INFORMACIONI SISTEMI ZA PODRŠKU MENADŽMENTU

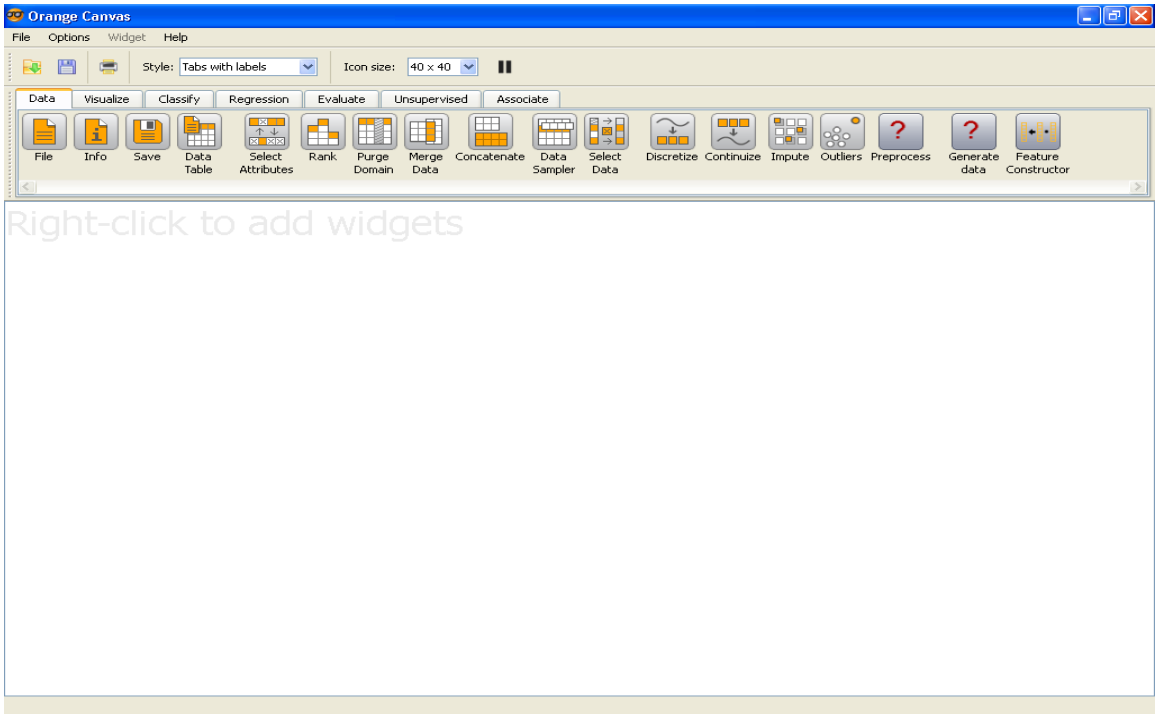
OBLAST:	**Classification**
ČVOROVI (WIDGET):	**Classification Tree, K-NN, Test learners, Predictions**
SKUPOVI PODATAKA:	**Titanic**
AUTOR:	**Jovana Mina Runić 141/07**

УНИВЕРЗИТЕТ У БЕОГРАДУ
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

2011, Beograd

Naredna skripta ilustrativno opisuje četiri čvora softvera Orange, primenjenog na bazi podataka „Titanik“.
Čvorovi koji će biti obrađeni su : Classification Tree, K Nearest Neighbours, Predictions i Test Learners.

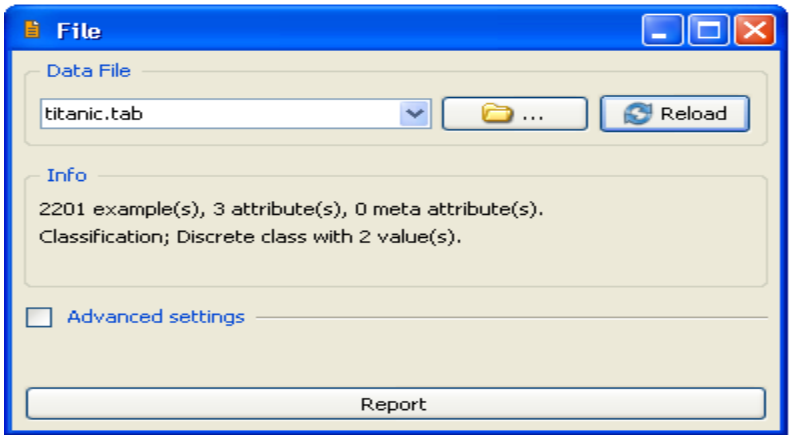
Nakon pokretanja programa Orange, pred nama se nalazi početni ekran aplikacije. (Slika 1.)



