

BIOLAB IN SODELAVCI

UVOD V ZNANOST O PODATKIH

BIOLAB

Avtorske pravice © 2021 Biolab in sodelavci

OBJAVLJENO S STRANI BIOLAB

TUFTE-LATEX.GOOGLECODE.COM

Licencirano pod licenco Apache, različica 2.0 ("licenca"); to datoteko lahko uporabljate zgolj v skladu z licenco. Kopijo licence lahko dobite na <http://www.apache.org/licenses/LICENSE-2.0>. Razen na zahtevo po veljavni zakonodaji ali podlagi pisnega dogovora, je pod licenco distribuirana programska oprema distribuirana na "KOT JE" PODLAGI, BREZ JAMSTVA ALI KAKRŠNIMI KOLI POGOJI, izrecnimi ali implicitnimi. Glej licenco za določen jezik, ki ureja dovoljenja in omejitve pod licenco.

Prvi natis, December 2021

Kazalo

<i>Delotoki v Orangeu</i>	5
<i>Raziskovanje podatkov</i>	8
<i>Shranjevanje delotokov</i>	11
<i>Nalaganje podatkov</i>	13
<i>Hierarhično razvrščanje v skupine</i>	15
<i>Raziskovanje gruč</i>	17
<i>Vaja: politika</i>	20
<i>Klasifikacija</i>	21
<i>Klasifikacijska drevesa</i>	22
<i>Pregled modela</i>	24
<i>Naivni Bayes</i>	25
<i>Logistična regresija</i>	27
<i>Klasifikacijska točnost</i>	28
<i>Kako goljufati</i>	30

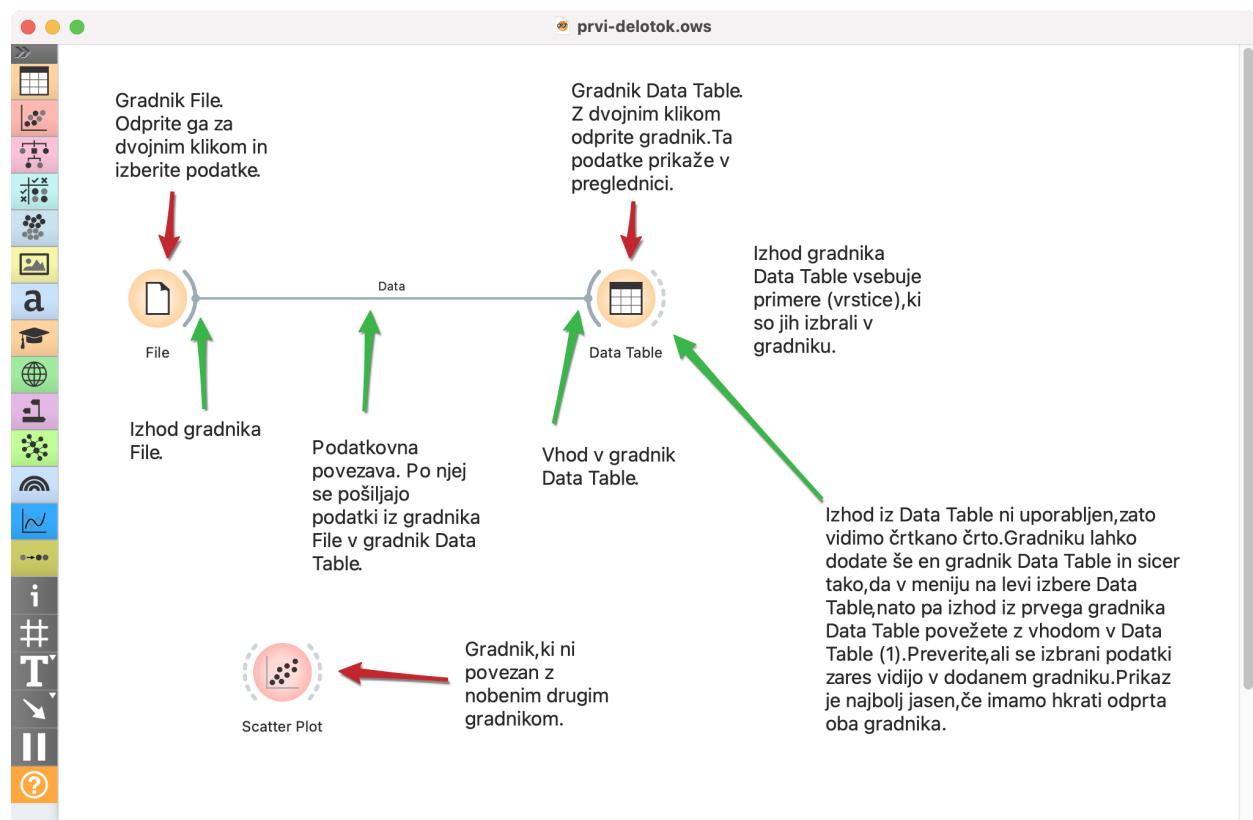
Prečno preverjanje 33

Vaja: kadrovska 34

Literatura 35

Delotoki v Orangeu

DELOTOKI V ORANGEU so sestavljeni iz komponent, ki berejo, procesirajo in prikazujejo podatke. Te komponente imenujemo gradniki. Na desni je prazen prostor, t.i. platno. Nanj polagamo gradnike. Gradniki v Orangeu komunicirajo preko komunikacijskih kanalov. Izhod iz enega gradnika je uporabljen kot vhod za drug gradnik.

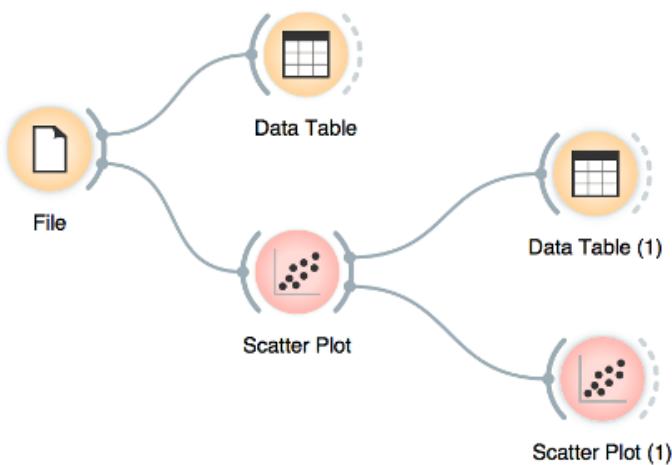


Delotoke sestavljamo tako, da polagamo gradnike na platno in jih povezujemo. Povezavo ustvarimo tako, da potegnemo črto od izhodnega v vhodni gradnik. Izhodi gradnika so na desni, vhodi pa na levi strani. V zgornjem delotoku gradnik *File* pošilja podatke v gradnik *Data Table*.

Slika zgoraj kaže preprost delotok z dvema povezanimi gradnikoma in enim gradnikom brez povezav. Izhodi gradnika so na desni strani, vhodi pa na levi.

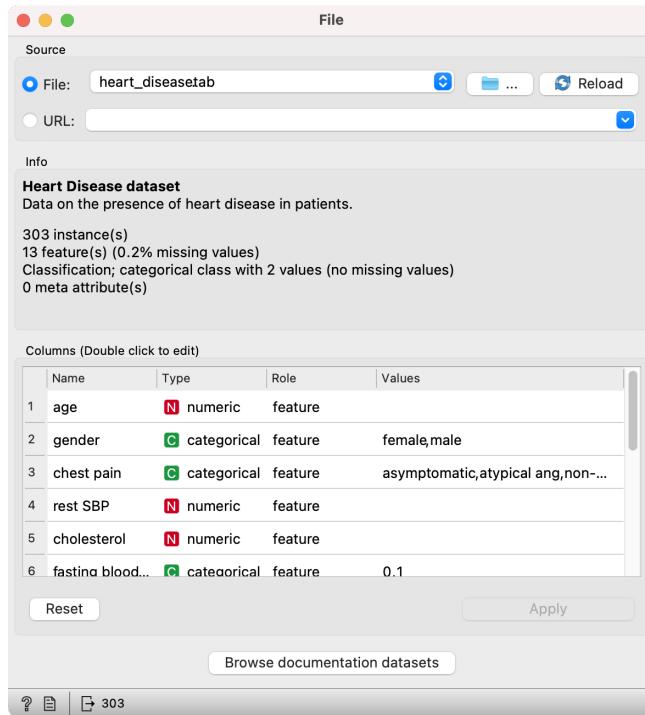
Pričnite z gradnjo delotoka, ki vsebuje gradnik File, dva gradnika Scatter Plot in gradnika Data Table:

Delotok z gradnikom File bere podatke iz računalnika in jih pošlje v gradnika Data Table in Scatter Plot. Data Table prikaže podatke v preglednici, Scatter Plot pa jih vizualizira. Izbrane točke iz Scatter Plota so poslane v naslednja gradnika, Data Table (1) in Scatter Plot (1).



Gradnik File bere podatke z lokalnega diska. Odprite File tako, da dvakrat kliknete na ikono. Orange že vsebuje nekaj prednaloženih korpusov. Iz spustnega menija izberite *heart-disease.tab*, podatke o prisotnosti srčno-žilnih bolezni pri pacientih.

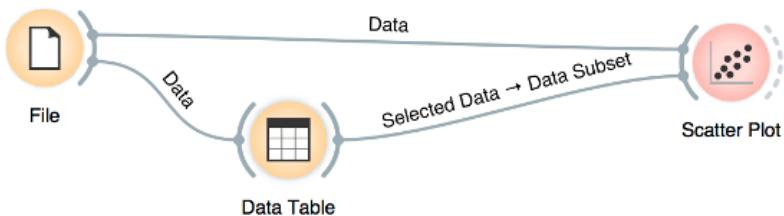
Oranjevi delotoki se pogosto pričnejo z gradnikom File. Podatki o srčno-žilnih boleznih vsebujejo 303 vrstice (paciente) in 14 spremenljivk. Od 14 stolpcev jih 13 vsebuje podatke o pacientu, en stolpec pa vsebuje informacijo o tem, ali ima pacient srčno bolezen ali ne.



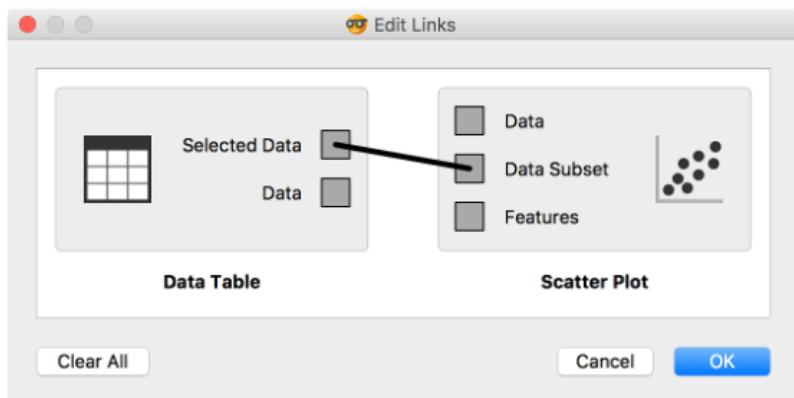
Ko naložimo podatke, odprimo preostale gradnike. V razsevnem diagramu izberimo nekaj točk in opazujmo, kako se pojavijo v gradniku Data Table (1). Uporabite kombinacijo dveh razsevnih diagramov, kjer drugi diagram prikaže podmnožico, ki ste jo izbrali v prvem.

Povežite izhod gradnika Data Table z gradnikom Scatter Plot (glej

spodaj). Odstranite preostala gradnike tako, da ju izberete in pritisnete Delete. Nato izberite vrstico v Data Table in preverite Scatter Plot. Primer, ki ste ga izbrali v preglednici, je sedaj označen v razsevnem diagramu! Uporabite puščice za sprehajanje po vrsticah ali pa s tipko Shift izberite več vrstic hkrati.



