Miloš Jovanović

Centar za poslovno odlučivanje
Fakultet organizacionih nauka
Univerzitet u Beogradu

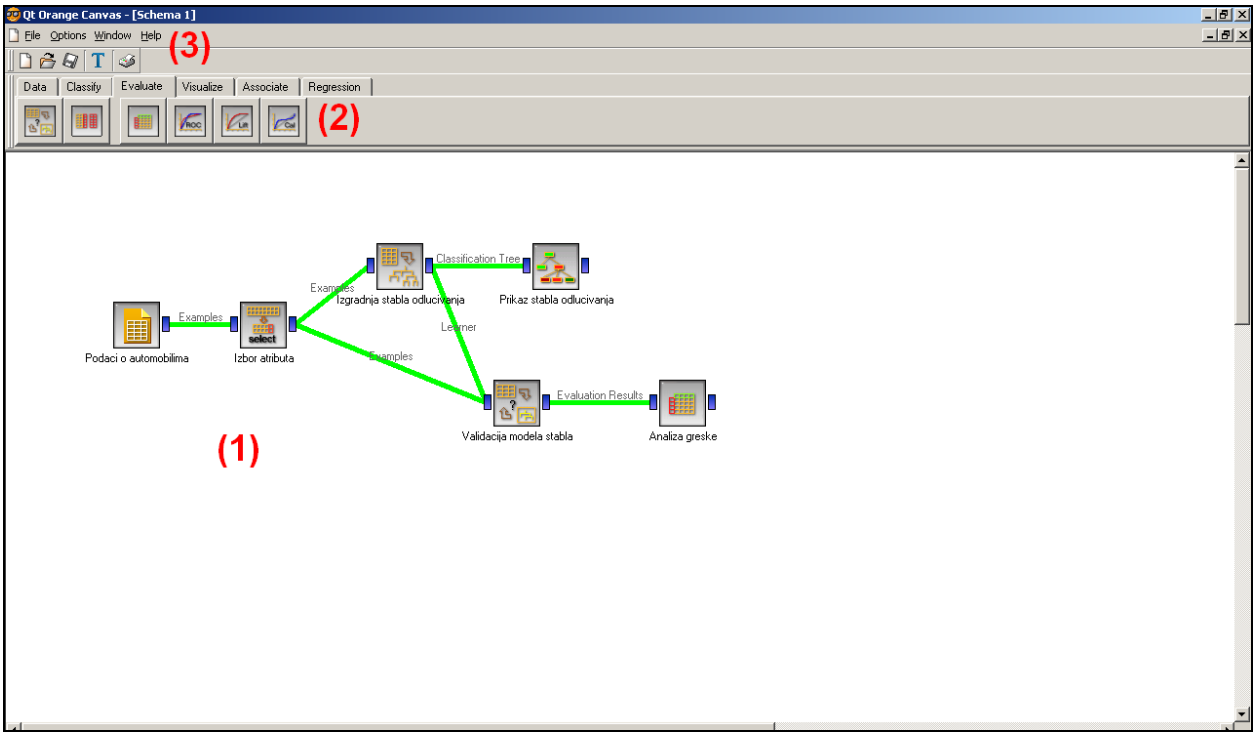
Data mining alat
Orange™

Program *Orange* namenjen je razvoju i primeni procesa otkrivanja zakonitosti u podacima (*Data mining*). Razvijen je od strane Univerziteta u Ljubljani (Fakulteta za računarstvo i informatiku), besplatan je i predstavlja program otvorenog koda (*open source*). Detalji o programu se mogu naći na Internet adresi www.ailab.si/orange.

U nastavku će biti prikazane osnovne funkcije programa. Takođe se prikazuju i primeri korišćenja, sa opisom rada u programu, kao i tumačenjem rezultata.

Radna površina

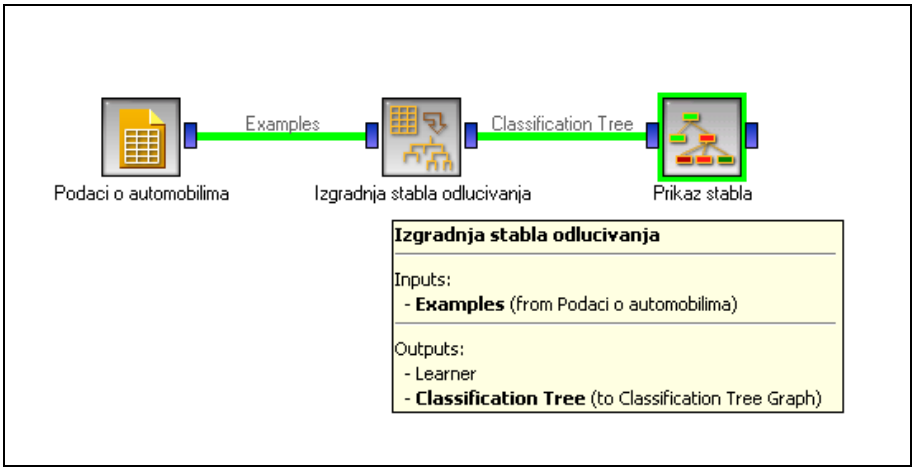
Radna površina programa je prikazana na Slici 1 i se sastoji od: površine za razvoj procesa za data-mining (1); skupa čvorova za procesiranje, podeljenih u grupe (2); glavnog menija za konfiguraciju programa i osnovne funkcije sa projektom (3).



Slika 1: Radna površina programa *Orange*

Proces za *data-mining* se kreira uklapanjem komponenti (čvorova) u tok u kome svaki čvor vrši deo funkcije procesiranja podataka. Primer jednog toka, sa nizom čvorova, prikazan je na Slici 2. Svaki čvor je definisan sa odgovarajućim ulazima, potrebnim za rad, i izlazima koji su rezultat procesiranja. Ulazi i izlazi čvorova definišu kako se ulančavaju čvorovi, tj. koje čvorove je moguće nadovezati na koje. Tako

su, na primer, čvorovi *File* i *Classification Tree* kompatibilni, jer izlaz prvog je definisan kao ulaz drugog, te je moguće ulančati ih (Slika 2). Opis i definiciju ulaza i izlaza čvora moguće je videti zadržavanjem strelice miša na nekom čvoru.



Slika 2: Primer toka i ulančavanja

Čvorovi su organizovani u nekoliko kategorija (grupa):

1. *Data* - čvorovi za osnovnu manipulaciju podacima
2. *Classify* - čvorovi algoritama za klasifikaciju
3. *Evaluate* - čvorovi za proveravanje kvaliteta modela
4. *Visualize* - čvorovi za vizuelni prikaz podataka
5. *Associate* - čvorovi algoritama za klasterovanje i asocijaciju
6. *Regression* - čvorovi algoritama za procenu

Čvorovi će biti opisani u daljem tekstu kako se budu spominjali u kontekstu.

Iz glavnog menija programa moguće je konfigurisati radno okruženje kroz podmeni *Options*. U podmeniju *File* moguće je sačuvati izgrađeni proces (u programu se zove šema – *Schema*), kao i učitati predhodno izgrađeni proces. Procesi se čuvaju u datoteci sa ekstenzijom „.ows“.

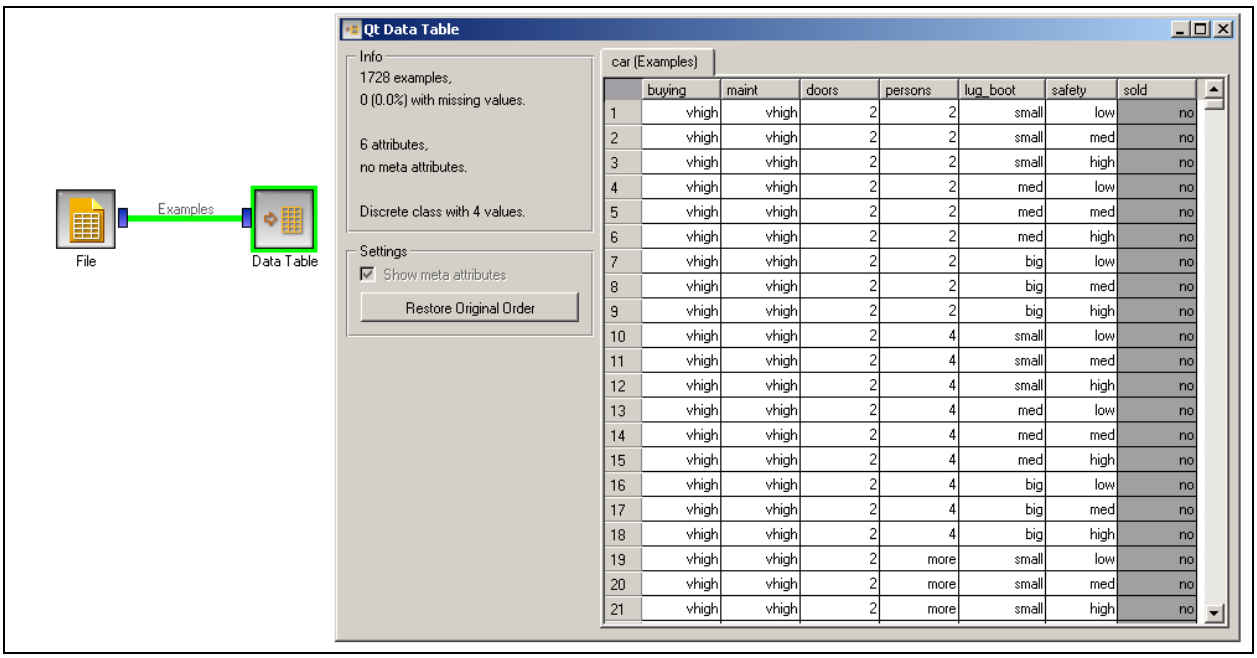
Učitavanje podataka i pregled podataka

Čvor odgovoran za učitavanje podataka u proces je čvor *File*, iz grupe *Data*. Čvor nema definisane ulaze (druge čvorove potrebne pre njega), a na izlazu se nalazi skup učitanih podataka, koji se označava kao *Examples* (Slučajevi). Duplim klikom na čvor se otvara konfiguracija čvora, gde je moguće podesiti izvornu putanju do datoteke sa podacima, kao i opcije za tretiranje nedostajućih vrednosti u podacima.