Slika 1. Početni izgled ekrana u softveru Orange

Nakon otvaranja aplikacije, izborom čvora 

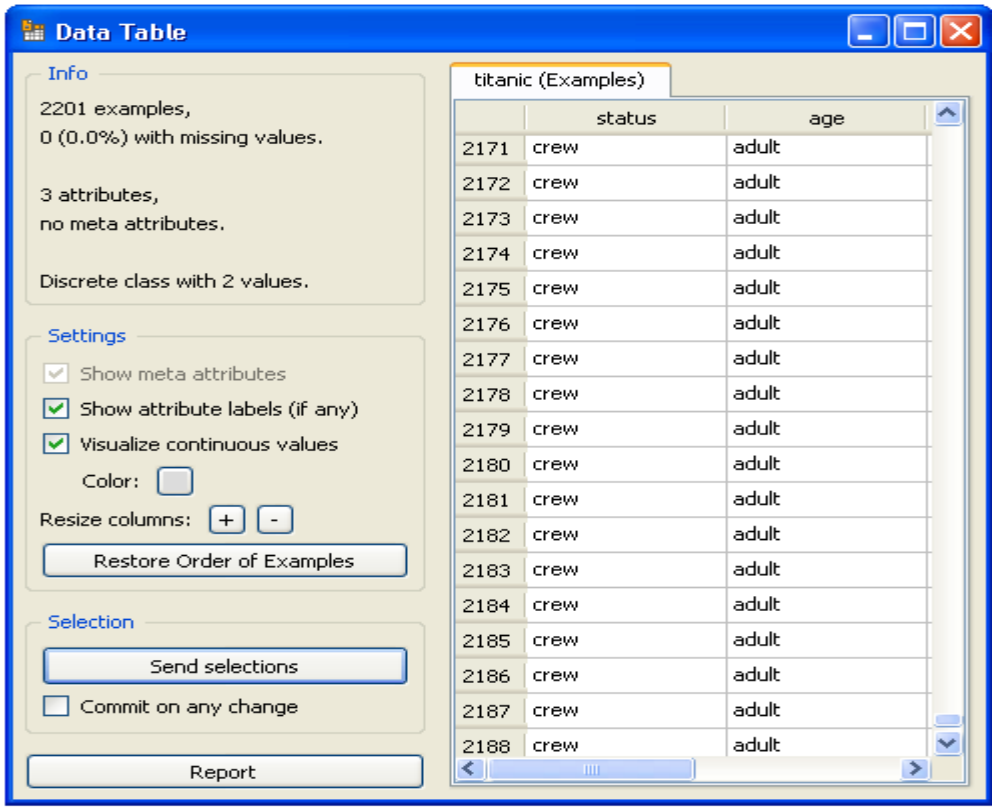
otvara nam se prozor u kome izaberemo bazu podataka koju ćemo koristiti, u ovom slučaju Titanik. (Slika 2.)



Slika 2. Izbor baze podataka

Povezivanjem čvora File sa čvorom 

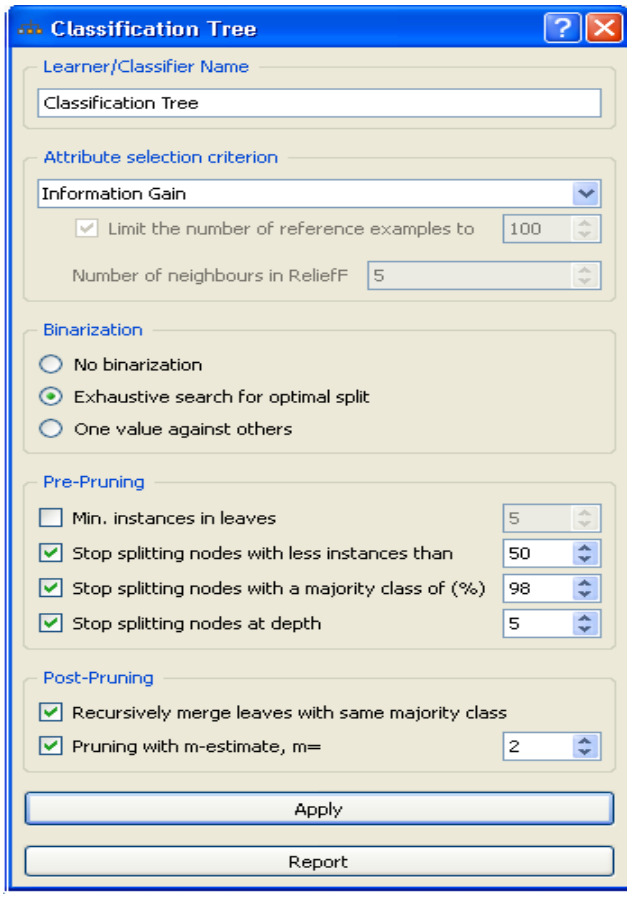
možemo da vidimo na pregledan način sa kojim podacima raspolažemo. U ovom primeru imamo 2201 slučaj i tri atributa (status, starost i pol), a izlaz pokazuje da li je osoba preživela ili ne. Nedostajućih podataka nema. (Slika 3.)



Slika 3. Pregled podataka

Kako bi izvršili klasifikaciju, gde povezujemo ulazne atribute sa izlaznim, kategoričkim (klasnim) atributom, da bi otkrili zakonitost po kojoj se određeni objekat svrstava u određenu klasu u okviru kartce Classify sa Slike1, biramo čvor 

i nakon povezivanja sa bazom podataka otvara nam se prozor u kome možemo izvršiti odgovarajuća podešavanja stabla (Slika 4):



Slika 4. Uređivanje stabla

Najpre biramo kriterijum selekcije atributa (**Information Gain**, **Gain Ratio**, **Gini Index**, **ReliefF**) da bi odlučili koji atribut treba da se koristi kao čvor pri račvanju stabla.

-Informaciona dobit nam pokazuje koliko se entropija sistema smanjuje, ako se za odlučivanje koristi određen atribut, tj. koji atribut nosi najviše informacija.

- Racio dobiti je sličan kriterijumu informaciona dobit, ali uzima u obzir broj kategorija koji poseduje određeni atribut i pogodniji je kada postoji veliki broj kategorija koje mogu uzeti atributi, što nije slučaj u ovom primeru.

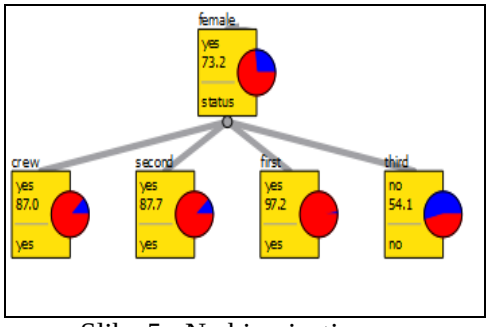
- Đini indeks meri nečistoću atributa, odnosno nemogućnost predviđanja izlaznog atributa na osnovu ulaznog.

- **ReliefF** meri korisnost atributa za račvanje na osnovu njegove sposobnosti da razgraniči jako slične slučajeve koji pripadaju različitim klasama.

