

Univerzitet u Nišu
Elektronski fakultet u Nišu



Seminarski rad
Prikupljanje i predobrada podataka za mašinsko učenje
Izbor atributa (Feature selection)

Mentor:

Doc. dr. Aleksandar Stanimirović

Student:

Mina Nikolić 1540

Sadržaj

1.	Uvod.....	3
2.	Kvalitet podataka i metode za obradu podataka	5
3.	Definisanje pojma izbora atributa	6
3.1.	Sličnosti i razlike između izbora atributa i srodnih oblasti	7
3.2.	Arhitektura za proces izbora atributa	8
4.	Traženje podskupa atributa	9
4.1.	Traženje podskupa atributa zasnovano na smeru traženja	10
4.2.	Traženje podskupa atributa zasnovano na strategiji pretrage	12
4.3.	Traženje podskupa atributa zasnovano na izboru kriterijuma selekcije.....	14
4.4.	Mere konzistentnosti	16
4.5.	Mere tačnosti.....	16
5.	Tipovi pristupa za proces izbora atributa	17
5.1.	Prikaz seta podataka za praktičnu demonstraciju procesa izbora atributa.....	17
5.2.	Embedded prisup	18
5.3.	Filter pristup.....	19
5.4.	Wrapper pristup	21
6.	Određivanje težine atributa	23
7.	Statističke metode za selekciju atributa na osnovu tipa podataka.....	24
7.1.	Numerički ulaz i numerički izlaz	26
7.2.	Numerički ulaz i kategorički izlaz	27
7.3.	Kategorički ulaz i kategorički izlaz	28
7.4.	Kategorički ulaz i numerički izlaz	28
8.	Interpretabilnost izbora atributa korišćenjem Explainable AI alata	29
8.1.	LIME (Local-Interpretable Model-Agnostic Explanations) biblioteka	30
8.2.	SHAP (Shapley Additive Explanations) biblioteka	32
8.3.	ELI5 biblioteka	34
9.	Zaključak.....	37
10.	Literatura.....	38

1. Uvod

Tendencija sve masovnije primene algoritama mašinskog učenja i veštačke inteligencije je dovela do razvoja naprednijih i efikasnijih načina na koje je moguće vršiti obradu podataka u cilju rešavanja raznih tipova problema. Problemi o kojima je reč su prisutni u okviru različitih domena koji se tiču nauke, zdravstva, edukacije, marketinga, ali i primene u okviru IT industrije.

Deo istraživačkog rada u okviru oblasti mašinskog učenja je posvećen kreiranju novih i modifikaciji postojećih algoritama sa ciljem postizanja boljih rezultata, dok je druge strane fokus i na adekvatnoj pripremi podataka nad kojima će biti vršena obrada. Stoga je bitno je naglasiti da rezultati primene algoritama ne zavise samo od njihovih karakteristika, već i od podataka nad kojima se vrši njihovo treniranje.

Za definisanje kvaliteta podataka se uglavnom koriste metrike kao što su tačnost, kompletnost, konzistenstnost, uverljivost, pravovremenost i interpretabilnost [1]. Podaci dobijeni iz realnih izvora podataka retko se nalaze u obliku koji je pogodan za primenu odgovarajućih algoritama mašinskog učenja, stoga se može reći da često ispoljavaju osobine koje nisu u skladu sa prethodno navedenim metrikama za ocenu kvaliteta podataka.

Da bi primena algoritama bila moguća, prethodno je potrebno izvršiti adekvatnu predobradu datih podataka na način koji odgovara nameni za koju će biti korišćeni. Kompleksnost predobrade podataka se ne ogleda samo u izboru metoda koje će biti korišćene, već i u načinu na koji se te metode mogu prilagoditi različitim tipovima podataka. Osim pomenutog, takođe se javlja i problem količine podataka koju je potrebno obraditi, budući da često biva slučaj da se zahteva rad nad podacima koji pripadaju Big Data domenu.

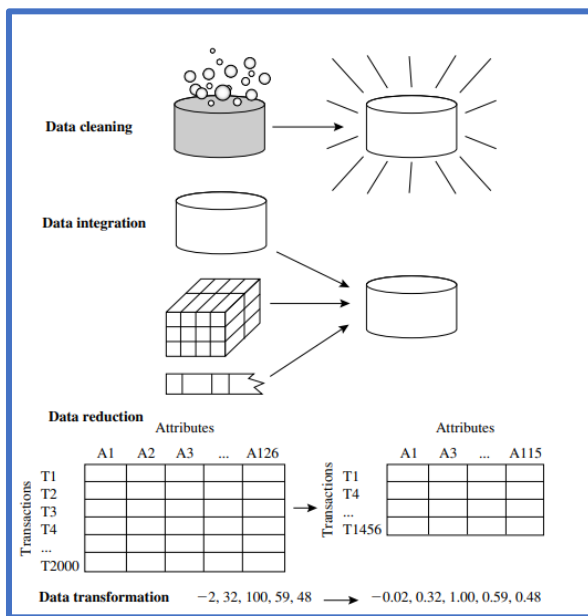
U dostupnoj literaturi je u velikom broju slučajeva dat prikaz obrade tabelarnih podataka, međutim potrebno je voditi računa o tome da se predobrada znatno razlikuje ukoliko je reč o multimedijalnim podacima poput slika i video zapisa. Uslovno rečeno, tip podataka diktira način na koji će se vršiti predobrada.

Tehnike koje se koriste za predobradu podataka imaju za cilj poboljšanje kvaliteta podataka kroz njihovu transformaciju (poput normalizacije, skaliranja, enkodiranja kategoričkih atributa), određivanje načina za rad sa nedostajućim podacima, redukciju dimenzionalnosti, rešavanje problema šuma u podacima, izbor instanci, ali i izbor atributa.

Očekivani ishod pravilne primene pomenutih metoda nad izabranim skupom podataka je unapređenje kvaliteta samih podataka, a samim tim i rezultata dobijenih nakon primene odabranih algoritama mašinskog učenja. U zavisnosti od seta podataka koji se koristi, sam proces predobrade može biti znatno duži i zahtevniji, dok je moguć i slučaj kada je potrebna minimalna modifikacija podataka nad kojima će se vršiti dalja obrada.

Predobrada podataka se može podeliti na nekoliko različitih celina poput čišćenja podataka, redukcije količine podataka, redukcije dimenzionalnosti, integracije podataka iz različitih izvora, ali i primene odgovarajućih transformacija nad podacima.

Cilj seminarskog rada je pre svega razumevanje različitih načina na koje je moguće vršiti predobradu podataka kroz izbor atributa (feature selection). U nastavku će biti dati teorijski prikazi odgovarajućih metoda, ali i praktični načini implementacije nad realnim skupom podataka kako bi bili prikazani efekti primene različitih metoda i tehnika za izbor atributa. Na slici 1 je moguće videti prikaz celina za vršenje predobrade podataka.



Slika 1 – Prikaz celina za vršenje predobrade podataka

Seminarski rad je organizovan na način da je prvo poglavlje posvećeno uvodnom delu u kome se pre svega vrši definisanje problema izbora atributa, nakon čega se u drugom poglavlju definišu pojam kvaliteta podataka i različite metode za obradu podataka. Fokus trećeg poglavlja je na definisanju izbora atributa uz pregled sličnosti i razlika između izbora atributa i srodnih oblasti, ali i tipične arhitekture za sam proces selekcije atributa.

Četvrto poglavlje se odnosi na proces traženja podskupa atributa zasnovanim na smeru traženja, strategiji pretrage, na izboru kriterijuma za pretragu, merama konzistentnosti i merama tačnosti. Cilj petog poglavlja je pregled tipova pristupa kroz Filter, Embedded i Wrapper metode. Rešavanje problema određivanja težine atributa je prikazano u okviru šestog poglavlja, dok su u sedmom poglavlju prikazane statističke metode za izbor atributa prema tipu podataka koji se koriste za analizu.

Osmo poglavlje je posvećeno problemu interpretabilnosti u domenu izbora atributa, stoga je dat prikaz različitih interpretacija dobijenih korišćenjem SHAP, LIME i ELI5 biblioteka. Na kraju, u okviru devetog poglavlja je dat zaključak, dok je poslednje, deseto poglavlje posvećeno pregledu literature.

2. Kvalitet podataka i metode za obradu podataka

Potreba za predobradom podataka se javlja pre svega zbog prirode realnih podataka nad kojima je potrebno vršiti dalju obradu. Podaci mogu biti mašinski generisani, ali i dobijeni putem unosa od strane ljudi. Kao što je prethodno napomenuto u uvodnom delu, faktori kojima se može vršiti procena kvaliteta podataka su:

- Tačnost (accuracy),
- Kompletnost (completeness),
- Konzistentnost (consistency),
- Pravovremenost (timeliness),
- Uverljivost (believability) i
- Interpretabilnost (interpretability)

Razlozi zbog kojih *tačnost* podata može biti problematična se odgledaju kroz pogrešno funkcionisanje mašina koje su zadužene za proces prikupljanja, neadekvatan unos od strane ljudi, namerni unos pogrešnih vrednosti radi prikrivanja ličnih informacija, kao i tehničke greške prilikom prenosa podataka.

Sa druge strane, *nekompletnost* podataka se može javiti usled nedostupnih ulaznih atributa, ali i pogrešne inicijalne pretpostavke da dati atributi neće biti od koristi. Konzistentnost podrazumeva da isti podaci koji se nalaze na različitim lokacijama ne smeju biti u konfliktu (moraju imati istu vrednost).

Pravovremenost se odnosi na činjenicu da su neki od podataka neophodni za obradu samo u određenim vremenskim perioda, a da je njihov značaj nakon toga minimalan. *Uverljivost* je metrika koja se može posmatrati iz ugla korisnika, odnosno može se reći da je data metrika zapravo prikaz poverenja koje korisnici ukazuju datim podacima.

Interpretabilnost se pre svega odnosi na činjenicu koliko su podaci i dobijeni rezultati razumljiviji za tumačenje i može se drugačije posmatrati u zavisnosti od tipa korisnika koji rukovodi podacima [1].

Nakon estimacije kvaliteta podataka potrebno je odlučiti koje tehnike i metode za obradu podataka će biti sprovedene u cilju poboljšanja njihovog kvaliteta. Metode koje se koriste za predobradu se mogu podeliti na podskupove koji se bave *čišćenjem podataka* (data cleaning), njihovom *integracijom* (data integration), *redukcijom količine podataka* koja će biti korišćena (data reduction), *redukcijom dimenzionalnosti* (dimensionality reduction), *smanjenjem brojnosti* (numerosity reduction) ali i *transformacijom podataka* (data transformation).

U literaturi postoji konflikt oko toga u kojem odnosu se redukcija dimenzionalnosti i izbor atributa nalaze. Stoga će u narednom poglavlju biti dat prikaz poređenja ovih pojmova.

3. Definisane pojma izbora atributa

Analiza setova podataka može podrazumevati suočavanje sa stotinama atributa od kojih je samo jedan deo relevantan i neophodan prilikom daljeg procesiranja. Rad sa velikim brojem atributa koji su irelevantni ili redundantni može imati za posledicu usporavanje celokupnog procesa obrade podataka.

Veliki broj algoritama mašinskog učenja funkcioniše primetno bolje ukoliko je dimenzionalnost, odnosno broj atributa u okviru skupa podataka manji. Razlog se ogleda u tome što se na taj način eliminišu atributi koji nisu od interesa odnosno koji predstavljaju izvestan šum. Još jedna od pogonosti rada sa manjim brojem atributa je i mogućnost lakšeg tumačenja modela odnosno može se reći da na taj način dolazi do povećanja interpretabilnosti modela i dobijenih rezultata.

Pored pomenutih, smanjenje broja atributa nad kojima se vrši obrada dovodi i do mogućnosti preglednijih vizuelizacija (pogotovo ukoliko se radi o dvodimenzionalnom ili trodimenzionalnom prostoru) [3].

Sa druge strane, problem može nastati i ukoliko se greškom eliminišu atributi koji su od značaja i samim tim dolazi do degradacije kvaliteta podataka nad kojima se vrši obrada. Kako bi se te potencijalne situacije izbegle, potrebno je primeniti različite tehnike i metode u okviru domena izbora atributa.

Jedan od načina za izbor atributa bi podrazumevao pomoć od strane domenskog eksperta, koji međutim može postati usko grlo sistema ako je reč o ogromnim količinama podataka koje je potrebno pregledati. Stoga, potrebno je okrenuti se i alternativnim rešenjima među kojima se nalazi redukcija dimenzionalnosti ali i izbor atributa.

Izbor atributa se može definisati kao *“proces odabira optimalnog podskupa atributa koji zadovoljavaju određeni kriterijum”* [2]. Razlozi zbog kojih se najčešće vrši izbor atributa su:

- Odbacivanje nerelevantnih podataka,
- Povećanje tačnosti kod modela mašinskog učenja,
- Redukovanje troškova podataka,
- Poboljšanje efikasnosti učenja kroz smanjenje memorije neophodne za čuvanje podataka i obradu od strane računara i
- Redukovanje kompleksnosti dobijenih rezultata u cilju poboljšavanja razumljivosti modela i podataka odnosno njihove interpretabilnosti [2].

Sam proces izbora atributa se može posmatrati kroz više različitih perspektiva u zavisnosti od domena primene. Prema tumačenju iz literature [2], kao perspektive sa najvećim značajem i primenom možemo izdvojiti:

- Traženje najboljeg podskupa atributa,
- Kriterijume za evaluaciju različitih podskupova
- Principe za selektovanje, oduzimanje, dodavanje i modifikaciju atributa prilikom samog procesa traženja
- Principe za selektovanje, oduzimanje, dodavanje i modifikaciju atributa prilikom primena u okviru konkretnih implementacija

U okviru rada, princip selekcije odnosno izbora atributa će biti prikazan na problemu koji se tiče binarne klasifikacije. Na taj način će biti omogućen prikaz različitih rešenja koje je moguće implementirati korišćenjem svake od pomenutih metoda za izbor atributa.