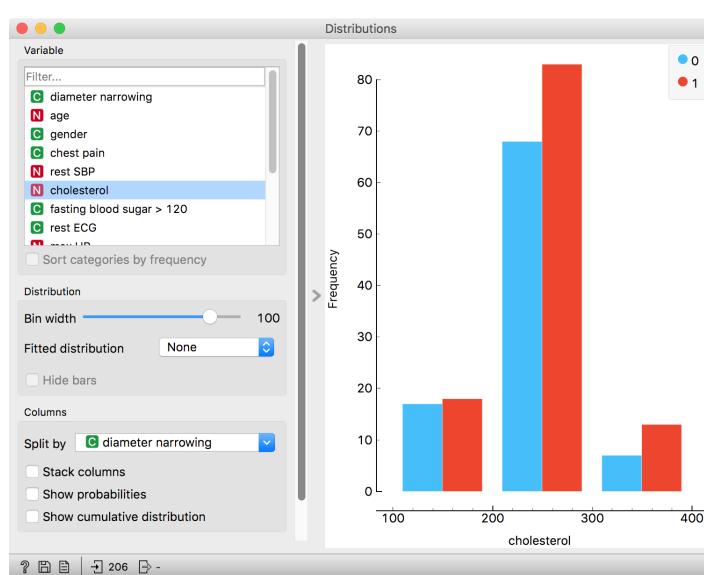
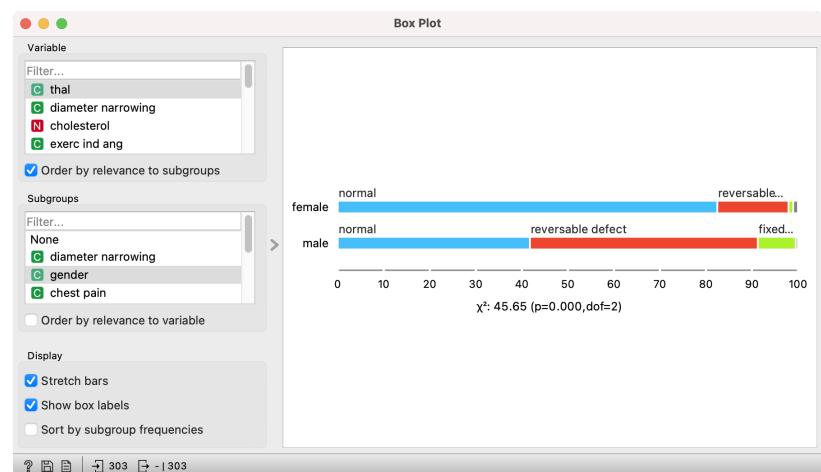
V zgornjem delotoku ima Scatter Plot dva vhoda, podatke iz gradnika File in izbor iz gradnika Data Table. Kako Orange razlikuje med primarnim virom podatkov in izborom? Prvi povezani signal uporabi kot celotne podatke, drugi signal pa kot podmnožico. Nastavite lahko spremenite ali preverite tako, da dvakrat kliknete na povezavo med gradnikoma.



Raziskovanje podatkov

SEDAJ BOMO ZAKOPALI GLOBOKO V PODATKE in pogledali, ali se v njih skriva kaj zanimivega. Najlažji način, da spoznamo podatke, je, da jih prikažemo v grafih. *Box Plot* (škatla z brki) nam pomaga odkriti osamelce, porazdelitve in zanimive spremenljivke.

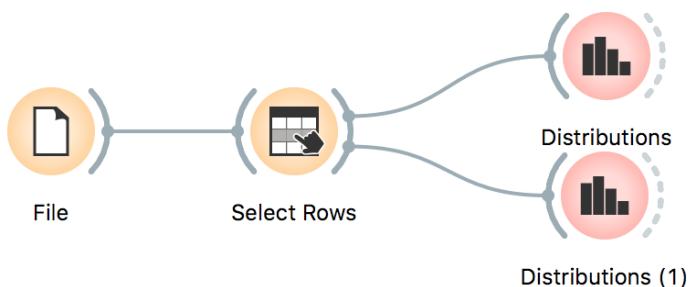
Poglejmo, ali nam Box Plot pove kaj zanimivega. Namig: poiščimo razlike med spoloma. Te so vedno zanimive in občasno celo resnične.



V razdelku *Subgroups* smo podatke razdelili po spolu (gender) in nato odklukali opcijo 'Order by relevance', ki spremenljivke razvrsti po tem, kako dobro ločijo med podskupinami. Moški v podatkih imajo slabše rezultate testa s talijem kot ženske. Poglejmo si še gradnik *Distributions*, ki je precej podoben Box Plotu. Prikazuje nam porazdelitve spremenljivk. Kakšna je porazdelitev starosti pacientov?

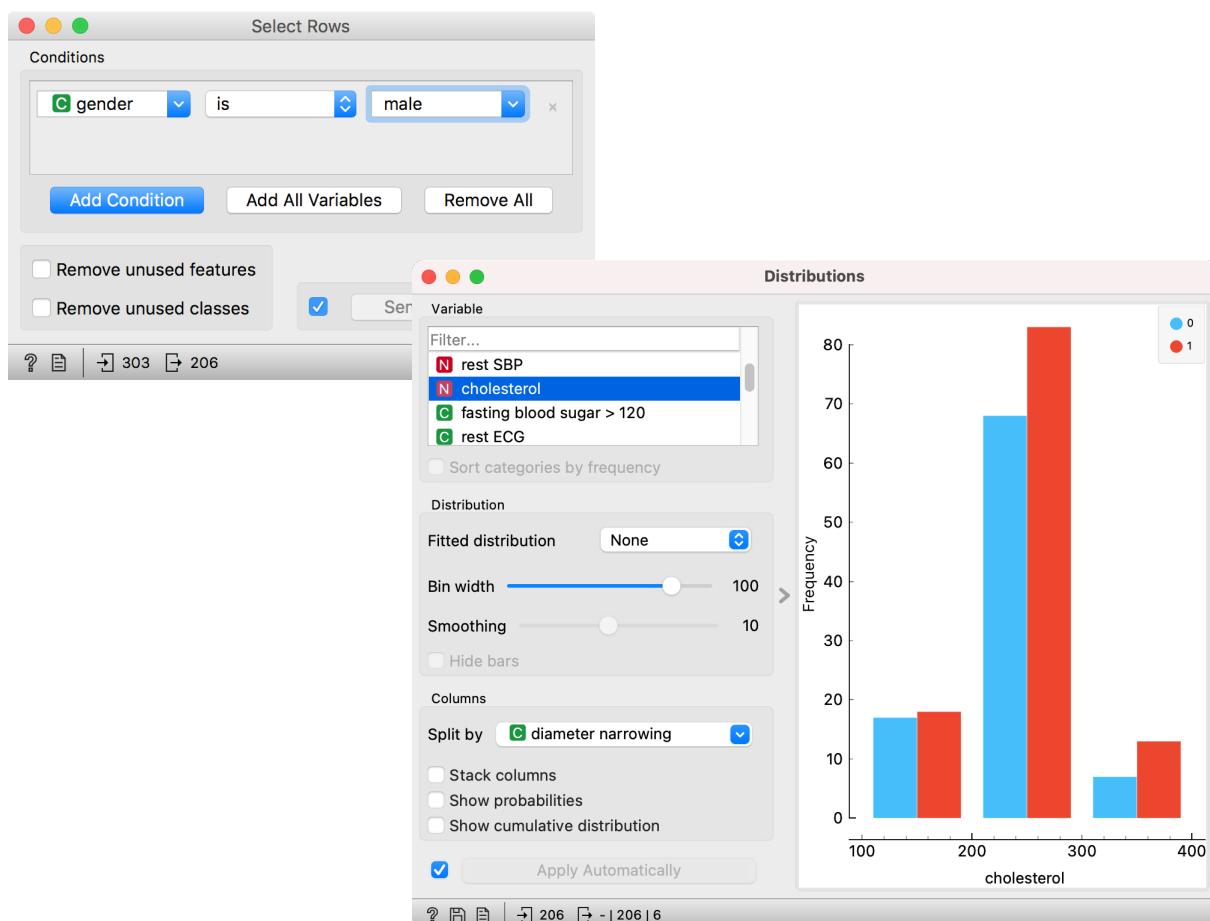
Podatke lahko razdelimo tudi po eni od spremenljivk — v našem primeru po spolu — in jih pregledamo ločeno.

V gradniku *Select Rows* izberemo moške paciente (gender is male), dodamo pa lahko še dodatne pogoje. Izbira podmnožic odlično deluje z vizualizacijami porazdelitev. Odprite oba gradnika hkrati in preiščite podatke.



Gradnika *Distributions* dobita različne podatke: zgornji dobi izbrane vrstice, spodnji pa preostale. Z dvojnim klikom na črto med gradniki lahko povezavo uredimo in Orangemu povemo, da želimo v spodnji gradnik poslati preostale podatke.

Tako Box Plot kot *Distributions* prikazujeta eno samo spremenljivko. Obstajajo pa tudi vizualizacije, ki lahko prikažejo več spremenljivk hkrati, tako da lahko vidimo povezave med njimi.

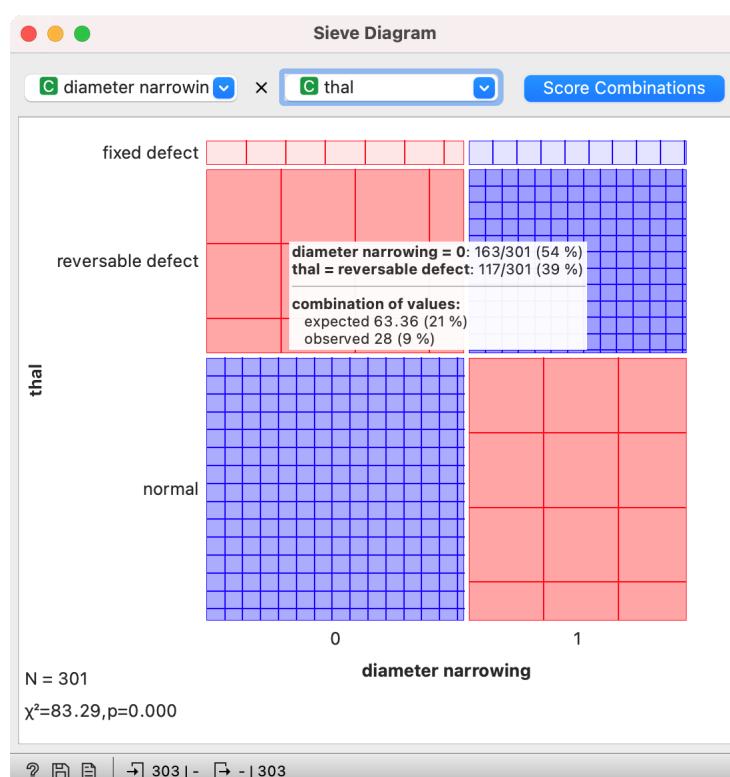




V gradniku lahko nastavite vizualizacije od ene do štirih spremenljivk. Preglejte različne vizualizacije.

Mosaic Display prikaže pravokotno razrezane stolpce, kjer je širina pravokotnika sorazmerna s pogostostjo različnih tipov bolečine v prsih. Vsak stolpec se nato naprej deli vertikalno po starosti. Pravokotniki se lahko delijo naprej še po spolu (y os). Znotraj pravokotnikov nam rdeča in modra polja pokažejo porazdelitev ciljne spremenljivke za podskupino, tanke črte ob strani pa prikažejo splošno porazdelitev.

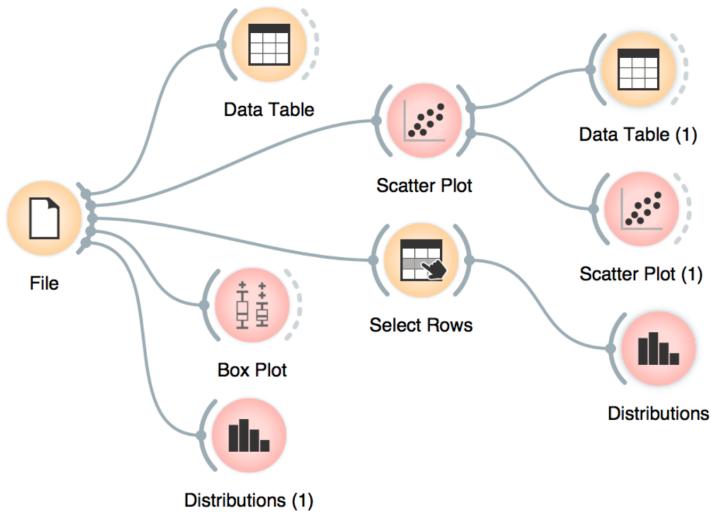
Kaj lahko razberemo iz tega diagrama?



Sieve Diagram takisto vodoravno in navpično deli pravokotnik, ki predstavlja množico pacientov, le da so rezi neodvisni, tako da je površina pravokotnika sorazmerna s pričakovanim številom pacientov, če predpostavimo, da so opazovane spremenljivke neodvisne. Npr. $1/2$ pacientov nima srčne bolezni in $2/5$ pacientov ima zaznano reverzibilno okvaro na testu s talijem. To pomeni, da je označen kvadrat velik $1/5$ celotne površine. Od približno 300 pacientov iz celotne zbirke bi v tem delu pravokotnika pričakovali $1/5 \times 300 = 60$ pacientov, ki nimajo srčne bolezni in imajo reverzibilno okvaro. Vendar je teh le 28. Ta kombinacija je torej redkejša, kot bi pričakovali. Sievov diagram tako kaže razliko med pričakovano in dejansko verjetnostjo z gostoto mreže in barvo polja.