Podaci koji se učitavaju mogu biti u „tab“ formatu ili u „csv“ formatu. „Tab“ format predstavlja podatke u tekstualnom formatu, u kome su vrednosti odvojene <tab> karakterom. U „CSV“ formatu vrednosti atributa svakog slučaja odvojene su zarezom[1]. U oba formata prva linija teksta ne predstavlja vrednosti, već nazive kolona, tj. atributa za opisivanje slučajeva.

[1] neke verzije programa imaju problema (bug-ove) sa učitavanjem „csv“ formata podataka, što je moguće zaobići konverzijom podataka u kompatibilni „tab“ format.

Učitane podatke u proces je moguće videti pomoću čvora *DataTable* iz grupe *Data*. Čvor na ulazu zahteva slučajeve (*Examples*), pa je moguće povezati ga sa čvorom *File*, što je prikazano na Slici 3.

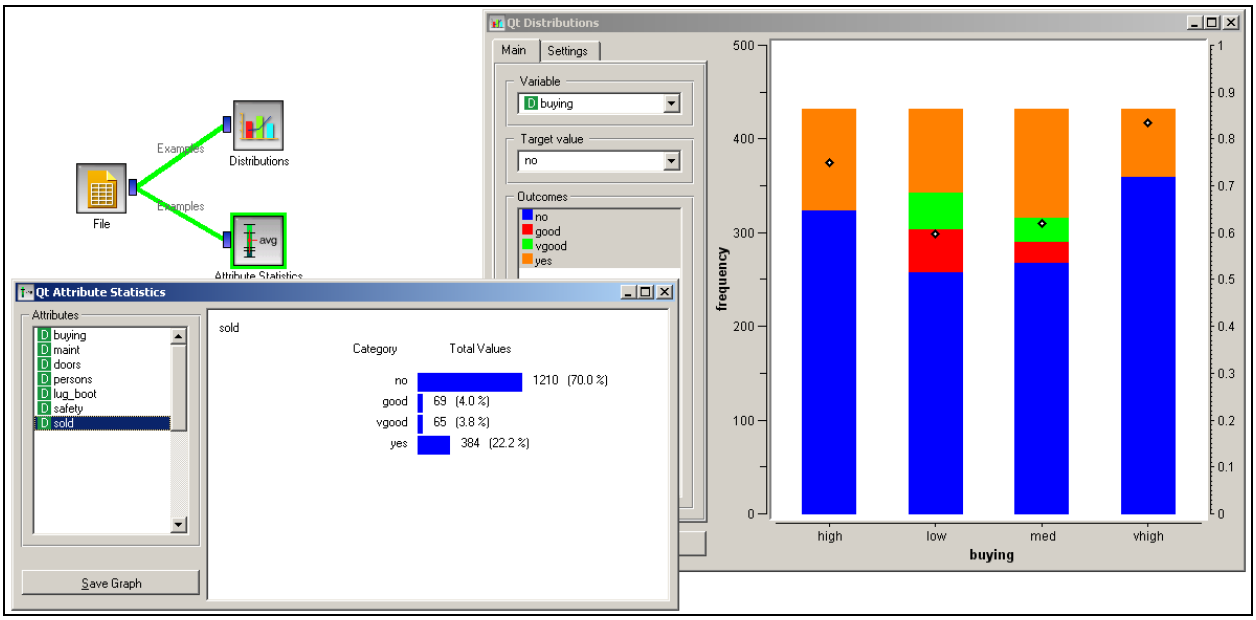


	buying	maint	doors	persons	lug_boot	safety	sold
1	vhigh	vhigh	2	2	small	low	no
2	vhigh	vhigh	2	2	small	med	no
3	vhigh	vhigh	2	2	small	high	no
4	vhigh	vhigh	2	2	med	low	no
5	vhigh	vhigh	2	2	med	med	no
6	vhigh	vhigh	2	2	med	high	no
7	vhigh	vhigh	2	2	big	low	no
8	vhigh	vhigh	2	2	big	med	no
9	vhigh	vhigh	2	2	big	high	no
10	vhigh	vhigh	2	4	small	low	no
11	vhigh	vhigh	2	4	small	med	no
12	vhigh	vhigh	2	4	small	high	no
13	vhigh	vhigh	2	4	med	low	no
14	vhigh	vhigh	2	4	med	med	no
15	vhigh	vhigh	2	4	med	high	no
16	vhigh	vhigh	2	4	big	low	no
17	vhigh	vhigh	2	4	big	med	no
18	vhigh	vhigh	2	4	big	high	no
19	vhigh	vhigh	2	more	small	low	no
20	vhigh	vhigh	2	more	small	med	no
21	vhigh	vhigh	2	more	small	high	no

Slika 3: Učitavanje i prikaz podataka

Povezivanje se obavlja jednostavnim prevlačenjem plavih krajeva čvorova jednog na drugi. Posle povezivanja, moguće je duplim klikom otvoriti čvor *DataTable*, posle čega se vidi prikaz učitanih podataka. Na levoj strani prikaza moguće je videti i statistike podataka, poput broja slučajeva, atributa, kao i broja klasa izlaznog atributa.

Dodatni uvid u podatke može se ostvariti čvorom *Distributions*, kao i čvorom *Attribute Statistics*, iz grupe *Visualize*. Oba čvora na ulazu imaju *Examples*, tako da se mogu vezati iza čvora *File*. *Distributions* čvor prikazuje raspodelu slučajeva po vrednostima izabranog atributa. Dodatno, na raspodeli se bojama ukazuje na broj slučajeva unutar svake od klasa (izlaznog atributa), što može nositi dosta informacija za analizu. Čvor *Attribute Statistics* koristi se isto kao i predhodni čvor, a služi za prikaz deskriptivnih statističkih pokazatelja svakog od atributa. Obe vizuelizacije se, posle povezivanja, mogu aktivirati duplim klikom na čvor. Primer rezultata vizuelizacije se može videti na Slici 4.



Slika 4: Vizuelizacija učitanih podataka

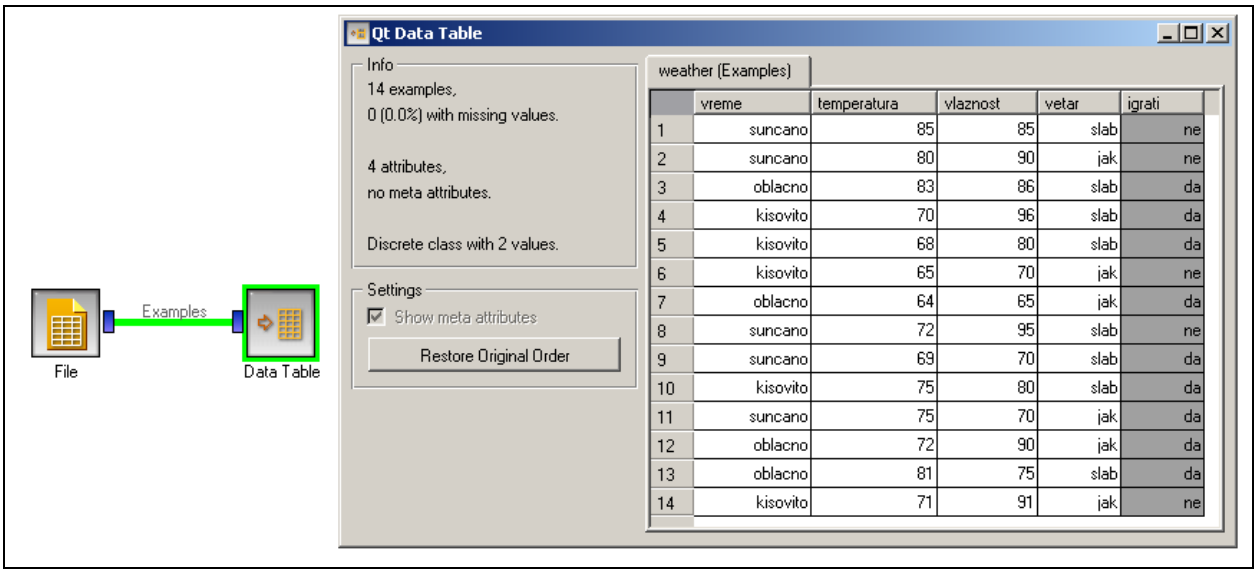
Analizom podataka kroz vizuelizaciju se preliminarno mogu uočiti neki paterni u podacima. Sa Slike 4 se, na primer, može uočiti da se automobili sa visokom prodajnom cenom (*buying* atribut sa vrednosti *high*) nikada ne prodaju dobro (nema slučajeva izlaznog atributa *good* i *vgood*, koji su na grafiku označeni crvenom i zelenom bojom).