Potrebno je naglasiti da u okviru domena izbora atributa ne postoji rešenje tipa “one fits all”, već je poželjno primeniti više različitih metoda i upoređivanjem dobijenih rezultata odabrati najpogodnije.

Sam proces izbora atributa će u mnogome zavisiti od konkretnog skupa podataka nad kojim se vrši obrada. Skup podataka nad kojim će u okviru rada biti izvršena demonstracija različitih metoda i načina za adekvatan izbor atributa se nalazi na adresi: [dodati adresu].

3.1. Sličnosti i razlike između izbora atributa i srodnih oblasti

Redukcija dimenzionalnosti predstavlja veliki problem prilikom procesa mašinskog učenja budući da što veći broj podataka podrazumeva i veće troškove prilikom procesiranja. Izbor atributa predstavlja jedan od načina na koji se može rešiti deo problema vezanih za redukciju dimenzionalnosti [1].

U literaturi često dolazi do nekonzistentnosti prilikom definisanja odnosa redukcije dimenzionalnosti i izbora atributa. Sa jedne strane se može smatrati da izbor atributa predstavlja podskup redukcije dimenzionalnosti, dok se sa druge strane može posmatrati i kao nezavisna celina.

Glavna razlika između redukcije dimenzionalnosti i izbora atributa se ogleda u tome što izbor atributa ne podrazumeva izmenu atributa, već samo izbor najpogodnijih, dok se kod redukcije dimenzionalnosti iz postojećih atributa dobijaju novi.

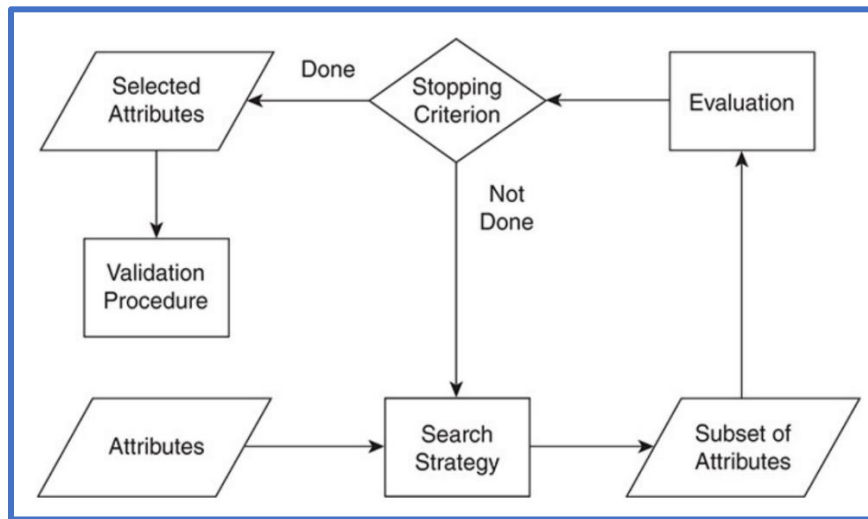
Kod redukcije dimenzionalnosti se prilikom implementacije najčešće koristi metoda ***Principal Component Analysis (PCA)***, dok je kod izbora atributa odabir metode usko povezan sa paradigmom odnosno perspektivnom iz koje posmatramo problematiku.

Različita tumačenja potiču pre svega od načina na koji dolazi do definisanja datih pojmova, ali se bez obzira na njihov međusobni odnos i podelu treba osvrnuti detaljno na operacije i metode koje se koriste u okviru izbora atributa.

3.2. Arhitektura za proces izbora atributa

Celokupan proces za selekciju odnosno izbor atributa se može posmatrati kroz nekoliko različitih celina koje se pre svega odnose na izbor mere za evaluaciju podskupa atributa, na strategiju za generisanje novih podskupova podataka, na kriterijum za prekid procesa pretrage, kao i način za validaciju dobijenih rezultata.

Arhitektura je prikazana kao generalni koncept koji daje način da se poveća opštost rešenja, međutim u zavisnosti od konkretne primene je poželjno izvršiti specijalizaciju i staviti fokus na optimizaciju konkretnih delova izvršavanja [3]. Na slici 2 je moguće videti prikaz arhitekture za proces izbora atributa.



Slika 2 – Prikaz arhitekture za proces izbora atributa

Traženje odgovarajuće strategije pretrage zasnovane na kriterijumu pretrage će biti detaljno prikazano u okviru poglavlja 4.2 pri čemu će akcenat pre svega biti na definisanju prednosti i mana algoritama koji se nalaze u domenu heurističke pretrage, sveobuhvatne pretrage, kao i nedeterminističke pretrage [3].

Nakon pronalaska podskupa atributa, potrebno je vršiti evaluaciju kako bi se dobijeni rezultati uporedili sa onim koji su prethodno predloženi i došlo da zaključka da li dolazi to tendencije poboljšanja ili pogoršanja. Proces evaluacije će se razlikovati u zavisnosti od toga da li je reč o Filter, Wrapper ili Embedded metodama za izbor atributa (detaljno objašnjenje datih metoda će biti dato u okviru poglavlja pod rednim brojem 5).

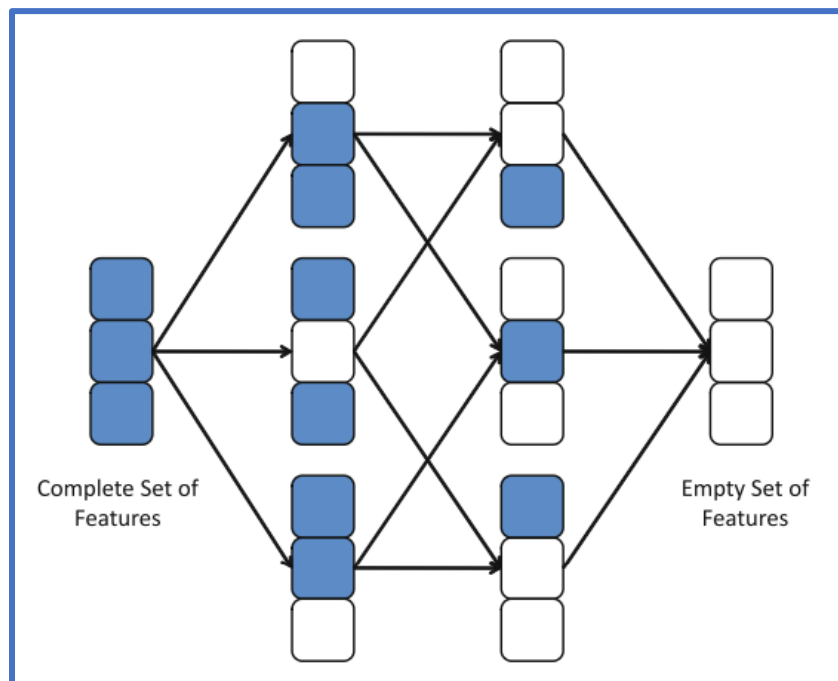
Budući da je broj mogućih podskupova atributa ogroman, potrebno je naći efikasan mehanizam za stopiranje pretrage. Mehanizam za stopiranje je uglavnom baziran na broju iteracija, unapred zadatom threshold-u (granici) ali i na veličini podskupa izdvojenih atributa.

Poslednja celina u okviru arhitekture za izbor atributa se ogleda u validaciji dobijenih rešenja. Najjednostavniji način za validiranje rezultata je upoređivanje pristupa pri kojem se koristi celokupan set podataka sa setom podataka nad kojim se koriste dobijeni podskupovi [3].

4. Traženje podskupa atributa

Budući da se problem izbora atributa može posmatrati u okviru domena problema traženja, svako stanje prostora traženja odgovara odabranom podskupu atributa. Odabir atributa može biti prikazan kao binarni niz, pri čemu elementi sa oznakom jedan predstavljaju odabrane attribute, dok oni sa oznakom nula bivaju odbačeni [2].

Ukupan broj podskupova će biti veličine 2^M , gde je M broj atributa u okviru seta podataka. Realni setovi podataka uglavnom poseduju veliki broj atributa, stoga se proces traženja retko počinje od celokupnog broja atributa pre svega zbog ogromnih troškova procesiranja. Na slici 3 se nalazi prikaz prostora traženja ukoliko set podataka poseduje tri atributa (ovakav prikaz je dat radi demonstracije, ali je teško u praksi naći set sa samo tri atributa) [2].



Slika 3 – Prikaz prostora traženja za izbor atributa

Proces traženja atributa se može posmatrati na različite načine, pri čemu se uzimaju u razmatranje:

- Smer traženja (sekvencijalno generisanje u napred, sekvencijalno generisanje unazad, bidirekciono traženje i nasumično traženje),
- Strategije traženja (strategije grubog traženja, sveobuhvatnog traženja, heurističke pretrage i nedeterminističke pretrage),
- Kriterijumi za selekciju (informacione mere, metrike za distancu, metrike zavisnosti, metrike konzistentnosti, kao i metrike za tačnost) i
- Filter, Wrapper i Embedded metode za izbor atributa.

4.1. Traženje podskupa atributa zasnovano na smeru traženja

Prilikom procesa traženja je bitno voditi računa i o smeru traženja, odnosno o tome da li se kreće od praznog skupa i vrši dodavanje atributa, ili se pak od celog skupa atributa eliminišu oni koji nisu od značaja. Prema smeru traženja, podela se može izvršiti na:

- Sekvencijalno generisanje unapred (Sequential Forward Generation),
- Sekvencijalno generisanje unazad (Sequential Backward Generation),
- Bidirekciono generisanje (Bidirectional Generation) i
- Nasumično generisanje (Random Generation)

Kada je reč o *Sekvencijalnom generisanju unapred*, bitno je naglasiti da pretraga počinje sa praznim skupom atributa S . Daljim radom algoritma dolazi do dodavanja novih atributa u skup S , pri čemu se koristi neki kriterijum koji prepoznaje one attribute koji su bolji od drugih. Kriterijum za zaustavljanje može biti broj relevantnih atributa, kako u skup S ne bi bili dodati svi postojeći atributi. Na slici 4 se nalazi pseudokod za algoritam Sekvencijalnog generisanja unapred.

Algorithm 1 Sequential forward feature set generation - SFG.

```
function SFG( $F$  - full set,  $U$  - measure)
  initialize:  $S = \{\}$  ▷  $S$  stores the selected features
  repeat
     $f = \text{FINDNEXT}(F)$ 
     $S = S \cup \{f\}$ 
     $F = F - \{f\}$ 
  until  $S$  satisfies  $U$  or  $F = \{\}$ 
  return  $S$ 
end function
```

Slika 4 – Pseudokod algoritma za Sekvencijalno generisanje unapred

Sa druge strane, algoritam *Sekvencijalnog traženja unazad* funkcioniše na suprotan način od prethodno pomenutog algoritma. Proces traženja kreće sa celokupnim skupom atributa, pri čemu se u toku izvršavanja eliminišu jedan po jedan atribut, prema određenom kriterijumu zaustavljanja. U ovom slučaju, koristi se kriterijum koji će pokazati i eliminisati attribute koji su najmanje relevantni za dati set podataka. Na slici 5 se nalazi pseudokod za algoritam Sekvencijalnog traženja unazad.

Algorithm 2 Sequential backward feature set generation - SBG.

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

Slika 5 – Pseudokod algoritma za Sekvencijalno generisanje unazad

Algoritmi sekvencijalnog generisanja unapred i unazad mogu biti kombinovani i na taj način se proces traženja može vršiti u oba smera, konkurentno. Algoritam koji koristi takav pristup se naziva *Bidirekciono generisanje*. Pretraga se zaustavlja u slučaju da jedna od pretraga rezultira nalaženjem najboljeg skupa od m atributa, pre nego da dođe do sredine ili kada obe pretrage dođu do sredine. Na slici 6 se nalazi pseudokod za algoritam Bidirekcionog traženja.

Algorithm 3 Bidirectional feature set generation - BG.

```

function BG( $F_f, F_b$  - full set,  $U$  - measure)
  initialize:  $S_f = \{\}$                                 ▷  $S_f$  holds the selected features
  initialize:  $S_b = \{\}$                                 ▷  $S_b$  holds the removed features
  repeat
     $f_f = \text{FINDNEXT}(F_f)$ 
     $f_b = \text{GETNEXT}(F_b)$ 
     $S_f = S_f \cup \{f_f\}$ 
     $F_b = F_b - \{f_b\}$ 
     $F_f = F_f - \{f_f\}$ 
     $S_b = S_b \cup \{f_b\}$ 
  until (a)  $S_f$  satisfies  $U$  or  $F_f = \{\}$  or (b)  $S_b$  does not satisfy  $U$  or  $F_b = \{\}$ 
  return  $S_f$  if (a) or  $F_b \cup \{f_b\}$  if (b)
end function

```

Slika 6 – Pseudokod algoritma bidirekcionog traženja

Pored prethodno pomenutih, postoji još jedan princip, koji se odgleda u tome da ne postoji prethodno definisan smer, već se traženje vrši po principu nasumičnosti. Reč je o *Nasumičnom generisanju*, pri čemu su smer pretrage, ali i odluka o tome da li se vrši dodavanje ili oduzimanje atributa iz skupa S nasumično odabrani. Na slici 7 se nalazi pseudokod za algoritam Nasumičnog traženja. Detalji o pomenutim algoritmima se mogu videti u okviru poglavlja 7.2.1.1 iz literature [2].

