Primena Random Forest algoritma kao podrška unapređenju postignuća studenata

Marija Mojsilović
Akademija strukovnih studija Šumadija
Odsek Trstenik
Trstenik, Srbija
mmojsilovic@asss.edu.rs

Enes Sukić
Univerzitet Nikola Tesla
Fakultet za informacione tehnologije i inženjerstvo
Beograd, Srbija
enes.sukic@fiti.edu.rs

Sažetak— **U radu će biti predstavljena primena metode veštačke inteligencije, odnosno Random Forest algoritma u ispitivanju faktor koji više utiče na savladavanje gradiva studenata na visokoškolskoj ustanovi. Parametar koji se posmatra i koji je krajnji ishod ovog istraživanja, određuje se na osnovu dve nezavisne promenljive, a to su aktivnost, odnosno interakcija na vežbama i predavanjima, kao i broj bodova sakupljen na predispitnim obavezama. Korišćene su dve faze Random Forest algoritma, prva faza podrazumeva generisanje nasumične šume pomoću kombinovanja n stabala odlučivanja. U drugoj fazi se izvode predviđanja za svako stablo stvoreno u prvoj fazi. Rezultat istraživanja pokazuje da broj bodova koje student sakupi tokom semestra na predispitnim obavezama ima veći uticaj na izlazni faktor a to je visoki nivo znanja, ocena 9 i 10.**

Ključne riječi - Veštačka inteligencija; Mašinsko učenje; Random Forest; Nastava; (key words)

I. UVOD

Za uticaj predispitnih obaveza i aktivnosti na nastavi na savladavanje gradiva na visokoškolskoj ustanovi korišćen je Random Forest algoritma. Istraživanje za ovaj rad je sprovedeno na predmetu Uvod u programiranje sa aspekta realizacije nastave. Programiranje je izvršenje instrukcija na računaru koje zadaje programer. Korišćenjem programskog jezika pišu se komande računaru koje računar razume i izvršava. Pored toga što se u programiranju ispisuju kodovi, neophodno je i razumeti zadatak koji treba rešiti. Kod rešavanja problema, potrebno je pronaći najbolje rešenje do samog problema. Kako programiranje ne bi bilo teško, potrebno je stalno usavršavanje. Predmet Uvod u programiranje, na Akademiji strukovnih studija Šumadija Odsek Trstenik ima za cilj da upozna i osposobi studente da koriste napredne tehnike programiranja, na primerima jezika C. Izrada zadataka na vežbama se radi u programu CodeBlocks.

Osnovu programiranja predstavlja metodologija pristupa rešavanju zadataka pomoću računara koja obuhvata analizu problema i definisanje matematičkog modela, izbor metode numeričkog rešavanja, projektovanje algoritma i definisanje strukture podataka i programskog jezika, editovanje programa, testiranje i ispravljanje grešaka i drugo. Takvim pristupom student se osposobljava za uspešno bavljenje programiranjem.

Krajnji ishod predmeta je da studenti znaju da kreiraju algoritam i napišu odgovarajući program, koristeći sintaksu i pravila pisanja programa u programskom jeziku C. Na laboratorijskim vežbama, koje prate tok teorijske nastave, studenti rešavaju programske probleme iz oblasti obrađenih na predavanjima u programskom jeziku C. Aktivnosti koje se obavljaju su postavka problema, izrada dijagrama toka i pisanje programa.

Korišćenjem metode veštačke inteligencije, odnosno, algoritma mašinskog učenja, Random Forest algoritma, dolazi se do detekcije faktora koji najviše utiče na krajnji ishod, odnosno ukupan broj bodova na završnom ispitu studenata.

II. VEŠTAČKA INTELIGENCIJA

Veštačka inteligencija ima mnogo podoblasti, od učenja pa do igranja. Dok je jedna od podoblasti računarstva, upravo veštačka inteligencija, koja računarima, pomoću softvera, omogućava da razmišljaju i razvijaju inteligenciju kao ljudi [1]. Prevashodno se oslanja na analizu podataka i mašinsko učenje, podrazumeva postojanje automatizovanih procedura za predviđanje fenomena, koji su predmet istraživanjana, pri čemu se nastoji da se otkriju osnovni obrasci u analiziranim podacima, odnosno pružanje uvida u problem koji se istražuje [2]. Pomaže naučnicima, inženjerima, lekarima u pokušaju da predvide određene pojave na osnovu zabeleženih iskustava (prikupljenih podataka).