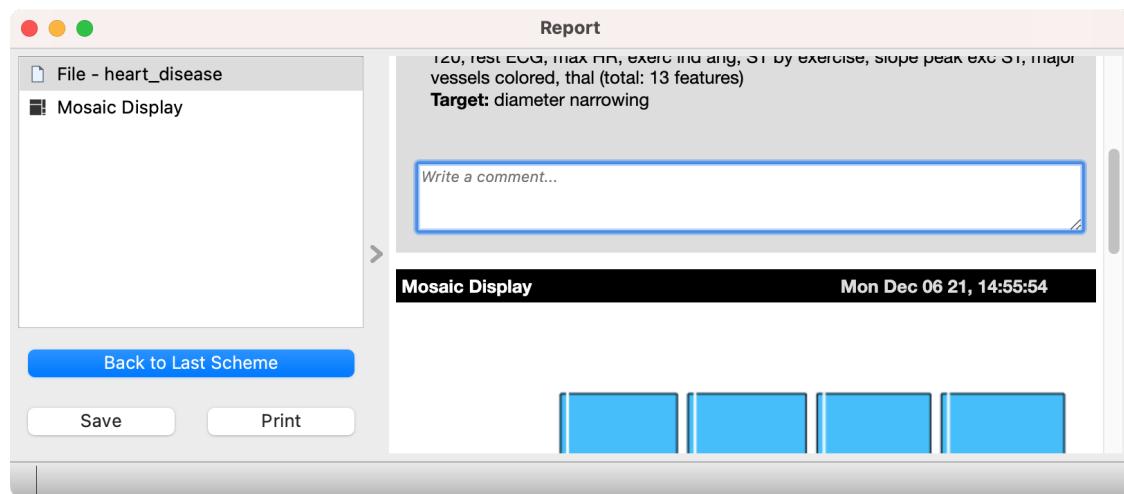
Shranjevanje delotokov

Če ste do sedaj sledili vsem navodilom - razen tistih seveda, kjer smo odstranjevali gradnike - potem vaš delotok izgleda nekako takole.



Še en trik: Ctrl-C (oz. or Cmd-C na Mac OS) prekopira vizualizacijo na odložišče (npr. iz Scatter Plota), od koder jo lahko s Ctrl-V (Cmd-V) prenesete v drugo aplikacijo (npr. Word).

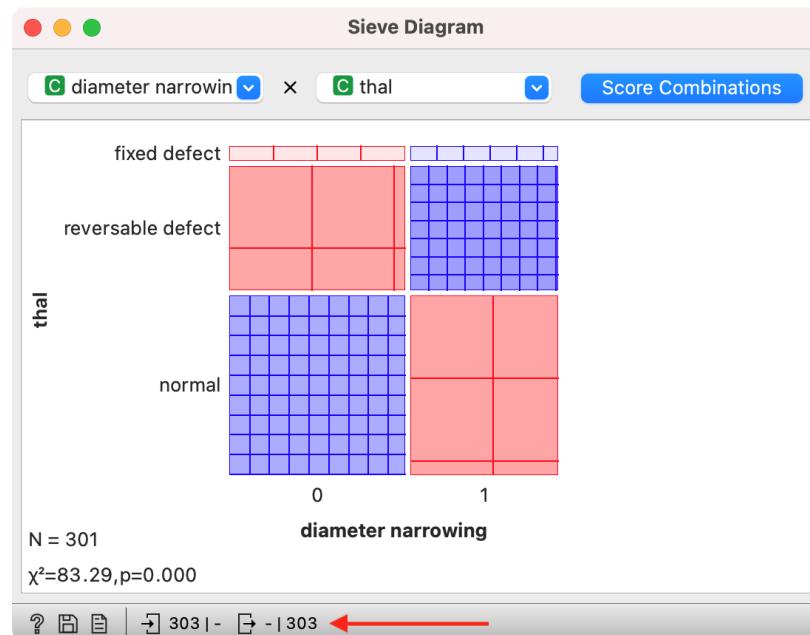
Delotok lahko shranite (File → Save) in ga delite s sodelavci. Vendar pa morate v isti direktorij priložiti tudi podatke, sicer bo Orange naložil zgolj shemo.



? | ⌂ | ↵ 303 | - | 303

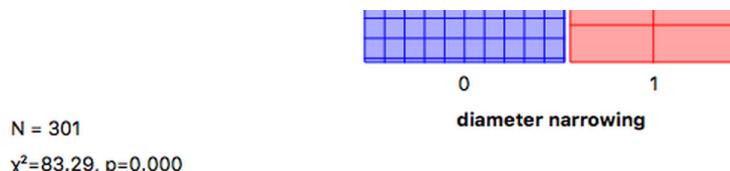
Orodna vrstica. Po vrsti si sledijo možnosti za pomoč, shranjevanje grafov in poročilo. V drugem delu vrstice so vhodi v gradnik in izhodi iz njega.

Gradniki imajo na dnu okna orodno vrstico. Tam lahko odpremo pomoč (opis uporabe gradnika), shranimo vizualizacijo in dodamo gradnik v poročilo.



Ko odkrijete kaj zanimivega v gradniku, kot je npr. nepričakovani Sievov diagram za paciente, kliknite *Report*. S tem dodate vizualizacijo v poročilo. V poročila lahko vnesemo vse gradnike, ki vodijo do zanimivega odkritja, saj tako natačno vemo, kako smo prišli do rezultatov.

Vsak del poročila omogoča tudi dodajanje komentarjev.



Očitno se s talijevim testom zlahka odkrije srčne bolezni, saj je precej več pacientov, ki so imeli odkrito napako na talijevem testu, tudi zares srčnih bolnikov.

Poročila lahko shranjujete v formatu HTML ali PDF ali pa kot datoteko, ki shrani vse delotoke, povezane z vizualizacijo. Te lahko kasneje tudi odprete in pregledate v Orangeu. Tako lahko vi in sodelavci enostavno ponovite analizo.

Nalaganje podatkov

Podatke, s katerimi smo delali do sedaj, smo dobili skupaj s programom Orange. Orange lahko bere tudi druge formate v obliki preglednic, npr. z vejico ali tabulatorjem ločene datoteke in Excelove preglednice. Pripravimo podatke (s šolskimi predmeti in ocenami) v Excelu in jih shranimo na računalnik.

V Orangeu podatke naložimo z gradnikom *File*.

The screenshot shows the Orange data mining software interface. On the left, there is a spreadsheet window titled 'Sheet1' containing data about students. The columns are labeled A through G, and the rows are numbered 1 through 8. The data includes student names (Jaka, Bine, Metka, Alenka, Maja) and their scores in English (angleščina), Mathematics (matematika), Physics (fizika), and Sports (športna vz.). Row 1 is a header row. The 'Študent' column is highlighted in green. The 'File' menu is open, showing the path 'File: študenti.xlsx'. The 'Info' section indicates there are 5 instances, 5 features, and 1 meta attribute. The 'Columns' section lists the attributes: angleščina, zgodovina, matematika, fizika, športna vz., and študent, with 'angleščina' through 'športna vz.' being features and 'študent' being a meta attribute. At the bottom, there are 'Reset' and 'Apply' buttons, and a link to 'Browse documentation datasets'.

	A	B	C	D	E	F	G
1	študent	angleščina	zgodovina	matematika	fizika	športna vz.	
2	Jaka	22	32	21	46	99	
3	Bine	91	65	89	11	29	
4	Metka	51	21	100	100	27	
5	Alenka	9	18	61	90	8	
6	Maja	93	39	12	17	63	
7							
8							

Orange je pravilno uganil, da so imena študentov besede in da je ta stolpec v podatkih nekaj posebnega - ponudi zgolj dodatne informacije in z njim ne delamo računskih operacij. Vsi ostali stolpci imajo številske vrednosti. Vedno je koristno preveriti, če je Orange pravilno prebral podatke. Gradnik File povežemo z gradnikom Data Table,



in dvakrat kliknemo na Data Table, da podatke prikažemo v preglednici.

The screenshot shows the 'Data Table' widget in the Orange interface. On the left, the 'Info' panel displays: 5 instances (no missing data), 5 features, No target variable, and 1 meta attribute. Under 'Variables', there are three checked options: Show variable labels (if present), Visualize numeric values, and Color by instance classes. Under 'Selection', the 'Select full rows' option is checked. At the bottom, there are buttons for 'Restore Original Order' and 'Send Automatically'. The main area shows a table with 5 rows and 6 columns. The columns are labeled: študent, angleščina, zgodovina, matematika, fizika, and športna vz. The rows represent students Jaka, Bine, Metka, Alenka, and Maja, with their respective scores.

	študent	angleščina	zgodovina	matematika	fizika	športna vz.
1	Jaka	22	32	21	46	99
2	Bine	91	65	89	11	29
3	Metka	51	21	100	100	27
4	Alenka	9	18	61	90	8
5	Maja	93	39	12	17	63

Odlično, vse je tako, kot mora biti.

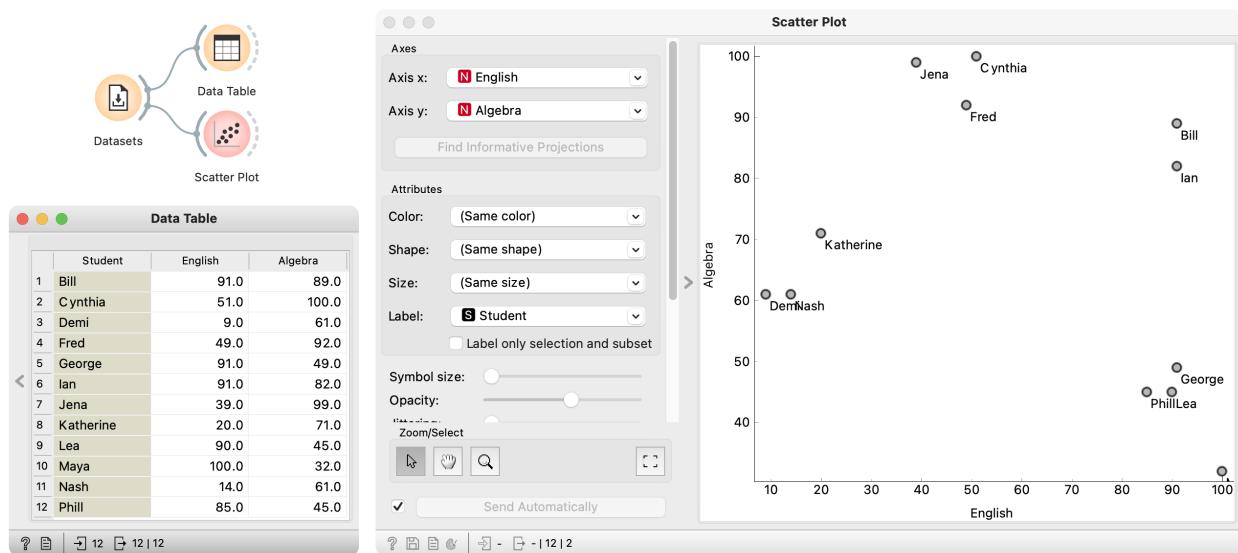
Uporabimo lahko tudi Google Sheets, prosto dostopen spletni urejevalnik preglednic. Takrat, namesto nalaganja datoteke z računalnika, vnesemo spletno povezavo do dokumenta v vrstico URL v gradniku File.

Nalaganje in oblikovanje podatkov je široka tema. Definiramo lahko tip in vrsto stolpca, dodamo, da je stolpec povezava do slike, itd. Ampak dovolj zaenkrat. Če bi radi izvedeli več, poglejte dokumentacijo o nalaganju podakov ali video na to temo.

Hierarhično razvrščanje v skupine

Ena od nalog rudarjenja besedil je iskanje zanimivih skupin dokumentov. Torej radi bi odkrili dokumente, ki so si podobni med sabo.

Poglejmo si preproste podatke z dvema stolpcema (glejte opombo) in jih prikažimo v gradniku *Scatter Plot*. Koliko skupin imamo? Kaj predstavlja različne skupine? Kateri primeri sodijo v posamezno skupino?



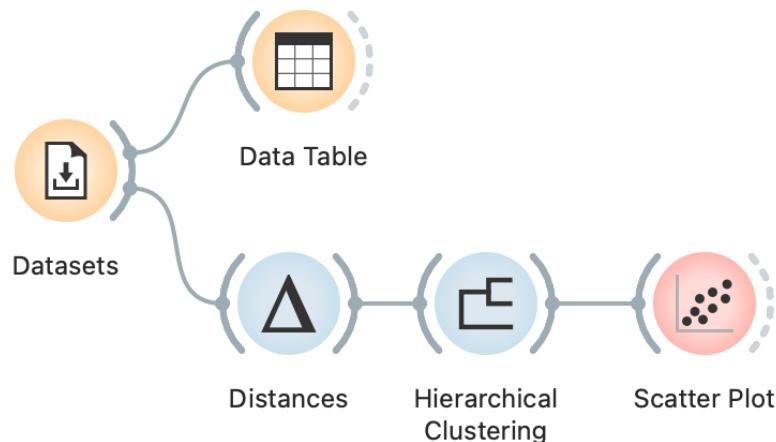
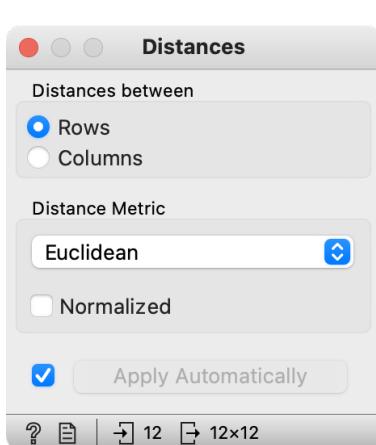
Kaj sploh pomeni "podobno"? Študenti so opisani s številskimi spremenljivkami, torej z ocenami pri predmetu. Ena od mer podobnosti je *evklidska razdalja*, ki preprosto izmeri razdaljo med dvema študentoma (točkama) v prostoru, kot bi to storili z metrom.

Sedaj definirajmo še postopek za razvrščanje v skupine. Recimo, da začnemo z vsakim dokumentom v svoji skupini, nato pa v vsakem koraku združimo skupini, ki sta si najbolj podobni. Razdaljo med skupinami izračunamo kot povprečje razdalj med posameznimi elementi skupine. Tak postopek imenujemo hierarhično razvrščanje v skupine.