Algorithm 4 Random feature set generation - RG.

```

function RG( $F$  - full set,  $U$  - measure)
  initialize:  $S = S_{best} = \{\}$                                 ▷  $S$  - subset set
  initialize:  $C_{best} = \#(F)$                                 ▷  $\#$  - cardinality of a set
  repeat
     $S = \text{RANDGEN}(F)$ 
     $C = \#(S)$ 
    if  $C \leq C_{best}$  and  $S$  satisfies  $U$  then
       $S_{best} = S$ 
       $C_{best} = C$ 
    end if
  until some stopping criterion is satisfied
  return  $S_{best}$                                 ▷ Best set found so far
end function

```

Slika 7 – Pseudokod algoritma nasumičnog traženja

4.2. Traženje podskupa atributa zasnovano na strategiji pretrage

Prilikom procesa pretrage generalno važi tvrdnja da što se više resursa potroši, to je veća verovatnoća da se dobiju bolji rezultati odnosno korisniji atributi. Budući da je česta situacija da se radi sa velikim količinama podataka, radi uštede resursa i vođenja računa o performansama, radi se pretraga koja za posledicu ima manje optimalna rešenja.

Pri tome postoji nekoliko različitih tipova algoritama koji se koriste tokom procesa pretrage i mogu se podeliti na sledeće kategorije:

- Sveobuhvatne pretrage (eng. Exhaustive Search),
- Heurističke pretrage (eng. Heuristic Search) i
- Nedeterminističke pretrage (eng. Nondeterministic search)

Korišćenje algoritama *Sveobuhvatne pretrage* se odgleda u nalaženju svih mogućih podskupova rešenja, stoga se garantuje i otkrivanje optimalnog rešenja. Međutim, potrebno je voditi računa o tome da je za skup koji čine svi atributi prostorna kompleksnost $O(2^M)$, stoga pristup nije pogodan za skupove podataka koji sadrže veliki broj atributa.

Jedna od najkorišćenijih metoda u okviru sveobuhvatne pretrage je *Focus* metoda. Korišćenje Focus metode podrazumeva prolazak kroz sve kombinacije od ukupnog broja atributa, pri čemu se generiše $\sum_{i=1}^M \binom{M}{i}$ podskupova kako bi se pronašao skup od m atributa koji zadovoljava prethodno definisane kriterijume [2]. Na slici 8 se nalazi pseudokod Focus algoritma.

```
Algorithm 7 Focus algorithm.  
function FOCUS( $F$  - all features in data  $D$ ,  $U$  - inconsistency rate as evaluation measure)  
  initialize:  $S = \{\}$   
  for  $i = 1$  to  $M$  do  
    for each subset  $S$  of size  $i$  do  
      if  $CALU(S, D) = 0$  then ▷  $CALU(S, D)$  returns inconsistency  
        return  $S$  - a minimum subset that satisfies  $U$   
      end if  
    end for  
  end for  
end function
```

Slika 8 – Pseudokod Focus algoritma za Sveobuhvatnu pretragu

Heuristička pretraga, kao što i samo ime kaže, za sam proces pretrage koristi heuristiku. U ovom slučaju je kompleksnost reda veličine $O(M)$ što će rezultovati boljim performansama, ali je gotovo sigurno da rezultat pretrage neće biti optimalno rešenje. Izbor heuristike ima najveći uticaj na nalaženje što boljeg rešenja.

Jedan od najjednostavnih načina za implementaciju heurističke metode bi bio upotreba modela koji omogućava izbor najpogodnijih atributa konkretno za dati algoritam (kao što je generalno slučaj sa stablima odluke).

Kompleksnije rešenje od prethodno pomenutog bi se moglo ogledati u primeni *MIFS algoritma* (eng. Mutual Information Based Feature Selection). Algoritam funkcioniše na način da računa MI meru između dva atributa, pri čemu bira kao sledeći atribut onaj koji maksimizira informacije o klasi do prosečne MI vrednosti nad prethodno izabrana dva atributa. Na slici 9 se može videti pseudokod MIFS algoritma [2].

```

Algorithm 8 MIFS algorithm.
function MIFS( $F$  - all features in data,  $S$  - set of selected features,  $k$  - desired size of  $S$ ,  $\beta$  -
regulator parameter)
  initialize:  $S = \{\}$ 
  for each feature  $f_i$  in  $F$  do
    Compute  $I(C, f_i)$ 
  end for
  Find  $f_{max}$  that maximizes  $I(C, f)$ 
   $F = F - \{f_{max}\}$ 
   $S = S \cup f_{max}$ 
  repeat
    for all couples of features  $(f_i \in F, s_j \in S)$  do
      Compute  $I(f_i, s_j)$ 
    end for
    Find  $f_{max}$  that maximizes  $I(C, f) - \beta \sum_{s \in S} I(f_i, s_j)$ 
     $F = F - \{f_{max}\}$ 
     $S = S \cup f_{max}$ 
  until  $|S| = k$ 
  return  $S$ 
end function

```

Slika 9 – Pseudokod MIFS algoritma za heurističku pretragu

Kada je reč o nedeterminističkoj pretrazi potrebno je naglasiti da je nastala kao kombinacija prethodna dva pristupa pri čemu je poznata i pod nazivom *Pretraga nasumičnim izborom*. Pretraga funkcioniše na način da se u svakom koraku generiše nasumični podskup podataka, uz to da se dobijeni rezultati tokom vremena poboljšavaju. Tipičan predstavnik ovog tipa algoritama je *Las Vegas Filter Feature Selection (LVF)*.

Algoritam funkcioniše na način da se vrši nasumično generisanje podskupa atributa i definiše se evaluaciona metoda koja ima za cilj da proveriti da li svaki dobijeni podskup zadovoljava prethodno definisane kriterijume [2]. Na slici 10 je prikazan pseudokod LVF algoritma za nedeterminističku pretragu.

```

Algorithm 9 LVF algorithm.
function LVF( $D$  - a data set with  $M$  features,  $U$  - the inconsistency rate,  $maxTries$  - stopping
criterion,  $\gamma$  - an allowed inconsistency rate)
  initialize: list  $L = \{\}$  ▷  $L$  stores equally good sets
   $C_{best} = M$ 
  for  $maxTries$  iterations do
     $S = \text{RANDOMSET}(\text{seed})$ 
     $C = \#(S)$  ▷  $\#$  - the cardinality of  $S$ 
    if  $C < C_{best}$  and  $\text{CALU}(S, D) < \gamma$  then
       $S_{best} = S$ 
       $C_{best} = C$ 
       $L = \{S\}$  ▷  $L$  is reinitialized
    else if  $C = C_{best}$  and  $\text{CALU}(S, D) < \gamma$  then
       $L = \text{APPEND}(S, L)$ 
    end if
  end for
  return  $L$  ▷ all equivalently good subsets found by LVF
end function

```

Slika 10 – Pseudokod LVF algoritma za nederminističku pretragu

4.3. Traženje podskupa atributa zasnovano na izboru kriterijuma selekcije

U okviru domena izbora atributa, pored pomenutih metoda potrebno je naći i načine za određivanje njihovog kvaliteta. Međutim, kriterijumi selekcije koje je potrebno odabrati pre svega zavise od namene za koju se vrše predobrada i analiza podataka.

Ukoliko je reč o klasifikaciji (praktični primeri u okviru rada prikazuju problem binarne klasifikacije), uglavnom se kao cilj uzima povećanje tačnosti, odnosno *accuracy* parametra. Kriterijumi selekcije će prema literaturi [2] biti posmatrani kroz sledeće celine:

- Informacione mere (eng. Information Measures)
- Mere rastojanja (eng. Distance Measures)
- Mere zavisnosti (eng. Dependence Measures)
- Mere konzistentnosti (eng. Consistency Measures)
- Mere tačnosti (eng. Accuracy measures)

Kada je reč o *informacionim merama*, cilj se pre svega ogleda u merenju neizvesnosti primaoca poruke, pri čemu porukom smatramo *Output klasu* prilikom procesa klasifikacije. Ukoliko primaoc razume poruku, smatra se da je neizvesnost niska, pri čemu važi i obrnuto, nerazumevanje poruke rezultuje visokim stepenom neizvesnosti. Na slici 11 se nalazi matematička interpretacija informacione mere:

$$IG(A) = \sum_i U(P(c_i)) - \mathbf{E} \left[\sum_i U(P(c_i|A)) \right]$$

Slika 11 – Matematička interpretacija informacione mere

Ukoliko se za proces izbora atributa koriste informacione mere, odabir odgovarajućeg atributa A_i se vrši umesto A_j ukoliko je $IG(A_i) > IG(A_j)$. Drugačije rečeno, atribut A_i se bira ukoliko više smanjuje neizvesnost od atributa A_j .

Diskusija o *merama rastojanja* se pre svega zasniva na činjenici da se traže atributi koji će što bolje izvršiti separaciju, odnosno razdvajanje klasa. Ukoliko se posmatra problem binarne klasifikacije, ukoliko se sa $D(A)$ predstavi distanca između $P(A|c_1)$ i $P(A|c_2)$, atribut A_i se bira umesto atributa A_j ukoliko važi $D(A_i) > D(A_j)$ [2].

U literaturi se kao popularne mere rastojanja navode usmerena divergentnost (eng. directed divergence) i varijansa. Njihova matematička interpretacije je data na slikama 12 i 13, respektivno. Često korišćene mere rastojanja su prikazane u tabeli na slici 14.

$$DD(A_j) = \int \left[\sum P(c_i | A_j = a) \log \frac{P(c_i | A_j = a)}{P(c_i)} \right] P(A_j = a) dx.$$

Slika 12 – Matematička interpretacija usmerene divergentnosti

$$V(A_j) = \int \left[\sum P(c_i) (P(c_i | A_j = a) - P(c_i))^2 \right] P(A_j = a) dx.$$

Slika 13 – Matematička interpretacija varijanse

	Mathematical form
Euclidean distance	$D_e = \left\{ \sum_{i=1}^m (x_i - y_i)^2 \right\}^{\frac{1}{2}}$
City-block distance	$D_{cb} = \sum_{i=1}^m x_i - y_i $
Cebyshev distance	$D_{ch} = \max_i x_i - y_i $
Minkowski distance of order m	$D_M = \left\{ \sum_{i=1}^m (x_i - y_i)^m \right\}^{\frac{1}{m}}$
Quadratic distance Q , positive definite	$D_q = \sum_{i=1}^m \sum_{j=1}^m (x_i - y_i) Q_{ij} (x_j - y_j)$
Canberra distance	$D_{ca} = \sum_{i=1}^m \frac{ x_i - y_i }{x_i + y_i}$
Angular separation	$D_{as} = \frac{\sum_{i=1}^m x_i \cdot y_i}{\left[\sum_{i=1}^m x_i^2 \sum_{i=1}^m y_i^2 \right]^{\frac{1}{2}}}$

Slika 14 – Tabelarni prikaz često korišćenih mera rastojanja

Svrha *mere zavisnosti* (mere asocijacije ili korelacije) se odgleda u računanju stepena korelacije između varijabli. Prilikom procesa odabira atributa, računa se korelacija između ulaznih atributa tj fičera i izlaznog atributa odnosno target fičera.

Ukoliko se oznaka $R(A)$ koristi da prikaže meru zavisnosti između atributa A i klase C , bira se atribut A_i naspram A_j ukoliko važi relacija $R(A_i) > R(A_j)$. To znači da se pre svega biraju atributi koji imaju veći stepen korelacije sa datim klasnim atributom. Mera zavisnosti koja se često koristi prilikom određivanja korelacije između ulaznih atributa i klasnog atributa je *Pearson-ov koeficijent korelacije*.

Na slici 15 je prikazana matematička interpretacija Pearson-ovog koeficijenta korelacije. Potrebno je naglasiti da ako su dve promenljive jako korelisane, jednu od njih je moguće izbaciti i na taj način izvršiti adekvatan odabir atributa [2].

$$\rho(X, Y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2 \right]^{\frac{1}{2}}}$$

Slika 15 – Matematička interpretacija Pearson-ovog koeficijenta korelacije

4.4. Mere konzistentnosti

Mere konzistentnosti pre svaga imaju za cilj da odrede minimalni broj atributa koji može da izvrši razdvajanje klasa koliko bi to bilo moguće korišćenjem celokupnog skupa atributa. U te svrhe koristi mehanizam prepoznavanja atributa koji su redundantni odnosno predstavljaju višak prilikom obrade i izračunavanja. Korišćenjem mera konzistentnosti se pokušava implementacija jednakosti $P(C/Potpuni\ skup\ atributa) = P(C/Dobijeni\ podskup\ atributa)$ [2].

4.5. Mere tačnosti

Mere tačnosti se posebno koriste u situacijama kada je reč o domenu nadgledanog učenja gde se radi o problemu klasifikacije. Međutim, prilikom implementacije rešenja za rad sa modelima koji se koriste za klasifikaciju, potrebno je voditi računa o problemu *overfitting-a*, o vremenu koje je potrebno za treniranje modela (generalno modeli sa većim brojem atributa zahtevaju i duži proces treniranja) ali i o tome da li su dobijeni atributi uslovljeni modelom mašinskog učenja koji se koristi prilikom treniranja. Na slici 16 je prikazan pregled formula koje se mogu koristiti prilikom računanja mere tačnosti.

	Mathematical form
Accuracy	$\frac{tp+fp}{tp+tn+fp+fn}$
Error rate	1 – Accuracy
Chi-squared	$\frac{n(fp \times fn - tp \times tn)^2}{(tp+fp)(tp+fn)(fp+tn)(tn+fn)}$
Information gain	$e(tp+fn, fp+tn) - \frac{(tp+fp)e(tp,fp)+(tn+fn)e(fn,tn)}{tp+fp+tn+fn}$ where $e(x, y) = -\frac{x}{x+y} \log_2 \frac{x}{x+y} - \frac{y}{x+y} \log_2 \frac{y}{x+y}$
Odds ratio	$\frac{tpr}{1-tpr} \bigg/ \frac{fpr}{1-fpr} = \frac{tp \times tn}{fp \times fn}$
Probability ratio	$\frac{tpr}{fpr}$

Slika 16 – Prikaz matematičkih formula za računanje mere tačnosti

5. Tipovi pristupa za proces izbora atributa

Kada je reč o različitim pristupima za proces izbora atributa, svi oni se pre svega mogu klasifikovati u odnosu na to da li pripadaju kategoriji nadgledanih (*eng. supervised*) ili nenadgledanih (*eng. unsupervised*) algoritama mašinskog učenja.

