Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Mark Lemut

Integracija pripomočka za uravnoteževanje podatkov v paket Orange

diplomsko delo

visokošolski strokovni študijski program
prve stopnje
računalništvo in informatika

Mentor: doc. dr. Aleksander Sadikov

Ljubljana, 2018

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Kandidat naj integrira pripomoček (angl. widget) za uravnoteževanje podatkov v sistem za strojno učenje Orange. Pripomoček je že sprogramiran (avtor: Aleš Smodiš), vendar se mu bo tako povečala enostavnost uporabe, predvsem za neračunalničarje, ki so tudi ciljni uporabniki tega pripomočka. Kandidat naj zasnuje ustrezne zaslonske maske (uporabniški vmesnik) ter ustrezno izkoristi in poveže ostale obstoječe pripomočke za delo s podatki v sistemu Orange. Napiše naj tudi kratka navodila za uporabo. Prav tako naj stestira pravilno delovanje implementacije/integracije na nekaj obstoječih podatkovnih domenah.

Zahvaljujem se mentorju doc. dr. Aleksandru Sadikovu za vse nastvete in pomoč pri izdelavi diplomskega dela.

Kazalo

Povzetek

Naslov: Integracija pripomočka za uravnoteževanje podatkov v paket Orange

Avtor: Mark Lemut

Statistične analize ali statistični testi se lahko izvajajo nad neuravnoteženimi podatkovnimi zbirkami, čeprav so statistični testi najbolj zanesljivi na uravnoteženih podatkih. Programov za uravnoteževanje podatkovnih zbirk ni veliko. Cilj naloge je bil narediti algoritem za uravnoteževanje, ki ga je razvil Aleš Smodiš dostopen za širšo množico, ki delajo s podatki in se ne ukvarjajo s programiranjem. Zato se je algoritem implementiral v brezplačni, odprtokodni program Orange, v obliko widgeta oz. pripomočka. Orange je preprost za uporabo in nudi veliko opcij za analizo, vizualizacijo in delo s podatki. Ponuja možnost uporabe strojnega učenja za klasifikacijo in regresijo.

Ključne besede: widget, Orange, uravnoteževanje podatkov, predpriprava podatkov.

Abstract

Title: Balancing widget integration into Orange

Author: Mark Lemut

Statistical tests can sometimes be performed on unbalanced data sets, even though they are most reliably performed on balanced data sets. There are not a lot of programs that can balance a data set. The assignments goal was to make the algorithm that was developed by Aleš Smodiš more accessible for an audience, that works with data and does not know how to program. Because of that, the algorithm was implemented into a free open source program called Orange. Orange is a simple program and offers a lot of options for data analysis, visualization, and for working with data in general. It offers the use of machine learning for classification and regression.

Keywords: widget, Orange, data balancing, data pre-preparation.

Poglavje 1

Uvod

V sodobnem svetu je veliko poklicev, pri katerih je pomembno delo s podatki. Z računalnikom lahko podatke analiziramo, vizualiziramo ali uporabimo pri napovedovanju vrednosti nekega atributa, s pomočjo algoritmov strojnega učenja [6].

Nad podatki se velikokrat izvajajo statističnih testi oz. analize. Statistične analize so posebni statistični postopki, s katerimi se preverja različne lastnosti zbranih podatkov. Statistični testi so najbolj zanesljivi na uravnoteženih podatkih. Bolj ko so si vzorci med seboj podobni, manj nas lahko skrbi vpliv razlik med vzorci na rezultat statističnih analiz oz. testov.

Veliko ljudi, ki se ukvarjajo s podatki, niso računalničarji in ne znajo programirati. Cilj naloge je bil implementirati algoritem, za uravnoteževanje podatkovnih zbirk v program, ki je dostopen vsem z računalnikom in internetno povezavo. Za program, v katerega se je implementiral algoritem, je bil izbran Orange [3]. Algoritem se je implementiral v obliko widgeta ali ka. Orange je bil izbran zaradi naslednjih razlogov:

- Je brezplačni, odprtokodni program za analizo in delo s podatki.
- Ima funkcionalnosti, ki so potrebne za pravilno delovanje algoritma.
- Ima funkcionalnosti za nadaljno obdelavo rezultatov uravnoteževalnega algoritma.

- Je zelo enostaven za uporabo.
- Orange in algoritem sta oba razvita v programskem jeziku python [10].
- Orange je bil razvit na fakulteti za računalništvo in infromatiko.

Primer: V zdravstvu. Widget bi koristili zdravniki, za uravnoteževanje podatkovne zbirke, ki vsebuje primere bolnikov z rakom na dojki [1]. Za bolnike obstajata dve različni zdravljeni, ki sta odvisni od nekega biomarkerja. Podatki, ki kažejo na odvisnost so močno neuravnoteženi po za zdravnika pomembnih lastnostih bolnika. Zato, da je možno bolj jasno uvideti morebitno povezavo med zdravljenjem in izraženostjo biomarkerja je koristno podatke uravnotežiti.

V okviru diplomske naloge se je razvil widget v programu Orange, ki neuravnoteženo podatkovno zbirko uravnoteži na podlagi vhodnih parametrov, ki jih nastavi uporabnik s pomočjo uporabniškega vmesnika.

Algoritem, ki ga uporablja widget za uravnoteževanje, je razvil in v svoji diplomski nalogi opisal Aleš Smodiš [8]. Deluje tako, da iz podatkov naredi štiri podskupine na podlagi dveh binarnih atributov. Vse štiri podskupine med seboj uravnoteži na podlagi enega ali večih atributov. Uravnoteženost skupin po nekem atributu se meri s pomočjo p vrednosti ali stopnje značilnosti. P vrednost predstavlja verjetnost, da med podskupinama na podlagi nekega atributa ni razlik. Ker p vrednost predstavlja verjetnost, vedno zavzema vrednost med 0 in 1. Slabost algoritma je, da se število primerov za nadaljno analizo po uravnoteževanju zelo zmanjša.

Widget se je testiral na treh podatkovnih zbirkah:

- Podatkovna zbirka titanic
- Podatkovna zbirka adult
- Podatkovna zbirka bolnikov z rakom na dojki [1].

Podatkovni zbirki titanic in adult sta dosegljivi na spletni strani UCI datasets [4]. Podatkovno zbirko bolnikov z rakom na dojki smo dobili od

zdravstva [1]. Močno anonimizirana verzija zbirke je bila na voljo izključno za testiranje diplome.

Poglavje 2

Implementacija

2.1 Opis uporabljenih orodij

2.1.1 Orange

Orange je program za analizo, vizualizacijo [9] in delo s podatki [3]. Je odprtokodni, kar pomeni, da lahko kdorkoli vidi njegovo izvorno kodo in si jo prilagodi svojim potrebam. Napisan je v programskem jezku python [10]. Deluje tako, da uporabnik kreira novo delovno shemo, na katero lahko dodaja različne widgete. Vsak widget ima svojo funkcionalnost. Med seboj se povezujejo. Povezave predstavljajo pot podatkov. Razdeljeni so v skupine. Skupina data vsebuje widgete, s pomočjo katerih uporabnik manipulira in analizira podatke. V to skupino spada najpomembnejši widget file, ki omogoča nalaganje podatkov v program. Ostale skupine so visualize, model, evaluate, unsupervised. Vsaka skupina vsebuje widgete, ki medseboj nudijo podobne funkcionalnosti. V program Orange je možno dodati lastne widgete.