Na primer:

- na osnovu prikupljenih podataka kliničkog lečenja, lekari mogu da ustanove koje su rizične grupe za određene kategorije bolesti,
- inženjeri na osnovu podataka o eksploataciji mašina ili uređaja mogu da predvide otkaze, odnosno da formiraju planove održavanja,
- meteorolozi, na osnovu podataka iz prošlosti, predviđaju prognozu vremena za određeni period [2].

A. *Mašinsko učenje*

Jedna od podoblasti veštačke inteligencije i računarstva je Mašinsko učenje, koristi se za upotrebu podataka i algoritama, koko bi se oponašao način na koji ljudi uče. U današnje vreme, Mašinsko učenje je sve zanačajnije u svetu informacionih

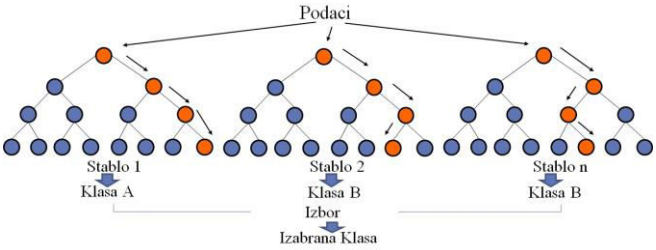
tehnologija. Korišćenjem raznih metoda, algoritmi su obučeni za predviđanje ili klasifikaciju podataka, a posebno kod analize velikog broja podataka.

Za dati problem, se na osnovu generičkih algoritama dobijaju informacije o nekom skupu podataka, bez pisanja programskog koda. Podatke je potrebno uneti u generički algoritam, kako bi on izgradio sopstvenu logiku zasnovanu na unetim podacima. Ne može se reći da jedan algoritam radi najbolje za svaki problem, pored mnogih drugih faktora, ima uticaj i veličina i struktura podataka. Kod rešavanja nekih problema bolje se pokazuju neuronske mreže, dok su kod nekih drugih bolja stabla odlučivanja [3].

III. RANDOM FOREST ALGORITAM

Random Forset je samo jedan od mnogobrojnih algoritama mašinskog učenja, pokazao se kao fleksibilan i jednostavan za korišćenje kod velikog broja problema. Šuma koju algoritam gradi, je skup stabla odlučivanja, koja se obučava metodom pakovanja. Kombinacijom modela učenja povećava se ukupan rezultat [4].

Random Forest ili Random Decision Forest Algorithm je namenjen za rešavanje problema klasifikacije i regresije, koji funkcionišu tako, što se formira mnoštvo stabala odluka u periodu učenja (treninga), slika 1 [4].



Slika 1. Stablo odlučivanja

Neke od osobina algoritma Random Forest, su sledeće:

- premašuje većinu algoritama stabala odlučivanja, u pogledu efikasnosti ali je njihova tačnost niža od stabala sa nagibom.
- može se koristiti za rangiranje važnosti promenljivih u regresionom ili klasifikacionom problemu.
- algoritam prolazi kroz dve faze, u prvoj stvara slučajnu šumu od n broja stabala odlučivanja, dok se u drugom delu generiše predviđanje za svako drvo iz prve faze.

Kombinovanjem više stabala odlučivanja, algoritam Random Forest je u stanju da predvidi klasu skupa podataka. Iako neka stabla pojedinačno možda neće dati tačne rezultate, kombinacija svih stabala odlučivanja vodi do tačnog predviđanja [5].