Usled jednostavnosti baze podataka, ovde ćemo koristiti kriterijum informacione dobiti.

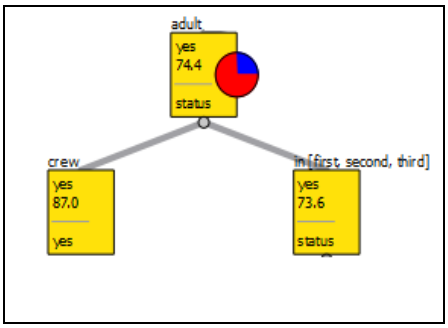
U okviru polja **Binarization** možemo izabrati neku od sledećih opcija :

- No binarization - ova opcija nam omogućava da stablo račvamo na maksimalan broj grana, koliko ima vrednosti u atributu. (Slika 5.)



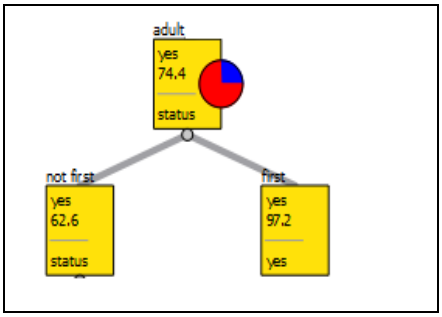
Slika 5. No binarization

- **Exhaustive search for optimal split** – ova opcija omogućava binarno račvanje stabla pretražujući sve kombinacije kako se vrednosti atributa mogu grupisati u dve grane i to na način da vrednosti po granama budu „optimalno“ grupisane, a na osnovu procenjivanja svih mogućih grupisanja. (Slika 6.)



Slika 6. Exhaustive search for optimal split

- **One value aganist others** – ova opcije takođe omogućava binarna račvanja stabla pri čemu je jedna grana vrednost samo jednog atributa, a druga grana je grupisanje svih ostalih vrednosti. (Slika 7.)



Slika 7. One value aganist others

Željeno orezivanje stabla tokom njegove izgradnje se vrši u polju **Pre Pruning**, gde možemo definisati :

- Minimalan broj slučajeva u listu (krajnjem čvoru u stablu odlučivanja) ;

- Prekid granjanja stabla kad broj slučajeva padne na određen nivo (u zavisnosti od broja slučajeva u bazi podataka definišemo broj slučajeva pri kome će stablo prestati sa grananjem, ali pri tom imati u vidu koliko je taj broj adekvatan za donošenje zaključka);

- Prekid granjanja stabla kad najzastupljenija klasa dostigne određen procenat ;

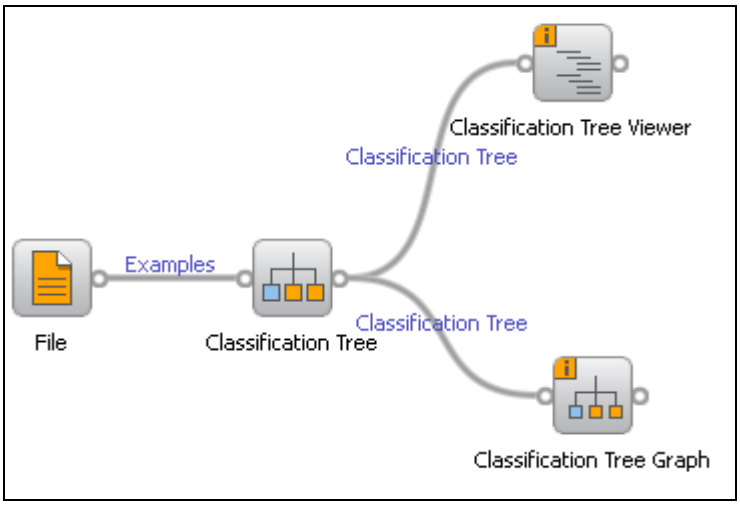
- Prekid granjana stabla na određenoj dubini (broju nivoa).

Kako bi nakon rasta stabla, našli stablo manjih dimenzija koje je ili povećalo ili u mimimalnoj meri izgubilo tačnost klasifikacije originalnog stabla, u okviru polja **Post-Pruring** izabraćemo prikazane opcije.

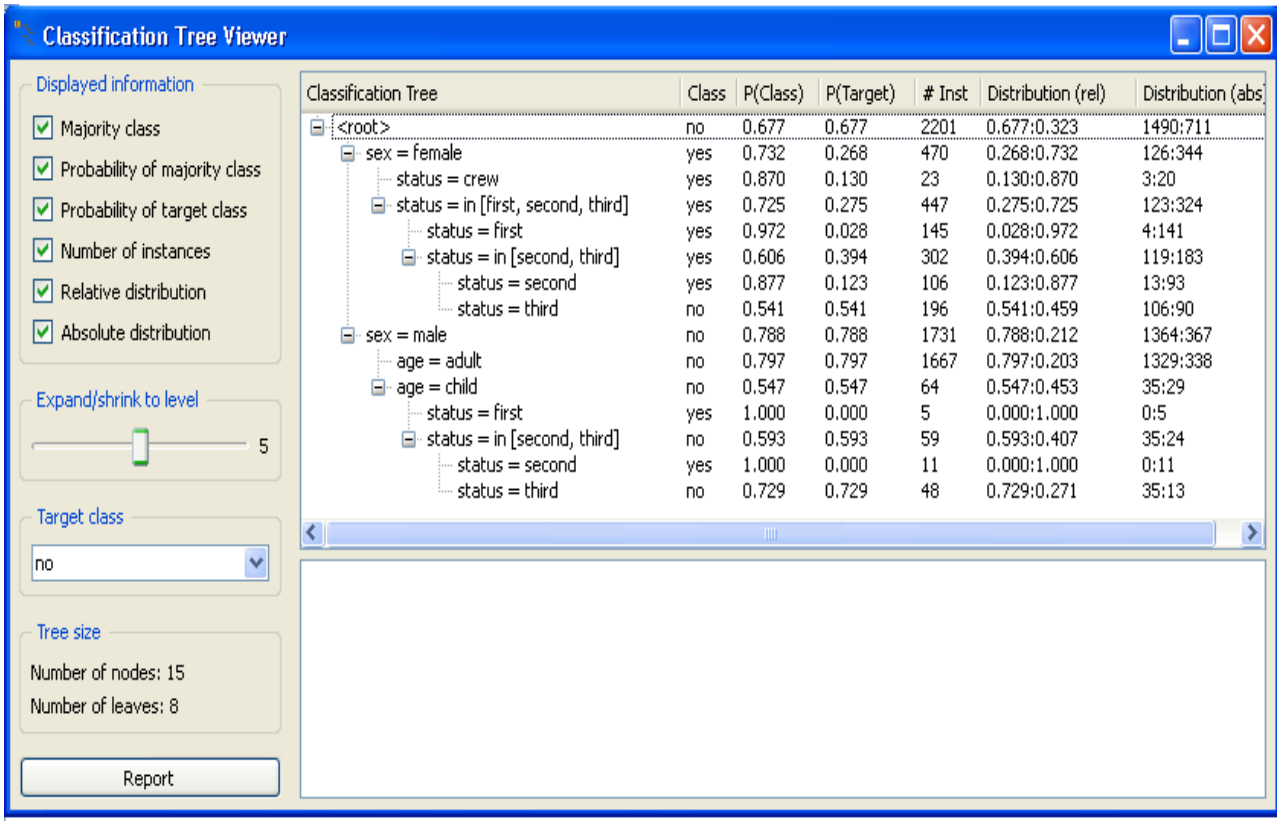
Klikom na Apply završavamo sa uređivanjem stabla, nakon čega biramo čvor 

