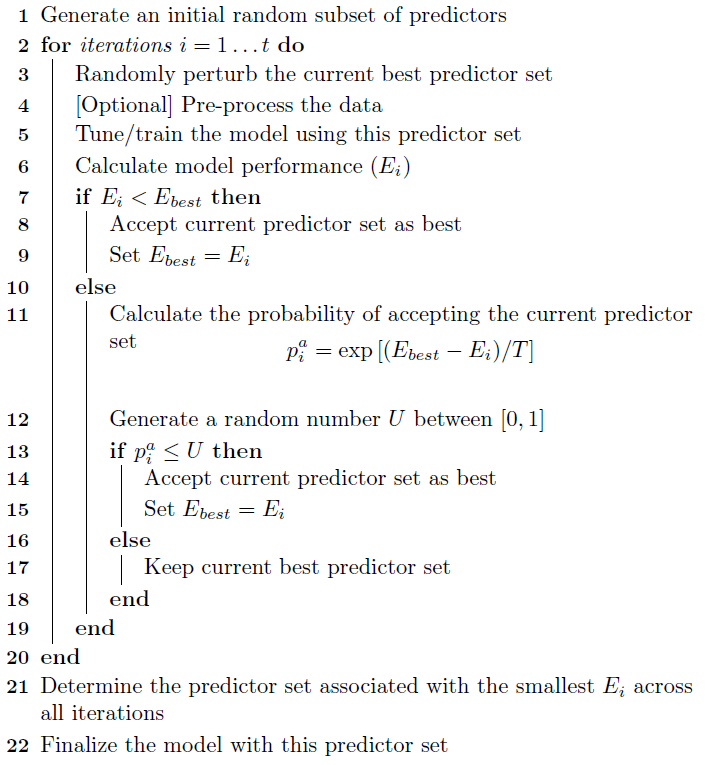
Ovaj algoritam je inspirisan evolucijom, tj. činjenicom da geni nekog organizma kroz generacije evoliraju kako bi se prilagodili i opstali. Ovaj algoritam se takođe negde ubraja i u wrapper metode.

Početna populacija se nasumično bira. Nakon ovog koraka se evaluira “fitness“ funkcija, i biraju se jedinke za reprodukciju na osnovu ove funkcije. Kombinacijom roditelja dobijamo dete/potomka, ali isto imamo i mutaciju koja unosi nasumične promene. Kada se ovo završi ovaj novi potomak se dodaje postojećoj populaciji. Sam algoritam se ponavlja od koraka evaluacije fitness funkcije sve dok se ne dođe do kriterijuma zaustavljanja.

## **5.2 Simulirano kaljenje**

Ovaj algoritam je takođe pohlepan (greedy) algoritam, ali je ovo ublaženo time što može prevazići lokalne minimume (ili lokalno “najbolja” rešenja) tako što uzima u obzir i rešenja koja su manje optimalna od trenutno najboljeg. Ova osobina nam omogućava da uzmemo u obzir veći broj potencijalnih rešenja, čime povećavamo šansu da nađemo (globalno) najbolje rešenje.

Ako posmatramo ovaj algoritam u kontekstu kretanja nad nekim poljem pretrage, pre nego što napravimo loš korak, izračuna se verovatnoća za prihvatanje tog koraka/poteza. Ako je ova verovatnoća veća od našeg dopuštenog praga, uzima se taj korak i istražuje se i stanje nakon tog koraka.[10]



*Slika 10,Izbor atributa upotrebom simuliranog kaljenja, pseudokod*[11]

## **Zaključak**

Izbor atributa može predstavljati veoma značajan korak u pripremi našeg skupa podataka. Zbog vremenskog i memorijskog ograničenja, ova tehnika se veoma često koristi, čak i kada se uzme u obzir da neka od njenih rešenja mogu da budu komputaciono skupa, što dodatno ide u prilog snazi ove tehnike.

Noviteti u ovoj oblasti se zasnivaju na automatizaciji izbora atributa, takođe sve veća upotreba dubokog učenja i pojma fokusa kod mreža može da se iskoristi za odabir najrelevantnijih karakteristika nekog skupa podataka.

# **Literatura**

[1] Garcia, S., et al, *Data preprocessing in data mining*. Springer, 2015.

[2] M. Haque. "Feature Engineering & Selection for Explainable Models A Second Course for Data Scientists," in *LULU Internacional*, 2022.

[3] J. Brownlee, *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*. Machine Learning Mastery, 2020.

[4] Wikipedia contributors, "Pearson correlation coefficient –- Wikipedia, The Free Encyclopedia," 2024

[5] Ž. Krneta, “Spirmanova Rang Korelacija (Spearman’s Ro),” PiStatistics, https://pistatistics.com/course/koeficijenti\_korelacije/lekcije/spirmanova-rang-korelacija-spearmans-ro/ (accessed Sep. 26, 2024).

[6] Wikipedia contributors, "Spearman's rank correlation coefficient –- Wikipedia, The Free Encyclopedia," 2024.

[7] C. Albon, *Machine learning with python cookbook: Practical solutions from preprocessing to deep learning*. O'Reilly Media, Inc., 2018.

[8] P. N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, Introduction to Data Mining. Pearson, 2019.

[9] Wikipedia contributors, Lasso (statistics) — Wikipedia, The Free Encyclopedia. 2024.

[10] Mohd. S. Haque, Md. Fahim, and M. Ibrahim, An Exploratory Study on Simulated Annealing for Feature Selection in Learning-to-Rank. 2023.

[11] M. Kuhn and K. Johnson, Feature Engineering and Selection: A Practical Approach for Predictive Models. CRC Press, Taylor & Francis Group, 2020