Glavna karakteristika nadgledanih algoritama se ogleda u tome da se prilikom procesa treniranja koristi izlazna, odnosno *target* varijabla, dok kod nenadgledanih to nije slučaj. Prilikom selekcije atributa se mogu koristiti različiti pristupi, stoga će u narednim poglavljima biti prikazana teorijska osnova i praktična primena rada sa *Embedded*, *Filter* i *Wrapper* metodama.

5.1. Prikaz seta podataka za praktičnu demonstraciju procesa izbora atributa

Kao što je prethodno napomenuto u okviru uvodnog dela, cilj seminarskog rada je prikaz teorijskih, ali i praktičnih primera u okviru samog procesa izbora podataka. Set podataka koji je izabran u ove svrhe, ali i celokupna implementacija su dostupni na sledećoj adresi: <https://github.com/minanikolic916/Feature-selection-/tree/master>.

Praktična implementacija se tiče rada na problemu binarne klasifikacije za određivanje toga da li su korisnici zavisni od upotrebe Interneta ili ne. U setu podataka je inicijalno dostupno 136 kolona koje predstavljaju rezultate dobijene anketiranjem korisnika, dok je broj instanci 2113. Što se tiče procesa predobrade podataka, pre svega su odrađene:

- Provera duplikata,
- Provera NaN vrednosti uz adekvatno čišćenje podataka,
- Sređivanje kolona uz eliminisanje onih koje nisu od značaja,
- Provera balansiranosti seta ali i sam proces balansiranja primenom SMOTE tehnike,
- Detekcija outliera uz vizuelizaciju i njihova eliminacija Isolation Forest metodom,
- Normalizacija i standardizacija podataka, kao i
- Podela na trening i test skup

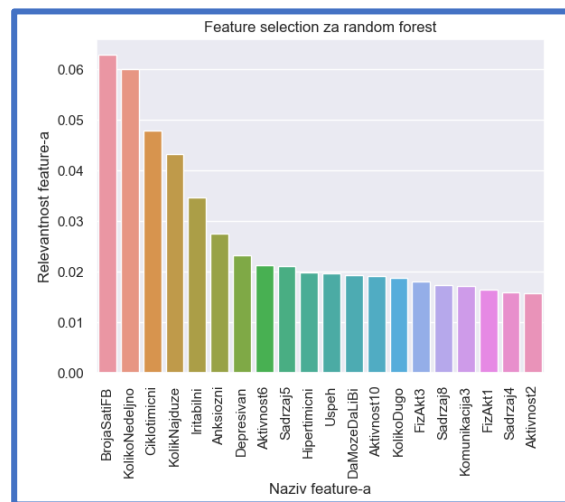
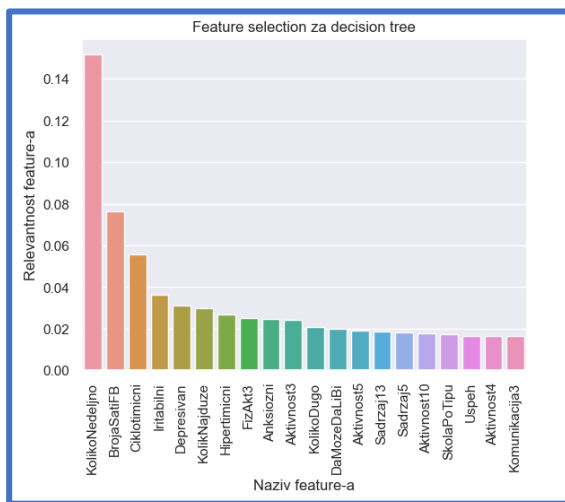
5.2. Embedded pristup

Glavna karakteristika *Embedded* pristupa za selekciju atributa se ogleda u tome da se proces izbora atributa vrši tokom samog treniranja modela. Takav pristup podrazumeva zavisnost izabranih atributa od konkretnog modela koji se koristi tokom treniranja, ali i znači da nikakav alat za eksternu selekciju atributa nije neophodan. Embedded modeli su poznati u literaturi i pod nazivom *Instristic*.

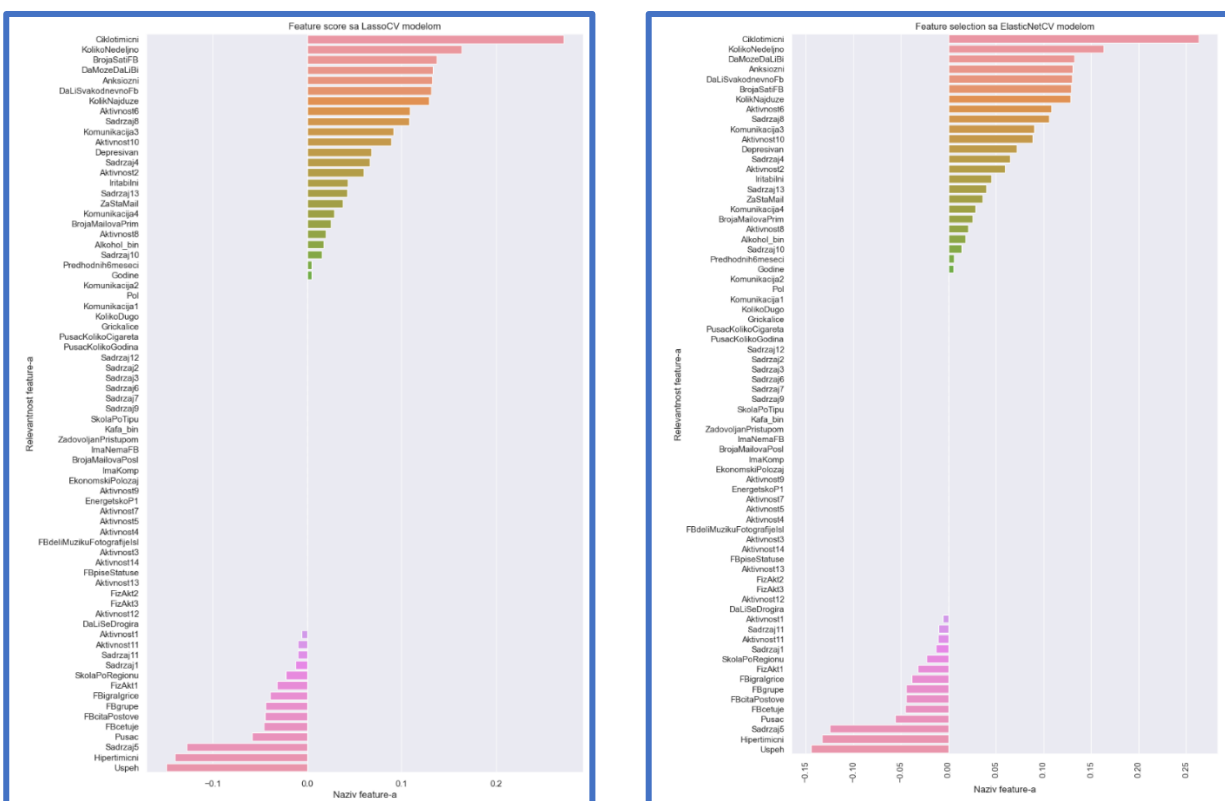
Primenom Embedded pristupa se smanjuje mogućnost overfitting-a, ali se i uglavnom postiže brže određivanje podskupa atributa budući da se postupak dešava prilikom procesa učenja. Nedostatak Embedded pristupa je u tome što direktno zavise od modela što može biti naročito problematično ukoliko se radi sa *tree-based* modelima koji koriste halapljiv (eng. greedy) pristup prilikom selekcije atributa.

Halapljivi pristupi uglavnom nalaze uži skup atributa koji može ispoljavati gore prediktivne performanse nego u slučaju rada sa celokupnim skupom atributa. Bitno je naglasiti i da nije moguće koristiti Embedded pristup sa svakim modelom mašinskog učenja, stoga se alternativa ogleda u primeni Filter i Wrapper metoda [4].

Algoritmi koji su izabrani za Embedded prikaz procesa izbora atributa su *Random Forest*, *Decision tree*, *LassoCV* i *ElasticNetCV*. Rezultati dobijeni korišćenjem datih algoritama su dati na slikama 17, 18, 19 i 20, respektivno.



Slika 17 i 18 – Prikaz najrelevantnijih atributa primenom Decision Tree i Random Forest modela



Slika 19 i 20 – Prikaz najrelevantnijih atributa primenom LassoCV i ElasticNet modela

5.3. Filter pristup

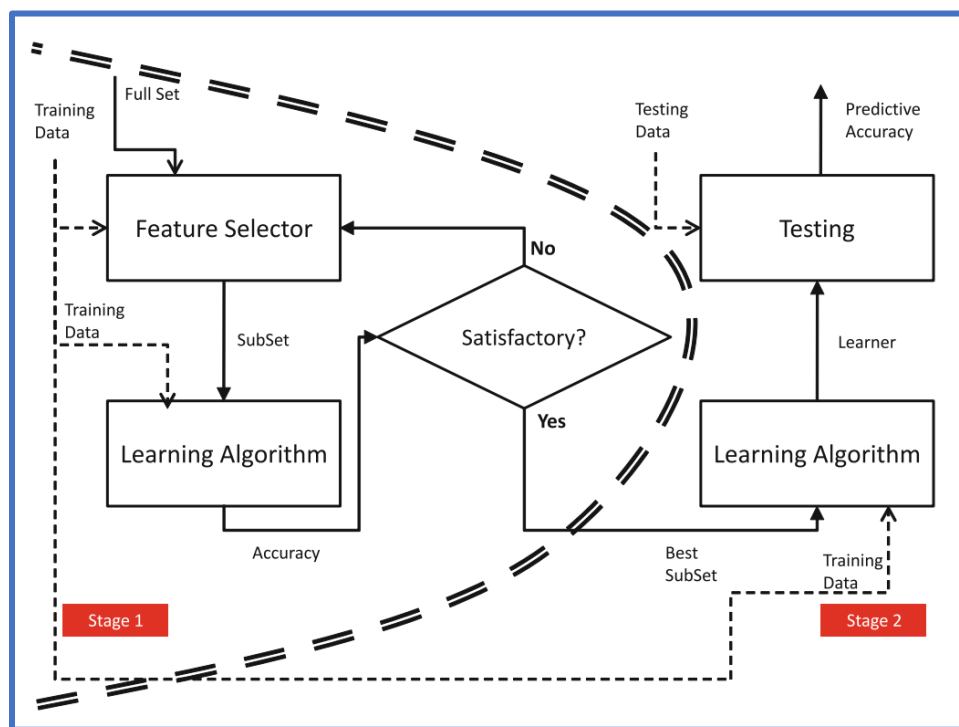
Za razliku od Embedded pristupa, *Filter* metode vrše izbor atributa nezavisno od konkretnog modela koji se koristi prilikom procesa mašinskog učenja. Naziv se ogleda u činjenici da se vrši filtriranje atributa pre samog procesa učenja. Način na koji se vrši filtriranje željenog skupa atributa zavisi od izabrane heuristike.

U okviru Filter metoda se može izdvojiti posebna kategorija pod nazivom *Rangeri*. Ime potiče od činjenice da se koriste metode koje primenjuju određeni kriterijum na osnovu kojeg će se vršiti rangiranje. Pri tome se onda shodno setu podataka bira određeni skup najpogodnijih atributa za dalje procesiranje [2].

Potrebno je naglasiti da Filter pristup funkcioniše adekvatno i sa setovima podataka koji imaju veliki broj dimenzija. Kao što je ranije pomenuto, ukoliko je moguće, potrebno je smanjiti dimenzionalnost kako bi sam proces mašinskog učenja i treniranja modela bio jednostavniji i efikasniji. Na taj način će i performanse dobijenog modela biti bolje.

Model za izbor atributa Filter pristupom se sastoji iz dve faze. Prva faza podrazumeva korišćenje mera (metrika) kao što su informacione mere, mere rastojanja, zavisnosti i konzistentnosti (navedene mere su detaljno objašnjene u prethodnim poglavljima). U drugoj fazi se vrši proces učenja i testiranja, pri čemu se dobijaju rezultati predikcija [2].

Na osnovu svih navedenih karakteristika može se zaključiti da je korišćenje Filter metoda pogodno pre svega zbog moguće primene za svaki tip modela (mogu se koristiti i u domenu dubokog učenja za rad sa neuronskim mrežama), zbog mogućnosti rada sa velikom količinom podataka (budući da je proces izračunavanja vremenski manje zahtevan jer se koriste evaluacione mere) ali i brzine izračunavanja (brže izračunavanje je takođe uslovljeno činjenicom da se radi sa evaluacionim metrikama poput distance, zavisnosti i konzistentnosti). Na slici 21 se može videti grafički prikaz procesa izbora atributa kada se radi sa Filter modelom [2].



Slika 21 – Proces izbora atributa primenom Filter metoda

5.4. Wrapper pristup

Izbor atributa primenom *Wrapper* pristupa se ogleda u ideji da se sam proces izbora vrši korišćenjem klasifikatora i metrika za prediktivne performanse datog klasifikatora. Drugačije rečeno, dobrota izabranih atributa će se ogledati kroz same performanse modela za klasifikaciju.