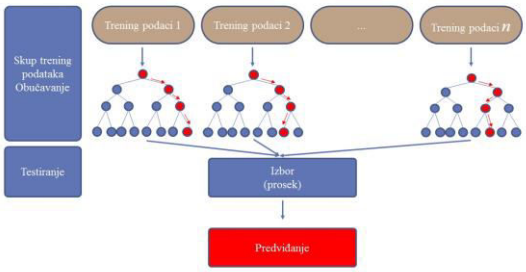
Pretpostavke za dobijanje boljeg klasifikatora:

- Potrebne su stvarne vrednosti skupa promenljivih, kako bi klasifikator predvideo tačne rezultate.
- Predviđanja, za svako stablo, moraju imati veoma nisku korelaciju.

Razlozi za primenu Random Forest algoritma:

- Zahteva manje sati obuke u poređenju sa drugim algoritmima.
- Predviđanje izlaza sa velikom tačnošću i efikasnošću, čak i za veliki skup podataka.
- Održavanje tačnosti, čak i kada nedostaje veliki deo podataka.

Random Forest Algoritam se odvija u dve faze. Prvi korak podrazumeva generisanje nasumične šume pomoću kombinovanja n stabala odlučivanja. U drugoj fazi se izvode predviđanja za svako stablo stvoreno u prvoj fazi, slika 2.



Slika 2. Način rada Random Forest algoritma [6]

Random Forest Algoritam se sastoji od sledećih koraka:

- Podaci iz skupa trening podataka se biraju nasumično.
- Kreiranje stabla odlučivanja uz pomoć izabranog skupa podataka (podskupova).
- Odabir vrednosti n za kreiranje stabala odlučivanja.
- Ponovno izvršavanje koraka 1 i 2.
- Slučajno odabrani novi skup podataka treba da se obradi tako što će se za svako stablo odlučivanja pronaći predviđanje i dodeliti nova tačka podataka [6].

Prednosti Random Forest algoritma:

- Sposobnost da se primeni i za klasifikaciju tipa problema i za regresiju.
- Sposobnost priemene na ogromnom (large – scale) višedimenzionalnom skupu podataka.
- Poboljšava tačnost modela i sprečava problem preteranog preklapanja.

Nedostaci Random Forest algoritma:

- Iako se može koristiti za razne vrste problema, nije pogodan za regresione probleme.
- Povećavanjem broja stabala, algoritam postaje spor i neefikasan u izvođenju scenarija u realnom vremenu.
- Potrebno je više resursa za obradu podataka i računanje.

IV. PRIMENA RANDOM FOREST ALGORITAM

Za primer primene Random Forest algoritma razmatraće se uticaj predispitnih obaveza i akivnosti, odnosno interakcije na nastavi na savladavanje gradiva na visokoškolskoj ustanovi.

Program je razvijen u softverskom paketu MatLab, primenom ugrađenih funkcija koje podržavaju Random Forest algoritam.

A. Opis problema

Cilj ovog istraživanja je proceniti koji faktor više utiče na postignuće studenata na predmetu Uvod u programiranje.

Parametar koji se posmatra je postignuto znanje na završnom ispitu, kodirano sa:

- 0 – slabo, ocena 5.
- 1 – srednji nivo znanja , ocena: 6 – 8.
- 2 – visoki nivo znanja, ocena: 9 – 10.

Ovaj indikator predstavlja pouzdanu vrednost koja zavisi od dve različite nezavisne promenljive koje su predstavljene u ovom istaživanju. Te promenljive su:

- Aktivnost na nastavi – interakcija – predstavlja da li je student zainteresovan i aktivan na nastavi i vežbama, da li učestvuje u izradi zadataka, to je procena profesora u intervalu od 0 do 10
- Predispitne obaveze – predstavljaju broj postignutih bodova tokom semestra, koje student sakupi kroz izradu kolokvijuma, u intervalu od 0 do 70

Podaci su prikupljani tokom dve školske godine, na prvoj godini strukovnih studija koju pohađa 240 studenata. Podaci su prikupljani kao zvanični statistički podaci škole.

Aktivnost na nastavi ima pozitivan uticaj na završnu ocenu odnosno savladavanje gradiva. Aktivno učenje tokom nastave pomaže učenicima da bolje razumeju gradivo. To može učiniti da se osećaju sigurnije pri rešavanju pitanja na ispitu. Učenici koji su aktivni tokom nastave često su motivisaniji za učenje. To može dovesti do veće posvećenosti učenju i boljeg uspeha na ispitu.

