RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte

Atskaite par otro praktisko darbu

Studiju kurss “Mākslīgā intelekta pamati”

Komandas numurs: 16

Darba izpildītāji:
Haralds Ķempelis, 221RDB205
Egija Kokoreviča, 221RDB288
Dāniels Čulka, 221RDB304
Eduards Seļakovs, 221RDB286
Ingus Alfs Āboltiņš, 221RDB256

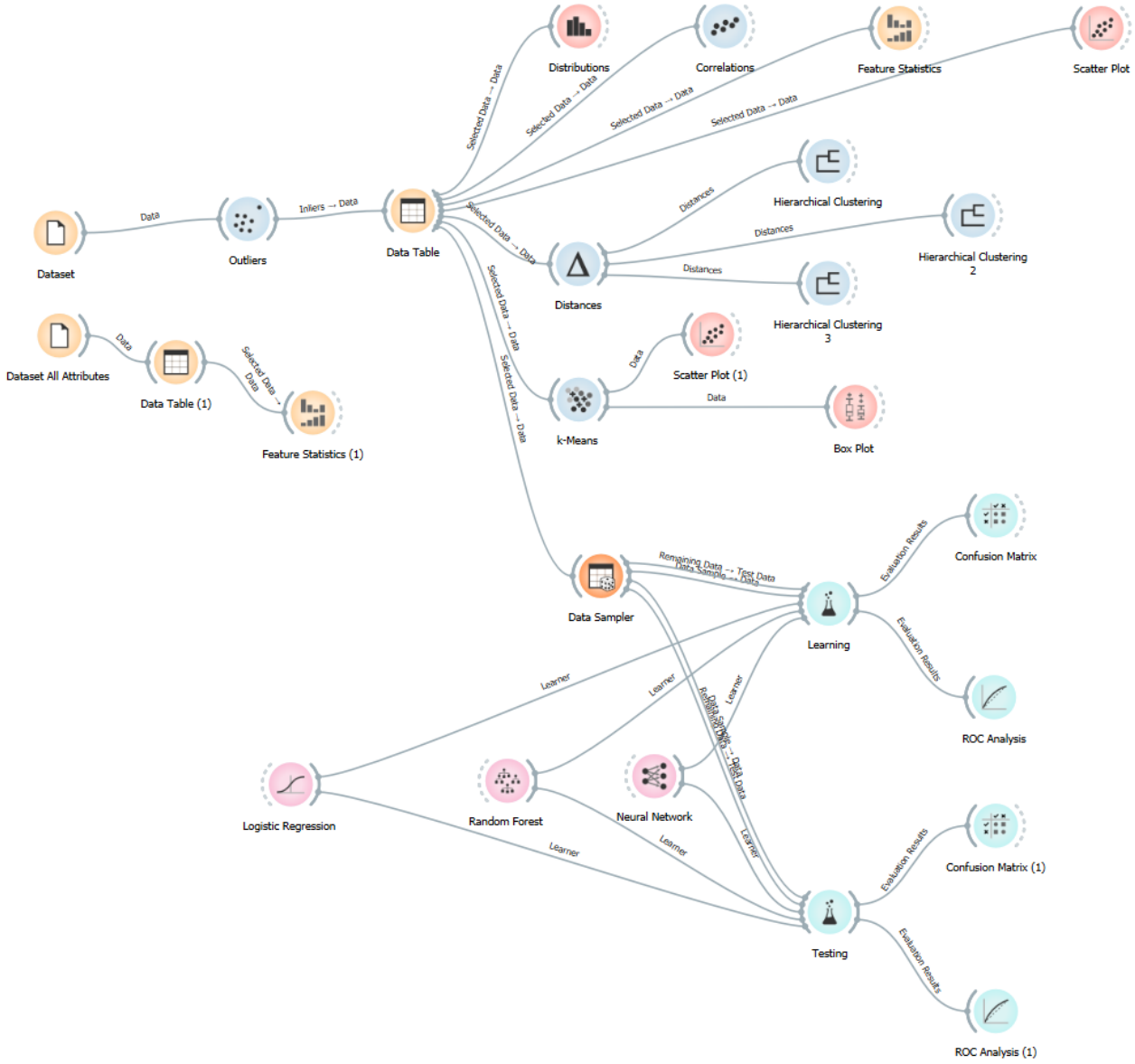
Mācībspēks:
Alla Anohina-Naumeca

Saite uz projektu:
https://github.com/HarKemp/MI_2PR

Saite uz datu kopu (.csv fails pieejams augstāk esošajā github saitē sadaļā “Dataset”):
https://archive.ics.uci.edu/dataset/145/statlog+heart