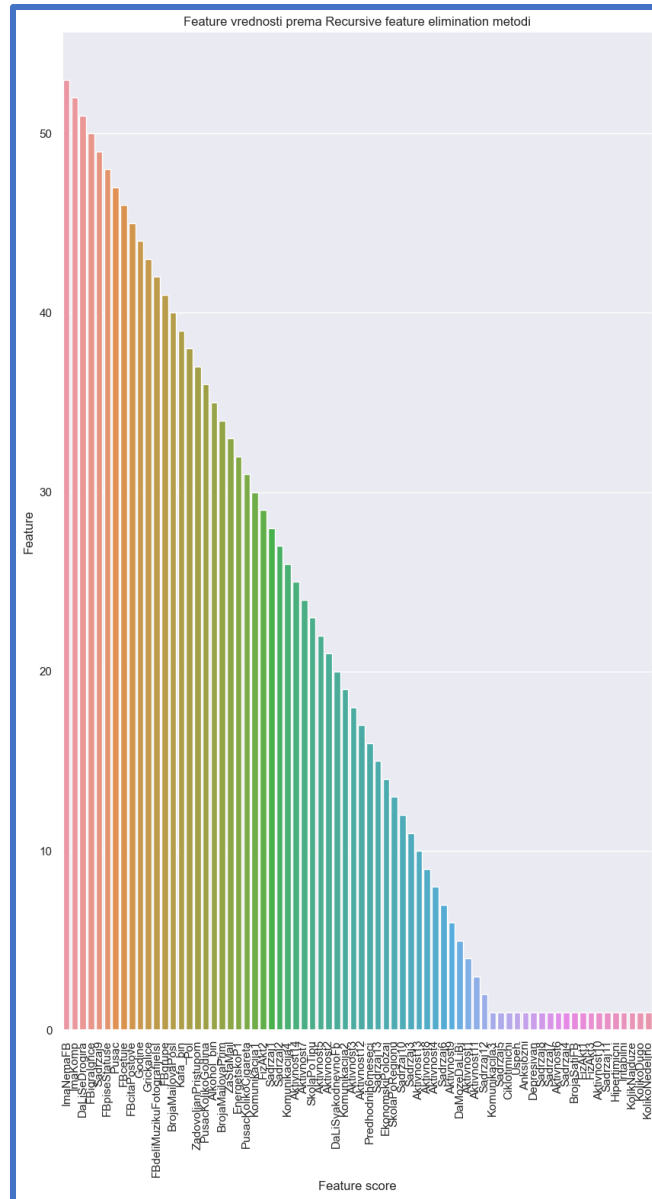
Primena Wrapper modela se može posmatrati kroz dve faze. Prva faza obuhvata proces odabira atributa pomoću *accuracy* mere za dati klasifikator, dok se u drugoj fazi vrši proces učenja i testiranja, što se poklapa sa drugom fazom Filter pristupa.

Funkcionisanje Wrapper metoda se može ogledati kroz *greedy* i *non-greedy* pristupe. Halapljiva pretraga podrazumeva biranje pravca pretrage u zavisnosti od toga koji pravac se čini najpogodnijim u datom trenutku što može rezultovati time da se pronađu lokalno, a ne globalno najbolja rešenja. Sa druge strane, non-greedy algoritmi ne bivaju zarobljeni u lokalnom optimumu jer mogu da promene pravac pretrage nakon ponovne evaluacije atributa. Kao tipičan predstavnik halapljivog algoritma se često pominje *Rekurzivna selekcija atributa* (eng. *Recursive Feature Selection*), dok u nehalapljive spadaju *Genetski algoritmi* [4].

Na slikama 22, 23 i 24 su dati rezultati procesa izbora atributa dobijeni primenom *Genetskog algoritma*, kao i *Recursive Feature Selection*-a (rekurzivni odabir atributa), respektivno.

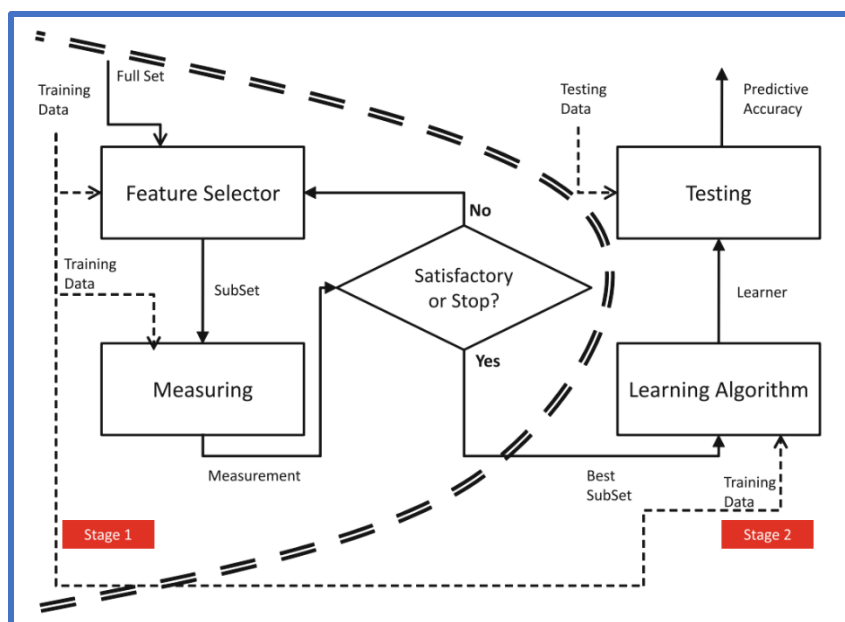
	feature	is_used			
0	Aktivnost1	True	41	Kafa_bin	True
1	Aktivnost10	True	42	KolikNajduze	True
2	Aktivnost11	True	43	KolikoDugo	False
3	Aktivnost12	False	44	KolikoNedeljno	True
4	Aktivnost13	False	45	Komunikacija1	False
5	Aktivnost14	True	46	Komunikacija2	True
6	Aktivnost2	False	47	Komunikacija3	True
7	Aktivnost3	True	48	Komunikacija4	True
8	Aktivnost4	False	49	Pol	True
9	Aktivnost5	False	50	Predhodnih6meseci	False
10	Aktivnost6	True	51	Pusac	True
11	Aktivnost7	False	52	PusacKolikoCigareta	True
12	Aktivnost8	True	53	PusacKolikoGodina	True
13	Aktivnost9	False	54	Sadrzaj1	True
14	Alkohol_bin	True	55	Sadrzaj10	False
15	Anksiozni	True	56	Sadrzaj11	True
16	BrojaMailovaPosl	True	57	Sadrzaj12	True
17	BrojaMailovaPrim	True	58	Sadrzaj13	True
18	BrojaSatiFB	True	59	Sadrzaj2	False
19	Ciklotimicni	True	60	Sadrzaj3	True
20	DaliSeDrogira	True	61	Sadrzaj4	False
21	DaliSvakodneвноFb	True	62	Sadrzaj5	False
22	DaMozeDaliBi	False	63	Sadrzaj6	True
23	Depresivan	False	64	Sadrzaj7	True
24	EkonomskiPolozaj	False	65	Sadrzaj8	False
25	EnergetskoP1	False	66	Sadrzaj9	False
26	FBcetuje	True	67	SkolaPoRegionu	True
27	FBcitaPostove	False	68	SkolaPoTipu	False
28	FBdeliMuzikuFotografijeIsl	True	69	Uspeh	True
29	FBgrupe	False	70	ZaStaMail	True
30	FBgrupaIgrice	True	71	ZadovoljanPristupom	True
31	FBpiseStatuse	False			
32	FizAkt1	True			
33	FizAkt2	True			
34	FizAkt3	True			
35	Godine	True			
36	Grickalice	False			
37	Hipertimicni	False			
38	ImaKomp	True			
39	ImaNemaFB	True			
40	Iritabilni	True			

Slike 22 i 23 – Prikaz rezutata dobijenih primenom genetskog algoritma



Slika 24 – Primena Rekurzivne eliminacije atributa

Negativna strana primene Wrapper pristupa se ogleda kroz rad sa velikim količinama podataka budući da se javlja problem načina na koji će se model izboriti sa svim tim podacima [2]. Sam proces načina funkcionisanja Wrapper pristupa se može videti na slici 25.



Slika 25 – Proces izbora atributa primenom Wrapper metoda

6. Određivanje težine atributa

Kada se radi o postupku odabira najrelevantnijih atributa, može se pribeći i izračunavanju težine atributa nasuprot metodama za njihovu eliminaciju. Atributima koji su relevantniji se dodeljuje veća težina, dok se onim manje relevantnijim vrši dodela nižih vrednosti.

Uglavnom je potrebno imati odgovarajuće domensko znanje kako bi metrici težine bio dodeljen smisao. Sa druge strane, moguće je i vršiti i automatsku dodelu težina, kao na primer u okviru nekih modela poput *Support Vector Machine-a*. U tom slučaju, atributi sa većom težinom imaju veći značaj u modelu [3].

Postupak određivanja težine atributa se može izvesti primenom *Relief* algoritma koji ima za cilj da odabere attribute koji su statistički najrelevantniji. U okviru Relief algoritma se ne vrši eksplicitno kreiranje podskupova atributa kao u prethodno pomenutim metodama.

Funkcionisanje se ogleda u semplovanju odnosno odmeravanju instanci pri čemu je ideja da su relevantni atributi oni na osnovu čijih vrednosti je moguće razlikovati instance koje su bliske jedna drugoj. Za svaku instancu I se vrši traženje dva najbliža suseda (koja pripadaju različitim klasama ukoliko se radi o problemu binarne klasifikacije). Jedan sused je definisan kao “bliski pogodak” (eng. near hit) H , a drugi kao “bliski promašaj” (eng. near miss) J .

Atribut će se smatrati relevantnim ukoliko su njegove vrednosti iste između I i H , a različite između I i J . Korisnim atributima se smatraju oni čija težina prelazi određenu, prethodno definisanu granicu koja se može statistički odrediti.

Glavnom prednošću algoritma se smatra činjenica da je moguće raditi i sa diskretnim i kontinualnim podacima, ali je mana mogućnost rada samo sa problemima koji obuhvataju domen binarne klasifikacije. Na slici 26 se nalazi pseudokod Relief algoritma.

Algorithm 11 Relief algorithm.

```
function RELIEF( $x$  - features,  $m$  - number of instances sampled,  $\tau$  - relevance threshold)
  initialize:  $w = 0$ 
  for  $i = 1$  to  $m$  do
    randomly select an instance  $I$ 
    find nearest-hit  $H$  and nearest-miss  $J$ 
    for  $j = 1$  to  $M$  do
       $w(j) = w(j) - dist(j, I, H)^2/m + dist(j, I, J)^2/m$   $\triangleright dist$  is a distance function
    end for
  end for
  return  $w$  greater than  $\tau$ 
end function
```

Slika 26 – Pseudokod Relief algoritma

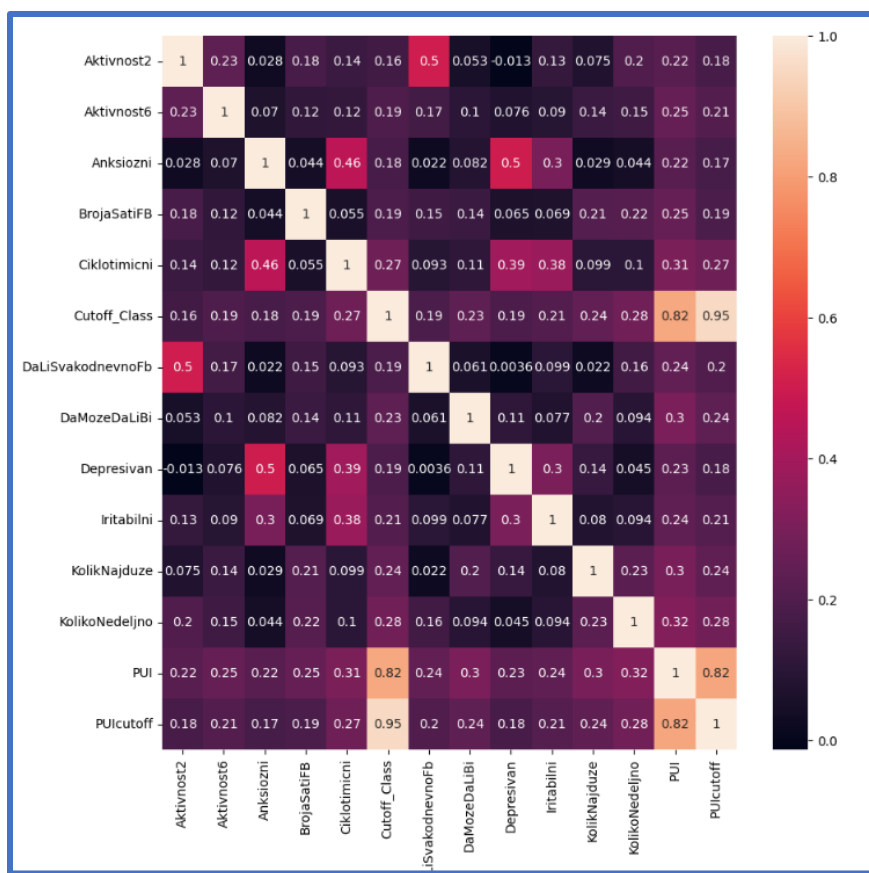
7. Statističke metode za selekciju atributa na osnovu tipa podataka

Kada je reč o upotrebi statističkih metoda za proces selekcije atributa, mora se pre svega voditi računa o tipovima podataka nad kojima se vrši obrada. Tipovi podataka nad kojima su u okviru datog rada demonstrirane različite tehnike za selekciju atributa se mogu podeliti na dve velike grupe, odnosno na numeričke i podatke kategoričkog tipa.