Međutim, važno je napomenuti da aktivnost na nastavi nije jedini faktor koji utiče na završnu ocenu na ispitu. Važno je da studenti imaju jasnu predstavu o gradivu i da redovno polažu kolokvijume i rade domaće zadatke što spada pod predispitne obaveze koje je neophodno da student savlada 50% kako bi imao prolaznu ocenu.

Kako predispitne obaveze nisu uslovljene, odnosno predispitne obaveze su nezavisne jedne od drugih, što znači da kroz polaganje kolokvijuma, ne znači da će student podjednako savladati kompletno gradivo.

Predispitne obaveze se obično sastoje od kolokvijuma, seminarskih radova, projekata, domaćih zadataka i drugih aktivnosti koje studenti trebaju da obave tokom semestra kako bi stekli znanje i veštine potrebne za polaganje završnog ispita.

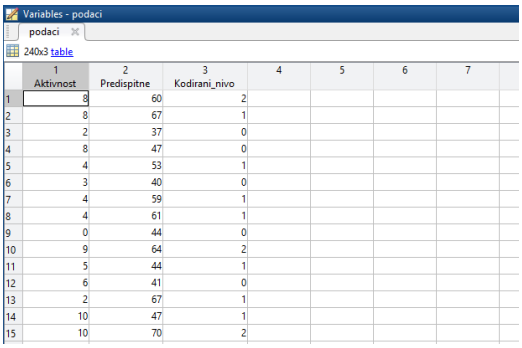
Predispitne obaveze mogu značajno uticati na završni ispit i ocenu tako što, redovno obavljanje predispitnih obaveza pomaže studentima da se bolje pripreme za završni ispit. Kroz rad na projektima, seminarima, testovima i drugim aktivnostima, studenti mogu proveriti svoje znanje i identifikovati oblasti u kojima su slabiji, što im daje vremena da se fokusiraju na te oblasti pre završnog ispita. U koloko studenti usešnije savladaju pređeno gradivo i ostvare bolje rezultate na predispitnim obavezama, to može povećati njihovo samopouzdanje i osećaj sigurnosti u svoje znanje i veštine. To ih može učiniti manje nervoznim i uplašenim tokom završnog ispita, što može poboljšati njihove performanse.

U većini slučajeva, osvojeni broj bodova koje studenti dobijaju na završnom ispitu nije jedini faktor koji utiče na njihovu završnu ocenu. Predispitne obaveze se često uzimaju u obzir prilikom konačnog ocenjivanja, što znači da studenti koji redovno i uspešno obavljaju svoje predispitne obaveze mogu dobiti bolju ocenu na kraju semestra.

Generalni zaključak sproveden u toku istraživanja pokazuje da oni koji su bolje savladali predispitne obaveze, više od 50%, imaju bolji uspeh na završnom ispitu.

B. Rezultati i diskusija

Za analizu su, u tabeli podaci.csv, dati realni podaci o aktivnosti na nastavi (Aktivnost) i ostvarenim predispitnim obavezama (Predispitne) za 240 studenta, slika 3.


Variables - podaci

podaci

240x3 table

	1 Aktivnost	2 Predispitne	3 Kodirani_nivo	4	5	6	7
1	8	60	2				
2	8	67	1				
3	2	37	0				
4	8	47	0				
5	4	53	1				
6	3	40	0				
7	4	59	1				
8	4	61	1				
9	0	44	0				
10	9	64	2				
11	5	44	1				
12	6	41	0				
13	2	67	1				
14	10	47	1				
15	10	70	2				

Slika 3. Podaci o aktivnosti i predispitnim obavezama (Podaci.csv)