Ponekad se u podacima pojavljuje veliki broj atributa, od kojih nemaju svi značaj za analizu. Atributi se ručno mogu filtrirati čvorom *Select Attribute*, iz grupe *Data*. Čvor i na ulazu i na izlazu ima *Examples*, a nudi mogućnost izbora atributa koji se koriste dalje u analizi. Uklonjeni atributi ostaju sakriveni za nastavak toka. Dodatno, ovim čvorom se može definisati i izlazni atribut. Bez ovog čvora, kao izlazni atribut se podrazumevano uzima poslednji atribut iz skupa podataka, što se vidi na Slici 3, gde je poslednji atribut zatamnjen kako bi se označilo da je izlazni (atribut klase).

Primer izgradnje modela - Klasifikacija

Problem klasifikacije jeste problem kreiranja načina za svrstavanje objekata (slučajeva) u ispravnu klasu. Postoji više algoritama za kreiranje modela za klasifikaciju, a u ovom programu su oni dostupni kroz čvorove grupe *Classify*.

Kao primer problema za klasifikaciju koristiće se podaci koji opisuju slučajeve igranja golfa, a dati su na Slici 5. Problem je odrediti način (model) klasifikacije slučajeva u ispravnu klasu. Informacija o klasi se nalazi u izlaznom atributu „igrati“, u kome vrednosti „da“ ili „ne“ određuju klasu slučaja (objekta). Svi slučajevi su opisani sa 4 atributa koji predstavljaju vremenske uslove slučaja iz prošlosti.



	vreme	temperatura	vlaznost	vetar	igrati
1	suncano	85	85	slab	ne
2	suncano	80	90	jak	ne
3	oblacno	83	86	slab	da
4	kisovito	70	96	slab	da
5	kisovito	68	80	slab	da
6	kisovito	65	70	jak	ne
7	oblacno	64	65	jak	da
8	suncano	72	95	slab	ne
9	suncano	69	70	slab	da
10	kisovito	75	80	slab	da
11	suncano	75	70	jak	da
12	oblacno	72	90	jak	da
13	oblacno	81	75	slab	da
14	kisovito	71	91	jak	ne

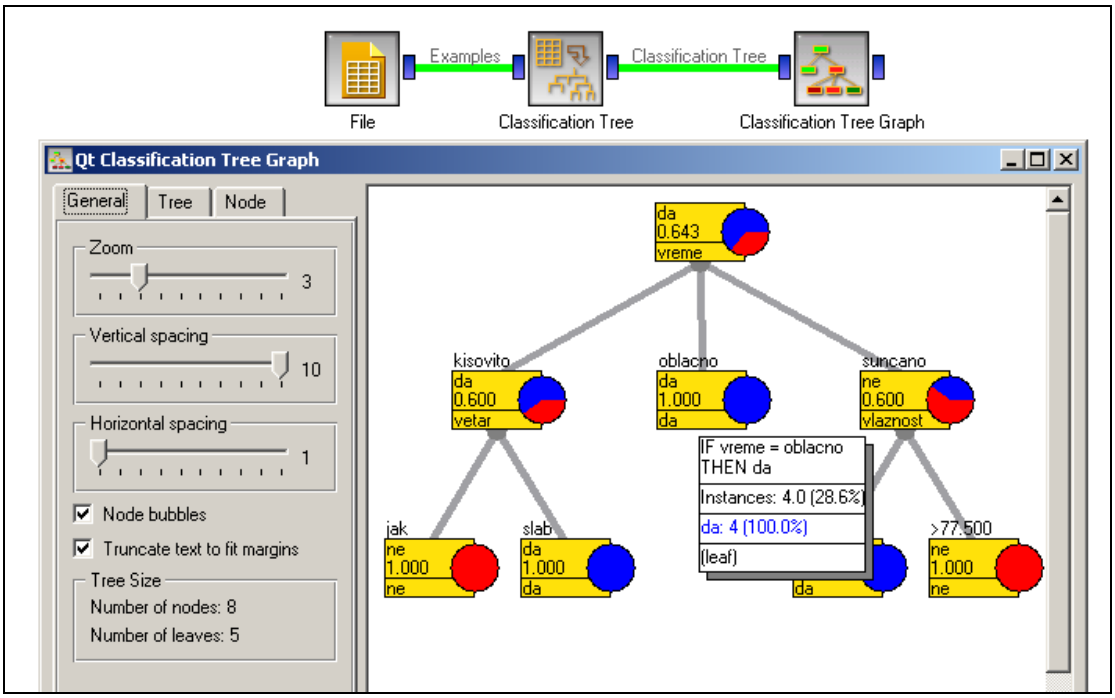
Slika 5: Istorijski podaci o igranju golfa

Izgradnja procesa za klasifikaciju se u programu može izvršiti na sledeći način:

1. Uvede se čvor *File*, kako bi se definisao izvor podataka;
2. Uvede se čvor *Classification Tree*, koji predstavlja algoritam za pravljenje stabla koje će biti klasifikator slučajeva;
3. Povežu se čvor *File* i čvor *Classification Tree*;

Posle ovoga, čvor *Classification Tree* će sadržati model za klasifikaciju, tj. stablo kojim je moguće izvršiti klasifikaciju, a koje je izgrađeno pomoću učitanih podataka. Kao što se može naslutiti, čvor *Classification Tree* na ulazu ima *Examples*, a na izlazu *ClassificationTree*, što znači da je izlaz iz čvora zapravo izgrađeno stablo.

Ako je potrebno vizuelizovati dobijeno stablo, to se može uraditi nadovezujući čvor *Classification Tree Graph* na čvor *Classification Tree*. Otvaranjem čvora vizuelizacije prikazaće se izgrađeno stablo koje predstavlja znanje na osnovu kojeg se slučajevi klasifikuju u klase, a što je prikazano na Slici 6.



Slika 6: Prikaz generisanog stabla

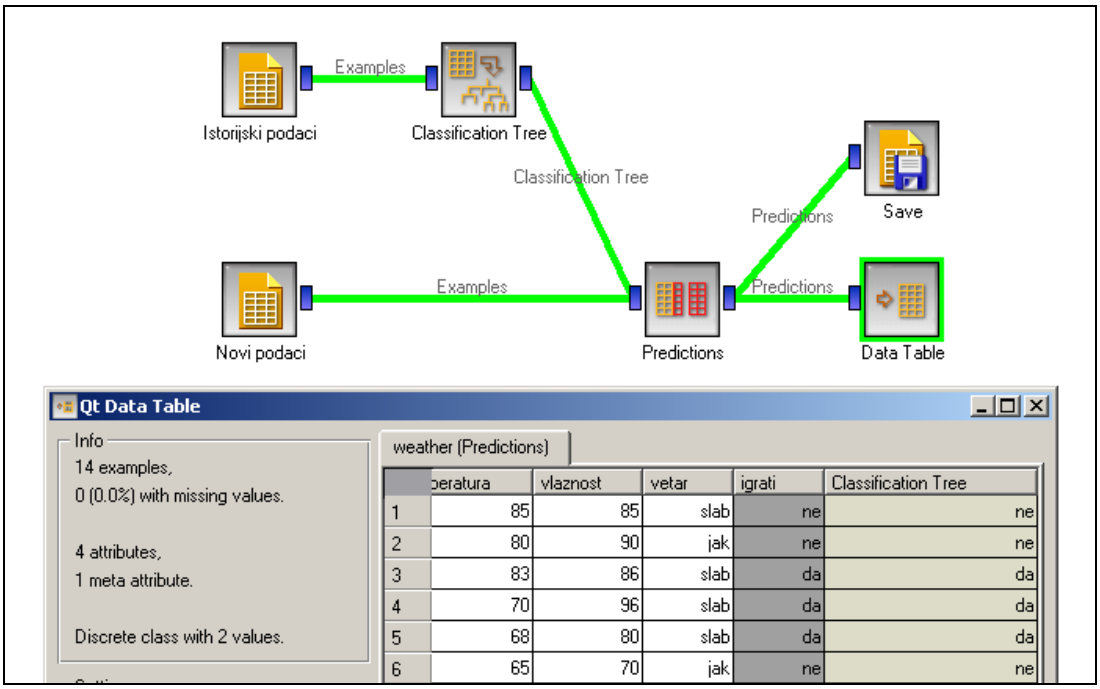
Na prikazanom stablu se vidi način na koji model odlučuje da li će se igrati golf na osnovu podataka o vremenu. Na primer, ako je vreme oblačno, zaključuje se da se igra golf, jer od četiri slučaja iz prošlosti, u sto odsto slučajeva se igralo u takvim vremenskim uslovima.