2.1.2 Python

Python je objektno usmerjen interpreterski programski jezik [10]. Objektno usmerjen programski jezik, je jezik, ki deluje na principu objektov, ki vsebujejo podatke v poljih ali atributih in kode v obliki procedur. Procedure

so bolj znane po imenu metoda. Interpreterski programski jezik, je jezik pri katerem se vsaka vrstica kode interpretira, oziroma prevede v drugi visoko nivojski jezik ali v strojni jezik [2] sproti. Poleg interpreterskih jezikov obstajajo jeziki, ki se pred izvedbo ukazov prevedejo v drugi visoko nivojski jezik ali v strojni jezik [2] v celoti. Python se je prvič pojavil leta 1991. Filozofija jezika je povzeta v dokumentu The Zen of Python [7].

2.2 Implementacija pripomočka

Algoritem v obliki widgeta je v program Orange bil implementiran v večih korakih.

2.2.1 Prvi korak

Orange in algoritem za uravnoteževanje sta oba razvita v programskem jeziku python. Zadnja verzija programa Orange je razvita v pythonovi verziji 3, zadnja verzija algoritma pa v pythonovi verziji 2. Med pythonovimi verzijami so majhne razlike, vendar dovolj velike, da je bilo pri prehodu algoritma na python 3 potrebno spremeniti določene dele kode.

Prehod za večino kode algoritma na python 3 ni bil problematičen. Izjema so bile funkcije cmp, ki se uporabljajo pri urejanju primerov. Potrebno je bilo uporabiti funkcijo cmp_to_key, ki staro cmp funkcijo pretvori v novejšo key funkcijo.

2.2.2 Drugi korak

Zasnoval se je uporabniški vmesnik, ki vsebuje vse potrebne grafične gradnike za uporabnikov vnos vhodnih podatkov [5]. Uporabniku omogoča:

- Izbiro dveh binarnih atributov, katerih vrednosti določajo štiri skupine primerov, ki se uravnotežujejo.
- Izbiro od enega do sedmih diskretnih atributov, na podlagi katerih se skupine uravnotežujejo.

- Vnos vrednosti k. K predstavlja število primerov v večjih skupinah, ki se ujemajo z enim primerom v manjših skupinah.

- Izbiro željene p vrednosti za vsak atribut, ki je bil izbran pri izbiri atributov za uravnoteževanje. P vrednost v tem primeru predstavlja željeno verjetnost, da med skupinam po izbranem atributu ni večjih razlik. Algoritem po uravnoteževanju vrne dejansko p vrednost med skupinami za vsak izbran atribut za uravnotežvanje.

- Izbiro uteži za vsak izbran atribut za uravnoteževanje. Utež je vrednost, ki hrani informacijo o pomembnosti atributa pri uravnoteževanju. Če se vsem atributom za uravnoteževanje nastavi vrednost uteži na 1, pomeni da so vsi atributi za uravnoteževanje upoštevajo kot enakovredni.

- Prikaz trenutnega stanja skupin.

2.2.3 Tretji korak

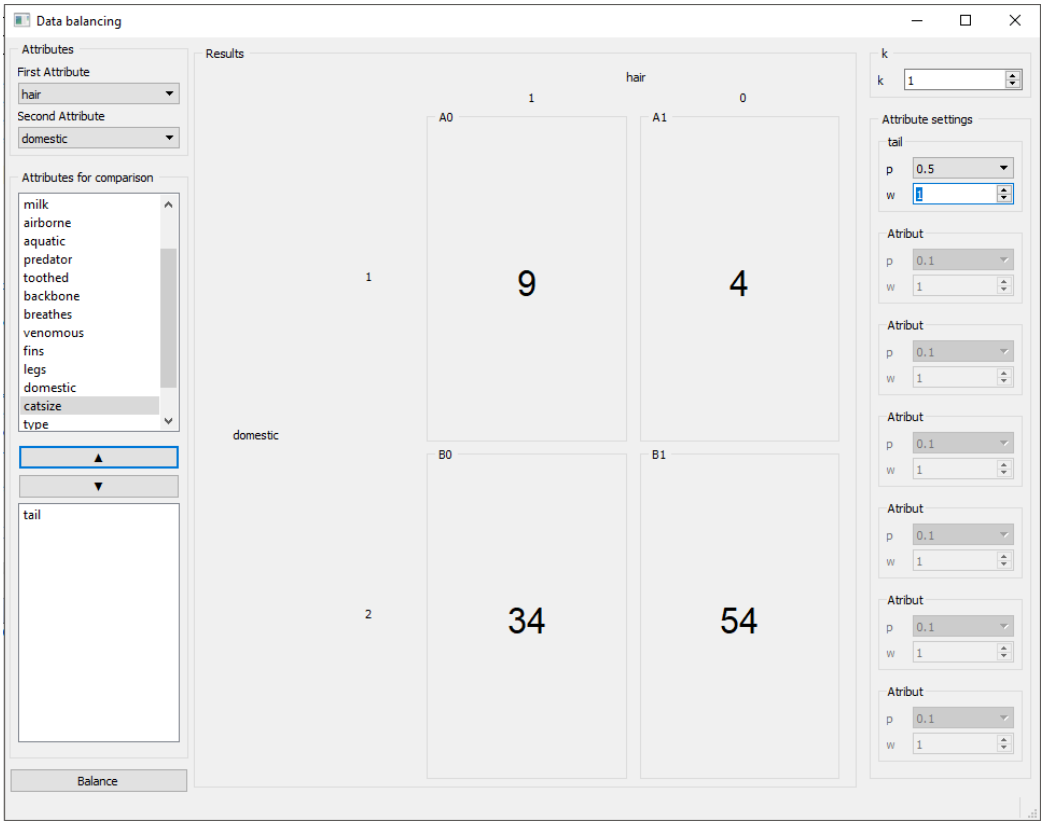
Zasnovani uporabniški vmesik se je implementiral v Orange. Vmesnik je razdeljen na tri dele:

- Levi del, ki vsebuje grafične gradnike za izbiro dveh binarnih atributov, na podlagi katerih se ustvarijo štiri skupine in grafične gradnike za izbor od enega do sedmih atributov za uravnoteževanje. Pod gradniki se nahaja gumb, ki ob pritisku izvede algoritem za uravnoteževanje.

- Srednji del, ki vsebuje grafične gradnike za prikaz trenutnega stanja skupin.

- Desni del, ki vsebuje grafične gradnike za izbor in vpis k vrednosti, p vrednosti in uteži. Z vsakim izbranim atributom za uravnoteževanje se omogoči skupina grafičnih vmesnikov za vnos p vrednosti in uteži.

Izgled uporabniškega vmesnika se vidi na sliki 2.1



Slika 2.1: Izgled uporabniškega vmesnika widgeta za uravnoteževanje

Za izbor dveh binarnih atributov, na podlagi katerih se ustvarijo štiri skupine, smo uporabili dva kombinirana seznama. Kombinirana seznama se inicializirata in dodata na okno widgeta v konstruktorju razreda, ki predstavlja widget. Pri branju vhodne podatkovne zbirke se seznama napolnita z imeni atributov zbirke. Izbor trenutnega atributa se hrani v spremenljivki v obliki indeksa.