ili 

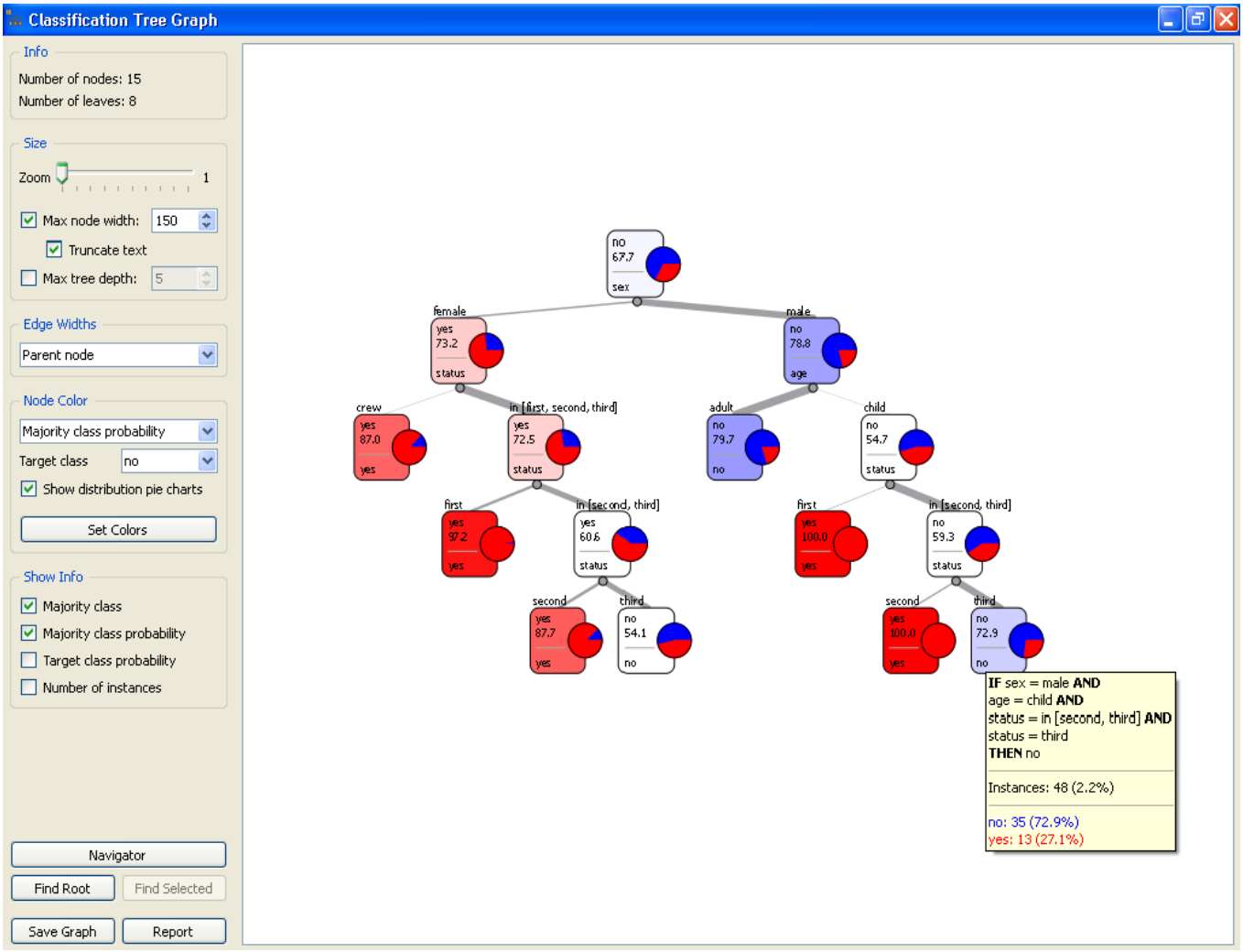
koje povezujemo sa čvorom Classification Tree (Slika 8.) da bismo videli izgrađeno stablo odlučivanja. (Slika 9. i Slika 10.)