Iz grupe čvorova *Classify* dostupni su i drugi algoritmi za izgradnju modela klasifikacije, koji ne moraju graditi stablo, već neki drugi model koji može klasifikovati objekte. Neki od dostupnih algoritama su:

- *C4.5* (predstavlja takođe algoritam za izgradnju stabla, a naslednik je popularnog ID3 algoritma)
- SVM (gradi kompleksni model vektora (hiperravni) koji najbolje razdvajaju podatke u klase)
- K-Nearest-Neighbours (gradi model koji klasifikuje objekte na osnovu sličnosti sa drugim objektima)

Upotreba modela klasifikacije

Sagrađeni model klasifikacije može se nadalje upotrebiti za klasifikovanje novih slučajeva koji se pojave u budućnosti. U datom primeru, to bi odgovaralo mogućnosti da se odredi u budućoj situaciji da li vremenski uslovi ukazuju na to da li treba igrati golf ili ne, a naučeno na prošlom iskustvu. Za izvođenje klasifikacije (predviđanja) nad novim slučajevima, u programu se koristi čvor *Predictions*, iz grupe *Evaluate*. Čvor na ulazu zahteva dve stvari: model za klasifikaciju (*Predictors*) i podatke (*Examples*) čiji izlazni atribut (klasu) treba odrediti. Na Slici 7 se vidi da dva ulazna toka ulaze u čvor *Predictions*, jedan iz čvora *Classification Tree* koji nosi model i jedan tok iz čvora *File* koji nosi podatke za klasifikaciju. Na izlazu iz čvora se nalaze slučajevi (*Examples*) koji nose novi atribut koji predstavlja klasu posle klasifikacije. Na Slici 7 je prikazano kako se rezultat klasifikacije može videti pomoću čvora *Data Table*, koji se stavlja na kraj toka.



Slika 7: Tok za prikaz i čuvanje predviđanja modela

Takođe, dobijena klasifikacija novih slučajeva se može sačuvati u datoteku, koristeći čvor *Save* iz grupe *Data*, kao što je prikazano na Slici 7.

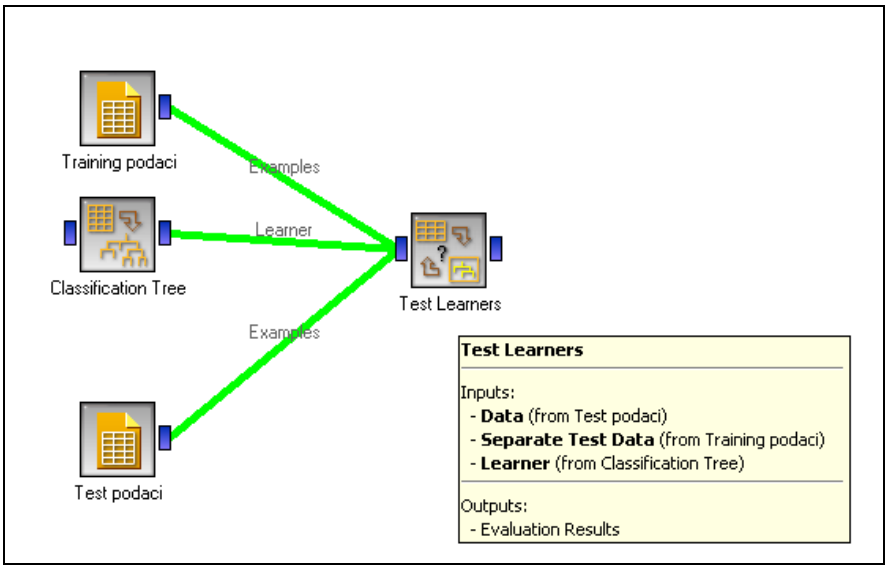
Validacija modela

Pre upotrebe modela, poželjno je ispitati kvalitet istog, kako bi se dobio nivo sigurnosti sa kojim se model može primenjivati. Proces u kome se kvalitet modela testira upotrebom nad podacima se zove validacija.

Kvalitet modela se najčešće meri procentom greške klasifikacije, kada se primeni nad podacima za koje se unapred zna kojoj klasi pripadaju. Tada se uporede prava klasa sa procenjenom od strane modela i izračuna na uzorku greška klasifikacije, kao procentualni odnos neispravno klasifikovanih slučajeva prema ispravno klasifikovanim. Postoje i složenije mere kvaliteta, što će se videti u programu, ali što ovaj tekst neće obrađivati.

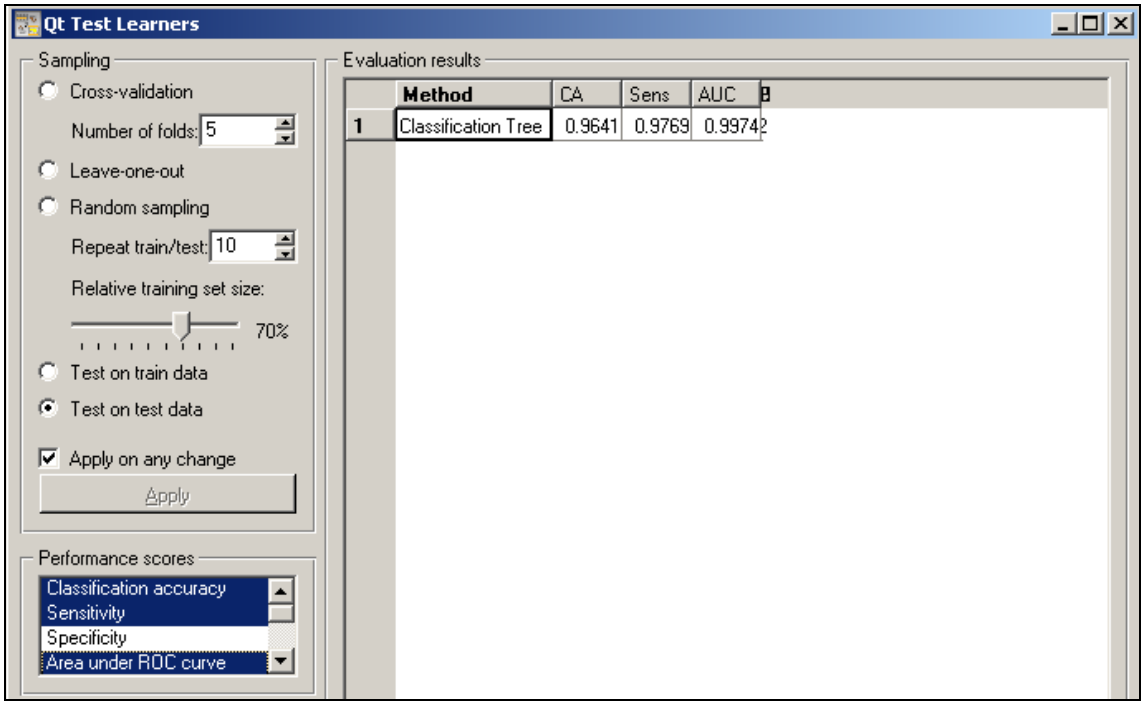
Za potrebe validacije, najčešće se iz početnog skupa podataka izdvaja jedan deo podataka koji se zove Test podaci (*Test Set*), nasuprot ostatku podataka koji se nazivaju Trening podaci (*Training set*). Ovim se omogućava da validacija bude ispravnija, pošto se model testira na podacima na kojimi nije građen. Tako se testira „generalizacija“ modela, što predstavlja osobinu da model daje dobre procene na novim slučajevima u budućnosti.

Za validaciju se u ovom programu koristi čvor *Test Learners*, iz grupe *Evaluate*. Ovaj čvor na ulazu zahteva tri stvari: podatke za učenje (Training set), model za klasifikaciju (Learner) i podatke za testiranje (Test set). Primer toka za validaciju prikazan je na Slici 8.



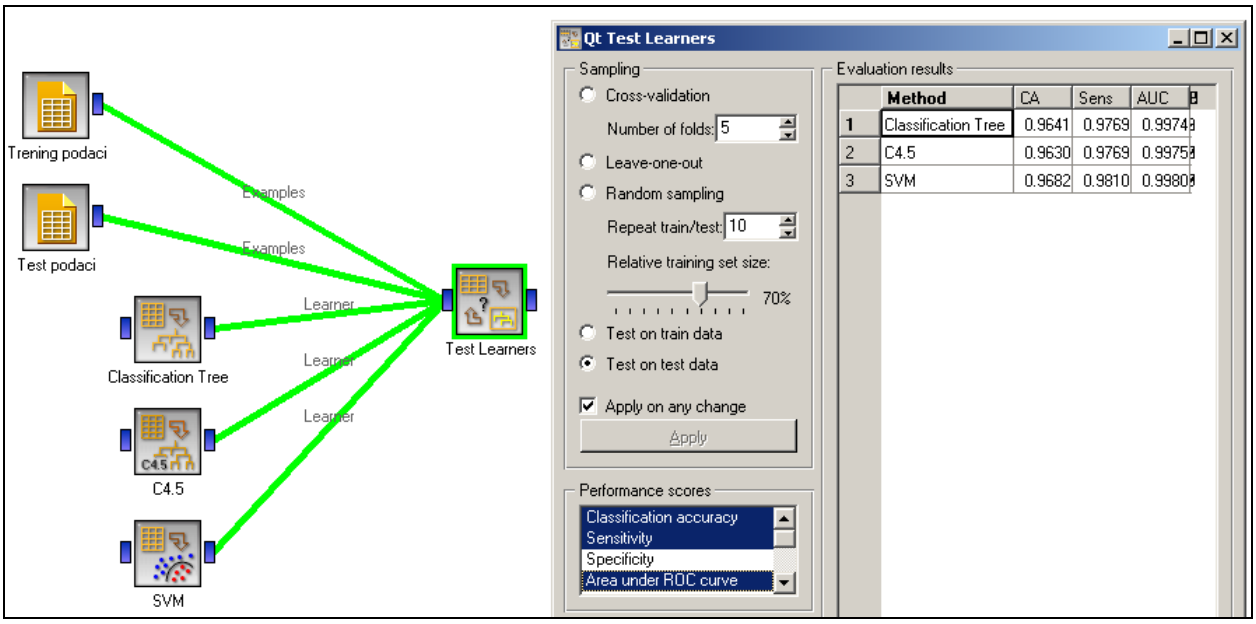
Slika 8: Tok za testiranje (validaciju) modela

Otvaranjem ovog čvora se, posle spajanja ulaza, mogu videti razne mere kvaliteta, što je prikazano na Slici 9. U donjem delu prozora se mogu izabrati mere kvaliteta koje se računaju, od kojih prva predstavlja tipičnu meru procenta tačnosti (*Classification Accuracy - CA*), što je procenat ispravnih klasifikacija na test podacima.