Za izbor enega ali do sedem atributov za uravnoteževanje, smo izbrali dva seznamska polja in dva gumba. Grafični vmesniki se inicializirajo in na okno widgeta dodajo v konstruktroju razreda, ki predstavlja widget. Prvo seznamsko polje se napolni z imeni atributov zbirke in predstavlja vse atribute, ki niso bili izbrani za uravnoteževanje. Drugo seznamsko polje predstavlja vse atribute, ki so bili izbrani za uravnoteževanje. Gumba se uporabljata pri od-

stranjevanju atributov iz enega seznamskega polja in dodajanje na drugega. Običajno, ko se nek objekt prestavi iz enega seznamskega polja v drugega in nato prestavi nazaj v prvega, se objekt doda na konec prvega seznamskega polja. Napisali smo metodo, ki vrne atribut na prvotno mesto v seznamskem polju. Posledica te metode je, da so v seznamskem polju, ki predstavlja izbrane atribute, atributi vedno urejeni po vrstnem redu atributov v podatkovni zbirki. Ko se atribut prestavi iz prvega seznamskega polja v drugega, se v novi seznam indeksov shrani njegov trenutni indeks. Če na seznamsko polje dodamo še en atribut, ki je bil prvotno pred atributom, ki smo ga dodali na drugo seznamsko polje, se pred indeks prvega atributa v seznam z indeksi zapiše trenutni indeks drugega atributa. Če indeks dodajamo na mesto, ki ni prvo v seznamu, je indeksu potrebno prišteti število indeksov, ki so pred njim. Ko prvi atribut vrnemo na prvo seznamsko polje, se njegovemu indeksu odšteje število indeksov, ki so v seznamu indeksov pred njegovim indeksom. Za pravilno dodajanje indeksov v seznam indeksov in atributov na drugo seznamsko polje, je potrebna še informacija o prvotnem vrstnem redu atributov.

Primer delovanja metode: Imamo seznam z objekti a, b, c, d, e in f. Objekt c, ki ima indeks 2 dodamo na drugi seznam in v seznam indeksov zapišemo njegov trenutni indeks, torej indeks 2. Nato objekt b prestavimo na drugi seznam objektov. Ker imamo informacijo o prvotnem vrstnem redu atributov, dodamo objekt b pred objekt c in v seznam indeksov dodamo trenutni indeks objekta b pred indeks objekta c. Trenutni indeks objekta b je 1. V drugi seznam premaknemo še objekt d, ki ima trenutni indeks 1. Objekt d se doda na konec seznama. Trenutnemu indeksu objekta d se prišteje številka 2, ker sta v seznamu indeksov dva indeksa pred njim. Ko se objekt c prestavi nazaj v prvi seznam objektov, se njegovemu indeksu odšteje število indeksov, ki so pred njim. Torej se indeksu 2 odšteje številka 1, ker je pred njim samo indeks objekta b. Objekt c se doda pred objekt e, ki ima trenutno indeks 1. V prvi seznam vrnemo objekt d. Indeksu objekta d se odšteje številko 1, ker je pred njim samo indeks objekta b. Objekt b se doda

na prvi seznam pred objektom e, ki ima trenutno indeks 2 in za objekt c. Na prvi seznam se nato še prestavi objekt b pred objekt c, ki ima trenutni indeks 1.

Na sredini so labele, ki prikazujejo trenutno stanje skupin. Ko uporabnik izbere nov atribut za kreacijo skupin, se na sredini izpis ustrezno popravi.

Na levem delu okna smo dodali grafični gradnik, za vpis vrednosti k in grafične gradnike za vpis p vrednosti in uteži za vsak izbran atribut za uravnoteževanje. Ko se atributi za uravnoteževanje dodajajo na drugo seznamsko polje in z njega odstranjujejo, se na levi strani omogočajo in onemogočajo gradniki za vpis p vrednost in uteži. Vrednosti se ob pričetku uravnoteževanja preberejo s spremenljivk, ki so z gradniki povezana.

Če med widgeti ni povezave in widget za uravnoteževanje nima vhodne podatkovne zbirke, so vsi grafični gradniki na oknu widgeta za uravnoteževanje onemogočeni.

2.2.4 Četrti korak

Spremenljivke, ki hranijo vhodne parametre in podatkovna zbirka, se uporabijo pri ustvarjanju objketa, ki se pošlje algoritmu za uravnoteževanje. Ko algoritem objekt obdela, vrne nov objekt, v katerem so izbrani primeri z njihovimi indeksi in p vrednosti za vsak izbran atribut za uravnoteževanje.

2.2.5 Peti korak

Iz rezultatov se preberejo indeksi primerov, ki še vedno pripadajo eni od skupin. Ustvarita se dve tabeli, ki se pošljeta naprej kot izhod. Prva tabela vsebuje primere, ki še vedno pripadajo eni od štirih skupin. Druga tabela vsebuje vse primere, katere je algoritem izločil iz skupin.

Poglavje 3

Navodila za uporabo

3.1 Zahteve za uporabo widgeta

Za uporabo widgeta za uravnoteževanje, je potrebno na računalnik naložiti program Orange verzijo 3. Potreben je tudi python 3 interpreter, ki se naloži pri nalaganju programa Orange. Ko je program Orange naložen, je potrebno prenesiti in naložiti še widget za uravnoteževanje. Koraki, ki so potrebni za dodajanje widgeta za uravnoteževanje programu Orange:

1. Zagnati program Orange.
2. Ugasniti dialog, ki se pokaže ob zagonu programa.
3. V menijski vrstici klikniti options in nato addons.
4. V oknu s seznamom dodatnih paketov za Orange poiskati in izbrati paket balancing.
5. Izbor obkljukati in klikniti gumb ok.

3.2 Priprava podatkov

Orange uporablja svoj tip datotek s končnico .tab. Poleg .tab datotek podpira uvoz velikega števila drugih datotek kot npr. excelove datoteke in da-

toteke, ki so ločene z vejico ali tabulatorjem. Podatki morajo biti v obliki tabele, ki ima primere ali vrstice in atribute ali stolpce.

3.3 Uvoz podatkov

V Orangeu se kreira in poimenuje novo shemo. S seznama z widgeti se potegne widget file na delovno območje sheme. Možen je tudi desni klik na delovno območje sheme, ki odpre kontekstni meni s seznamom vseh widgetov. Z dvoklikom na widget file se odpre okno, ki ponuja možnosti za uvoz datoteke s podatki. Zgornji del okna ponuja možnost za uvoz datoteke s podatki, ki se nahaja na lokalnem računalniku in možnost za uvoz datoteke z interneta. Ob uspešnem uvozu podatkov, spodnji del okna prikazuje seznam atributov. Prikazuje ime, tip in vlogo atributa. Orange sam pri uvozu nastavi tip in vlogo atributov. S klikom na tip ali vlogo se lahko ročno spremeni. Orange podpira zvezni, diskretni, niz znakov in časovni tip atributa. Če je atribut diskreten, se tu prikazujejo še možne vrednosti atributa. Možne vloge atributa so meta, class, skip in feature.

Tip in vlogo atributa je možno nastaviti že v sami datoteki na dva načina. Prvi način je, da se pred ime atributa doda c, m, i, w za vlogo, C, D, T, S za tip in nato znak #. Znak c predstavlja vlogo class, m predstavlja vlogo meta, i predstavlja vlogo skip in w predstavlja feature. Znak C predstavlja zvezni tip, D predstavlja diskretni tip, T predstavlja časovni tip in S predstavlja niz znakov. Drugi način je za imena atributov dodati dve vrstici. V prvo dodatno vrstico se vpiše tip atributa v drugo, se vpiše vlogo atributa.

3.4 Uravnoteževanje podatkov

S seznama z widgeti se doda widget za uravnoteževanje. Z desne strani widgeta file, se potegne povezavo v levo stran widgeta za uravnoteževanje. Z dvoklikom na widget za uravnoteževanje, se odpre okno z nastavitvami za uravnoteževanje podatkov. Okno je razdeljen v tri dele. Na levi strani

je območje za izbiro atributov za ustvarjanje skupin in izbiro atributov za uravnoteževanje. Na sredini je območje za prikaz trenutnega stanja skupin in rezultatov. Na desni strani je območje za nastavitev vrednosti k in za vsak atribut za uravnoteževanje posebej p vrednost in utež.