Slika 8. Povezivanje čvorova



Slika 9. Classification Tree Viewer



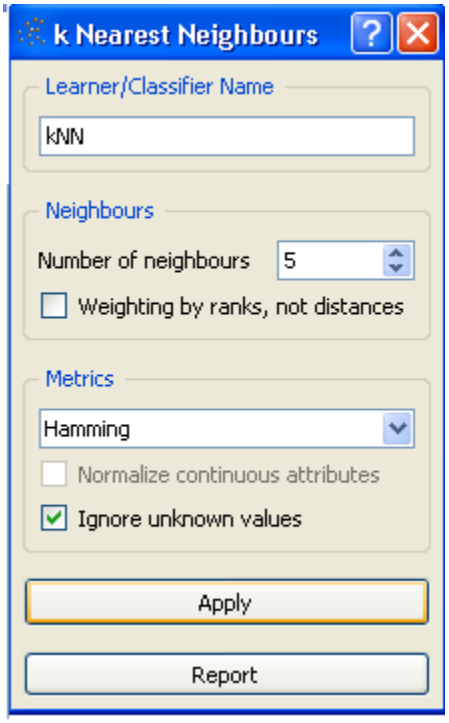
Slika 10. Classification Tree Graph

Krećući se od korena do listova stabla vidimo da kad je osoba bila ženskog pola i u posadi preživela je (njih 87,7%), ako je bila u prvoj klasi 92,2%, u drugoj 87%, a u trećoj nije. S obzirom da smo koristili kriterijum informacione dobiti stablo je na osnovu atributa pol i status imalo dovoljno informacija za dobijanje krajnjeg rezultata, bez uključivanja atributa uzrast. S druge strane, ako je osoba muškog pola i odrasla nije preživela, a ako je bila dete i iz prve klase preživela je, isto važi i za drugu klasu, ali opet u trećoj klasi nailazimo na to da osoba nije preživela.

Imajući u vidu da smo izabrali opciju Exhaustive search for optimal split (Slika 4.), stablo se binarno račva pri čemu vidimo da za atribut *status* jedna grana uzima vrednosti *crew*, dok druga ostale vrednosti tog atributa, odnosno *first, second, third.*. Zatim stablo nastavlja da raste račvajući se i na te vrednosti. Takođe, definsano je da stablo prestane sa račvanjem kad broj slučajeva padne na manje od 50, kada najzastupljenija klasa dostigne 98% i na dubini od 5 nivoa, što se jasno vidi na prethodnoj slici.

Stablo odlučivanja nam na ovaj način omogućava hijerarhijski uređeno, pregledno i jednostavno tumačenje.

Sada prelazimo na čvor u okviru kartice Classify, koji slično metodi zaključivanja na osnovu slučajeva traži za svaki entitet u bazi određen, unapred definisan broj njegovih najbližih suseda koristeći neku od ponuđenih metrika, dajući pritom jedan izlaz kao prosek izlaza tih najbližih suseda. Metod najbližih suseda nam ustvari određuje sličnost jednog slučaja sa ostalim, zbog čega je pogodan za relativno male baze podataka, s obzirom da je brzina metode linearno zavisna od broja slučajeva.
Na narednoj slici je prikazan prozor koji se otvara pokretanjem ovog čvora, gde možemo izvršiti određena podešavanja :



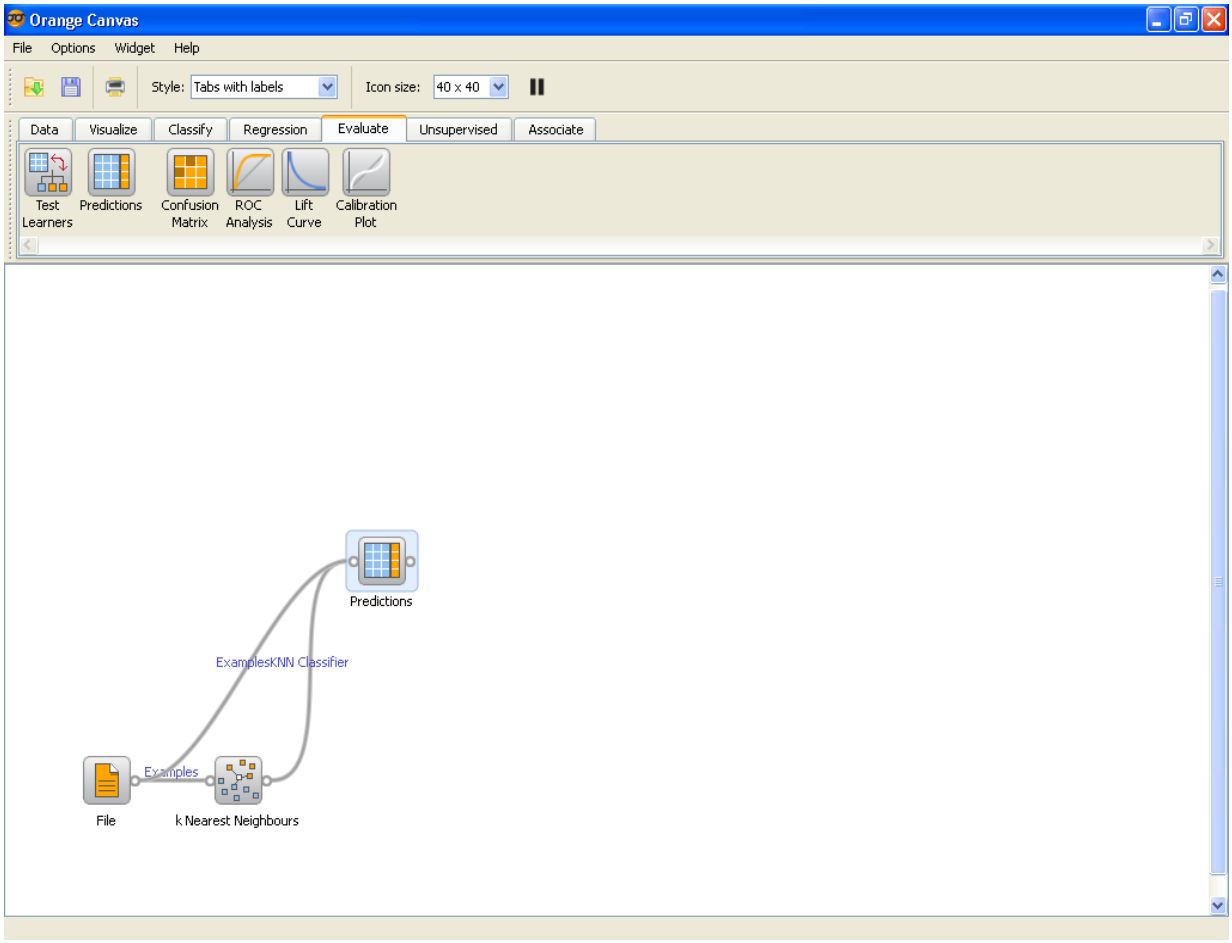
U polju **Neighbours** definišemo željeni broj najbližih suseda, kao i to da li želimo da se blizina meri na osnovu rangova, a ne udaljenosti.