Pre svega mora da se izvrši priprema podataka za analizu ovim algoritmom:

```
clc; clear all; close all;
warning off;
%% Formiranje Stabla odluka
podaci=readtable('podaci.csv');
k=["VISOKO","SREDNJE","SLABO"];
l=[2,1,0];
g=podaci.Nivo_Znanja;
brojac=zeros(length(g),1);
for i=1:length(k)
rs=ismember(g,k(i));
brojac(rs)=l(i);
end
podaci.Kodirani_nivo=brojac;
podaci.Nivo_Znanja=[];
```

Promenljiva *podaci*, učitava pripremljene podatke iz tabele podaci.csv. Dok se promenljiva k i l koristi za kodiranje nivoa

znanja. Sledeći korak je slučajna (random) podela vektora, odnosno skupa podataka za unakrsnu proveru, funkcija *cvpartition*:

```
cv=cvpartition(size(podaci,1),"HoldOut",0.3);
```

Parametar "HoldOut", znači da se skup deli na ne – slojevitu particiju za n zapažanja. Ovaj postupak deli podatke na skup za obuku i skup za trening.

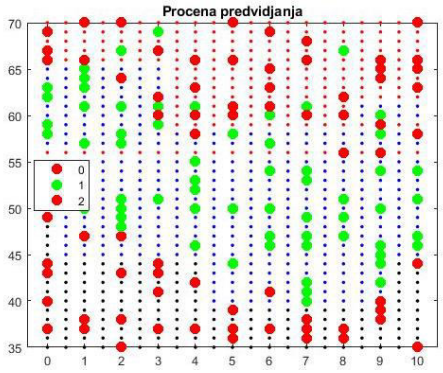
Nakon podele na particije koristi se funkcija za automatsko generisanje stabla *fitctree*, odnosno funkcija za generisanje predviđanja, *predict*:

```
dataTrain=podaci(~idx,:);
dataTest=podaci(idx,:);
testing=dataTest(1:end,1:end-1);
model=fitctree(dataTrain,'Kodirani_nivo');
prediction=predict(model,testing);
```

U kodu koji je prikazan, pored generisanja trening podataka, generisani su i podaci za testiranje. U trećoj liniji nalaze se test podaci, nakon toga se generiše stablo i funkcijom *predict* izvršava se predviđanje. Grafička interpretacija rezultata:

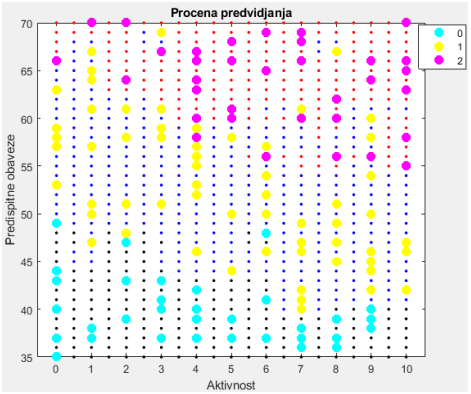
```
ms=(sum(prediction==table2array(dataTest(:,end)))/
size(da   taTest,1))*100
e=min(podaci.Aktivnost):0.5:max(podaci.Aktivnost);
f=min(podaci.Predispitne):1:max(podaci.Predispitne);
[x1 x2]=meshgrid(e,f);
x=[x1(:) x2(:)];
ms=predict(model,x);
figure(1)
gscatter(x1(:), x2(:),ms,'kbr');
hold on
gscatter(dataTrain.Aktivnost,dataTrain.Predispitne,
dataTrain.Kodirani_nivo,'rg','.',30);
view(model,'Mode','graph');
title('Procena predvidjanja');
```

Na dijagramu se uočava da visok nivo znanja, odnosno maksimalan broj bodova na završnm ispitu, postižu oni studenti koji su pokazivali aktivnost na času i imali visok broj bodova na predispitnim obavezama. Međutim, postoje i određena odstupanja u modelu, što je posledica slučajne podele skupa podataka. Ova odstupanja se rešavaju optimizacijom algoritma.



Slika 4. Dijagram procene predviđanja

Ukoliko se ne pristupi optimizaciji, može se koristiti funkcija: *fitensemble*. Prilagođavanje skupa učenika (engl. leaners) za klasifikacione probleme i regresiju. Ovom funkcijom se već dobija realna procena da studenti koji imaju visok broj bodova na predispitnim ocenama i pokazuju aktivnost na nastavi dobijali su visoke ocene 2, slika 5.