3.4.1 Izbor atributov za ustvarjanje skupin

Leva stran okna je razdeljena na zgornji in spodnji del in se jo vidi na levi strani slike 3.2. Zgornji del je namenjen za izbor dveh atributov. Z njunim izborom se ustvari štiri skupine. Atributa ne smeta biti enaka in morata biti diskretna in binarna. Binarni atributi imajo samo dve vrednosti. Ob izboru dveh ustreznih atributov se območje za prikaz trenutnega stanja skupin posodobi. Za vsako skupino se prikazuje število primerov.

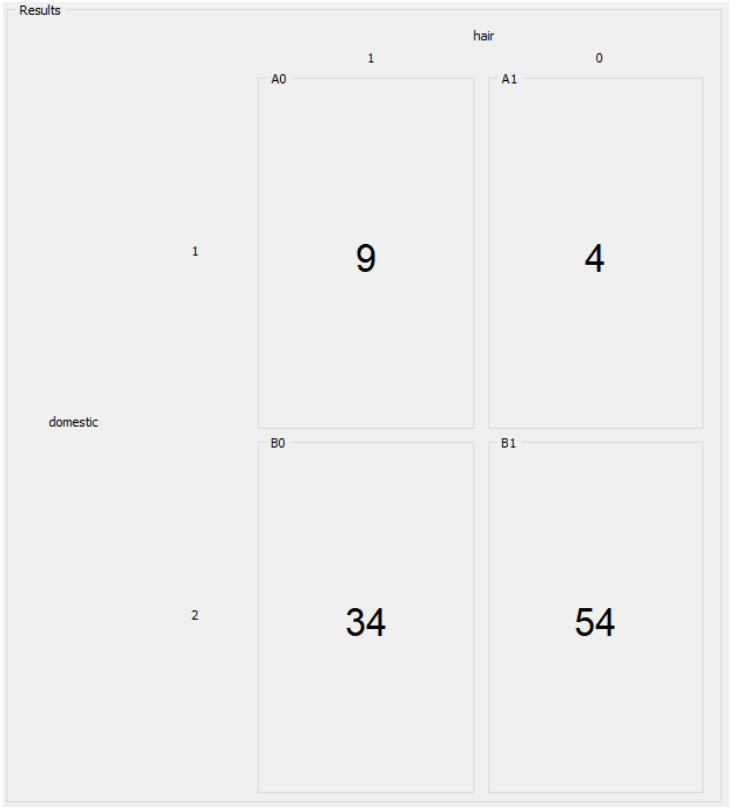
3.4.2 Izbor atributov za uravnoteževanje

Spodnji del leve strani je namenjen za izbor atributov, ki se upoštevajo pri uravnoteževanju ustvarjenih štirih skupin. Možno je uravnoteževati samo po diskretnih atributih. Izbrati je potrebno vsaj en atribut in največ sedem. Podatki se lahko uravnotežujejo po največ sedmih atributih, ker je uravnoteževanje po več kot sedmih atributih kombinatorično izjemno zahtevno in običajno nepotrebno.

3.4.3 Prikaz trenutnega stanja skupin

Na sredini okna je prikaz matrike skupin, ki se ga vidi na sliki 3.1. Matrika je sestavljena iz štirih območij. Območja so poimenovana A0, A1, B0 in B1. Nad temi območji se prikazuje trenutno izbrani prvi atribut, z obema možnima vrednostima. Levo od teh območij se prikazuje trenutno izbrani drugi atribut, z obema možnima vrednostima. Na sredini vsakega območja je številka, ki predstavlja število primerov, v pripadajoči skupini. Vsak primer spada v to skupino, ker ima vrednost prvega atributa enako vrednosti, ki se prikazuje nad svojim območjem in vrednost drugega atributa enako

vrednosti, ki se prikazuje levo od svojega območja. Manjši skupini naj se nahajata v območjih A0 in A1, večji skupini pa v območjih B0 in B1.



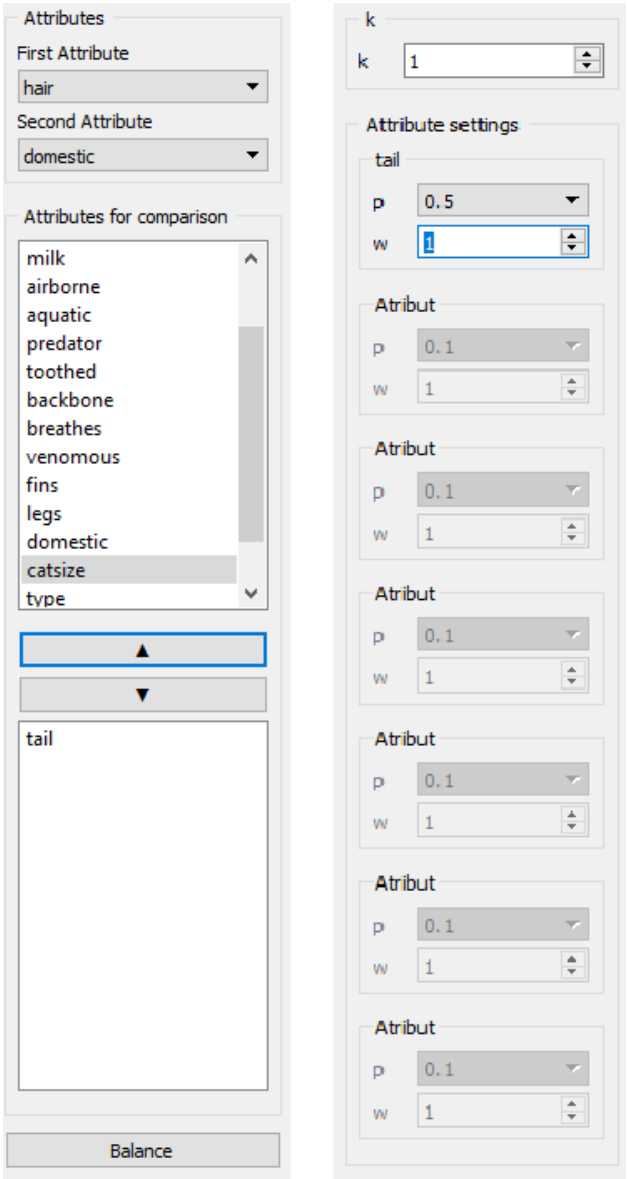
Slika 3.1: Sredina widgeta za uravnoteževanje

3.4.4 Nastavitev vhodnih parametrov

Na desni strani okna je izbor nastavitev vhodnih parametrov. Desno stran se vidi na desni strani slike 3.2. Ta stran je razdeljena na dva dela. Zgornji del služi izboru k vrednosti. K vrednost predstavlja število primerov iz večjih dveh skupin, ki ustreza enemu primeru manjših dveh skupin. Privzeta vrednost je 1. Vrednost 1 pomeni, da bo po uravnoteževanju število primerov v skupinah enako. Vrednost 2 ali n pomeni, da je v večjih skupinah lahko dvakrat ali n krat več število primerov kot v manjših skupinah. Spodnji del desne strani je namenjena za izbor nastavitev izbranih atributov za uravnoteževanje. Število omogočenih nastavitev določa število izbranih atributov za uravnoteževanje, saj vsak sklop nastavitev pripada atributu. Vsak sklop ima nastavitev za p vrednost in utež.