Numerički podaci se dalje mogu podeliti na celobrojne podatke (eng. integer), podatke sa pokretnim zarezom (eng. floating point), dok se kategorički mogu podeliti pre svega na ordinalne i nominalne.

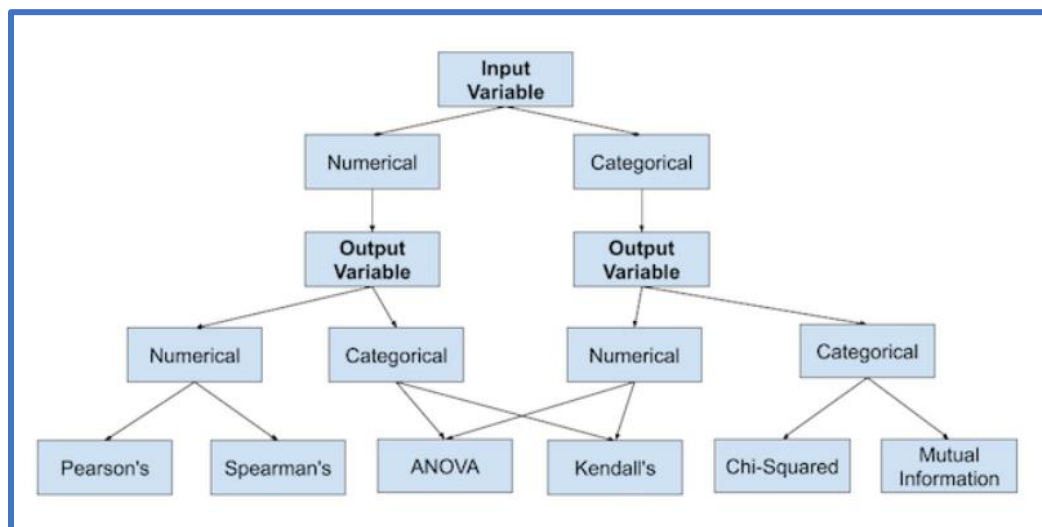
Odabir odgovarajuće statističke metode osim tipa podataka, zavisi i od toga da li je reč o ulaznim podacima (eng. input) ili izlaznim (eng. output). Ulaznim podacima se smatraju oni koji predstavljaju ulaze u model pomoću kojeg se vrši treniranje, dok su izazni oni čiju je vrednost potrebno predvideti (drugačije poznati pod nazivom target varijabla).

Kada je reč o statističkim metodama, korisno je pre svega posmatrati matricu korelacije. Na slici 27 se nalazi matrica korelacije na osnovu koje se može uočiti stepen korelacije između ulaznih atributa i target klase.



Slika 27 – Prikaz matrice korelacije za izabrani set podataka

Ukoliko je izlazna promenljiva numeričkog tipa (ne uzimajući u obzir podatke koji su prethodno kodirani na vrednosti 0 i 1), radi se o problemu regresije, dok je izlaz kategoričkog tipa rezervisan za domen klasifikacije [5]. Na slici 28 se može videti jedan od načina za podelu statističkih metoda za selekciju atributa, predložen u literaturi [5].



Slika 28 – Podela statističkih metoda prema tipu korišćenih podataka

Na osnovu slike 28 se može zaključiti da se može vršiti posmatranje četiri različite grupe, pri čemu je podela izvršena na osnovu tipa podataka, ali i toga da li se radi o ulaznim ili izlaznim podacima [5]. Grupe dobijene na ovaj način su:

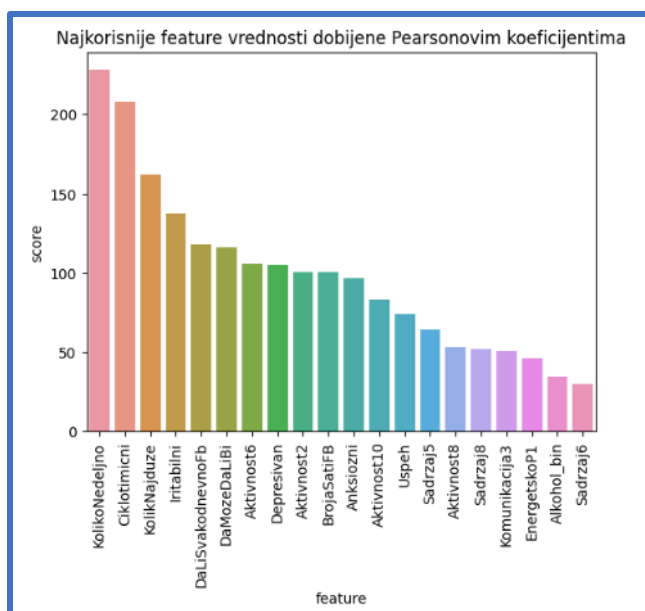
1. Numerički ulaz i numerički izlaz
2. Numerički ulaz i kategorički izlaz
3. Kategorički ulaz i kategorički izlaz
4. Kategorički ulaz i numerički izlaz

7.1. Numerički ulaz i numerički izlaz

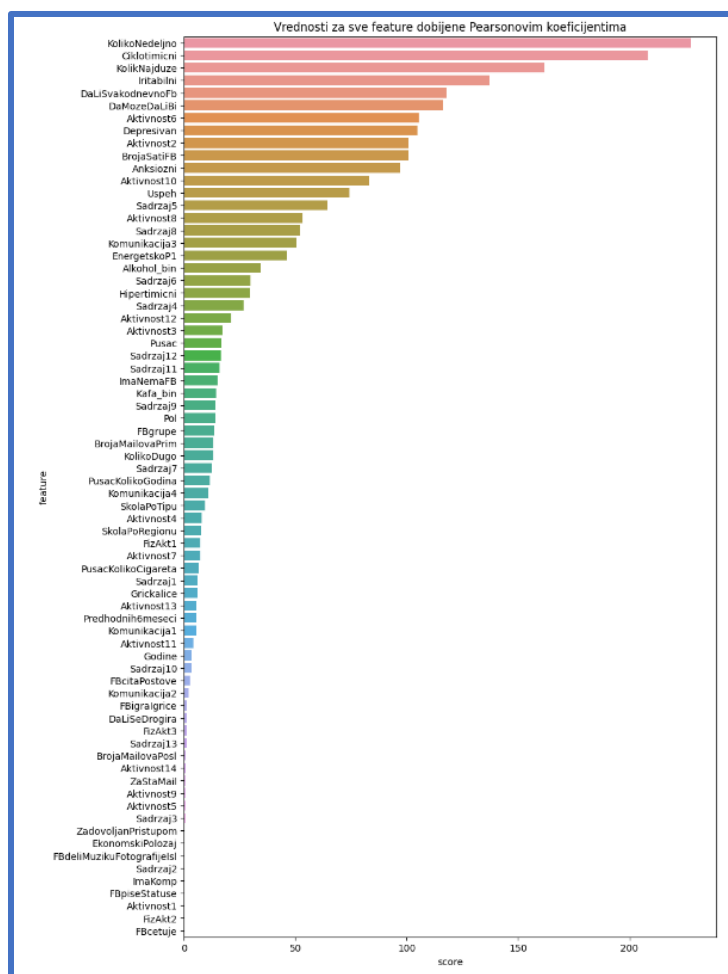
Kada je reč o podacima čiji su i ulaz i izlaz numeričkog tipa, radi se o rešavanju problema iz domena logističke regresije. U tom slučaju se prema literaturi [5] preporučuje korišćenje koeficijenata korelacije:

- *Pearsonov koeficijent* (eng. Pearson), ukoliko je reč o lineranoj korelaciji,
- *Spirmanov koeficijent rangiranja*, ukoliko je reč o nelinearnoj korelaciji

Budući da je u izabranom setu podataka situacija takva da su i ulaz i izlaz numeričkog tipa (izlaz je kodiran sa 0 i 1 u zavisnosti od toga da li je instanca klasifikovana kao nezavisna ili zavisna od Interneta, respektivno). Na slici 29 se nalazi prikaz najkorisnijih atributa izabranih metodom Pearsonovog koeficijenta, dok je na slici 30 prikaz vrednosti za sve attribute iz seta podataka.



Slika 29 – Prikaz najkorisnijih atributa dobijenih metodom Pearsonovih koeficijenata



Slika 30 – Prikaz vrednosti za sve atribute metodom Pearsonovih koeficijenata

7.2. Numerički ulaz i kategorički izlaz

Podaci čiji je ulaz numeričkoj tipa, dok je izlaz kategorički pripadaju domenu rešavanja problema vezanih za klasifikaciju. U tom slučaju se mogu koristiti sledeće tehnike, take bazirane na principima korelacije. Potrebno je voditi računa o tome da li se radi o lineranog ili nelinearnoj korelaciji, stoga se podela može izvršiti na sledeći način:

- *ANOVA koeficijent korelacije*, u slučaju rada sa lineranom korelacijom
- *Kendalov (eng. Kendall) koeficijent ranga*, u slučaju rada sa nelinearnom korelacijom

Prilikom korišćenja Kendalovog koeficijenta ranga, bitno je voditi računa o tome da je kategorički izlaz ordinalnog tipa [5].

7.3. Kategorički ulaz i kategorički izlaz

Ukoliko su i ulaz i izlaz kategoričkog tipa, takođe se radi o domenu koji je vezan za rešavanje problema klasifikacije. Najčešća korelaciona mera koja se koristi kada je reč o ovom slučaju je Hi-Kvadrat (eng. Chi-Squared). Moguće je i korišćenje metoda dostupnih u okviru teorije informacija kao što je Mutual Information pristup [5]. Stoga, podela se definiše na sledeći način:

- *Hi-Kvadrat* (eng. Chi-Squared) test
- *Mutual Information* pristup

7.4. Kategorički ulaz i numerički izlaz

Kategorički ulaz i numerički izlaz su karakteristični za domen logističke regresije. Mogu se koristiti iste metode koje su opisane u delu za numerički ulaz i kategorički izlaz, podrazumevajući suprotan redosled pristupa prilikom korišćenja (ANOVA koeficijent korelacije i Kendalov koeficijent ranga).

8. Interpretabilnost izbora atributa korišćenjem Explainable AI alata

Ubrzani razvoj veštačke inteligencije i mašinskog učenja u prethodnim godinama je rezultirao njihovom sve većom upotrebom u domenu nauke, istraživačkog rada, ali i primene u okviru raznih grana industrije.

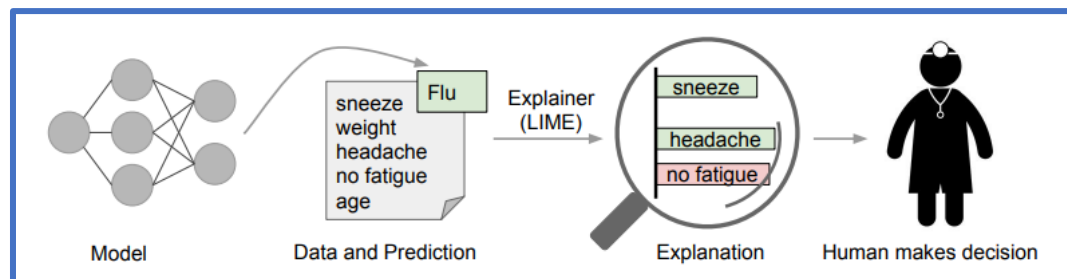
Potencijalna opasnost i neizvesnost prilikom primene se ogleda u tome što način funkcionisanja većine modela nije u potpunosti jasan, već se smatra da rade po principu crne kutije (eng. Black box). Drugačije rečeno, vrši se unos podataka, a nakon toga se sam proces treniranja apstrahuje od korisnika, pri čemu se nakon završetka obrade korisniku prikazu dobijeni rezultati.

Prednost takvog pristupa je pre svega jednostavnost primene, dok se kao negativna strana može navesti nepotpuno razumevanje načina na koji model funkcioniše, donosi odluke, ali i nepostojanje informacija o tome na koji način izbor atributa doprinosi dobijenim rezultatima.

Kako bi ti problemi bili prevaziđeni, predloženi način je razvoj alata koji se bave interpretabilnošću i objašnjivošću (eng. Explainable AI – XAI) rezultata dobijenih prilikom primene algoritama mašinskog učenja. Naročit značaj pomenutih alata se ogleda pre svega u domenima gde se vrši procesiranje i analiza osetljivih podataka, poput rada sa medicinskim podacima. Upotreba Explainable AI alata u okviru medicinskom domena je detaljno prikazana u radovima [6], [7], [8] i [9].

U okviru Explainable AI domena je predložen veliki broj implementacionih rešenja, međutim kao rešenja sa najvećom primenom su se izdvojile biblioteke LIME (Local Interpretable Model-Agnostic Explanations), SHAP (Shapley Additive explanations) i ELI5.

Korišćenjem prethodno pomenutih biblioteka se može vršiti interpretacija i vizuelizacija rezultata čak i na nivou pojedinačnih instanci iz seta podataka. Za svaki od atributa korišćenih prilikom procesa treniranja se može videti koliki značaj je imao prilikom donošenja predikcije, ali i da li je reč o pozitivno ili negativno korelisanim vrednostima. Na slici 31 se može videti generalni koncept logike koja stoji iza upotrebe Explainable AI alata.



Slika 31 – Logika iza upotrebe Explainable AI alata

U okviru Explainable AI oblasti postoji veliki broj istraživačkih radova koji se bave prednostima i manama različitih biblioteka i detaljnim upoređivanjem dobijenih rezultata. Međutim, fokus ovog poglavlja nije na komparaciji svih postojećih rešenja, već na prikazu načina na koji se može vršiti adekvatna interpretacija, ali i implementacija pomoću prethodno navedenih biblioteka. Stoga će praktičan primer realizacije biti prikazani korišćenjem LIME [10], SHAP [11] i ELI5 biblioteka.