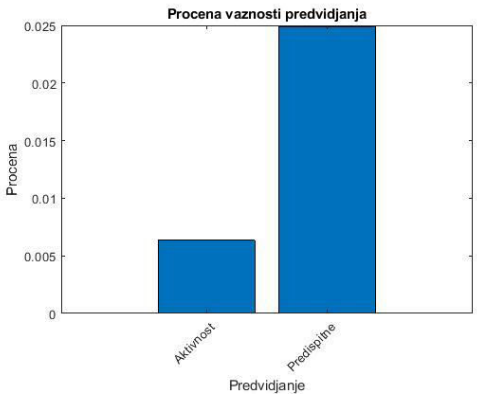


Slika 5. Dijagram procene predviđanja

Pre optimizacije algoritma generiše se dijagram uticaja analiziranih faktora na postignuće studenta:

```
imp = predictorImportance(model);
figure(2)
bar(imp);
title('Procena vaznosti predvidjanja');
ylabel('Procena');
xlabel('Predvidjanje');
```

Iz dijagrama se uočava da predispitne obaveze imaju najveći uticaj na završnu ocenu, slika 6.



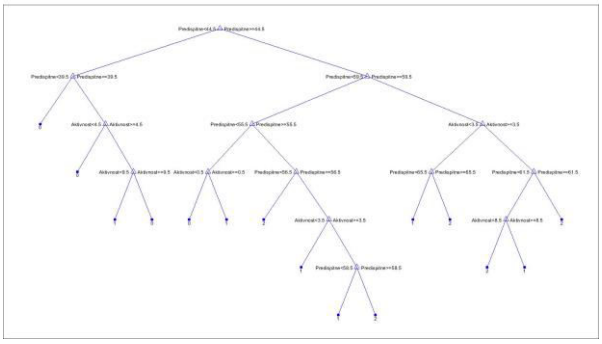
Slika 6. Dijagram važnosti predviđanja

Optimizacija i formiranje stabla odlučivanja postiže se funkcijom *fitctree*. Ovom funkcijom se prilagođavaju binarna stabla odlučivanja za klasifikaciju problema sa više klasa:

```
h = gca;
h.XTickLabel = model.PredictorNames;
h.XTickLabelRotation = 45;
h.TickLabelInterpreter = 'none';
%% optimizacija
```

```
model1=fitctree(dataTrain.Aktivnost,dataTrain.Predispitne,'
OptimizeHyperparameters','auto')
```

Maksimalan broj podele je 5, koji je automatski dodeljen na osnovu kompleksnosti modela, slika 7.



Slika 7. Stablo odlučivanja

Funkcija fitctree koristi parametar Optimize Hyperparameters za optimizaciju gubitka unakrsne validacije klasifikatora koristeći podatke iz mera za predviđanje:

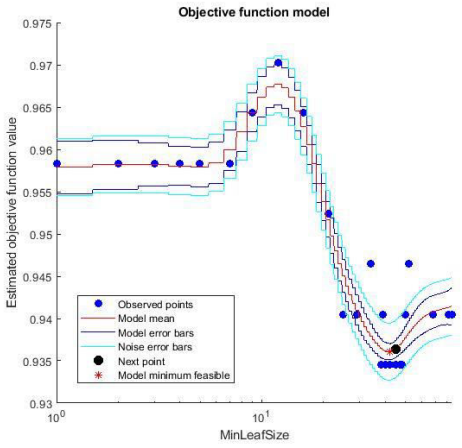
```
Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 19.2819 seconds.
Total objective function evaluation time: 1.9341
Best observed feasible point:
  MinLeafSize
  ___________
     19

Observed objective function value = 0.92857
Estimated objective function value = 0.92856
Function evaluation time = 0.064104
Best estimated feasible point (according to models):
  MinLeafSize
  ___________
     21
Estimated objective function value = 0.92856
Estimated function evaluation time = 0.037524
```