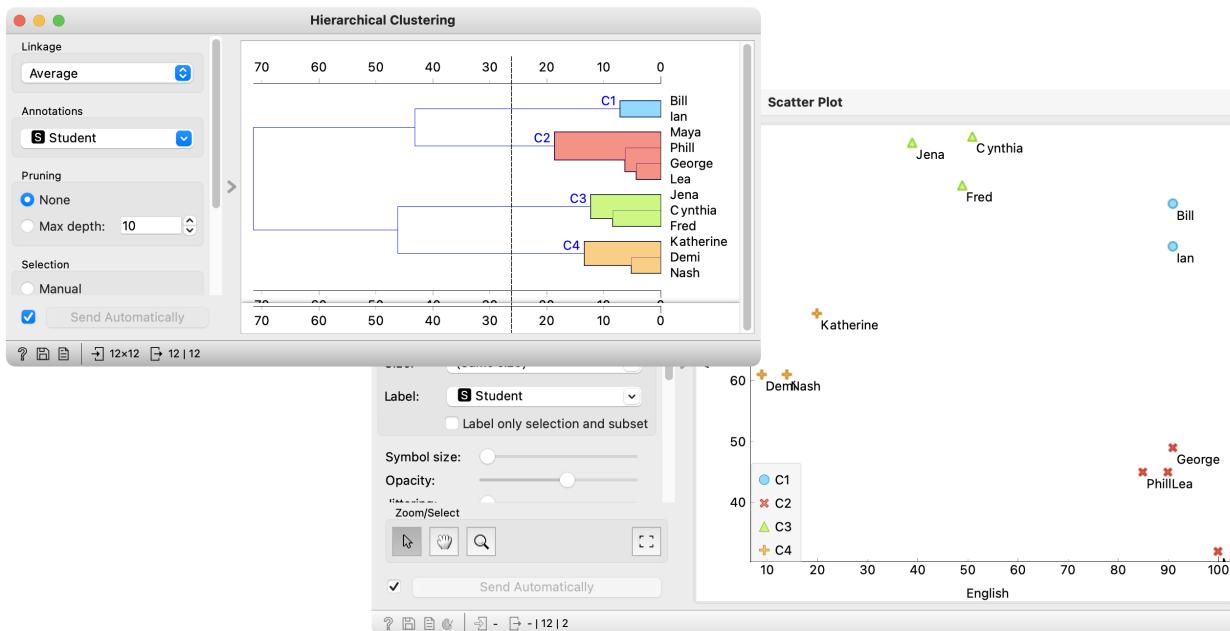
Razvrščanje v skupine bomo predstavili s preprostimi podatki o študentih in njihovih ocenah pri angleščini in matematiki. Podatki so dostopni v gradniku Datasets.

Načinov merjenja razdalj med skupinami je več. Način, ki smo ga opisali, se imenuje *povprečna razdalja (average linkage)*. Lahko bi računali tudi *razdaljo med najbližnjima točkama v skupini (single linkage)* ali pa med točkama, ki sta si *najbolj oddaljeni (complete linkage)*.

Rezultate razvrščanja v skupine na primeru naših študentov si lahko pogledamo v sledečem delotoku:



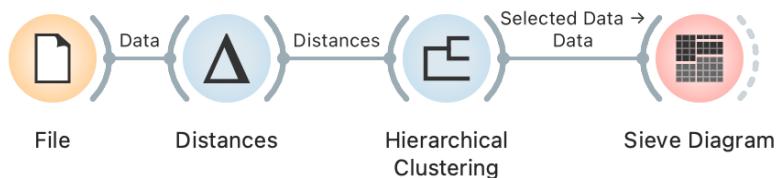
Naložite podatke z gradnikom *Datasets*, izračunajte razdalje z gradnikom *Distances*, uporabite *Hierarchical Clustering* in si poglejte rezultate v gradniku *Scatter Plot*. Gradnik Hierarchical Clustering omogoča, da hierarhijo skupin odrežemo pri določeni meri podobnosti in tako definiramo skupine.



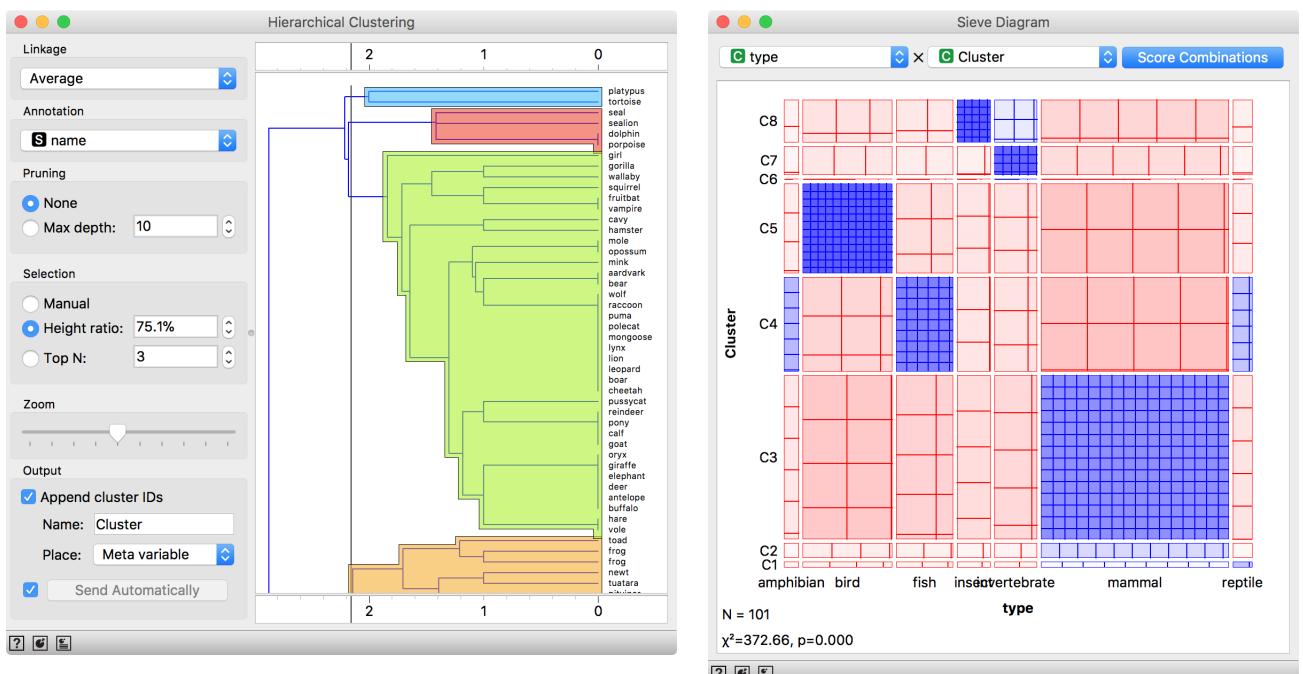
Raziskovanje gruč

Pričajoči zapiski so prevod zapiskov, ki jih uporabljamo po vsem svetu. Na tem mestu poslušalcem predstavimo hrvaško čokolado Životinjsko carstvo. V tem okolju to najbrž ni potrebno.

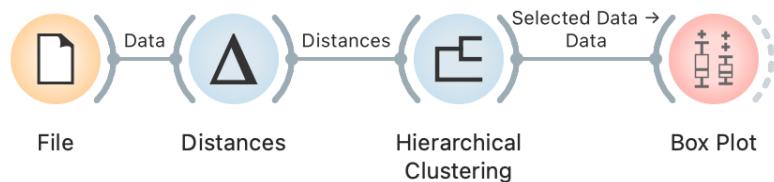
Sličice živali smo že kot otroci lepili v album, nikoli pa nismo zares razumeli, kako so bile karte razporejene po albumu. Kasneje smo se naučili nekaj o taksonomiji, vendar smo inženirji, zato smo jo raje odkrili sami, s podatkovnim rudarjenjem. Taksonomija naj izhaja iz podobnosti (torej merjenja razdalj) med vrstami.



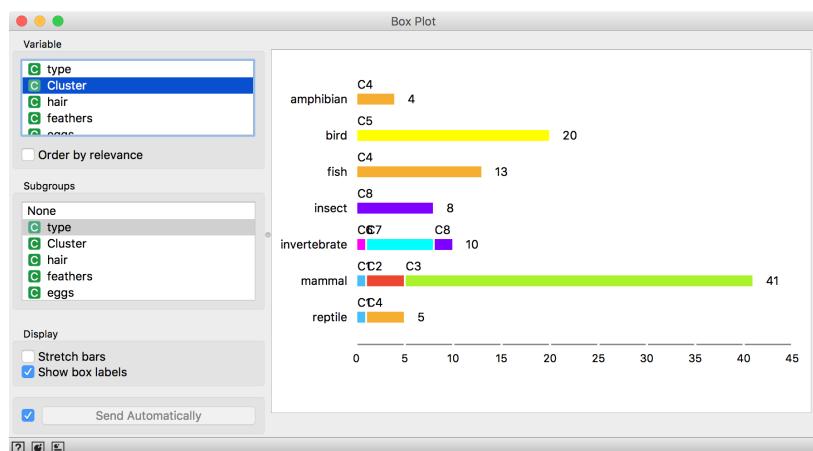
Uporabili bomo podatke `zoo.tab` (iz Orangeve dokumentacije). Podatki vsebujejo različne lastnosti živali (ima dlako, ima perje, valjajca). Izmerimo razdaljo in izračunamo gručenje. Živali v teh podatkih imajo tudi stolpec z razredom, ki mu pripadajo (sesalci, žuželke, ptice, itd.). Ne bi bilo imenitno, če bi gručenje odkrilo te iste razrede? To lahko preizkusimo z gradnikom Hierarchical Clustering, rezultate pa potem opazujemo v Sievovem diagramu.



Rezultati so zanimivi. Vse ptice, na primer, je gručenje prepoznalo kot eno skupino (označilo jo je s C6). Skupina C4 pa je mešana in vsebuje dvoživke, ribe in plazilce.

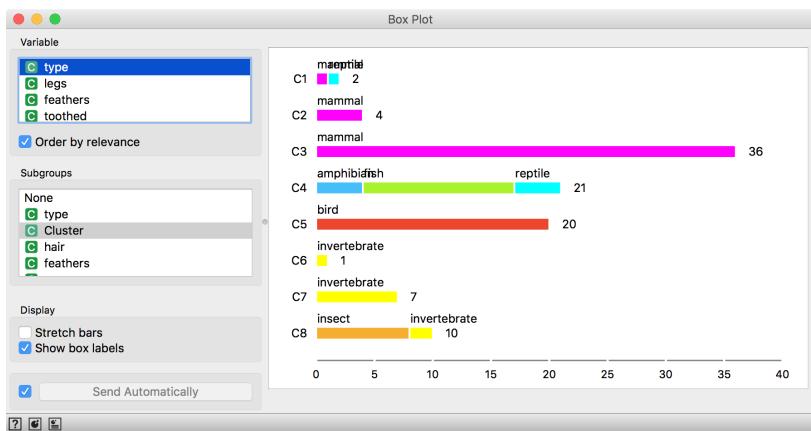


Še lepše je, če to pogledamo v gradniku *Box Plot*. Vidimo lahko porazdelitev razredov živali po izbranih gručah. Ali pa vizualizacijo obrnemo in pogledamo, v katerih gručah so kateri razredi.

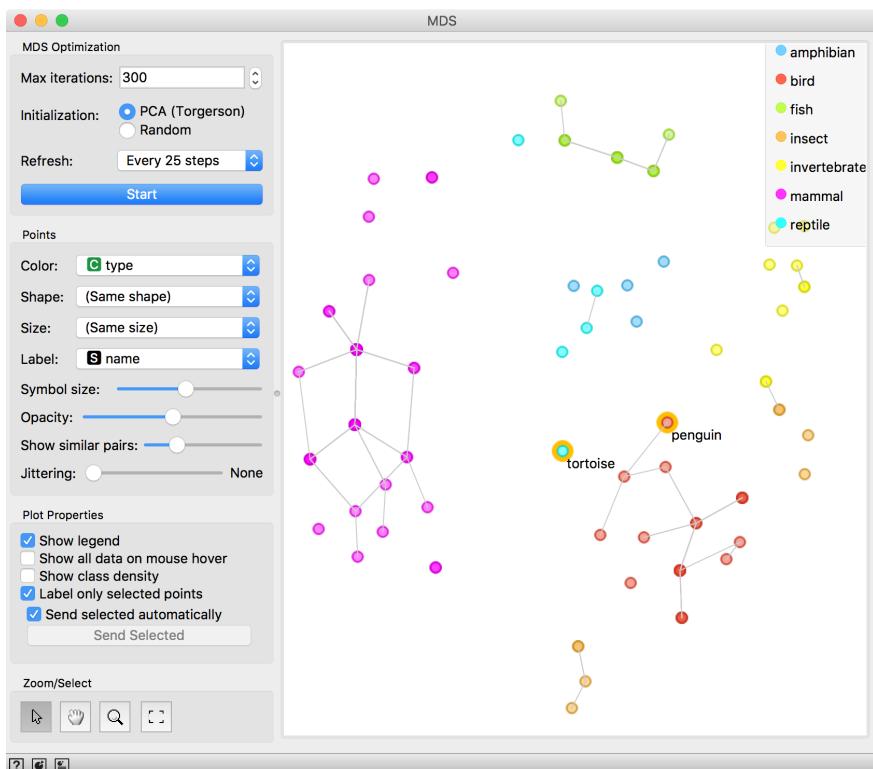
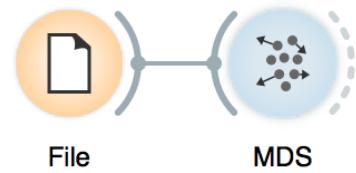


Kaj pa je narobe s sesalci? Zakaj niso v eni sami skupini? Za to sta dva razloga. Prvi je, da predstavljajo kar 40% primerov. Drugi pa, da vsebujejo nekaj čudakov. Izberite skupino v škatli z brki in preverite, katere živali so to.