- P vrednost ali stopnja značilnosti se uporablja pri preverjanju domnev oz. hipotez. Predstavlja verjetnost dogodka, zato so možne vrednosti med 0 in 1. Nizka p vrednost pomeni, da lahko z 1 - p vrednost verjetnosti rečemo, da med dvema skupinama na podlagi izbranega atributa obstajajo razlike. Željena p vrednost se nastavi za vsak atribut posebej. Izbira se lahko med naslednjimi vrednostmi: 0.1, 0.2, 0.25, 0.5, 0.9, 1. Možno je vpisati drugo vrednost, ki mora biti med 0 in 1. Željena je večja p vrednost, saj se s tem zagotavlja, da med skupinama z veliko verjetnostjo ni velike razlike pri izbranem atributu.

- Utež predstavlja pomembnost atributa pri uravnoteževanju. Možne vrednosti so cela števila od 1 naprej. Privzeta vrednost za vsak atribut je 1. 1 pri vsakem atributu pomeni, da je vsak atribut enako pomemben pri uravnoteževanju. Če enemu atributu določimo večjo vrednost, pomeni, da je ta atribut bolj pomemben pri uravnoteževanju kot ostali.



Slika 3.2: Leva in desna stran widgeta za uravnoteževanje

3.4.5 Rezultati in izhodni podatki

Ob končanem izboru vseh nastavitev, je levo spodaj gumb balance, ki ob pritisku izvede algoritem za uravnoteževanje. Na delovno območje sheme se iz seznama widgetov doda widget data table. Iz desne strani widgeta za uravnoteževanje, se potegne povezava v levo stran widgeta data table. Dvojni klik na najnovejšo povezavo odpre okno, kjer se izbira kateri izhod iz widgeta za uravnoteževanje naj gre v widget data table. Možna sta dva izhoda. Prvi izhod je seznam vseh primerov, ki so še vedno v eni izmed štirih skupin, ki so sedaj uravnotežene. Drugi izhod je seznam primerov, ki niso več v nobeni izmed skupin.

V primeru, da je povezava z widgetom na levi strani prekinjena, je izbira nastavitev onemogočena. Izbira nastavitev se omogoči takrat, ko je povezava na levi strani vzpostavljena in ima widget za uravnoteževanje vhodno podatkovno zbirko.

3.5 Možni vplivi na rezultate

Vrstni red primerov lahko vpliva na končno število primero v skupinah, ker algoritem za uravnoteževanje po vrsti izloča primere, ki povzročajo, da si dve skupini nista dovolj podobni po nekem izbranem atributu. Skupini sta si dovolj podobni po izbranem atributu, ko je p vrednost višja ali enaka p vrednosti, ki se jo nastavi za ta atribut. Primeri se najprej urejajo po prvem atributu, če je potrebno se urejajo po nadaljnih atributih.

Orange vsebuje widgete, s pomočjo katerih se lahko vpliva na rezultate uravnoteževanja. Najbolj vplivajo widgeti, ki omogočajo spreminjanje podatkovne zbirke.

3.5.1 Widget select columns

Pripomoček select columns omogoča uporabniku odstranjevanje atributov iz podatkovne zbirke. Možno ga je uporabiti za spreminjanje vrstnega reda

atributov, kar lahko vpliva na vrstni red primerov.

3.5.2 Widget select rows

Pripomoček select rows omogoča izločevanje primerov s pomočjo pravil, ki jih nastavi uporabnik. Za vsak atribut lahko nastavi, katere vrednosti so dovoljene oz. katere vrednosti niso dovoljene.

3.5.3 Widget discretize

Widget discretize omogoča diskretizacijo zveznega atributa, po katerem se lahko uravnotežuje skupine. Če ga diskretiziramo v binaren atribut, se lahko uporabi tudi pri ustvarjanju skupin.

3.6 Widgeti za vizualizacijo rezultata in izpis primerov v rezultatu

3.6.1 Widget data table

Widget data table se lahko uporabi za izpis vseh primerov, ki so po uravnoteževanju v eni izmed uravnoteženih skupin ali izpis vseh primerov, ki niso v nobeni skupini.

3.6.2 Widget distributions

Widget distirbutions se lahko uporabi za izris diagrama, ki prikazuje porazdelitev primerov na podlagi nekega atributa. Omogoča tudi izris diagrama, ki porazdelitev primerov prikazuje na podlagi dveh atributov.

Poglavje 4

Testiranje

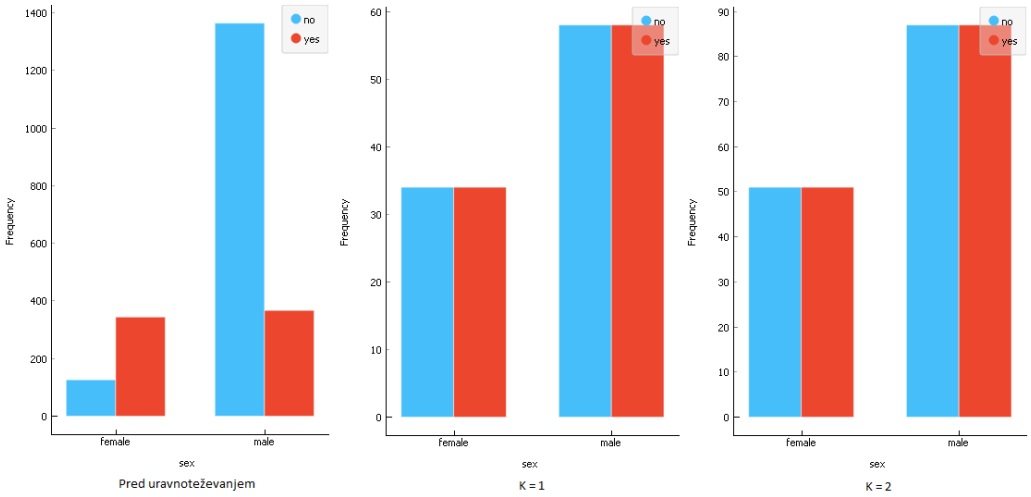
Pravilnost implementacije algoritma za uravnoteževanje v obliko widgeta je bilo potrebno testirati. Glavni razlog za testiranje je prenos algoritma za uravnoteževanje na python 3. Poleg testiranja pravilnega delovanja algoritma, se je testiralo tudi pravilni izvoz podatkov. Testirali smo na treh podatkovnih zbirkah. Podatkovne zbirke na katerih smo testirali:

- Podatkovna zbirka titanic, ki ima 2201 primerov.
- Podatkovna zbirka adult, ki ima 32561 primerov.
- Močno anonimizirana verzija podatkovne zbirke bolnikov z rakom na dojki, ki predstavlja stanje velikega števila podatkovnih zbirk v realnem svetu.

4.1 Podatkovna zbirka titanic

V podatkovni zbirki Titanic je 2201 primerov in 4 atributi. Atributi so status, spol, starost in preživetje. Status ima štiri vrednosti: prvi, drugi, tretji razred in posadka ali osebje. Spol ima dve vrednosti: ženska in moški. Starost ima vrednosti: 1, ki predstavlja otroke in 2, ki predstavlja odrasle. Preživetje hrani podatek o preživetju osebe potopa ladje Titanika.