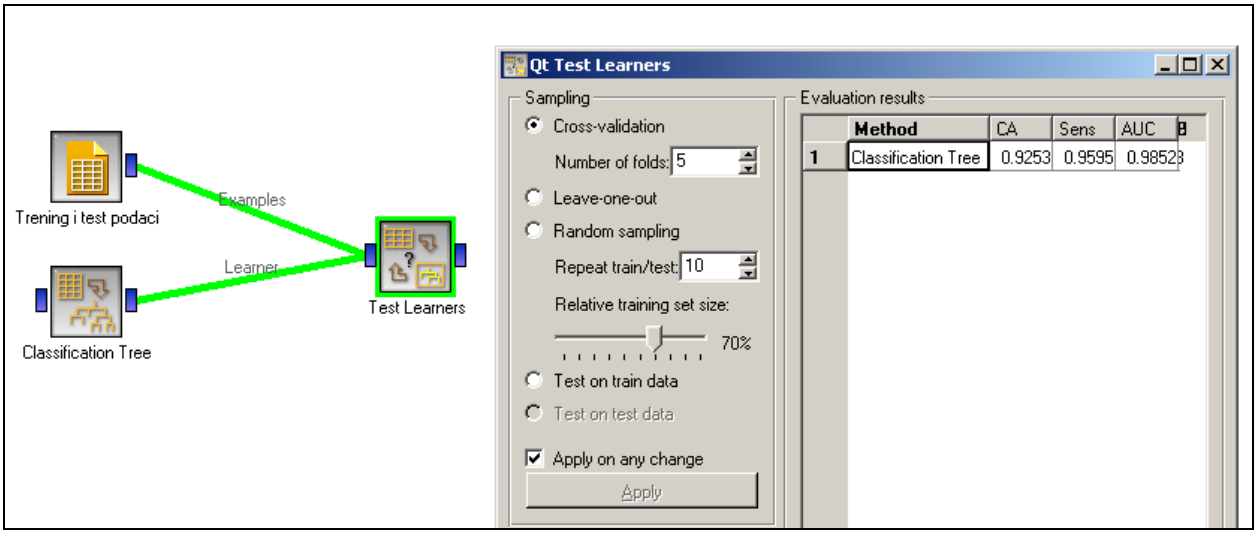
Slika 9: Prikaz kvaliteta modela (rezultata validacije)

Na Slici 10 se može videti i mogućnost da se različiti modeli mogu testirati paralelno, čime se omogućuje laka uporedna analiza kvaliteta različitih modela. U primeru je upoređen kvalitet čvorova za klasifikaciju: *Classification Tree*, *C4.5* i *SVM*, nad konkretnim podacima.



Slika 10: Poređenje kvaliteta različitih algoritama

Treba još napomenuti da u slučajevima kada je skup podataka relativno mali, deljenje podataka na trening i test podatke može biti neefektivno, jer će se dodatno smanjiti količina podataka za izgradnju modela, što ozbiljno može ugroziti mogućnost dobrih rezultata modela. Tada se umesto podele na trening/test podatke, validacija može sprovesti postupkom „kros validacije“ (*Cross-Validation*). Tada se na ulaz čvora *Test Learners* donose svi podaci, kao i algoritam za učenje. Otvaranjem čvora *Test Learners*, sa leve strane prozora se može izabrati opcija *Cross Validation*, koja će proveriti kvalitet modela bez razdvajanja skupa podataka na trening/test skup. Proces za ovakvu validaciju se može videti na Slici 11. Pošto se postupak kros-validacije neće detaljno opisivati u ovom tekstu, zainteresovani čitaoci se upućuju na obilne izvore na Internetu (e.g. Wikipediju).



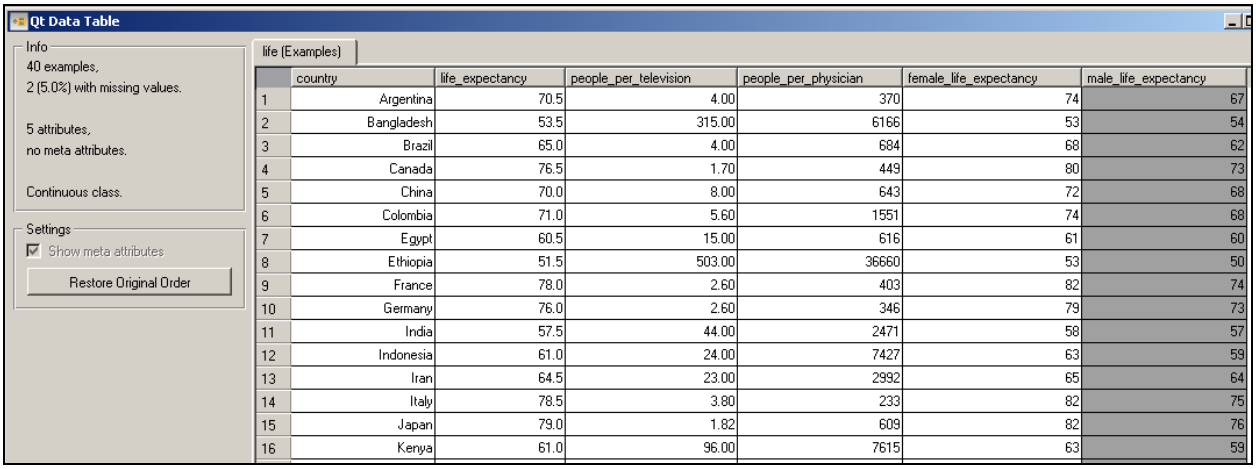
Slika 11: Kros-validacija

Primer izgradnje modela - Klasterovanje

Kao primer problema koji može ilustrovati korist od klasterovanja, koristiće se primer zemalja opisanih demografskim podacima. Pretpostavlja se da su po korišćenim atributima neke zemlje slične, kao i da poznavanje sličnosti zemalja može koristiti za razne analize, zbog čega se nad podacima traže klasteri.

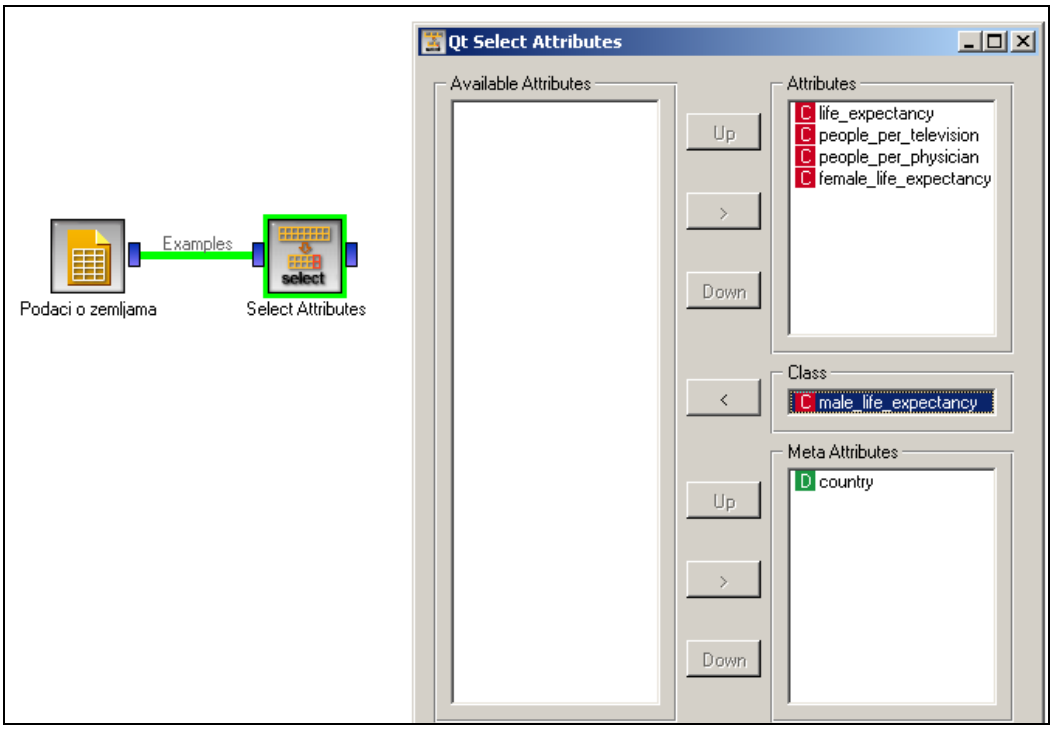
Klasteri su definisani kao grupe objekata (slučajeva) koji su međusobno dovoljno slični, a dosta različiti od objekata iz drugih klastera.