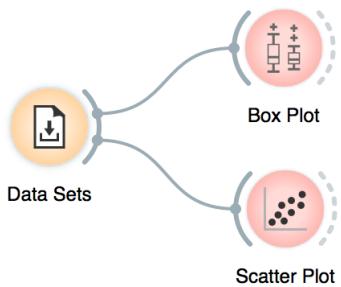
Posebnosti izbranih skupin lahko preverimo tako, da gradniku Box Plot odkljukamo opcijo 'Order by relevance', v Subgroups razdelku pa izberemo zopet spremenljivko Cluster. Očitno se skupine zares najlepše ločijo po razredih živali. Kako zanimivo!



S pomočjo hierarhičnega razvrščanja težko vidimo, kako podobna sta si, npr. želva in pingvin. Za to obstaja vizualizacija, ki primeri, ki so si med seboj podobni, nariše blizu skupaj, različne pa daleč. Tako vizualizacijo imenujemo večrazsežnostno lestvičenje (multidimensional scaling, MDS).



Vaja: politika



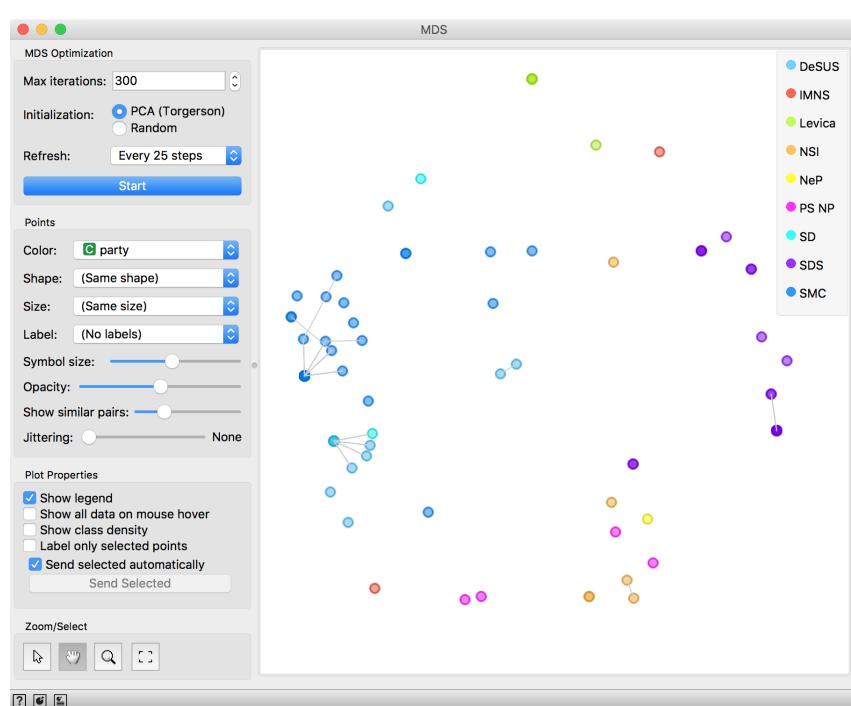
Čas je, da preizkusimo novo znanje. V gradniku *File* z namizja naložite podatke *slovenski-parlament.tab*.

1. Z Box Plotom ugotovite, kakšna je povprečna starost poslancev.
2. S Scatter Plotom odkrijte, kdo največ govoril na sejah parlamenta (ima največ govorov na sejo in besed na sejo). Zakaj je tako?
3. V Scatter Plotu pripravite vizualizacijo prisotnosti poslancev na sejah in njihovih pobud. Kaj lahko ugotovite iz teh dveh spremenljivk?

Čas je, da ugotovimo, kateri poslanci so si med sabo najbolj podobni. Računali bomo zgolj razdaljo med glasovanjem poslancev na sejah parlamenta. Najprej moramo izmeriti razdalje med njimi in uporabiti hierarhično razvrščanje, da odkrijemo skupine. Ročno izberite število skupin tako, da povlečete razmejitveno črto levo ali desno v vizualizaciji.



4. Kakšno bi bilo primerno število skupin? Kako bi interpretirali te skupine? Zapišite predlagano število skupin in ključno lastnost posamezne skupine (po čem so si poslanci podobni).



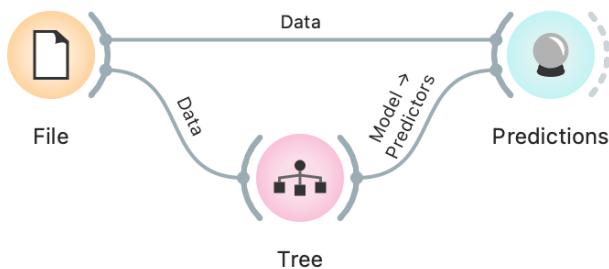
Iz dendrograma težko ugotovimo, kako blizu si je naključen par poslancev. Z vizualizacijo MDS bomo prikazali poslance takoj, da bodo podobni poslanci blizu skupaj. Dobili bomo nekakšen zemljevid poslancev.

5. Kateri poslanec glasuje izrazito drugače kot njegova stranka?

Klasifikacija

V prejšnjih lekcijah smo raziskali in gručili podatke, sedaj pa bi želeli zgraditi napovedni model. Tokrat bomo iz stopnje izraženosti genov poskusili napovedati funkcijo gena. Za to bomo uporabili podatke *brown-selected.tab* iz Orangeve dokumentacije. Nato bomo gradnik *File* povezali z gradnikom *Predictions*.

Predictions sam po sebi ne pokaže ničesar. Potrebuje namreč še model - nekakšna navodila, kako naj napoveduje iz podatkov. Pripeljmo v *Predictions* še gradnik *Tree*, kot vidite spodaj.



Podatki pridejo v *Tree*, ki na podlagi podatkov zgradi napovedni model. *Predictions* dobi podatke iz gradnika *File*, iz gradnika *Tree* pa model. To je nekaj novega; v prejšnjih delotokih so si gradniki pošiljali zgolj podatke, tukaj pa imamo kanal, ki pošilja model.

Predictions uporabi model za izdelavo napovedi na podatkih in te napovedi prikaže v tabeli.

Kako točne so napovedi? Je ta model dober? Kako vemo? Še prej pa - kaj sploh je drevo (*Tree*)? Kako izgleda? In kako ga Orange zgradi? Je to dober način gradnje algoritmov?

Screenshot of the Orange 'Predictions' widget interface:

Predictions

Show probabilities for: Proteas, Resp, Ribo

Tree

	function	gene	alpha 0	alpha 7
9	Proteas	YFR050C	0.093	0.027
10	Proteas	YDL097C	0.062	0.002
11	Proteas	YOR259C	-0.037	-0.122
12	Proteas	YPR108W	-0.016	-0.051
13	Proteas	YER021W	0.012	0.008
14	Proteas	YGR253C	-0.053	0.167
15	Proteas	YGL011C	0.011	-0.017
16	Proteas	YMR314W	-0.022	-0.048
17	Proteas	YGR135W	-0.002	-0.009
18	Proteas	YER012W	0.045	0.041

Model AUC CA F1 Precision Recall

Tree 0.998 0.978 0.978 0.978 0.978

Restore Original Order

?

186 | 1x186

186 | 1x186

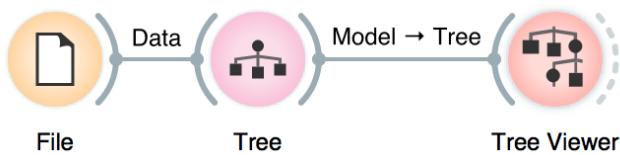
Klasifikacijska drevesa

Klasifikacijska drevesa so bila izredno priljubljena metoda v zgodnjih dneh stronjega učenja, ko so jih neodvisno predlagali inženir Ross Quinlan (C4.5) in skupina statistikov (CART), vključno z očetom naključnih gozdov Leonom Breimanom.

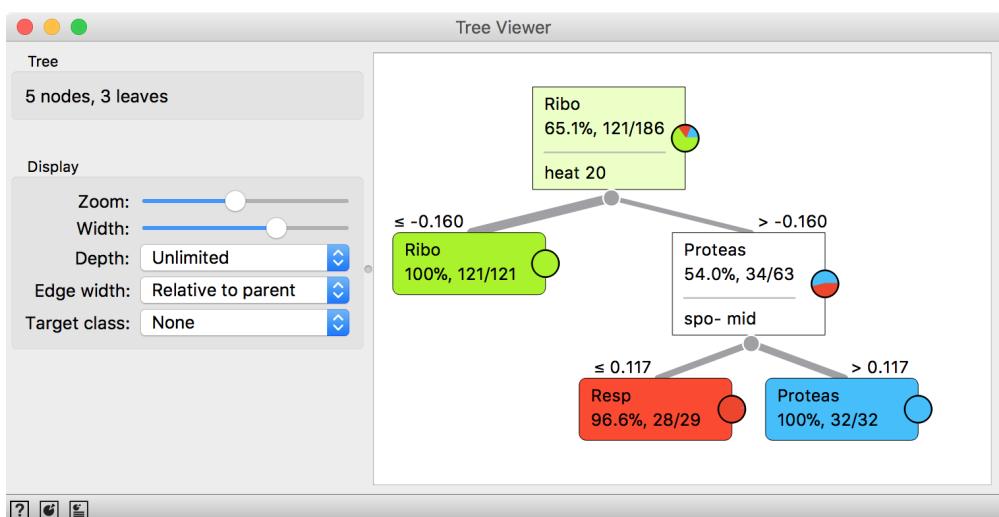
Nekatera pravila izvedena s klasifikacijskim drevesom niso najbolj zanesljiva za majhne podatke. Koliko podatkov potrebujemo, da lahko pridemo do trdnejših zaključkov?

GRADNJA KLASIFIKACIJSKIH DREVES JE ENA OD NAJSTAREJŠIH, A ŠE VEDNO PRILJUBLJENIH METOD STROJNEGA UČENJA. Kakšna model je torej drevo?

Poglejmo si, kako je videti drevo na naših podatkih o genih. Za prikaz drevesa bomo uporabili gradnik *Tree Viewer*.



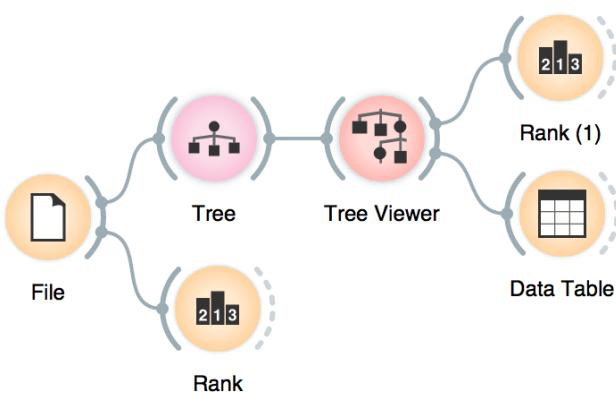
Drevesa beremo od zgoraj navzdol. Večina genov, 65 %, je po funkciji ribosomov, kar označimo tudi v deblu drevesa. Spremenljivka, ki najlepše loči podatke na dva dela, tako da v njih prevlada en razred, je *heat 20*. Če je izraženost gena manjša od -0.16, gremo v levo vejo, sicer pa v desno. Poglejmo si levo vejo drevesa. 100 % genov v tej veji je ribosomov; veja je povsem enotna. Na desni strani moramo deliti naprej in sicer po spremenljivki *spo-mid*. Če je vrednost *spo-mid* manjša od 0.117, gremo levo, sicer desno. Obe končni veji sta izredno enotni. Drevo torej vzame najbolj informativne spremenljivke in po njih hierarhično razdeli drevo v podmnožice.