8.1. LIME (Local-Interpretable Model-Agnostic Explanations) biblioteka

LIME (Local-Interpretable Model-Agnostic Explanations) se može definisati kao Explainable AI alat koji ima mogućnost da odredi predikcije bilo kog klasifikatora ili regresora, korišćenjem globalne i lokalne perspektive.

Bitna odlika LIME-a je pre svega *interpretabilnost*, odnosno mogućnost određivanja kvalitivne mere za razumevanje odnosa između ulaznih atributa i izlaza. Ukoliko postoje stotine ili hiljade atributa na osnovu kojih je potrebno vršiti predikciju, logično je tvrditi da je u tom slučaju mogućnost interpretabilnosti jako niska. Potrebno je naglasiti i da je interpretabilnost mera koja dosta zavisi od ciljane korisničke grupe.

Pored interpretabilnosti je potrebno definisati i *lokalnu vernost* (local fidelity), koja se ogleda u tome da objašnjenje (eng. prediction) mora odgovarati onome kako se model ponaša u blizini instance koja se predviđa, odnosno koju je potrebno objasniti. Lokalna vernost ne podrazumeva globalnu vernost tj atributi koji su globalno važni mogu biti nerelevantni u lokalnom kontekstu i obrnuto [10].

Još jedna bitna karakteristika LIME-a je mogućnost pružanja objašnjenja za bilo koji modela, odnosno drugačije rečeno *model-agnostički* pristup. Pristup tog tipa omogućava upotrebu modela koji su ručno implementirani i podešeni prema specifičnim namenama.

Način na koji LIME vrši izračunavanje može biti prikazan putem sledeće matematičke formule date na slici 32:

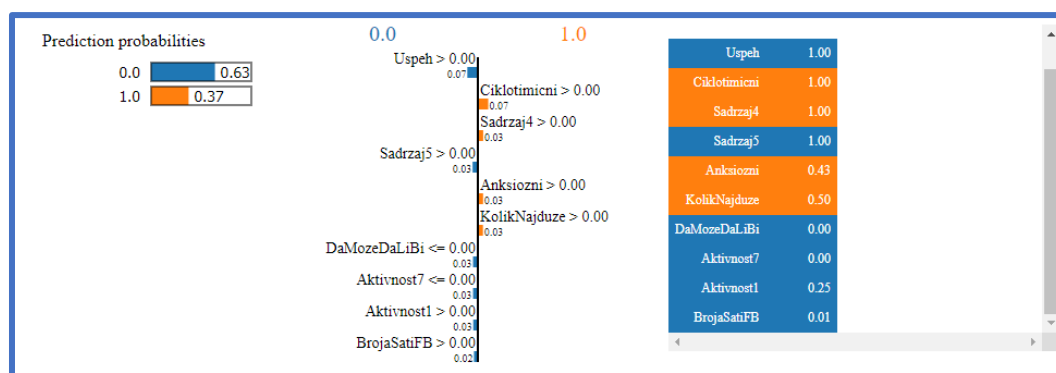
$$\xi(x) = \operatorname{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Slika 32 – Matematička interpretacije načina funkcionisanja LIME-a

Što se tiče notacije vezane za prethodno navedenu formulu, g predstavlja objašnjenje koje će biti posmatrano kao model, dok je G klasa potencijalno interpretabilnih modela. Budući da nije svako g dovoljno jednostavno za interpretaciju, uvodi se $\Omega(g)$ koje predstavlja meru kompleksnosti (na primer za stabla odluke to može biti dubina stabla).

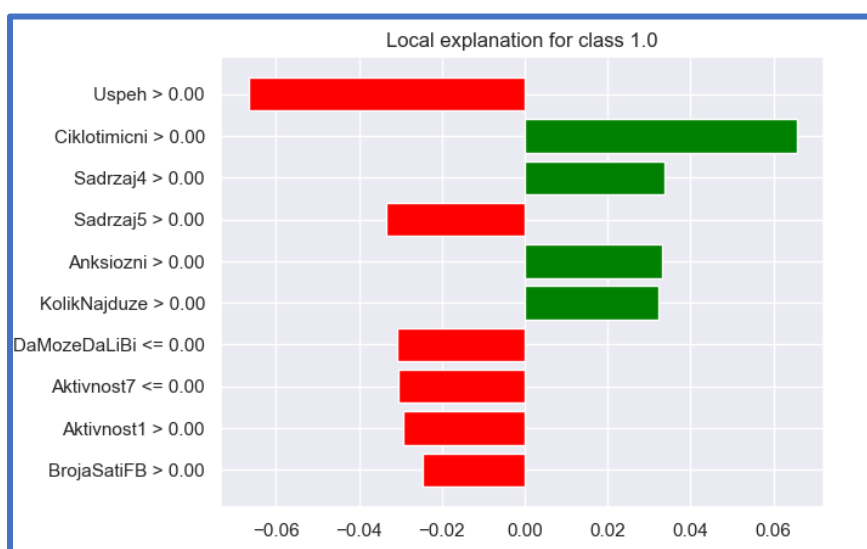
Funkcija $f(x)$ će u slučaju klasifikacije označavati verovatnoću da x pripada datoj klasi. Kao mera udaljenosti instance x od z se koristi $\pi_x(z)$. Sa $L(f, g, \pi_x)$ je označena mera koja označava koliko g greši u aproksimaciji f u lokalitetu definisanom od strane π_x [10]. Kako bile omogućene i interpretabilnost i lokalna vernost, tendencija je na minimizaciji $L(f, g, \pi_x)$, dok $\Omega(g)$ i dalje ima dovoljno nisku vrednost za interpretaciju od strane ljudi.

Budući da je pomoću LIME biblioteke moguće interpretirati i predikcije na nivou pojedinačnih instanci, u okviru praktične implementacije je dato objašnjenje za instancu sa indeksom 20. Na slici 33 su prikazani rezultati izvršenja *explain_instance* funkcije dostupne u okviru LIME biblioteke. Narandžasta i plava boja na slici pokazuju pozitivne i negativne korelacije, respektivno.



Slika 33 – Prikaz objašnjenja za pojedinačnu instancu primenom *expain_instance* funkcije

Drugačiji način vizuelizacije, dobijen primenom *as_pyplot_figure()* funkcije je prikazan na slici 34. Potrebno je naglasiti da je rađena vizuelizacija za istu instancu kao u predthodnom primeru.



Slika 34 – Prikaz objašnjenja za pojedinačnu instancu primenom *as_pyplot_figure()* fukcije

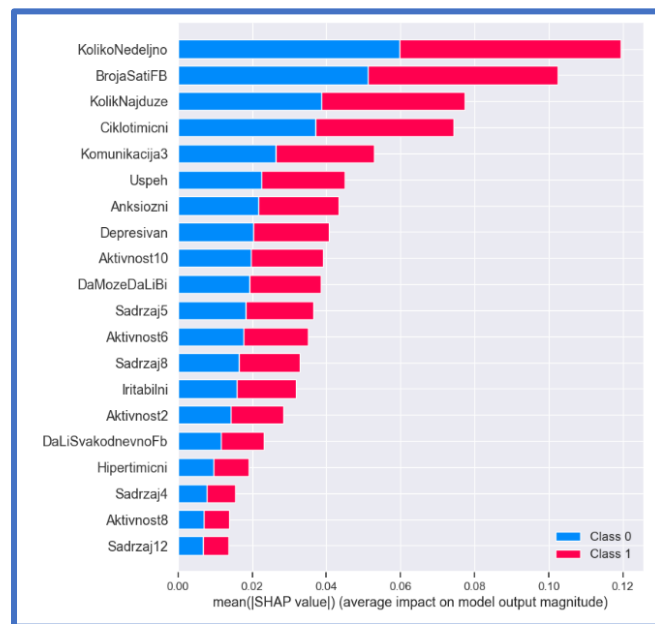
8.2. SHAP (Shapley Additive Explanations) biblioteka

SHAP (Shapley Additive Explanations) je biblioteka koja ima za cilj da prikaže predikciju za pojedinačnu instancu na način da vrši izračunavanje doprinosa svakog atributa prilikom donošenja odluke za predikciju vrednosti [12].

Shapley vrednosti se koriste da odrede efekte koje pojedinačni atributi imaju na određivanje output-a za dati model mašinskog učenja, pri čemu se vrši dodela vrednosti, odnosno definisanje težine za svaki atribut u datom setu podataka.

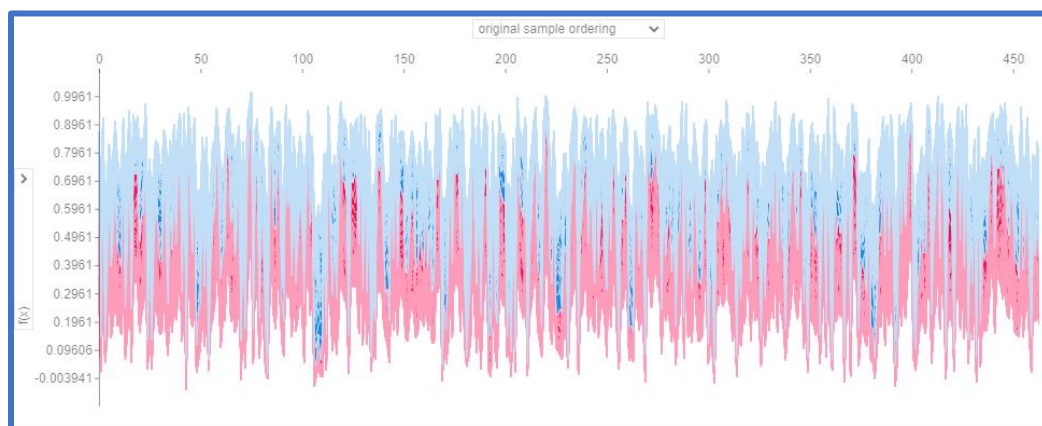
Budući da se u obzir uzimaju sve moguće kombinacije ulaznih vrednosti, SHAP garantuje konzistentnost i lokalnu tačnost. Postoje dve varijante biblioteke, *KernelSHAP* i *TreeSHAP* pri čemu je Kernel SHAP model agnostička varijanta [12]. Kao glavni nedostatak primene SHAP biblioteke se navodi relativno sporo izvršenje u odnosu na ostale biblioteke iz Explainable AI domena.

Na slici 35 je dat praktičan primer primene SHAP biblioteke uz korišćenje *summary_plot* funkcije pomoću koje je moguće videti na koji način pojedinačni atributi doprinose dobijenom izlazu iz odabranog modela mašinskog učenja (u ovom slučaju je korišćen Random Forest model).



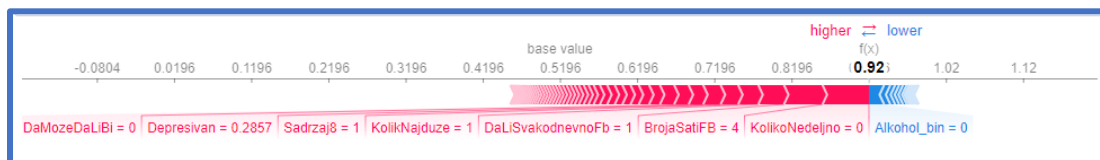
Slika 35 – Prikaz objašnjenja doprinosa izlazu modela mašinskog učenja za svaki atribut

Drugačiji vid vizuelizacije je moguće postignuti korišćenjem *force_plot* funkcije pri čemu se dobijeni rezultati mogu videti na slici 36. Dobijena vizuelizacija je interaktivna, stoga je moguće videti vrednosti za svaku od instanci koje se nalaze u okviru seta podataka.



Slika 36 – Prikaz dobijen korišćenjem *force_plot* funkcije

Kao i u slučaju LIME biblioteke, SHAP omogućava određivanje i pregled predikcije na nivou pojedinačnih instanci. U te svrhe je potrebno koristiti funkciju *shap.force_plot* pri čemu je *shap_values* funkciji prethodno prosleđena istanca od interesa (u datom primeru korišćena je instanca sa indeskrom 20). Na slici 37 je moguće videti rezultate za vrednosti atributa u slučaju predikcije za pojedinačnu instancu.



Slika 37 – Prikaz dobijen korišćenjem *force_plot* funkcije nad jednom instancom

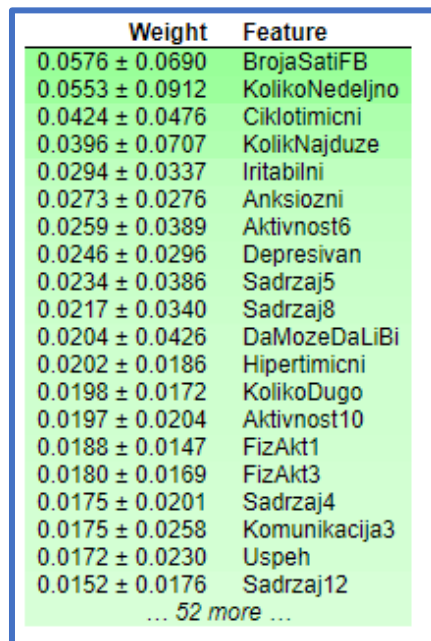
Crvenom bojom su označene one vrednosti koje verovatnoću podižu ka jedinici, dok su plavom prikazane one koje smanjuju vrednost, odnosno pomeraju je dalje od jedinice. Za odabranu instancu je dobijena vrednost 0.92 što bi značilo da je prilično velika verovatnoća da je vrednost target atributa 1. Na ekvivalentan način je moguće dobiti predikcije i za ostale instance.

8.3. ELI5 biblioteka

Eli5(Explain like im 5) je open-source Python biblioteka koja omogućava korisnicima pregled objašnjenja predikcija dobijenih odabranim modelima mašinskog učenja. Koristi se u svrhe debugiranja modela mašinskog, ali i dubokog učenja, pri čemu je upotreba ograničena na linearne i tree-based modele [12].