Podaci iz primera se mogu videti na Slici 12. Pošto je program automatski prepoznao poslednji atribut kao „izlazni“, to treba izmeniti, jer u zadatku klasterovanja ne postoji izlazni atribut klase, već se grupe formiraju na osnovu sličnosti svih atributa, a ne na osnovu predodređene (apriori) pripadnosti klasi. U programu se koristi čvor *Select Attributes*, da se isključi izlazni atribut, što je prikazano na Slici 13.



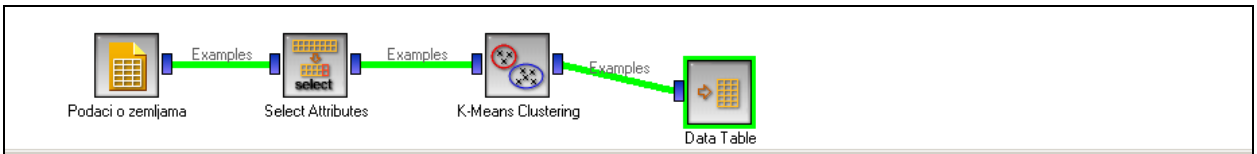
	country	life_expectancy	people_per_television	people_per_physician	female_life_expectancy	male_life_expectancy
1	Argentina	70.5	4.00	370	74	67
2	Bangladesh	53.5	315.00	6166	53	54
3	Brazil	65.0	4.00	684	68	62
4	Canada	76.5	1.70	449	80	73
5	China	70.0	8.00	643	72	68
6	Colombia	71.0	5.60	1551	74	68
7	Egypt	60.5	15.00	616	61	60
8	Ethiopia	51.5	503.00	36660	53	50
9	France	78.0	2.60	403	82	74
10	Germany	76.0	2.60	346	79	73
11	India	57.5	44.00	2471	58	57
12	Indonesia	61.0	24.00	7427	63	59
13	Iran	64.5	23.00	2992	65	64
14	Italy	78.5	3.80	233	82	75
15	Japan	79.0	1.82	609	82	76
16	Kenya	61.0	96.00	7615	63	59

Slika 12: Demografski podaci iz raznih zemalja



Slika 13: Izbor podataka za analizu i definisanje izlaznog (Class) atributa

Za zadatak klasterovanja, koristiće se čvor *K-means Clustering*, iz grupe *Associate*, koji po povezivanju u tok pronalazi zadati broj klastera. Ako se otvori podešavanje čvora, moguće je podesiti i željeni broj klastera. Slika 14 prikazuje tok za izgradnju klastera algoritmom *K-means*. Ako treba pogledati kako su slučajevi dodeljeni klasterima, može se nadovezati čvor *Data Table*, posle čega se dobija početni skup podataka, proširen za kolonu koja predstavlja pripadnost slučaja određenom klasteru (Slika 14).



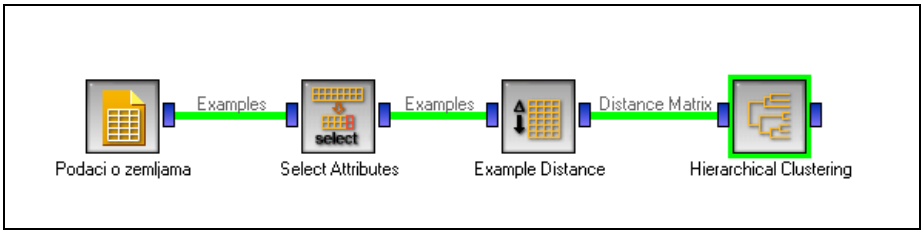
(Examples)

	life_expectancy	people_per_television	people_per_physician	female_life_expectancy	male_life_expectancy	cluster	country
1	70.5	4.00	370	74	67	1	Argentina
2	53.5	315.00	6166	53	54	2	Bangladesh
3	65.0	4.00	684	68	62	1	Brazil
4	76.5	1.70	449	80	73	1	Canada
5	70.0	8.00	643	72	68	1	China
6	71.0	5.60	1551	74	68	1	Colombia
7	60.5	15.00	616	61	60	1	Egypt
8	51.5	503.00	36660	53	50	3	Ethiopia
9	78.0	2.60	403	82	74	1	France

Slika 14: Tok za primenu *K-means* klasterovanja i prikaza rezultata

Pored *K-means* klasterovanja, moguće je svrstati slučajeve u klaster i pomoću čvora *Hierarchical Clustering*. Slika 15 prikazuje tok za izgradnju klastera algoritmom hijerarhijskog klasterovanja. Ako se otvori čvor *Hierarchical Clustering*, posle uvezivanja u tok, može se videti i grafički prikaz (Dendrogram) spajanja klastera, od sitnijih ka krupnijim klasterima. Takvim uvidom sa dendrograma se može steći uvid

koji slučajevi su sličniji, pošto su se ranije spojili u manji klaster prilikom izgradnje većih klastera. Prikaz dodeljenih slučajeva klasterima se takođe može videti *Data Table* čvorom, slično kao sa Slike 14.



Slika 15: Tok za primenu hijerarhijskog klasterovanja

Primer izgradnje modela - Asocijativna pravila

Zadatak otkrivanja asocijativnih pravila predstavlja težnju za otkrivanjem svih relevantnih veza u istovremenom pojavljivanju nekih osobina pojava. Cilj je otkriti veze (asocijacije) između bilo kojeg podskupa atributa, koje će ukazati da kada neki objekat poseduje jednu osobinu, on istovremeno poseduje i drugu osobinu koja je u vezi (asocijaciji) sa prvom. Asocijacije se predstavljaju u formi AKO-ONDA pravila, gde u delu uslova (AKO delu) može biti više atributa. Primer za pravilo asocijacije, nad podacima koji opisuju povrede na skijalištima, može biti:
AKO (slucaj = povreda noge) i (kolicina snega = mala) ONDA (staza = stazaBr2)

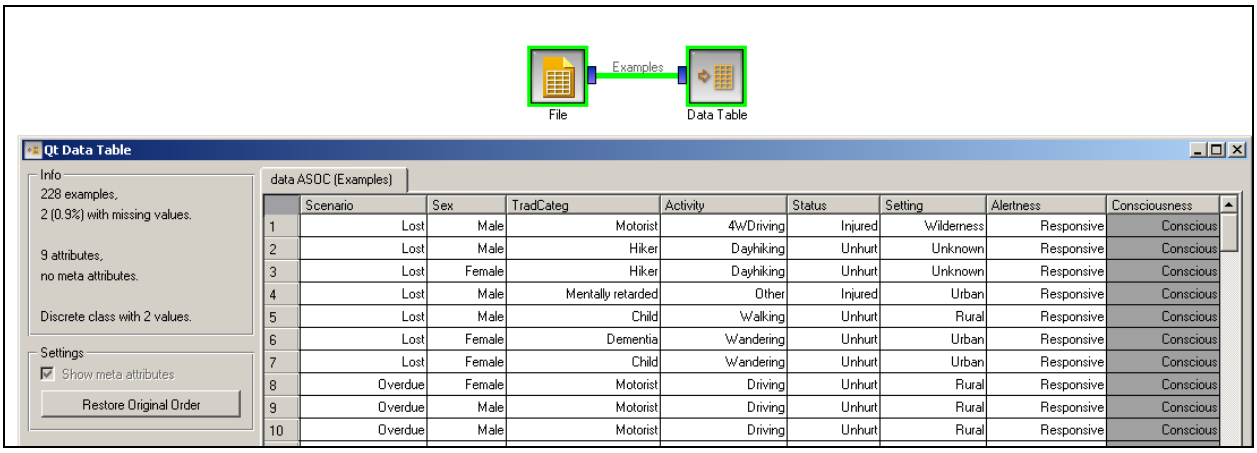
U opštem slučaju asocijativno pravilo ima formu:

AKO (atribut1=vrednost1) i (atribut2=vrednost2) i … i (atributN=vrednostN) ONDA (atributM=vrednostM)

Za razliku od klasifikacije i ostalih prediktivnih zadataka, proces otkrivanja asocijativnih pravila nije usmeren ka jednom izlaznom atributu. To znaci da izlazni atribut u pravilu moze biti bilo koji atribut iz skupa, što, pored fleksibilnosti, ima i posledicu da su algoritmi često spori u izvršavanju.

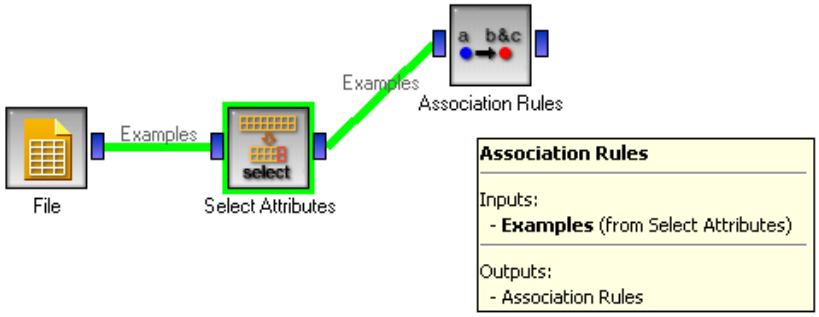
Još jedna osobina, tj. nedostatak, je što algoritmi za otkrivanje asocijativnih pravila funkcionišu samo sa kategoričkim (nenumeričkim) atributima. Tako numerički atributi koji opisuju objekte ili ostaju neupotrebljivi, ili ih je potrebno tehnikama diskretizacije prevesti u kategoričke.