Drevo se začne z najbolj uporabno spremenljivko. Kaj pa naj bi ta bila? To je spremenljivka, ki loči podatke v najbolj čisti podmnožici. Nato se podmnožice delijo naprej, po novih najboljših spremenljivkah, dokler vsi podatki v veji niso pripadniki enega razreda (močno

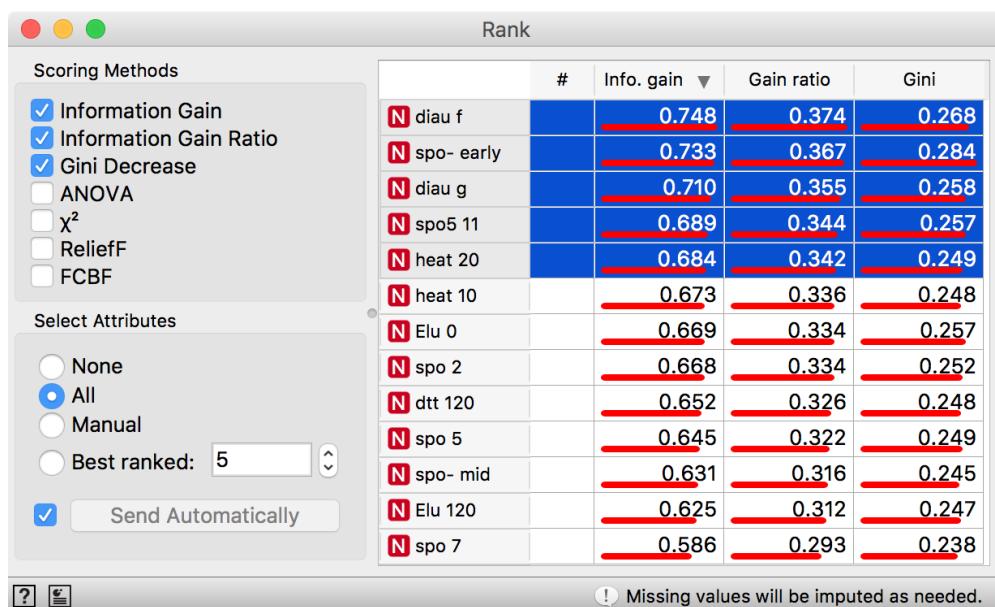
rdeče, modre, zelene veje) oz. dokler ni v podmnožici premalo podatkov oz. dokler ne zmanjka uporabnih spremenljivk za deljenje.

Še vedno nismo bili čisto jasni glede 'najbolj uporabne spremenljivke'. Mer kvalitete spremenljivk je veliko, temeljijo pa na tem, kako lepo ločijo razrede. Splošen koncept bomo prikazali z mero informacijskega prispevka. Mero lahko zračunamo z gradnikom *Rank*, ki oceni kvaliteto spremenljivk in jih razvrsti po tem, koliko nam povedo o razredu. Informacijski prispevek lahko ocenimo na celotnih podatkih ali pa zgolj na eni od vej drevesa iz gradnika Tree Viewer.



Na tem tečaju se ne bomo učili, kako izračunati informacijski prispevek. Na stackoverflow.com obstaja dobra razlaga koncepta s formulami in grafi (pogugljajte).

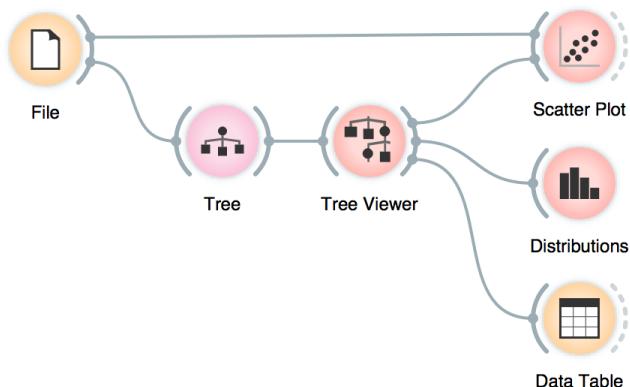
Poleg informacijskega prispevka Rank prikaže še druge mere (npr. relativni informacijski prispevek in giniijev indeks), ki so velikokrat skladne med sabo in so bile izumljene za boljšo podporo kategoričnim spremenljivkam z več vrednostmi.



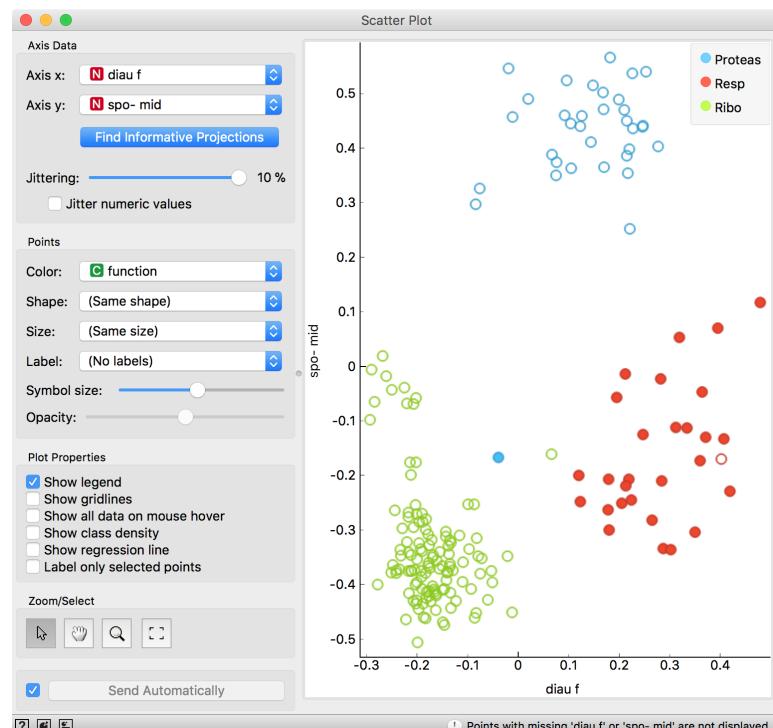
Pregled modela

Kadar je mogoče, vizualizacije v Orange podpirajo izbor in pošiljanje podmnožic. Iskanje zanimivih podmnožic in analiziranje njihovih lastnosti je ključen del raziskovalne analize podatkov, pristopa, ki ga priporoča guru vizualizacije podatkov Edward Tufte.

Poglejmo si še eno zanimivo kombinacijo gradnikov: Tree Viewer in Scatter Plot. V Scatter Plotu najdite najboljšo vizualizacijo podatkov, torej tako, ki najlepše loči med razredi. Nato povežite Tree Viewer v Scatter Plot. Ob izboru katerekoli veje iz drevesa, se bodo v Scatter Plot obarvale točke, ki pripadajo izbrani veji.



Za zabavo smo vključili še nekaj gradnikov v delotok. Tree Viewer izbere podmnožico, ki ustreza pravilom, kot jih poda model.



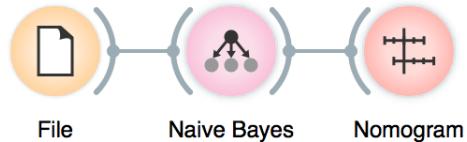
Naivni Bayes

Poglejmo si še nekaj drugih klasifikacijskih metod, npr. naivnega Bayesa. Tokrat bomo poskušali napovedati, kateri od potnikov Titanika, ki je potonil sredi Atlantskega oceana leta 1912, so preživeli. *titanic.tab* opisuje 2201 potnika, njegovo vozovnico (prvi, drugi, tretji razred, osebje), starost in spol.

Naložimo podatke in jih peljimo v gradnika *Naive Bayes* in *Nomogram*.

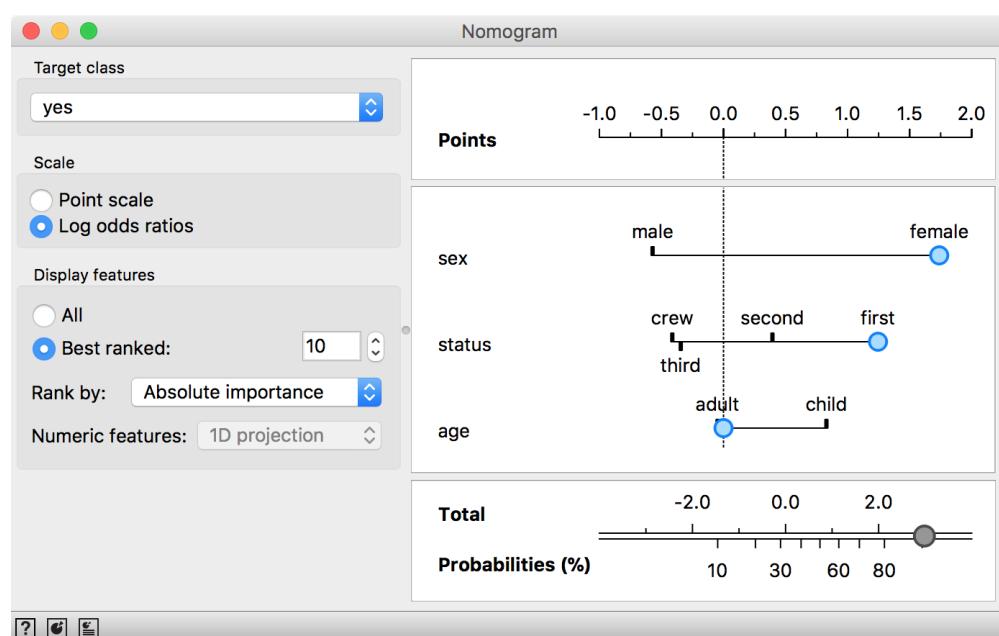
V Nomogramu vidimo lestvico 'Points' na vrhu in lestvice za vsako spremenljivko posebej (skupaj tri). Spodaj je dvojna lestvica verjetnosti. Pozorni boste na 'Target class' v zgornjem levem kotu. Če nastavimo ciljno vrednost na 'yes', potem bo nomogram prikazal verjetnost, da je potnik preživel.

Iz nomograma je razvidno, da je za naivnega Bayesa najpomembnejša spremenljivka spol. Če premaknemo modro točko na 'female', vidimo, da se verjetnost za preživetje poveča na 73 %. Če je ta ženska potovala v prvem razredu, se njena možnost preživetja poveča na 90 %. Za napovedi potrebujemo vsoto vseh prispevkov k ciljni vrednosti (preživetje) in nato to pretvorimo v verjetnost. Pretvorba je prikazana na spodnji lestvici.

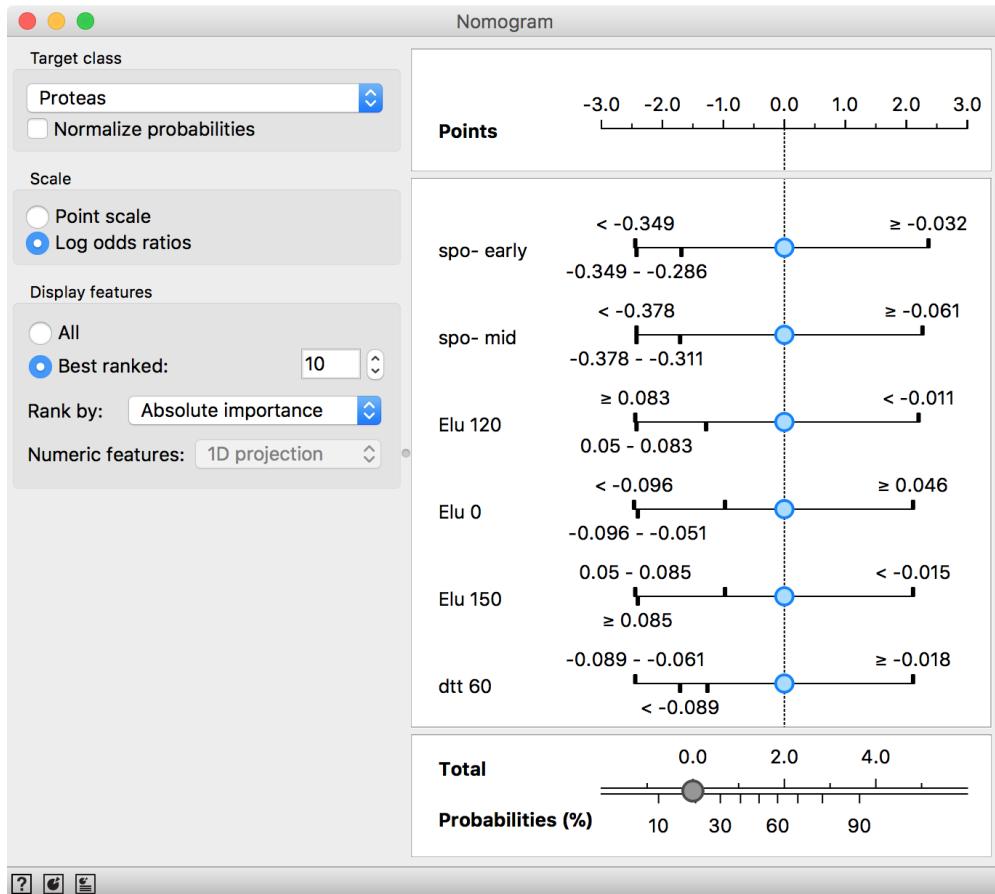


Naivni Bayes je metoda, ki predpostavlja (glede na razred) med seboj neodvisne spremenljivke. Če bi bile spremenljivke med samo neodvisne - kar so v praksi redko - bi bil naivni Bayes idealni klasifikator.

Nomogram sešteje prispevke vsake spremenljivke k ciljni vrednosti (preživetje) in potem to pretvori v verjetnost.



Vrnimo se k podatkom *brown-selected.tab*. Da bi Naive Bayes delal na teh podatkih, moramo številske spremenljivke pretvoriti v kategorične.
Ali še vedno razumemo model?