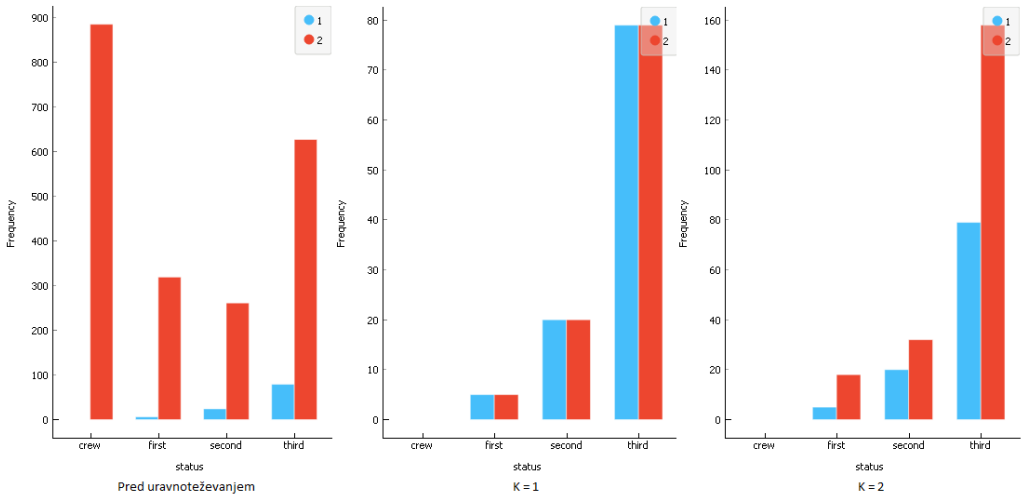
Pri uravnoteževanju podatkovne zbirke smo za atributa, s pomočjo katerih se ustvarijo skupine izbrali preživetje in starost. Preživelo je 57 otrok in 654 odraslih. Umrlo je 52 otrok in 1438 odraslih. 21% vseh primerov je ženskega spola. Med izbrane atribute smo dodali samo atribut spol. K vrednost smo nastavili na 1. Po uravneteževanju je algoritem vrnil p vrednost za atribut spol 1, kar pomeni, da med skupinami ni velikih razlik pri atributu spol. Uravnoteževanje smo ponovili s tem, da smo k nastavili na 2. Algoritem je ponovno vrnil p vrednost za atribut spol 1. Porazdelitev primerov po spolu in preživetju pri k je 1 in k je 2 se vidi na sliki 4.1.



Slika 4.1: Porazdelitev primerov po spolu in preživetju pred in po uravnoteževanju.

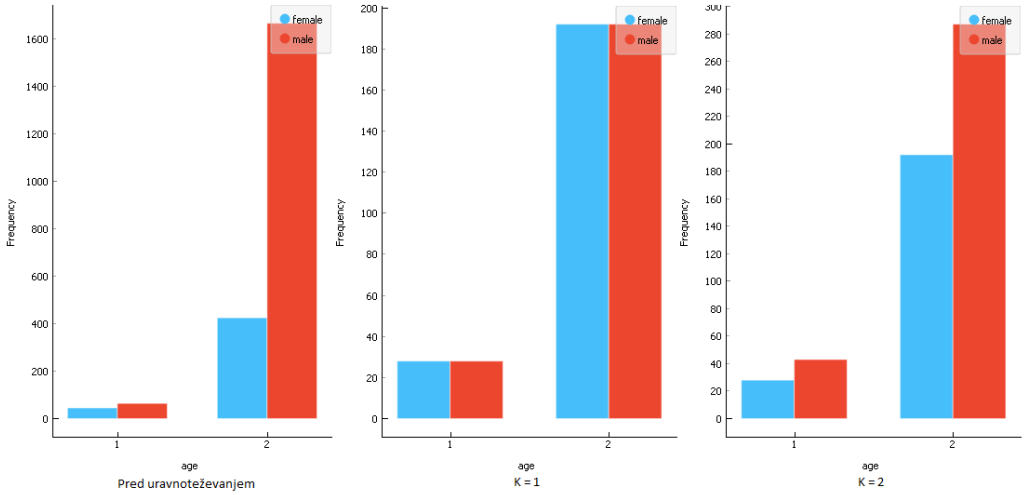
Nato smo atributom za uravnoteževanje dodali atribut status. Atributoma spol in status smo nastavili p vrednost na 1. Vrednost k smo nastavili na 1. Algoritem je p vrednost za atribut spol vrnil 1, za atribut status pa 0. To pomeni, da med skupinami pri atributu spol ni velikih razlik, pri atributu status pa so. Uravnoteževanje smo ponovili z nastavljanjem p vrednosti pri atributu status na 0.9, 0.5, 0.25, 0.2 ter 0.1. Pri vsakem poizkusu je algoritem vrnil p vrednost za atribut status 0, kar pomeni, da algoritem ni zmogel uravnotežiti skupin po atributu status. Razlog za to je, da ni primera, ki

bi bil otrok in hkrati del posadke. Porazdelitev po statusu in starosti na sliki 4.2.

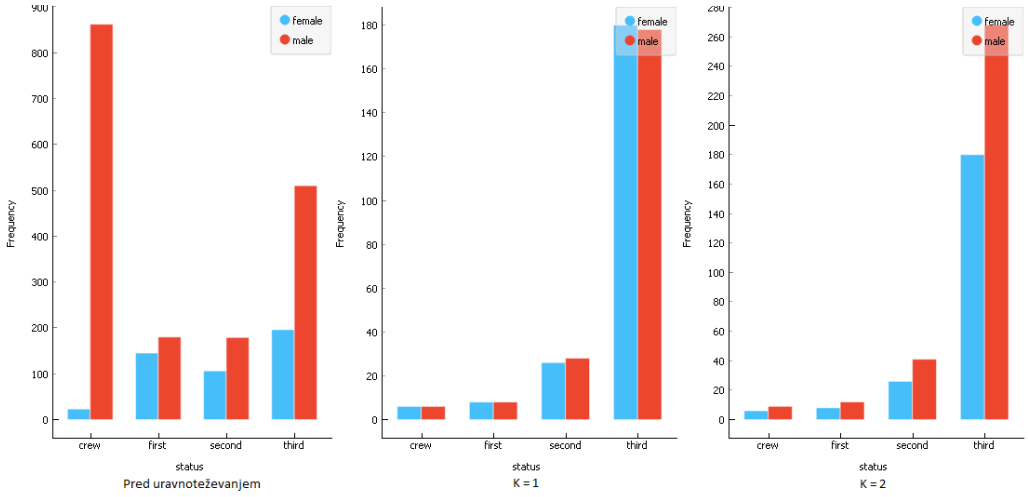


Slika 4.2: Porazdelitev primerov po statusu in starosti pred in po uravnoteževanju.