U polju **Metrics** imamo ponuđene metrike za računanje blizine (**Euclidean, Hamming, Manhattan, Maximal**).
-Euklidska metrika definiše razdaljinu između dve tačke podataka računanjem kvadratnog korena zbira kvadrata razlika između odgovarajućih vrednosti. Koristi se kao mera odstojanja kod numeričkih podataka.
-Hamming metrika za računanje udaljensti meri broj atributa na osnovu kojih se slučajevi razlikuju, pritom uzima vrednosti : 0 kada su atributi isti i 1 kada se atributi razlikuju ;
- Manhattan metrika (odstojanje tipa „gradskog bloka“) za računanje udaljenosti koristi sumu apsolutnih razlika između atributa ;
- Maximal koristi maksimalne razlike između svih atributa.

Pritom se mora voditi računa koja metrika radi sa numeričkim, a koja sa kategoričkim podacima. Ovde je izabrana Hamming metrika, koja se izdvaja kao najbolja za rad sa diskretnim vrednostima.
Takođe, izabrana je opcija ignorisanja nepoznatih vrednosti.

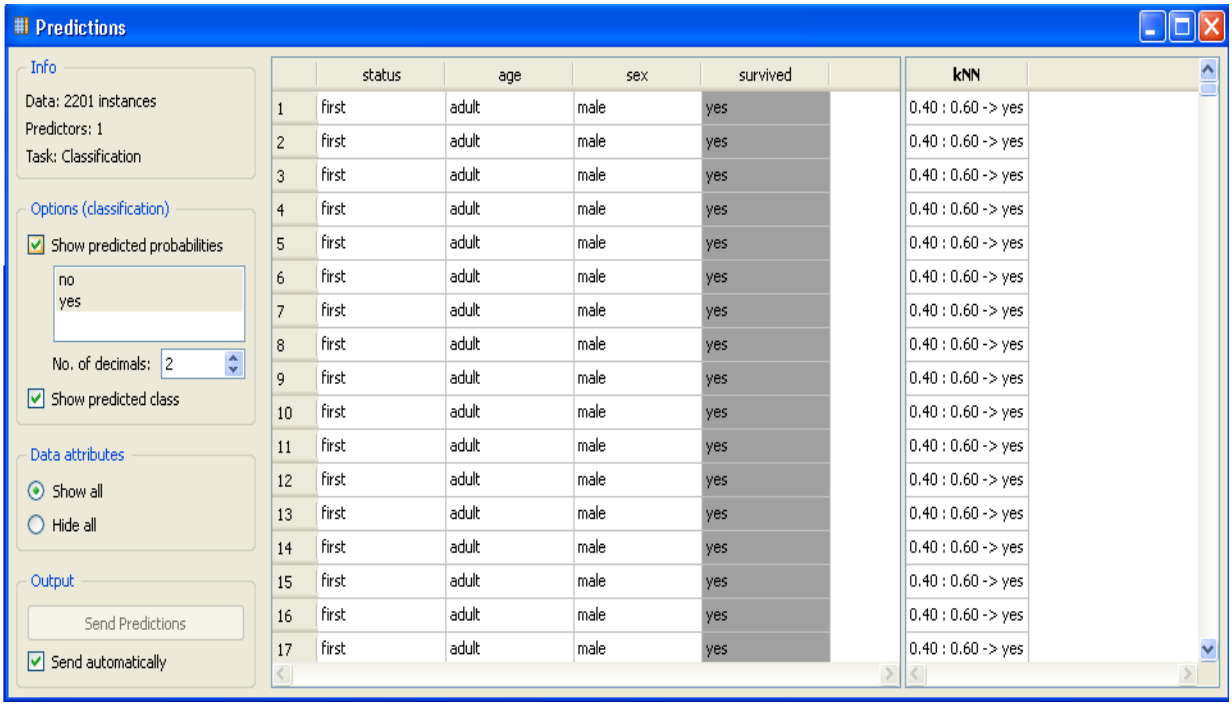
Slika 11. Čvor k Nearest Neighbours

Nakon toga u okviru kartice Evaluate biramo čvor koji povezujemo sa bazom podataka i čvorom k Nearest Neighbours. (Slika 12.) Čvor Predictions može se povezati i sa bazom i čvorom Classification Tree. Ovde je prikazan rezultat u slučaju povezivanja sa čvorom k Nearest Neighbours. Inače čvor Predictions koristimo za prikazivanje i upoređivanje izlaza koje nam pruža baza podataka i primenjen model.