Podaci koji će se koristiti predstavljaju slučajeve reagovanja spasilačke ekipe, opisane sa atributima koji opisuju vrste slučaja, osobine oštećenih, itd. Podaci su prikazani *Data Table* čvorom na Slici 16.



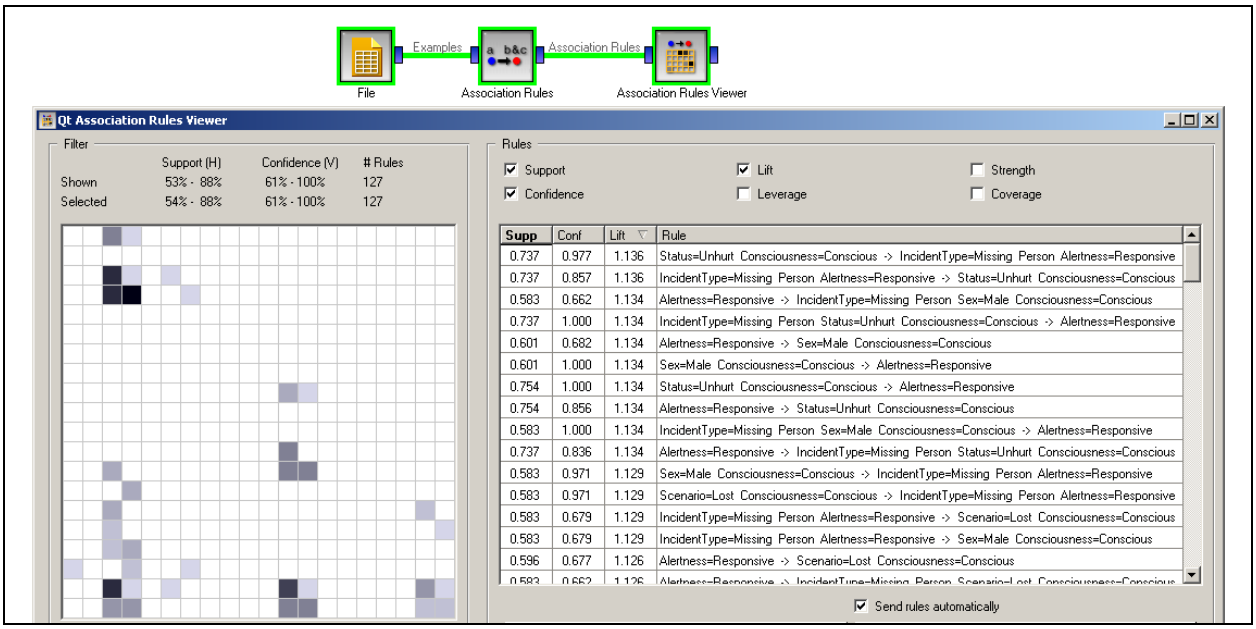
Slika 16: Podaci za otkrivanje asocijativnih pravila

U programu se otkrivanje asocijativnih pravila vrši čvorom *Association Rules* iz grupe *Associate*. Primer toka za izgradnju procesa za tu svrhu prikazan je na Slici 17. Slično kao kod zadatka klasterovanja, čvor *Select Attributes* se uvodi da bi filtrirao atribute između kojih se traži asocijacija, kao i da ukloni izlazni atribut, pošto asocijativna pravila ne poznaju pojam atributa klase (izlaznog atributa), jer spadaju u deskriptivne, a ne prediktivne algoritme.



Slika 17: Tok za otkrivanje asocijativnih pravila

U nastavku toka se može vezati čvor *Association Rules Viewer*, koji omogućava prikaz otkrivenih pravila. Otvaranjem tog čvora se može dobiti lista okrivenih pravila, što se vidi na desnoj strani Slike 18. Vidi se da je otkriven veliki broj pravila (tačnije 127), kao i da je teško razaznati koja od pravila su značajna i korisna.



Slika 18: Tok i prikaz otkrivenih pravila

Dodatna informacija o svakom pravilu su i njegove mere kvaliteta, koje opisuju koliko je pravilo tačno, upotrebljivo, značajno i neočekivano. Jedne od osnovnih mera kvaliteta asocijativnih pravila su poverenje (*confidence*) i podrška (*support*). Poverenje predstavlja verovatnoću da se desi posledica iz pravila (ONDA deo), ako je poznato da se desio uzrok pravila (AKO deo). Predstavlja preciznost pravila u zaključivanju, a računa se po sledećoj formuli:

$$\text{conf}\ \ A \Rightarrow B\ = \frac{|A \cap B|}{|A|}$$, gde su A i B skupovi slučajeva sa određenim osobinama.

Podrška je druga mera kvaliteta koja procenjuje koliko je pravilo upotrebljivo, tako što računa verovatnoću da se ispune uslovi iz uzroka pravila. Ta mera ukazuje na to u kolikoj relativnoj meri će biti moguće primeniti pravilo, a računa se po sledećoj formuli:

$$\text{supp}\ \ A \Rightarrow B\ = \frac{|A|}{|S|}$$, gde je S celokupan skup slučajeva.

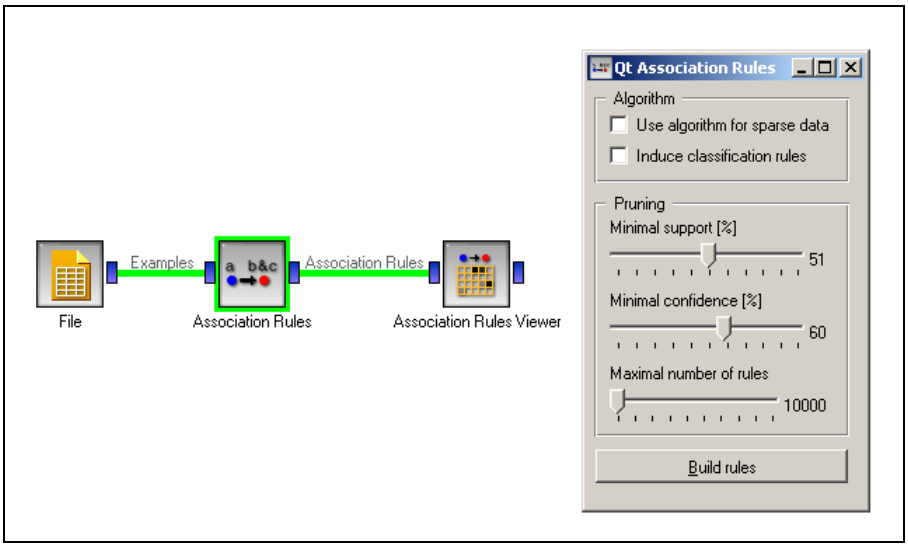
U programu su u listi pravila uključene i mere kvaliteta, prdružene svakom pravilu, što se vidi na Slici 18 (oznake *conf* i *supp*). Pravila je moguće i sortirati po merama kvaliteta (jednostavnim pritiskom na zaglavlje), što može olakšati izdvajanje relevantnih pravila iz skupa svih pronađenih pravila.

Uz listu pravila se na levoj strani prikaza (Slika 18) vidi i mogućnost filtriranja pravila po merama kvaliteta, i to kroz matricu koja na vertikalnoj dimenziji ima poverenje, a na horizontalnoj dimezniji podršku otkrivenih pravila.

Dodatno, uz poverenje i podršku, pravila se mogu opisati i drugim merama kvaliteta, među kojima je i mera Lift. Lift predstavlja meru koja ocenjuje neočekivanost pravila, a računa se po sledećoj formuli:

$$\text{lift}\ \ A \Rightarrow B\ = \frac{\frac{|A \cap B|}{|A|}}{\frac{|B|}{|S|}}$$

Pošto algoritam za pronalaženje asocijativnih pravila može biti dugotrajan jer pretražuje veliki prostor pravila, može se uticati na nekoliko načina na efikasnost algoritma. Jedan način je da se odrede donji pragovi kvalita pravila, što je moguće podesiti otvaranjem čvora *Association Rules*, a što je prilazano na Slici 19. Dodatno, efikasnost se može poboljšati izborom podskupa atributa od početnog skupa, kako bi algoritam istražio asocijacije na samo tom izabranom podskupu. Ovo je moguće uraditi čvorom *Select Attributes*, pre čvora *Association Rules*. Kao posledica ove težnje za efikasnošću može biti umanjenje efektivnosti algoritma u pronalaženju svih relevantnih pravila, ali to je kompromis koji je potrebno svesti na pravu meru.



Slika 19: Podešavanje čvora *Association Rules*

Pretprocesiranje podataka

Zadaci otkrivanja zakonitosti u podacima otkrivaju znanje koje može biti potencijalno korisno za unapređenje poslovnih procesa. Nažalost, otkriveno znanje može imati i suviše niske pokazatelje kvaliteta da bi bilo primenljivo, što može biti posledica više uzroka.