Skupine smo spremenili. Atribut starost smo zamenjali z atributom spol. Preživelo je 344 žensk in 367 moških. Umrlo je 126 žensk in 1364 moških. Uravnoteževali smo po atributih starost in status. Obema atributoma smo nastavili p vrednost na 1. K vrednost smo prvič nastavili na 1, v drugem poizkusu pa na 2. Algoritem je za oba atributa za uravnoteževanje vrnil vrednost 1, kar pomeni, da med skupinami po atributih status in starost ni velikih razlik. Porazdelitev primerov po starosti in spolu pred uravnoteževanjem in po uravnoteževanju pri k je 1 in k je 2 se vidi na sliki 4.3 in porazdelitev po statusu in spolu na sliki 4.4



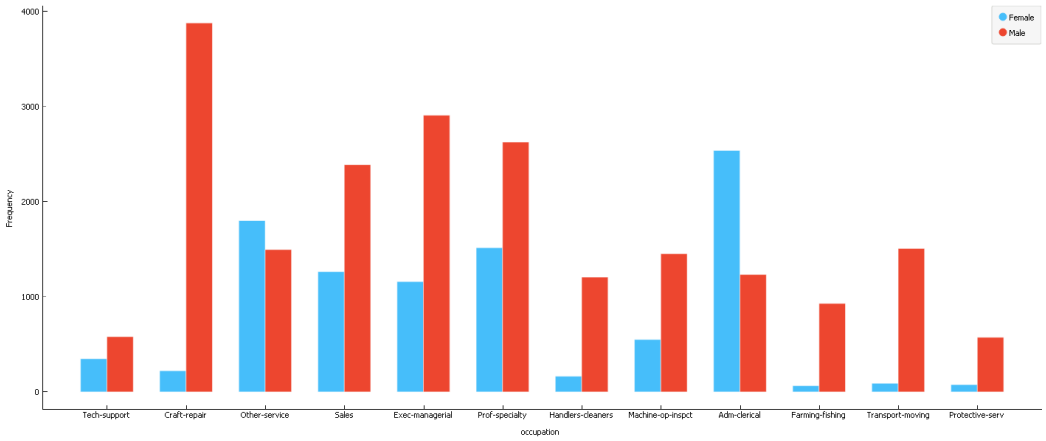
Slika 4.3: Porazdelitev primerov po starosti in spolu pred in po uravnoteževanju.



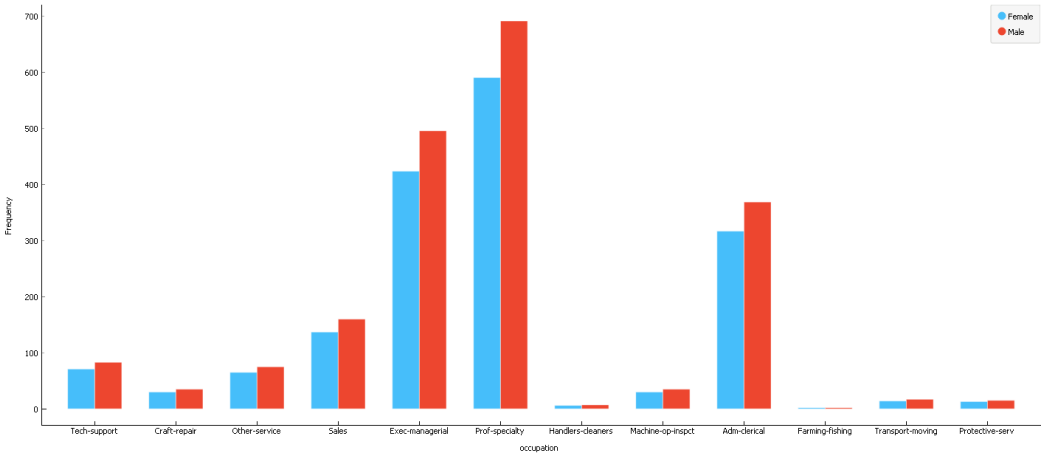
Slika 4.4: Porazdelitev primerov po statusu in spolu pred in po uravnoteževanju.

4.2 Podatkovna zbirka Adult

V podatkovni zbirki je 32561 primerov. Za ustvarjanje skupin smo izbrali atributa spol in y. Atribut spol predstavlja spol primera. Atribut y hrani podatek, če oseba na leto zasluži več ali manj kot 50000 denarja. Valuta v zbirki ni podana. Za uravnoteževanje smo uporabili atribute rasa, poklic in zakonski status. Izločili smo primere, ki imajo pri atributu poklic vrednost oborožene sile ali zasebne hišne storitve. Če bi obdržali primere s temi vrednostmi, ne bi mogli uravnoteževati po tem atributu, ker v podatkovni zbirniki npr. ni ženske, ki bi bila v vojski. Atributu poklic in atributu zakonski status smo določili p vrednost na 1. Atributu rasa pa na 0.5. Vsem atributom smo določili utež na 1. K smo nastavili na 1. Po uravnoteževanju je algoritem za vsak atribut vrnil p vrednosti 0.99. To pomeni, da lahko z 99% verjetnostjo rečemo, da med skupinami po izbranih treh atributih ni velike razlike. Porazdelitev primerov po poklicu in spolu pred uravnoteževanjem se vidi na sliki 4.5. Porazdelitev po uravnoteževanjem se vidi na sliki 4.6



Slika 4.5: Porazdelitev primerov po poklicu in spolu pred uravnoteževanjem.

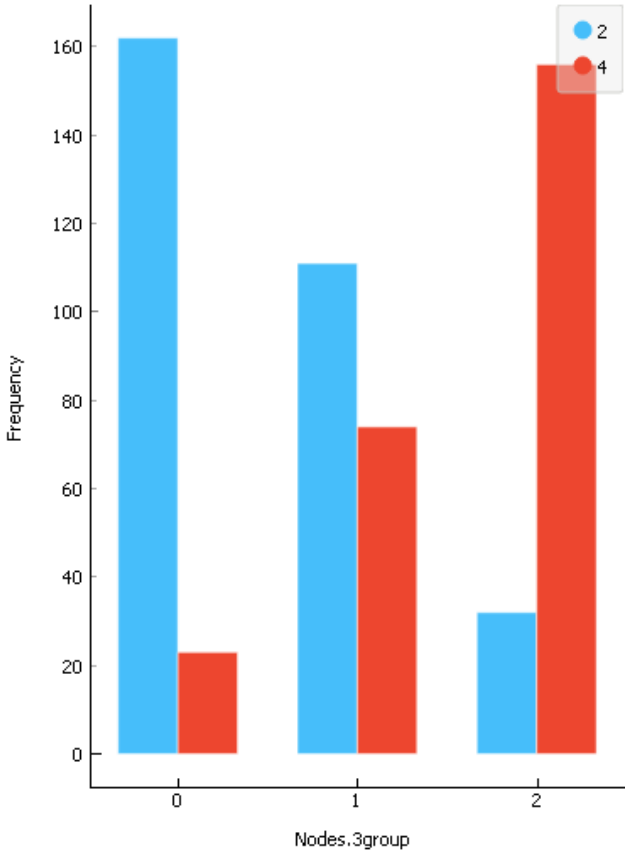


Slika 4.6: Porazdelitev primerov po poklicu in spolu po uravnoteževanju.

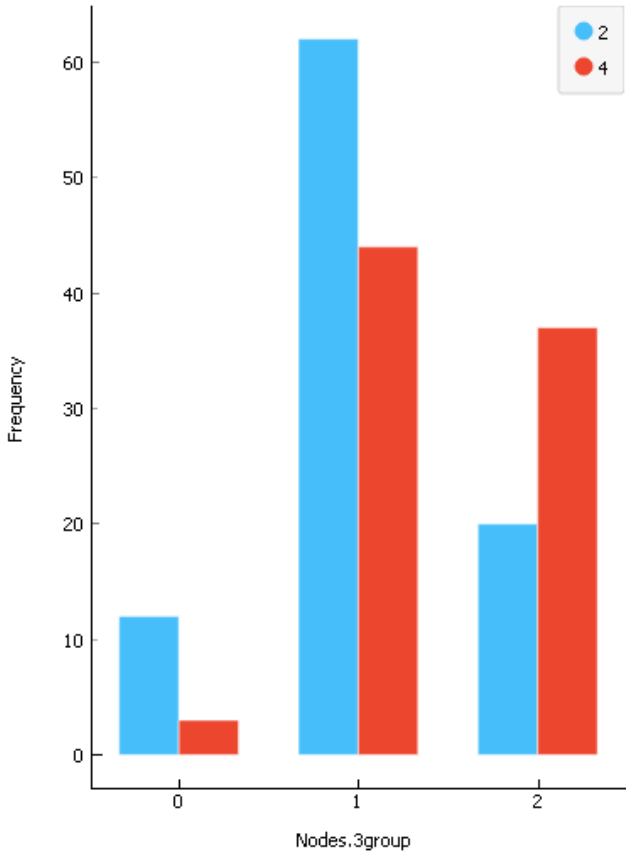
4.3 Podatkovna zbirka bolnikov z rakom na dojki