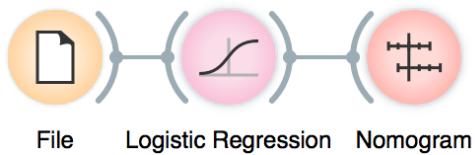
Ko pretvorimo številke v kategorije, izgubimo določeno mero preciznosti, ki lahko vpliva na točnost modela. In ker so številke sedaj predstavljene kot grobi intervali, iz nomograma težko razložimo model.

Poglejmo si raje metodo, ki odlično dela s številskimi vrednostmi, hkrati pa jo lahko enako lepo prikažemo.

Logistična regresija

Logistična regresija je ena najpreprostejših metod strojnega učenja. Deluje s številskimi vrednostmi in je odlična za modeliranje naših genov. V našem delotoku naivnega Bayesa preprosto zamenjamo z gradnikom *Logistic Regression*.

Logistična regresija je v resnici klasifikacijska in ne regresivna metoda. Regresija se imenuje zgolj zato, ker je metodološko podobna linearni regresiji.

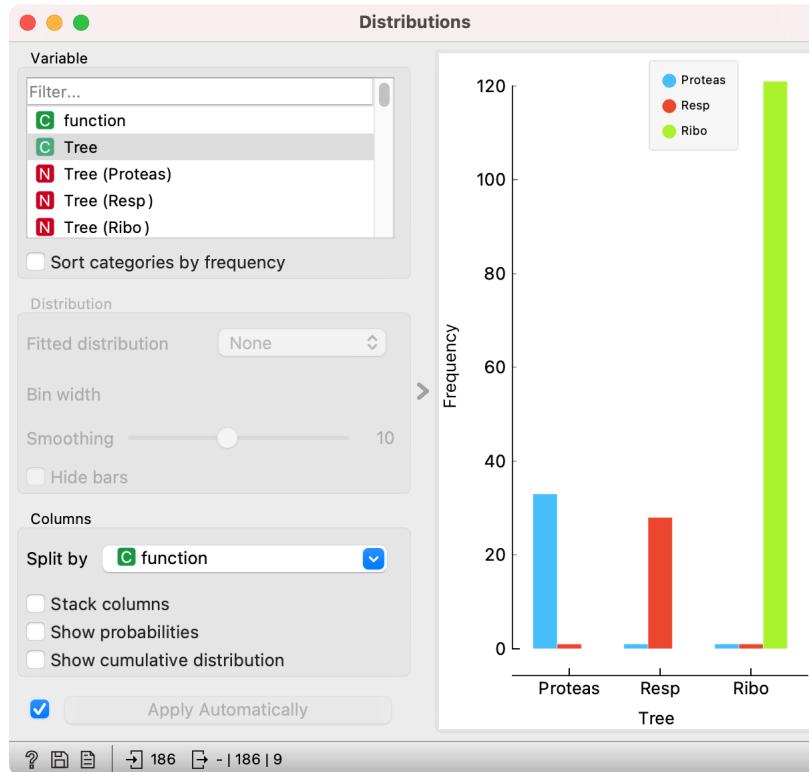
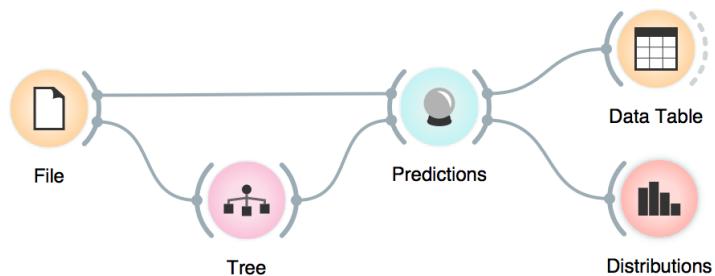


V nomogramu so naše vrednosti enakomerno razporejene. To naredi nomogram bolj pregleden, kot je bil prej z naivnim Bayesom. Ena spremenljivka, spo-mid, izstopa: očitno je za napovedni model dvakrat bolj pomembna kot naslednja najpomembnejša spremenljivka. Dolžina lestvice je namreč odvisna od pomembnosti spremenljivke.

V modelu logistične regresije ima vsaka spremenljivka svojo utež oz. pomembnost. Ko uporabimo model za napovedovanje novega primera, model pomnoži vrednosti spremenljivk z njihovimi utežmi, jih sešteje in nato pretvori v verjetnost. Končna vsota se pretvori v verjetnost z logistično funkcijo, ki jo vidimo na dnu nomograma.

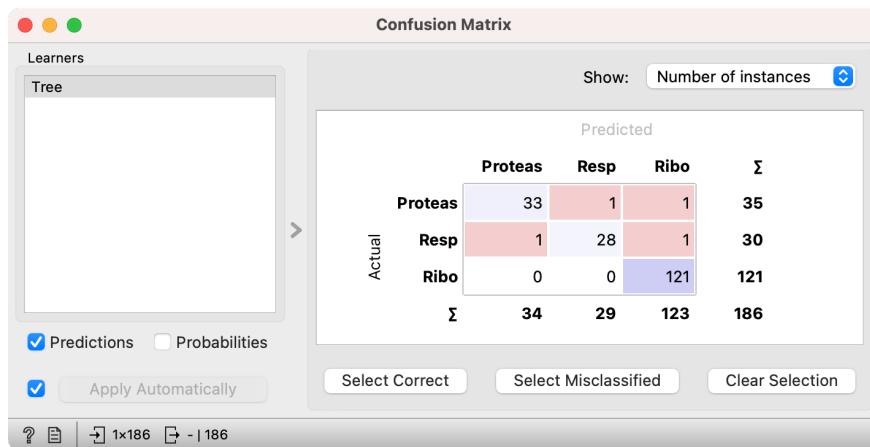
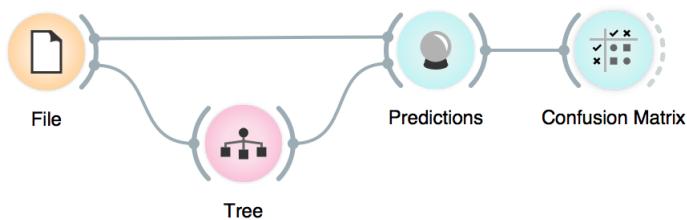
Klasifikacijska točnost

Sedaj ko vemo, kaj klasifikacijska drevesa so, se pojavi vprašanje, kakšna je točnost njihovih napovedi. Za začetek moramo definirati, kaj razumemo pod točnost. Najpreprostejša mera točnosti v klasifikaciji je klasifikacijska točnost, ki je izražena kot razmerje primerov, za katere je model pravilno napovedal razred. Poglejmo, če lahko ocenimo ali pa vsaj dobimo občutek za klasifikacijsko točnost z gradniki, ki jih že poznamo.



Predictions pošlje na izhod preglednico z dodanim stolpcem z napovedmi. V gradniku *Data Table* lahko razvrstimo podatke po katerem koli od dveh stolpcov (Tree ali Function) in ročno izberemo primere, kjer sta vrednosti v stolpcih različni (kar bi bilo težavno na velikih podatkih). Če pravilno nastavimo spremenljivke, lahko v grobem ocenimo napovedno točnost kar v gradniku *Distributions*.

Včasih pa bi potrebovali kaj bolj natančnega kot zgolj grobo oceno iz Distributions gradnika. Statistiko pravilno in nepravilno klasificiranih primerov vidimo v matriki zmot (ang. *Confusion Matrix*).

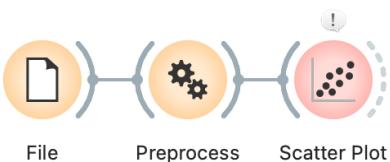


Vidimo, da je drevo za 33 proteaz, 28 respiratornih genov in 121 ribosomov pravilno napovedalo rezultat. Vendar pa je napačno uvrstil 4 gene. Ker je klasifikacijska točnost razmerje pravilno napovedanih primerov, jo izračunamo kot število pravilnih napovedi ($33 + 28 + 121 = 182$) deljeno s številom vseh primerov (186). Klasifikacijska točnost je torej $182/186 = 97\%$. Sliši se dobro, pa je res?

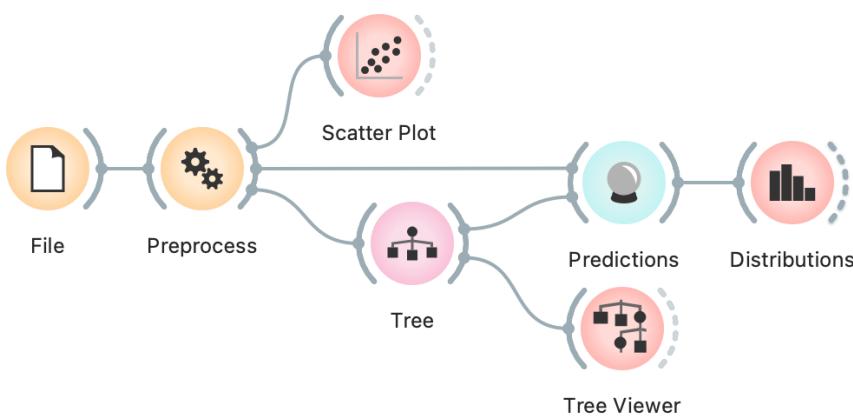
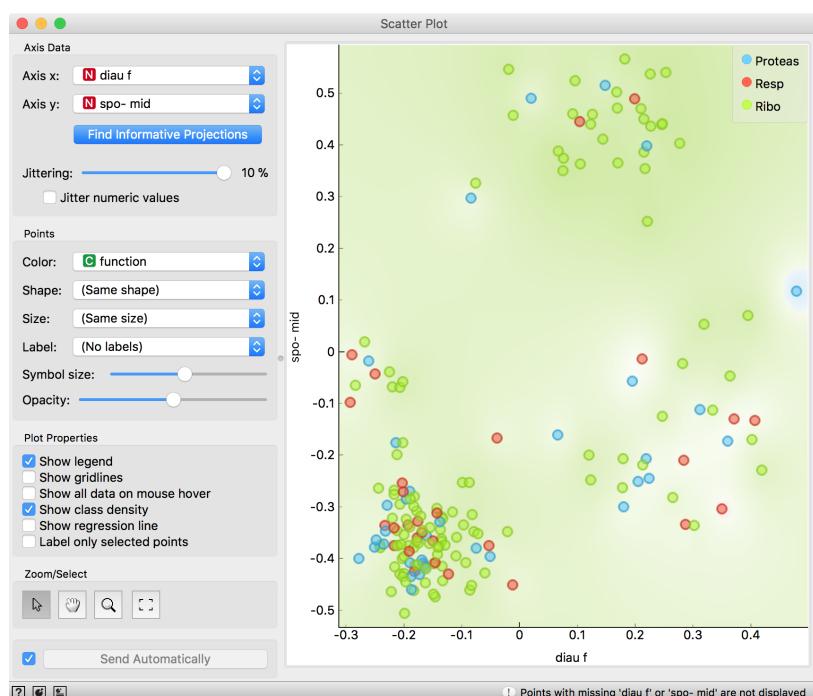
Kako goljufati

Ta lekcija ima čuden naslov in ni čisto jasno, zakaj smo ga izbrali. Morda nam ti, bralec, lahko poveš, kaj ima ta lekcija z goljufanjem.

Zakaj je ozadje razsevnega diagrama popolnoma zeleno? Kam sta šli drugi dve barvi, modra in rdeča?

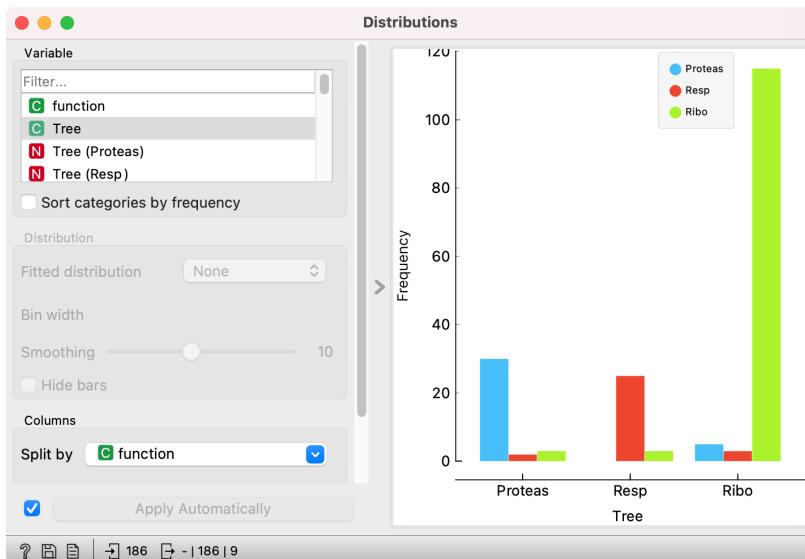


Trenutno klasifikacijsko drevo daje odlične rezultate. Zmoti se le v nekaj primerih. Ali lahko podatke tako zelo pokvarimo, da bo drevo neuporabno? Na primer, odstranimo lahko kakršno kolikso povezavo med genskimi ekspresijami in njihovo funkcijo. Za to bomo uporabili gradnik *Preprocess* in metodo najključnega razporejanja razredov (*randomize class*). Poglejmo si kaos, ki ga ustvari v razsevnem diagramu, kjer smo prej imeli lepe gruče!