Slika 12. Prikaz povezanih čvorova

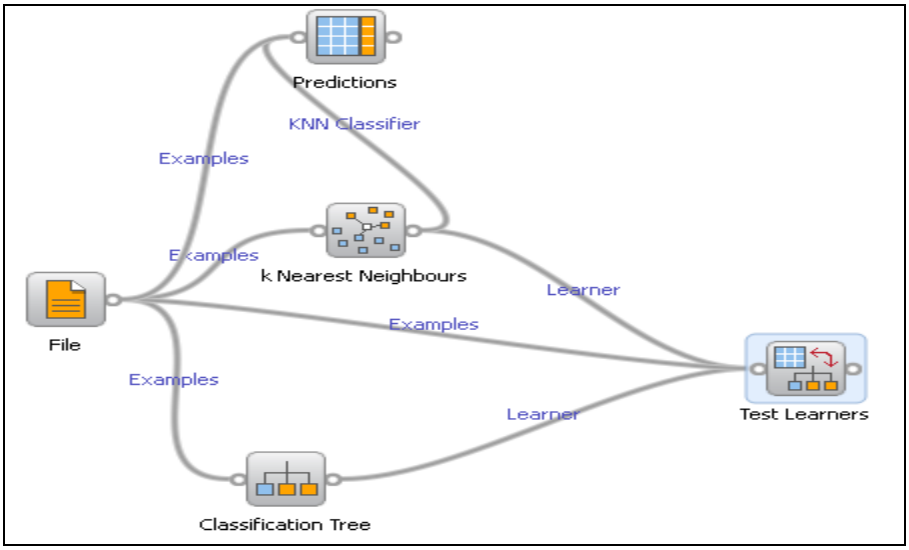
Otvaranjem čvora Predictions možemo sagledati odgovarajuće izlaze. (Slika 13.)
Naredna slika pokazuje na pregledan način atribute, izlaze u bazi i izlaze našeg modela. Može se videti gde se ti izlazi poklapaju, a gde ne, kao i predviđene verovatnoće svake klase. Ovaj čvor koristimo i kada imamo nove slučajeve pa treba da utvrdimo kojoj klasi pripadaju. Poredeći novi slučaj sa ostalima iz baze, možemo preko čvora Predictions da izvršimo i klasifikaciju novog slučaja.



Slika 13. Prikaz izlaza

Sledeći čvor koji će biti obrađen je 

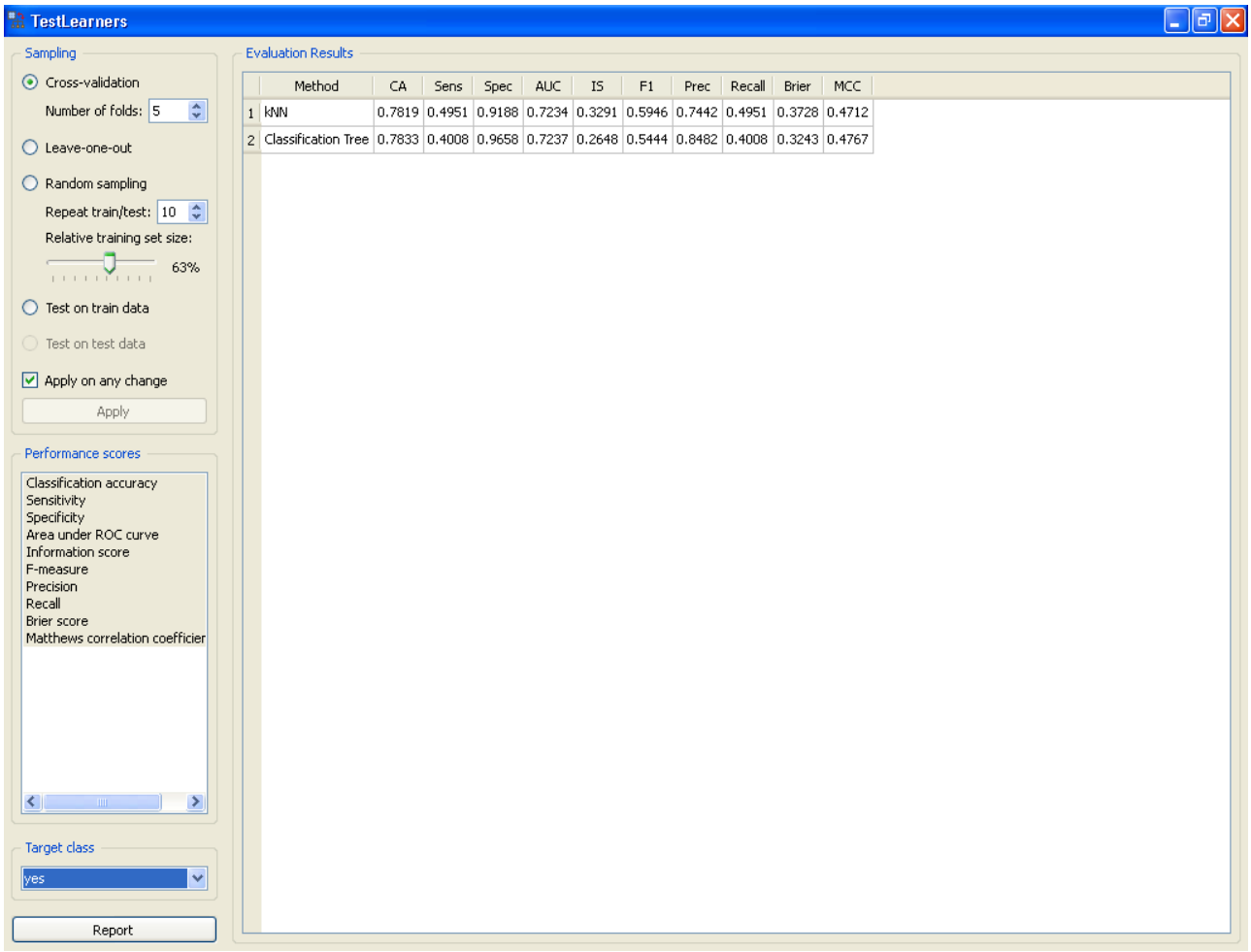
u okviru kartice Evaluate i on takođe može biti povezan sa bazom i čvorom Classification Tree, kao i čvorom k Nearest Neighbours, što je ovde slučaj. (Slika 14.)