Najkorišćenije funkcije u okviru eli5 biblioteke su *eli5.show_weights()*, čija je namena inspekcija parametara datog modela, dok *eli5.show_prediction()* ima za cilj da otkrije zbog čega je model odredio takvu predikciju za odabranu instancu.

Na slici 38 je moguće videti primenu show_weights funkcije, što u domenu eli5 modela označava prikaz globalne interpretabilnosti modela.



Weight	Feature
0.0576 ± 0.0690	BrojaSatiFB
0.0553 ± 0.0912	KolikoNedeljno
0.0424 ± 0.0476	Ciklotimicni
0.0396 ± 0.0707	KolikNajduze
0.0294 ± 0.0337	Iritabilni
0.0273 ± 0.0276	Anksiozni
0.0259 ± 0.0389	Aktivnost6
0.0246 ± 0.0296	Depresivan
0.0234 ± 0.0386	Sadrzaj5
0.0217 ± 0.0340	Sadrzaj8
0.0204 ± 0.0426	DaMozeDaLiBi
0.0202 ± 0.0186	Hipertimicni
0.0198 ± 0.0172	KolikoDugo
0.0197 ± 0.0204	Aktivnost10
0.0188 ± 0.0147	FizAkt1
0.0180 ± 0.0169	FizAkt3
0.0175 ± 0.0201	Sadrzaj4
0.0175 ± 0.0258	Komunikacija3
0.0172 ± 0.0230	Uspeh
0.0152 ± 0.0176	Sadrzaj12
... 52 more ...	

Slika 38 – Prikaz rezultata dobijenih primenom show_weights funkcije iz eli5 biblioteke

Prikaz globalne interpretabilnosti je moguće posmatrati i nakon primene **PermutationImportance-a**. Permutation importance (srp.važnost permutacije) omogućava način za računanje važnosti atributa za bilo koji black-box estimator tako što se meri koliko se score smanjuje kada atribut nije dostupan. Kao score može biti korišćena tačnost (eng. accuracy), f1 mera ili neka druga metrika po izboru.

Kao što je prethodno napomenuto, ideja je da se prati promena score-a kada atribut od interesa nije prisutan. To bi podrazumevalo eliminisanje atributa iz skupa atributa koji se koriste i ponovno treniranje modela nad novim skupom atributa.

Međutim, očigledno je da bi takav pristup bio jako zahtevan što se tiče resursa, pogotovo u slučajevima kada set podataka poseduje veliki broj ulaznih kolona, odnosno atributa. Da bi ponovno treniranje modela bilo izbegnuto, atribut se može eliminisati samo iz test dela seta podataka.

Budući da bi izbacivanje kolone dovelo do drugih problema, predloženo rešenje je da se umesto toga zamene vrednosti nekim šumom koji ne nosi korisne informacije. U tom slučaju je potrebno voditi računa o tome da korišćeni šum ima istu distribuciju vrednosti kao originalne vrednosti atributa.

Najjednostavniji način postizanja takve funkcionalnosti je da se za šum izaberu promešane (eng. shuffled) vrednosti atributa, stoga odatle ideja iza upotrebe Permutation Importance funkcije odnosno wrapper-a. Na slici 39 je moguće videti rezultate dobijene primenom prethodno pomenute funkcije.

Weight	Feature
0.0129 ± 0.0028	BrojaSatiFB
0.0093 ± 0.0022	KolikoNedeljno
0.0053 ± 0.0020	Ciklotimicni
0.0017 ± 0.0008	DaMozeDaLiBi
0.0009 ± 0.0011	Iritabilni
0.0008 ± 0.0005	Sadrzaj8
0.0006 ± 0.0008	Depresivan
0.0004 ± 0.0004	Sadrzaj5
0.0003 ± 0.0005	Aktivnost6
0.0003 ± 0.0005	Aktivnost10
0.0003 ± 0.0013	Anksiozni
0.0002 ± 0.0005	KolikNajduze
0.0002 ± 0.0005	Alkohol_bin
0 ± 0.0000	EnergetskoP1
0 ± 0.0000	FBpiseStatuse
0 ± 0.0000	FBigralgrice
0 ± 0.0000	FBcetuje
0 ± 0.0000	FBdeliMuzikuFotografijesi
0 ± 0.0000	FBcitaPostove
0 ± 0.0000	EkonomskiPolozaj
... 52 more ...	

Slika 39 – Prikaz rezultata nakon upotrebe Permutation Importance wrapper-a

Potrebno je naglasiti da je Permutation Importance poželjno koristiti uglavnom u slučajevima kada broj atributa izabranog seta podataka nije preveliki, budući da je sam proces izračunavanja zahtevan što se tiče računarskih resursa.

Nalik LIME i SHAP bibliotekama, ELI5 takođe ima mogućnost prikaza dobijene predikcije na nivou pojedinačne instance. U te svhre je potrebno koristiti *show_prediction* funkciju. Na slici 40 je moguće videti dobijene rezultate za instancu sa brojem indeksa 20. Pozitivne korelacije su prikazane zelenom bojom, dok su one sa negativnim vrednostima obojene crvenom.

Contribution?	Feature	Value
+0.496	<BIAS>	1.000
+0.057	KolikoNedeljno	1.000
+0.056	DaMozeDaLiBi	0.848
+0.049	KolikNajduze	0.576
+0.028	Ciklotimicni	0.606
+0.024	BrojaSatiFB	0.125
+0.024	Alkohol_bin	0.848
+0.018	Sadrzaj8	0.924
+0.015	Uspeh	0.616
+0.015	Predhodnih6meseci	0.848
+0.013	Hipertimicni	0.682
+0.013	Aktivnost10	1.000
+0.013	Iritabilni	0.736
+0.011	BrojaMailovaPrim	0.152
+0.011	Aktivnost7	0.038
+0.010	Komunikacija3	1.000
+0.010	Aktivnost14	0.076
+0.010	EkonomskiPolozaj	0.712
+0.009	Sadrzaj5	0.250
+0.009	Aktivnost13	0.326
+0.009	Sadrzaj2	0.212
+0.007	Aktivnost8	0.962
+0.006	Sadrzaj6	0.326
+0.006	Aktivnost3	0.288
+0.006	Komunikacija4	0.924
+0.006	FBpiseStatuse	0.000
+0.005	Aktivnost5	0.538
+0.005	SkolaPoRegionu	0.250
+0.005	FBdeliMuzikuFotografijel	0.000
+0.004	Anksiozni	0.627
+0.004	Sadrzaj10	0.326
+0.004	Sadrzaj3	0.788
+0.004	FBcitaPostove	0.000
+0.003	Aktivnost6	0.326
+0.003	Sadrzaj4	0.288
+0.003	Sadrzaj7	0.288
+0.003	PusacKolikoGodina	0.000
+0.003	Aktivnost9	0.538
+0.003	Aktivnost2	1.000
+0.003	Pol	0.000
+0.002	DaLiSvakodneвноFb	1.000
+0.002	FizAkt1	0.857
+0.002	DaLiSeDrogira	0.000
+0.002	Sadrzaj12	0.462
+0.002	Sadrzaj11	0.500
+0.002	KolikoDugo	0.638
+0.002	PusacKolikoCigareta	0.000
+0.002	Aktivnost12	0.538
+0.001	Aktivnost11	0.114
+0.001	Aktivnost1	0.288
+0.001	Sadrzaj9	1.000
+0.001	Komunikacija1	0.500
+0.001	BrojaMailovaPosl	0.152
+0.001	FBgralgrice	0.000
+0.001	FBgrupe	0.000
+0.001	Kafa_bin	0.000
+0.001	Pusac	0.000
+0.000	ImaKomp	0.000
+0.000	ZaStaMail	0.500
+0.000	ImaNemaFB	1.000
-0.001	Sadrzaj13	0.250
-0.001	Aktivnost4	0.076
-0.002	FizAkt3	0.667
-0.002	SkolaPoTipu	0.750
-0.004	Sadrzaj1	0.000
-0.006	FizAkt2	1.000
-0.007	Godine	0.000
-0.008	Grickalice	0.152
-0.015	EnergetskoP1	0.000
-0.016	Komunikacija2	1.000
-0.036	Depresivan	0.000

Slika 40 – Rezultati predikcije dobijeni primenom show_prediction funkcije iz eli5 biblioteke

9. Zaključak

U okviru domena prikupljanja i predobrade podataka postoji veliki broj tehnika i metoda čiji je cilj da omoguće rad sa što kvalitetnijim podacima, radi dobijanja boljih i preciznijih rezultata. Korak predobrade podataka je neophodan i potrebno je odabrati odgovarajuće pristupe koji pre svega zavise od prirode podataka nad kojima se vrši dalja analiza.

Bitno je napomenuti da upotreba kvalitetnih algoritama sama po sebi neće dovesti do dobrih predikcija i rezultata budući da najčešće ne može da nadomesti rad sa podacima koji prethodno nisu adekvatno preprocesirani i obrađeni. Stoga, potrebno je staviti poseban akcenat na korišćenje i implementaciju neophodnih metoda u okviru domena prikupljanja i predobrade podataka.

Fokus datog seminarskog rada je bio na detaljnom prikazu oblasti izbora atributa (Feature selection), kao jednoj od najčešće korišćenih tehnika za predobradu podataka. Nakon uvodnog dela, najpre je u drugom poglavlju definisan sam pojam kvaliteta podataka, uz navođenje metoda koje se u tu svhu koriste.

U okviru trećeg poglavlja je uveden pojam izbora atributa, pri čemu su prikazane sličnosti i razlike između izbora atributa i srodnih naučnih oblasti. Pored navedenog, dat je i prikaz potencijalne arhitekture za proces izbora atributa.

Fokus četvrtog poglavlja je bio na prikazu različitih načina za proces traženja podskupa atributa, pri čemu je izvršen pregled procesa traženja zasnovan na smeru traženja, na strategiji pretrage, na izboru kriterijuma selekcije, ali je dat i prikaz mera konzistentnosti i mera tačnosti.

U okviru petog poglavlja su obrađeni pristupi za proces izbora atributa pri čemu je najpre dat prikaz seta podataka za demonstraciju samog procesa izbora atributa. Nakog toga je izvršena podela na Embedded, Filter i Wrapper metode, gde su navedeni različiti algoritmi mašinskog učenja pomoću kojih je dat prikaz načina funkcionisanja metoda iz gore navedenih skupova.

Cilj šestog poglavlja je bio na određivanju težine atributa, uz dat pseudokod Relief algoritma koji se upravo u te svrhe i koristi. U okviru sedmog poglavlja je dat pregled statističkih metoda za selekciju atributa na osnovu tipa podataka, dok je poslednje, osmo poglavlje posvećeno interpretabilnosti samog procesa izbora atributa korišćenjem Explainable AI alata. Konkretno, dat je prikaz implementacije pomoću SHAP, LIME i ELI5 biblioteka koje se koriste u cilju interpretabilnosti i objašnjivosti rezultata (eng. Interpretability and explainability) dobijenih primenom algoritama mašinskog učenja.

Na kraju je dat prikaz korišćene literature, dok se praktična implementacija gde su obrađeni primeri za gore navedena poglavlja nalazi na sledećem linku: <https://github.com/minanikolic916/Feature-selection-/tree/master>. Cilj samog rada je bio na prikazu odgovarajućih teorijskih osnova i koncepata vezanih za sam proces izbora atributa, ali i praktične implementacije čija je svrha praćenje teorijskog dela.

10. Literatura

- [1] Jiawei Han, Micheline Kamber and Jian Pei, [Data Mining: Concepts and Techniques](#), Third Edition, 2012, Morgan Kaufmann
- [2] Salvador García , Julián Luengo and Francisco Herrera, Data Preprocessing in Data Mining, 2015, Springer
- [3] P.-N.Tan, M. Steinbach, A. Karpatne, V. Kumar: Introduction to data mining, Addison Wesley, Second edition, 2019.
- [4] Max Kuhn, Kjell Johnson, Feature Engineering and Selection: A practical Approach for Predictive Models, 2019
- [5] Jason Brownlee, Data Preparation for Machine Learning: Data Cleaning, Feature Selection and Data Transforms in Python, 2020, Machine Learning Mastery
- [6] H. W. Loh, C. P. Ooi, S. Seoni, P. D. Barua, F. Molinari, and U. R. Acharya, “Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011-2022),” *Comput. Methods Programs Biomed.*, vol. 226, no. 107161, p. 107161, 2022
- [7] S. Knapič, A. Malhi, R. Saluja, and K. Främling, “Explainable Artificial Intelligence for Human Decision Support System in the Medical Domain,” *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 740–770, Sep. 2021, doi: 10.3390/make3030037.
- [8] J. Duell, X. Fan, B. Burnett, G. Aarts, and S.-M. Zhou, “A comparison of explanations given by explainable artificial intelligence methods on analysing electronic health records,” in 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 1–4, 2021.
- [9] D. Dave, H. Naik, S. Singhal, and P. Patel, “Explainable AI meets healthcare: A study on heart disease dataset,” *arXiv:2011.03195 [cs.LG]*, 2020
- [10] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
- [11] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 4768–4777.
- [12] M. Mehta, V. Palade and I. Chatterje, “Explainable AI: Foundations, Methodologies and Applications”, 2022
- [13] Koriščenje Permutation Importance wrapper-a, Eli5 dokumentacija, dostupna na adresi: [https://eli5.readthedocs.io/en/latest/blackbox/permutation_importance.html#:~:text=eli5%20provides%20a%20way%20to,Decrease%20Accuracy%20\(MDA\)%E2%80%9D](https://eli5.readthedocs.io/en/latest/blackbox/permutation_importance.html#:~:text=eli5%20provides%20a%20way%20to,Decrease%20Accuracy%20(MDA)%E2%80%9D).