Dobro. Gotovo ni modela, ki bi lahko napovedal tole zmedo. Prepričajmo se. (Ko povežete *Preprocess* z gradnjikom *Tree*, bo Orange povzel signala *Preprocessor*. Tukaj boste morali ročno prevezati povezavo iz *Preprocessed Data* v *Data*. Povezave v dialogu odstranite tako, da kliknete nanje.)

In rezultat? Tukaj so naše distribucije:



Signali iz gradnika *Data Sampler* nimajo imen, ker smo želeli prihraniti prostor. *Data Sampler* je razdelil podatke na vzorčne in izven-vzorce (t.i. preostale podatke). Vzorčni so bili poslani v *Tree*, medtem ko so bili preostali poslani direktno v *Predictions*. Nastavite *Data Sampler* tako, da bosta množici približno enako veliki.

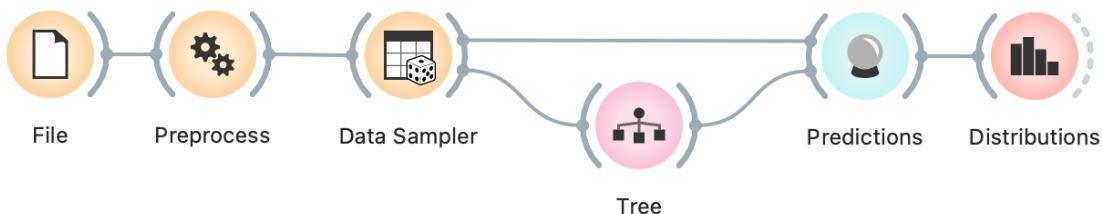
Nadvse nenavadno. Skoraj brez napak. Kako je to mogoče? In to na naključnih podatkih?

Uganko bomo rešili tako, da odpremo *Tree Viewer* in pogledamo drevo. Koliko vej ima? Ali končne veje vsebujejo veliko primerov?

Očitno si je drevo preprosto zapomnilo vsak primer iz podatkov. Nič čudnega, da so bile napovedi tako dobre. Drevo nima nobenega smisla in je kompleksno, ker si je zapomnilo vse skupaj.

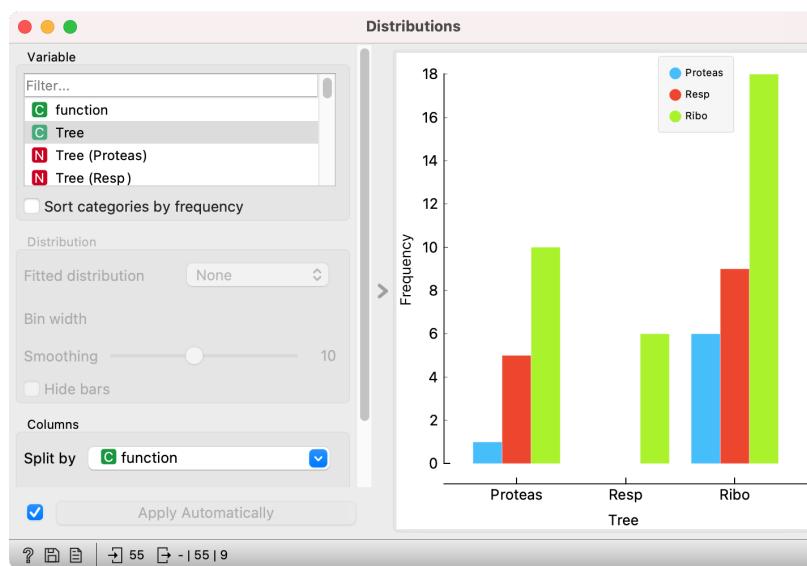
Če je temu tako, če si klasifikator zapomni podatke, ne da bi odkril splošne vzorce, bo na novih podatkih popolnoma zanič. Preverimo. Podatke bomo razdelili na dva dela, na učno in testno množico. Klasifikator bomo naučili na učnih podatkih, točno pa bomo ocenili na testnih.

Signali iz gradnika *Data Sampler* nimajo imen, ker smo želeli prihraniti prostor. *Data Sampler* je razdelil podatke na vzorčne in izven-vzorce (t.i. preostale podatke). Vzorčni so bili poslani v *Tree*, medtem ko so bili preostali poslani direktno v *Predictions*. Nastavite *Data Sampler* tako, da bosta množici približno enako veliki.



Izgleda, da je bila večina genov napovedana kot ribosomnih. Zakaj? Zopet je vse zeleno (tako kot na razsevnem diagramu). Namig: odgovor najdete v građniku Box Plot.

Poglejmo, kako izgleda naš Distributions gradnik po testiranju klasifikatorja na testnih podatkih.



Napovedi funkcije genov so popoln polom. Na naključno zmesanih učnih podatkih naš klasifikator odpove. Končno to, kar smo pričakovali.

Da bi res preizkusili kvaliteto (točnost) klasifikacijske metode, moramo naučiti model na učnih podatkih in ga potem preveriti na testnih. S testom lahko razlikujemo med klasifikatorji, ki si rezultate zapomnijo na pamet, in temi, ki se naučijo splošen model.

Učenje ni piflarija. Nasprotno, učenje je odkrivanje vzorcev, ki vladajo podatkovm, in prenašanje teh vzorcev na nove podatke. Da lahko ocenimo točnost klasifikatorja, potrebujemo torej testne podatke. Ta ocena pa ne sme biti odvisna od samo ene delitve podatkov na učne in testne (tudi tukaj je prostor za goljufanje). Nasprotno, postopek testiranja moramo ponoviti večkrat, vsakič z novima učnostnima množicama in potem poročati o povprečnem rezultatu.

Da bi klasifikator odpovedal, smo morali naključno zmešati samo učne podatke. Poskuite spremeniti delotok tako, da bodo vrednosti razredov premešane zgolj v učnih podatkih, ne pa tudi v testnih.

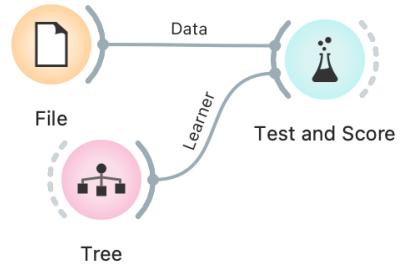
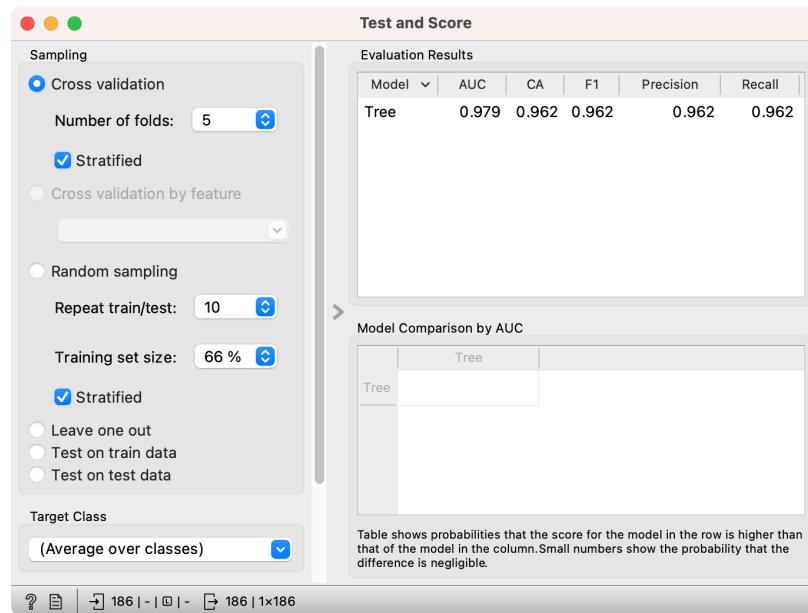
Prečno preverjanje

Ocena točnosti je odvisna od podatkov v obeh podmnožicah. Da bi povečali robustnost ocene, lahko ta postopek ponovimo večkrat, vsakič z drugačno testno podmnožico. Ena takih metod je prečno preverjanje. V Orangeu je na voljo v gradniku *Test and Score*.

V vsaki ponovitvi bo Test and Score vzel del podatkov za učenje, zgradil napovedni model na njih z izbrano metodo in potem preveril točnost modela na preostali, testni množici. Za to bo gradnik potreboval na vhodu podatke, iz katerih bo vzorčil učno in testno množico, in pa učno metodo, ki jo bo uporabil na učnih podatkih za grajenje napovednega modela. V Orangeu učnim metodam pravimo kar učenci (ang. learner). Torej Test and Score potrebuje učenca na vhodu.

To je nov način, kako uporabiti gradnik *Tree*. V prejšnjih delotekih smo uporabili gradnikov drug izhod, imenovan *Model*; njegova gradnja je zahtevala podatke. Tokrat podatkov ne potrebujemo; potrebujemo zgolj učenca.

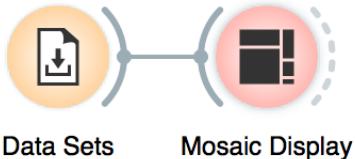
Tako izgleda Test and Score gradnik. CA pomeni klasifikacijsko točnost in zaenkrat je to vse, kar nas zanima.



Za tehnične tipe: učenec je objekt, ki ob prisotnosti vhodnih podatkov vrne model. Točno to, kar Test and Score potrebuje.

Prečno preverjanje razdeli podatke na, recimo, 10 ločenih podmnožic, ki jim bomo rekli zavihki. V vsaki iteraciji se en zavihek uporabi za testiranje, ostalih 9 pa za učenje. Na ta način bo model preverjal vsak primer le enkrat.

Vaja: kadrovská



Sedaj boste sami gradili modele in z njimi napovedovali. V gradniku File z namizja naložite podatke *attrition-train.tab*.

1) Z Mosaic Display najdite najlepšo projekcijo za dve spremenljivki? Kaj vam ta projekcija pove? Kakšen kader bo najhitreje odšel?



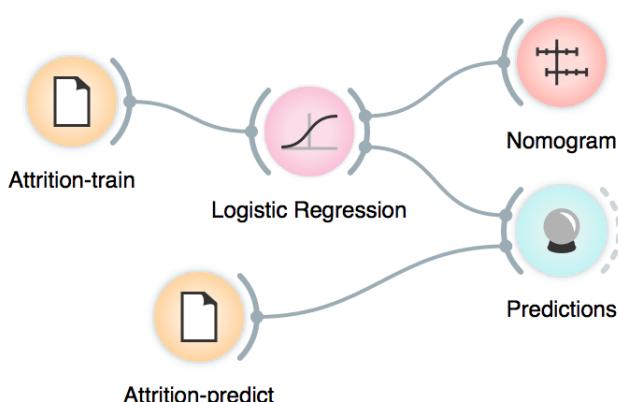
2) Uporabite vse znane metode za napovedovanje in jih povežite v Test and Score. Katera metoda ima najvišjo klasifikacijsko točnost?

3) Katere tri spremenljivke so najpomembnejše za logistično regresijo? (Pozor! Target class morate nastaviti na 'yes'.) Bi lahko razložili model? Kateri zaposleni bodo bolj verjetno dali odpoved?

Sedaj so nam iz kadrovske dostavili podatke za tri nove uslužbence. Kadrovsko službo zanima, kakšna je verjetnost, da bodo ti uslužbenci dali odpoved (za vsakega uslužbenca posebej). Uporabite podatke *attrition-predict.tab*, jih naložite v File in napovljte verjetnosti z logistično regresijo.

4) Kdo od treh uslužbencev je najmanj zadovoljen (oz. bo verjetno dal odpoved)?

5) Kaj mora kadrovská storiti, da bo uslužbenca zadržala? Nāmig: poglejte v Nomogram, zakaj uslužbenci odhajajo in povljete, s čim jih lahko prepričate, da ostanejo.



Literatura

[https://github.com/biolab/orange3.](https://github.com/biolab/orange3)

Robert Bringhurst. *The Elements of Typography*. Hartley & Marks, 3.1 edition, 2005. ISBN 0-88179-205-5.

Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, et al. Orange: data mining toolbox in python. *the Journal of machine Learning research*, 14(1):2349–2353, 2013.

Primož Godec, Matjaž Pančur, Nejc Ilenič, Andrej Čopar, Martin Stražar, Aleš Erjavec, Ajda Pretnar, Janez Demšar, Anže Starič, Marko Toplak, et al. Democratized image analytics by visual programming through integration of deep models and small-scale machine learning. *Nature communications*, 10(1):1–7, 2019.

Primož Godec, Nikola Đukić, Ajda Pretnar, Vesna Tanko, Lan Žagar, and Blaž Zupan. Explainable point-based document visualizations. *arXiv preprint arXiv:2110.00462*, 2021.

Edward R. Tufte. *Envisioning Information*. Graphics Press, Cheshire, Connecticut, 1990. ISBN 0-9613921-1-8.

Edward R. Tufte. *Visual Explanations*. Graphics Press, Cheshire, Connecticut, 1997. ISBN 0-9613921-2-6.

Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, 2001. ISBN 0-9613921-4-2.

Edward R. Tufte. *Beautiful Evidence*. Graphics Press, LLC, first edition, May 2006. ISBN 0-9613921-7-7.