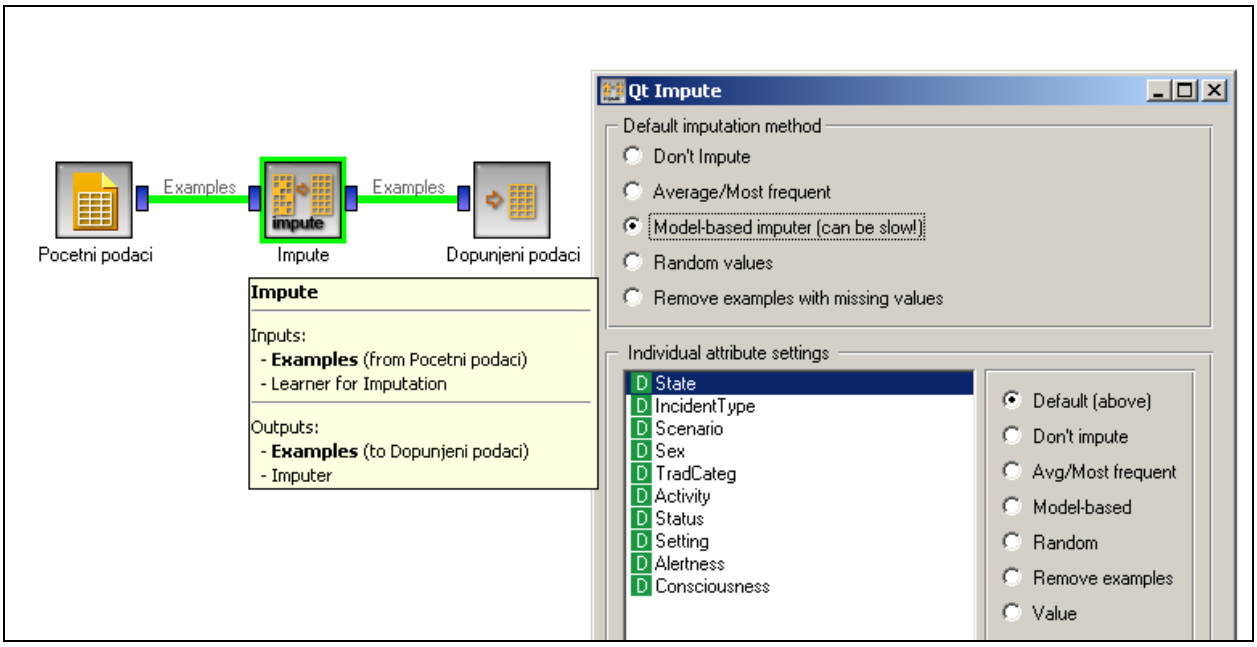
Jedan od značajnih uzroka kvaliteta znanja jeste i kvalitet podataka na osnovu kojih se otkriva znanje. Kvalitet podataka zavisi od broja slučajeva, broja atributa, izbora pravih atributa a zanemarivanja nepotrebnih, od grešaka u podacima, nestandardnih slučajeva, nedostajućih podataka, itd. Neke od ovih nedostataka u podacima se mogu otkriti i ispraviti pre procesa za izgradnju modela, što se postiže tehnikama pretprocesiranja podataka.

Jedan od mogućih problema u podacima su nedostajući podaci. Slučajevi sa vrednostima atributa koje nedostaju ne mogu biti korišćeni za izgradnju modela, a kod nekih algoritama mogu i da zaustave ili

ometu proces. Zato je poželjno rešiti taj problem pre puštanja algoritama za izgranju modela. U programu se ovaj problem rešava čvorom *Impute Data*, koji na ulazu ima slučajeve (*Examples*) sa nedostajućim podacima, a na izlazu takođe slučajeve, ali bez tog problema, koji se može rešiti na više načina. Otvaranjem čvora *Impute Data* se mogu definisati načini za rešavanje, a neki od njih su:

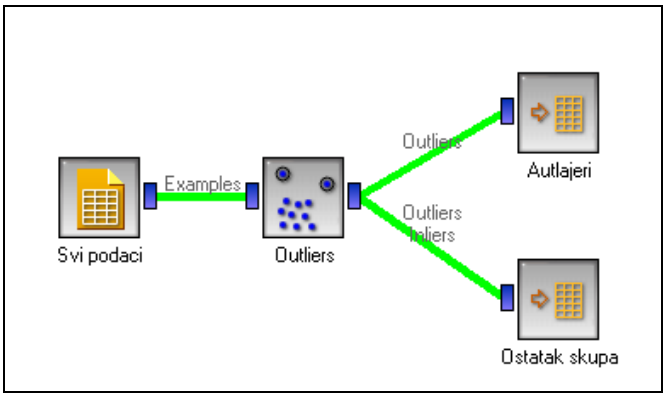
- izbacivanje slučajeva koji imaju nedostajuću vrednost atributa,
- popunjavanje nedostajućih vrednosti sa prosečnim vrednostima atributa,
- popunjavanje nedostajućih vrednosti sa slučajnim vrednostima.

Tok za rešavanje problema nedostajućih podataka, kao i izbora tehnike za to, je prikazan na Slici 20.



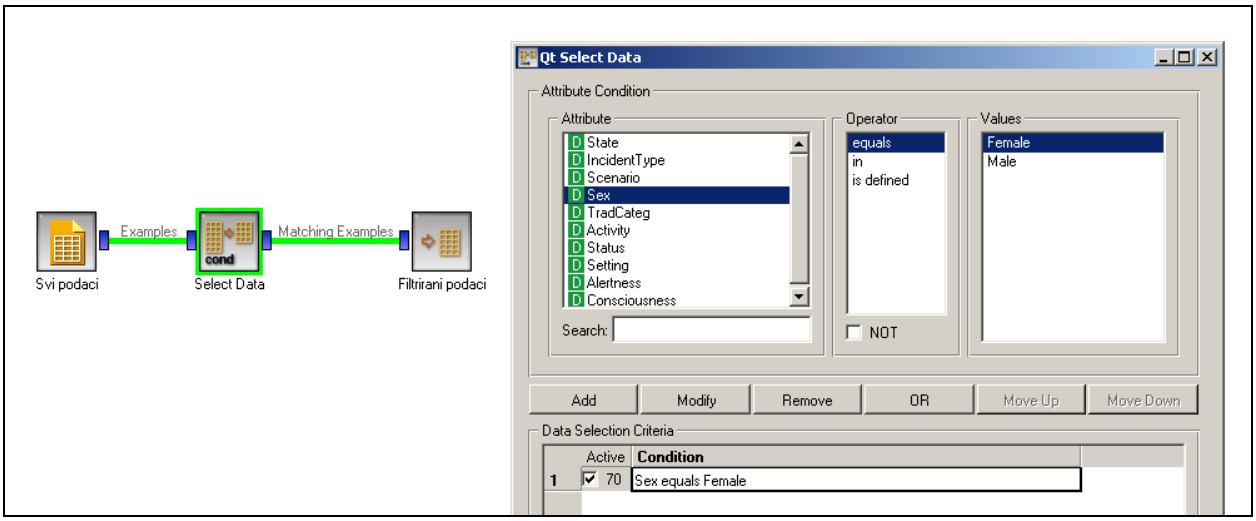
Slika 20: Tok za rešavanje problema nedostajućih vrednosti

Problem u izgradnji modela mogu napraviti i nestandardni podaci, koji se u statistici nazivaju autlajeri (*outliers*). Oni predstavljaju retke događaje, koji su izuzeci od pravila u podacima. Mogu ukazati na greške, ali mogu biti i jednostavno slučajevi koji se razlikuju dosta od ostalih slučajeva, iz drugih razloga. Pošto su oni izuzeci od pravila, mogu uticati da izgrađeni model nad svim podacima ne bude kvalitetan, jer je algoritmima teško da uoče pravilnosti u prisustvu izuzetaka. Način da se u programu autlajeri otkriju i uklone je korišćenje čvora *Outliers*. Primajući slučajeve na ulazu, ovaj čvor izdvaja podatke koji se smatraju autlajerima (statistički) koji se onda mogu ukloniti iz ukupnog skupa podataka. Primer toka koji koristi ovaj čvor je dat na Slici 21.



Slika 21: Tok za oktrivanje i izolovanje autlajera

Greške u podacima i autlajeri se mogu otkriti i na druge načine, na primer raznim vizuelizacijama ili pregledanjem podataka. Uočeni nedostaci se mogu otkloniti *Select Data* čvorom. Taj čvor nudi mogućnost filtriranja podataka koji zadovoljavaju neki uslov, pa se u uslovu mogu definisati problemi u podacima koji će biti filtrirani. Primer toka i definicije uslova za filtriranje *Select Data* čvorom dat je na Slici 22.



Slika 22: Tok i definisanje filtriranja podataka po uslovu

Pre puštanja algoritama za otkrivanje znanja, moguće je i izvršiti određene transformacije podataka, među kojima i konverzije tipova atributa. U programu postoje dva čvora, *Discretize* i *Continuize*, koji omogućavaju da se numerički atributi pretvore u kategoričke (diskretne), kao i da se kategorički pretvore u numeričke (kontinualne), respektivno. Potreba za konverzijom tipova je najčešće uslovljena ograničenjima algoritama, poput algoritama za pronalaženje asocijativnih pravila.

Zaključak

Program Orange predstavlja platformu za izgradnju procesa otrkivanja zakonitosti u podacima koja je dosta jednostavna, ali i moćna i vrlo ilustrativna za potrebe učenja ove oblasti. Iako je program akademski i besplatan, okruženje dosta podseća na komercijalne alate i merljivo je sa njima. Jedini nedostatak programa je izostanak kvalitetne dokumentacije, kao i zajednice koja bi ovaj program podstakla na brži razvoj.