Slika 14. Prikaz čvorova

Test Learners nam omogućava da sagledamo i uporedimo različite modele koje smo koristili, kao i da vidimo koliko smo u pravu kad je u pitanju naše predviđanje, a na osnovu odgovarajućih pokazatelja. Takođe, podržava različite metode uzorkovanja, omogućavajući nam da određen deo podataka bude za treniranje, generisanje modela, a deo za testiranje, proveru ispravnosti modela. U okviru polja **Samling** možemo izabrati : (Slika 15.)

- **Cross-validation** se koristi za proveru tačnosti modela i deli celokupne podatke u više delova. Algoritam se testira na jednom delu, a na ostalim trenira, ponavljajući taj postupak na svim delovima.
- **Leave One Out** je metod sličan prethodnom, ali se koristi kad imamo malo podataka, tj. mali uzorak, a potrebna nam je velika tačnost.
- **Random sampling** omogućava da proizvoljno izaberemo npr. 70% uzorka za treniranje, a 30% za testiranje.



Slika 15. Test Learners

Odgovarajući pokazatelji su :

- **Classification accuracy** nam pokazuje procenat tačno klasifikovanih slučajeva, tj. stavlja u odnos broj predviđenih i stvarnih slučajeva, što u našem primeru, gde je korišćena klasa preživeli, znači da je kNN u 78,19% od ukupnog broja slučajeva predvideo tačno, a Classification Tree u

78,33% slučajeva. Obzirom na ove rezultate treba nastojati da se model poboljša i dostigne viši nivo tačnosti.

- **Senzitivity** nam pokazuje procenat tačnih pozitivno predviđenih slučajeva u onosu na stvaran broj pozitivnih slučajeva, tj. pokazuje sposobnost testa da identifikuje one slučajeve koji su stvarno preživeli (koliko puta je model rekao da kada je trebao da kaže da).
U našem slučaju kNN je u 49,51% slučajeva tačno predvideo da će osoba preživeti od ukupnog broja slučajeva koji su preživeli, a Classification Tree u 40,08% slučajeva, pokazujući malo lošiju tačnost. I u ovom slučaju model pokazuje relativno nizak procenat tačnih klasifikacija.

- **Specificity** suprotno Senzitivity-u pokazuje procenat negativno predviđenih slučajeva u odnosu na stvaran broj negativnih slučajeva, tj. pokazuje sposobnost testa da identifikuje one slučajeve koji stvarno nisu preživeli (koliko puta je model rekao da nije da kada je trebao da kaže da nije da).

- **Area under ROC curve** se koristi kod ROC analize (razvija se kriva u koordinatnom sistemu gde ordinata predstavlja Sensitivity, a apcisa Specificity, odnosno stopu lažno pozitivnih). Cilj je da ova vrednost bude što veća jer bi to značilo da je ROC kriva više pomerena ka gornjem levom uglu što znači veću senzitivnost. Pojednostavljena definicija AUC- a bi bila da je to verovatnoća da ćemo jedan slučaj, koristeći model, više tačno da predvidimo nego netačno. Ovaj pokazatelj se često i koristi pri komparaciji modela. Za uzumanje neke od sledećih vrednosti se može reći da je izvršena klasifikacija :
  0,90-1,00 – Odlična
  0,80-0,90 – Vrlo dobra
  0,70- 0,80 – Prosečna
  0,60-0,70 – Loša
  0,50-0,60 – Jako loša

Ovde oba modela sa vrednostima, kNN-0,7234 i Clas.Tree-0,7237 daju prosečne rezultate, odnosno verovatnoće da će tačno predvideti slučaj.

- **Information score** nam pokazuje prosečan iznos informacija po svakom klasnom slučaju.

- **F-measure** predstavlja harmonijsku sredinu pokazatelja Precision i Recall. Uzima vrednosti od 0 do 1, pri tom 0 označava najlošiju, a 1 najbolju tačnost klasifikacije. S obzirom da kNN ima vrednost 0,5946, a Clas. Tree 0,5444 može se reći da je klasifikacija i ovde prosečna.

- **Precision** pokazuje za svaki izlazni atribut procenat pozitivno predviđenih slučajeva u odnosu na ukupan broj slučajeva koje smo klasifikovali datim izlazom. Tj. u našem primeru može se reći da je kNN od ukupnog broja ljudi koje je klasifikovao da su preživeli u 74,42% slučajeva bio u pravu, a Clas.Tree pokazuje bolji rezultat od 84,42%.

- **Recall** predstavlja isto što i Senzitivity.

- **Brier score** predstavlja prosečno odstupanje između predviđene i stvarne verovatnoće odigravanja događaja. Poželjno je da ta vrednost bude što manja.U ovom primeru se može reći da je zadovoljavajuća, gde kNN iznosi 0,3728, a Clas.Tree 0,3243.

- **Matthews correlation coefficient** predstavlja koeficijent korelacije između predviđenih i stvarnih izlaza. Može uzeti vrednosti od -1 do 1, pri tom -1 je loše, 0 prosečno i 1 odlično predviđanje. Ovde i kNN-0,4712 i Clas.Tree-0,4767 pokazuju prosečno do odlično predviđanje.

Na osnovu prethodnog sagledavanja rezultata, model u većini slučajeva ne pokazuje zadovoljavajuće rezultate obzirom na vrednosti koje pokazatelji mogu uzeti.
Međutim, uvek treba imati u vidu da je donosilac odluka taj, koji poredeći vrednosti pokazatelja, može da uvidi kolika je tačnost modela koji su mu na raspolaganju i na njemu ostaje da utvrdi da li model koji npr. predviđa tačnost od 75% zadovoljava njegove kriterijume. tj. da li će taj model koristiti prilikom donošenja odluka ili će zahtevati odgovarajuće izmene i poboljšanja modela dok pokazatelji ne dostignu željenu tačnost.