V podatkovni zbirki je 655 primerov. Primeri predstavljajo bolnike z rakom na dojki [1]. Atributa vrsta_kt in uPA_PAI1 smo uporabili za ustvarjanje skupin. Atribute zdravlje, Menopause, T.Grade, T-Size, T.HR, Invasion in Node.3group pa za uravnoteževanje. Pred uravnoteževanjem smo odstranili primere, ki so imeli pri kateremkoli izmed atributov za uravnoteževanje vrednost, ki predstavlja manjkajočo vrednost, kot so npr. 9 in 999. Izločili smo 99 primerov. V skupini, pri kateri ima vrsta_kr vrednost 4 in uPA_PAI1 vrednost 0, je 68 primerov. Pri skupini z vrednostjo atributa vrsta_kt je 2 in uPA_PAI1 je 0, je 61 primerov, pri vrednosti atributa vrsta_kt je 4 in uPA_PAI1 je 1, je 185 primerov in v zadnji skupini je 244 primerov. K vrednost smo nastavili na 3. Vsem atributom razen Menopause in Invasion smo določili p vrednost na 0.5. Atributoma Menopause in Invasion smo določili p vrednost na 0.25. Atributu Nodes.3group smo določili uteži vrednost 10. Vsem ostalim atributom za uravnoteževanje smo nastavili utež na 1. To pomeni, da je Nodes.3group pri uravnoteževanju bolj pomemben od ostalih atri-

butov. Algoritem je vrnil p vrednosti 0.81 za atribut zdravlje, 0.51 za atribut T.Grade, 0.55 za atribut Menopause, 0.92 za atribut T.HR, 0.96 za atribut T.Size, 0.93 za atribut Invasion in 0.95 za atribut Nodes.3group. Vsaka p vrednost v tem primeru predstavlja s kakšno verjetnostjo lahko rečemo, da med skupinam ni velike razlike po atributu, npr. z 93% verjetnostjo lahko rečemo, da med skupinami ni velike razlike po atributu T.Size. Porazdelitev primerov po atributih Vrsta_kt in Nodes.3group se pred uravnoteževanjem vidi na sliki 4.7. Porazdelitev primerov po uravnoteževanju se vidi na sliki 4.8.



Slika 4.7: Porazdelitev primerov po vrsti_kt in Nodes.3Group pred uravnoteževanjem.



Slika 4.8: Porazdelitev primerov po vrsti_kt in Nodes.3Group po uravnoteževanju.

Poglavje 5

Zaključki

V okviru diplomske naloge, je bil v program Orange implementiran algoritem za uravnoteževanje neuravnoteženih podatkovnih zbirk v obliko widgeta. Algoritem, ki se je implementiral je razvil in v svoji diplomi opisal Aleš Smodiš [8].

Algoritem za uravnoteževanje na podlagi dveh binarnih atributov ustvari štiri podskupine. Podskupine se uravnotežujejo na podlagi večih atributov za uravnoteževanje. Pri uravnoteževanju, se algoritmu podajo željene p vrednosti in uteži za vsak atribut za uravnoteževanje. Če se npr. atributu a nastavi p vrednosti na 0.5, to pomeni, da želimo da je vsaj 50% verjetnosti da med skupinami ni velikih razlik. Po uravnoteževanju algoritem za vsak atribut za uravnoteževanje vrne dejansko p vrednost.

Namen widgeta je ljudem, ki delajo s podatki in niso programerji, omogočiti enostavno uporabo algoritma za uravnoteževanje. Algoritem se je implementiral v program Orange, ker je program popolnoma brezplačen in preprost za uporabo. Nudi funkcionalnosti, ki so za uporabo algoritma za uravnoteževanje tako nujno potrebna, kot npr. nalaganje datoteke s podatki v program, kot uporabna pri nadaljni analizi rezultatov uravnoteževalnega algoritma.

Orange in algoritem sta oba razvita s programskim jezikom python. Algoritem je bilo potrebno prestaviti na python 3. Potrebno je bilo zasnovati in

nato implementirati uporabniški vmesnik. Uporabnik lahko sedaj s pomočjo uporabniškega vmesnika nastavi vse potrebne nastavitve za algoritem. Nato je bilo potrebno povezati vhodne parametre in podatkovno zbirko v objekt, ki se je podal algoritmu. Rezultate algoritma se uporabi za ustvarjanje dveh tabel. Prva tabela vsebuje vse primere, ki so ostali v eni izmed sedaj uravnoteženih skupinah. Druga tabela vsebuje vse primere, ki niso več v nobeni izmed skupin. Rezultati se lahko uporabijo za nadaljno analizo. Pri analizi se lahko uporabijo tudi druge funkcionalnosti programa Orange, npr. vizualizacijo podatkov [9] s pomočjo widgeta distributions.

Widget za uravnoteževanje se je testiral na treh podatkovnih zbirkah. Dve podatkovni zbirki adult in titanic je možno dobiti na spletni strani UCI datasets [4]. Zadnja podatkovna zbirka, ki hrani podatke o bolnikih z rakom na dojki pa smo dobili od zdravstva.

Widget za uravnoteževanje ni popoln. Nekatere izmed možnih izboljšav:

- Pri velikem številu primerov, algoritem dolgo uravnotežuje skupine. Možna izboljšava je proces uravnoteževanja prestaviti na drugo nit, zato da uporabniški vmesnik ne postane neodziven.
- Možno je uravnoteževati samo po diskretnih atributih. Izboljšava bi bila dodati možnosti uravnoteževanja po zveznih atributih.
- Uporabniški vmesnik za izbor atributov za ustvarjanje skupin, vedno ponuja vse atribute. Možna izboljšava je v kombiniranih seznamih prikazovati samo diskretne atribute, ki so binarni.

Literatura

[1] Simona Borštnar, Aleksander Sadikov, Barbara Možina, and Tanja Čufer. High levels of uPA and PAI-1 predict a good response to anthracyclines. *Breast cancer research and treatment*, 121(3):615–624, 2010.

[2] Sivarama P Dandamudi. *Introduction to assembly language programming: from 8086 to Pentium processors*. Springer Science & Business Media, 2013.

[3] Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, Miha Štajdohar, Lan Umek, Lan Žagar, Jure Žbontar, Marinka Žitnik, and Blaž Zupan. Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, 14:2349–2353, 2013.

[4] Dua Dheeru and Efi Karra Taniskidou. *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, 2017.

[5] Deborah J Mayhew. *Principles and guidelines in software user interface design*. Prentice-Hall, Inc., 1991.

[6] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[7] Tim Peters. The zen of python. In *Pro Python*, pages 301–302. Springer, 2010.

[8] Aleš Smodiš. *Strategije za uravnoteženo izbiro retrospektivnih podatkov za simulacijo prospektivnih raziskav*. Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2016.

[9] Alexandru C Telea. *Data visualization: principles and practice*. CRC Press, 2014.

[10] John M Zelle. *Python programming: an introduction to computer science*. Franklin, Beedle & Associates, Inc., 2004.