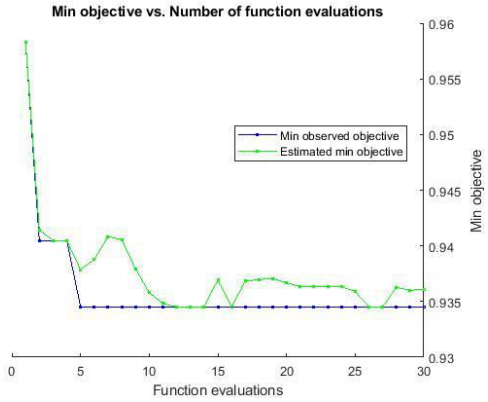
Rezultat optimizacije stabla odlučivanja:

```
model1 =
  ClassificationTree
  ResponseName: 'Y'
  CategoricalPredictors: []
  ClassNames: [1×36 double]
  ScoreTransform: 'none'
  NumObservations: 168
  HyperparameterOptimizationResults:
 [1×1 BayesianOptimization]
```

Nakon formiranja stabla odlučivanja, prikazan je tok optimizacije, na slici 8 prikazano je formiranje broja listova, dok je na slici 9 prikazana funkcija evaluacije.



Slika 8. Minimalan broj listova



Slika 9. Funkcija evaluacije

V. ZAKLJUČAK

U ovom radu je istraživano koji od faktora više utiče na posignuće studenta na predmetu Uvod u programiranje, primenom Random Forest algoritma. Svaki od faktora koji su predstavljeni kao ulaz, su međusobno nezavisni i ne utiču jedan na drugi, ali svaki od njih utiče na izlaznu promenljivu. Dobijeni rezultat, pokazuje da broj bodova koje studenti steknu kroz semestar više utiče na izlaznu veličinu, odnosno dobro savladavanje gradiva, nego aktivnost i interakcija na nastavi i vežbama. Dalje unapređenje ovog rada, bilo bi istraživanje više ulaznih faktora koji utiču na izlaznu veličinu, odnosno razlaganje predispitnih obaveza na više segmenata, kao i korišćenje drugih metoda veštačke inteligencije i upoređivanje dobijenih rezultata u cilju dobijanja najoptimalnijeg rešenja u postizanju boljeg uspeha studenata.

LITERATURA

[1] M. Milosavljević, "Veštačka inteligencija," Univerzitet Singidunum, Beograd, 2015.

[2] Z. Nagy, "Osnove veštačke inteligencije i mašinskog učenja," Kompjuter biblioteka, Beograd, 2019.

[3] R. Zhong, C. Salehi, R. Johnson, "Machine learning for drilling applications: A review," Journal of Natural Gas Science and Engineering, 2022.

[4] H. Azimi, H. Shiri, M. Mahdianpari, "Iceberg-seabed interaction analysis in sand by a random forest algorithm," Polar Science, 2022.

[5] Y. Fang, L. Ma, Z. Yao, W. Li, S. You, "Process optimization of biomass gasification with a Monte Carlo approach and random forest algorithm," Energy Conversion and Management, 2022.

[6] Random Forest Algorithm: https://www.javatpoint.com/machine-learning-random-forest-algorithm

ABSTRACT

The paper will present the application of the artificial intelligence method, i.e. the Random Forest algorithm in examining the factor that has a greater influence on the mastering of the material of students at a higher education institution. The observed parameter, which is the final outcome of this research, is determined on the basis of two independent variables, namely the activity, that is, the interaction in exercises and lectures, as well as the number of points collected in pre-examination tasks. Two phases of the Random Forest algorithm were used, the first phase involves the generation of a random forest by combining n decision trees. In the second stage, predictions are made for each tree created in the first stage. The result of the research shows that the number of points a student collects during the semester on the pre-examination obligations has a greater influence on the exit factor, which is a high level of knowledge, grades 9 and 10.

APPLICATION OF THE RANDOM FOREST ALGORITHM AS A SUPPORT FOR THE IMPROVEMENT OF STUDENT ACHIEVEMENT

Marija Mojsilović, Enes Sukić