2023./2024.studiju gads

Orange rīka darbplūsma



1. attēls. Darba plūsmas attēlojums Orange rīkā

I daļa

Datu kopas apraksts

Datu kopas nosaukums:

Heart Disease – Sirds slimības

Datu kopas avots:

Datu kopa tika atrasta un “.csv” paplašinājuma fails tika iegūts avotā [15], taču tā tika paņemta no avota [16] . Savukārt šīs datu kopas, tika iegūtas no plašākas datu kopas[17], kas kopā satur vairāk par 70 atribūtiem ar informāciju par pacientiem. Attiecīgās datu kopas aprakstā norādīts, ka klasifikācijas uzdevumu un citu mākslīgā intelekta algoritmu pielietošanai iespējams lietot tikai dažus konkrētus atribūtus, kuri arī veido to datu kopu, kas tiks izmantota šajā darbā.

Iekļauj sevī datu kopas par sirds slimībām no 4 vietām - Klīvlenda, Ungārija, Šveice un "VA Long Beach". Datu kopu ziedoja Peter Turney

Datu kopas izveidotājs un/vai īpašnieks:

Andras Janosi, William Steinbrunn, Matthias Pfisterer, Robert Detrano

Datu kopas problēmsfēras apraksts:

Datu kopa atspoguļo to vai pacientam ir konstatēta kāda sirds slimība balstoties uz dažādu analīžu veiktiem rezultātiem. Mērķis ir izprast, kuri faktori visvairāk norāda uz sirds slimības esamību.

Datu kopas licencēšanas nosacījumi:

Creative Commons Attribution 4.0 International (CC BY 4.0)

Informācija par datu kopas savākšanas veidu vai procedūru:

Nav informācijas

Datu kopas satura apraksts

Datu objektu skaits datu kopā:

270

Datu kopas pazīmju (atribūtu) atspoguļojums kopā ar to lomām Orange rīkā:

	Name	Type	Role	Values
1	age	N numeric	feature	
2	sex	C categorical	feature	0, 1
3	**chest pain type**	C **categorical**	**feature**	
4	resting blood ...	N numeric	feature	
5	serum cholestoral	N numeric	feature	
6	fasting blood ...	C categorical	feature	0, 1
7	**resting ...**	C **categorical**	**feature**	
8	max heart rate	N numeric	feature	
9	exercise induce...	C categorical	feature	0, 1
10	oldpeak	N numeric	feature	
11	**ST segment**	C **categorical**	**feature**	
12	major vessels	N numeric	feature	
13	**thal**	C **categorical**	**feature**	
14	**heart disease**	C **categorical**	**target**	**1, 2**

1.1. attēls. Datu kopas pazīmju saraksts

Klašu skaits datu kopā:

2

Klašu apraksts:

Datu kopā ir 2 klases:

1 - norāda, ka pacientam nav novērota nekāda sirds slimība

2 - norāda, ka pacientam ir novērota sirds slimība

Datu objektu skaits, kas pieder katrai klasei:

1.1. tabula

Objektu skaits katrā klasē

Klases iezīme	Datu objektu skaits
1	150
2	120

Pazīmju apraksts:

1.2. tabula

Pazīmju apraksts

Pazīmes apzīmējums/nosaukums	Pazīmes skaidrojums	Vērtību tips	Vērtību diapazons
age	Pacienta vecums	Numeric	29 - 77 gadi
sex	Pacienta dzimums	Categorical (Binary)	0 – sieviete 1 - vīrietis
chest-pain	Norāda krūts reģiona sāpju veidu	Categorical	1 - tipiska angīna 2 – atipiska angīna 3 – sāpes, kuras neizraisa angīna 4 - asimptomātiski
Rest-bp	Asinspiediens miera stāvoklī	Numeric	94 – 200 mm Hg
Serum-chol	Holesterīns	Numeric	126 – 564 mg/dl
Fasting-blood-sugar	Vai cukura līmenis asinīs pārsniedz 120 mg/dl	Categorical (Binary)	0 - nepārsniedz 1 - pārsniedz
electrocardiographic	Elektrokardiogrāfijas rezultāti miera stāvoklī	Categorical	0 - normāls 1 - ar ST viļņa anomāliju (T viļņa inversijas un/vai ST pacēlums vai depresija > 0,05 mV)

			2 - norāda uz iespējamu vai noteiktu kreisā kambara hipertrofiju pēc "Estes" kritērijiem
Max-heart-rate	Maksimālais novērotais sirds ritms	Numeric	71 – 202 bpm (sitieni minūtē)
angina	Vai vingrošanas laikā ir novērota angīna	Categorical (Binary)	0 - Nē 1 - Jā
oldpeak	ST sašaurinājums, ko izraisa vingrošana attiecībā pret miera stāvokli	Numeric	0 – 6,2
slope	ST segmenta slīpums vingrinājuma laikā	Categorical	1 - augšupejošs 2 - plakans 3 - lejupejošs
Major-vessels	Lielo asinsvadu skaits, kuri iekrāsojās floroskopijā	Numeric	0 – 3 (skaits)
thal	Kādi defekti novēroti sirdī	Categorical	3 - normāls 6 - nelabojams defekts 7 - atgriezenisks defekts
Heart-disease	Vai ir sirds slimība	Categorical	1 – nav sirds slimība 2 – ir sirds slimība

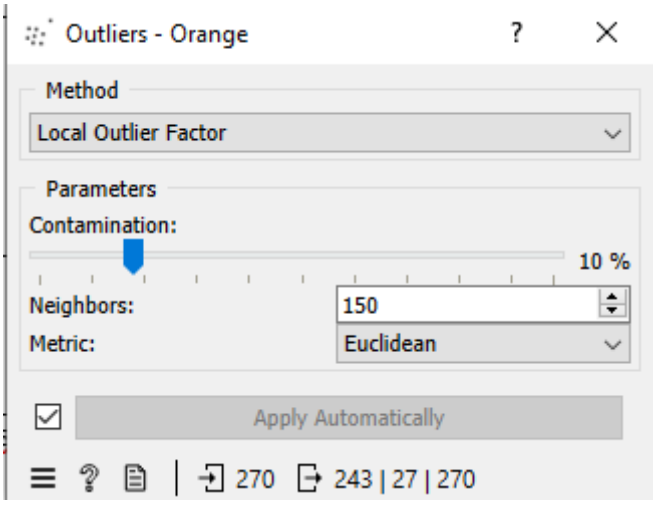
Datu faila struktūra:

	heart disease	age	sex	chest pain type	sting blood pressu	serum cholestoral	fasting blood sugar	electrocardiographi	max heart rate	ercise induced angi	oldpeak	ST segment	major vessels	thal
1	2	70	1	4	130	322	0	2	109	0	2.4	2	3	3
2	1	67	0	3	115	564	0	2	160	0	1.6	2	0	7
3	2	57	1	2	124	261	0	0	141	0	0.3	1	0	7
4	1	64	1	4	128	263	0	0	105	1	0.2	2	1	7
5	1	74	0	2	120	269	0	2	121	1	0.2	1	1	3
6	1	65	1	4	120	177	0	0	140	0	0.4	1	0	7
7	2	56	1	3	130	256	1	2	142	1	0.6	2	1	6
8	2	59	1	4	110	239	0	2	142	1	1.2	2	1	7
9	2	60	1	4	140	293	0	2	170	0	1.2	2	2	7
10	2	63	0	4	150	407	0	2	154	0	4.0	2	3	7
11	1	59	1	4	135	234	0	0	161	0	0.5	2	0	7
12	1	53	1	4	142	226	0	2	111	1	0.0	1	0	7

1.2. attēls. Datu faila struktūra

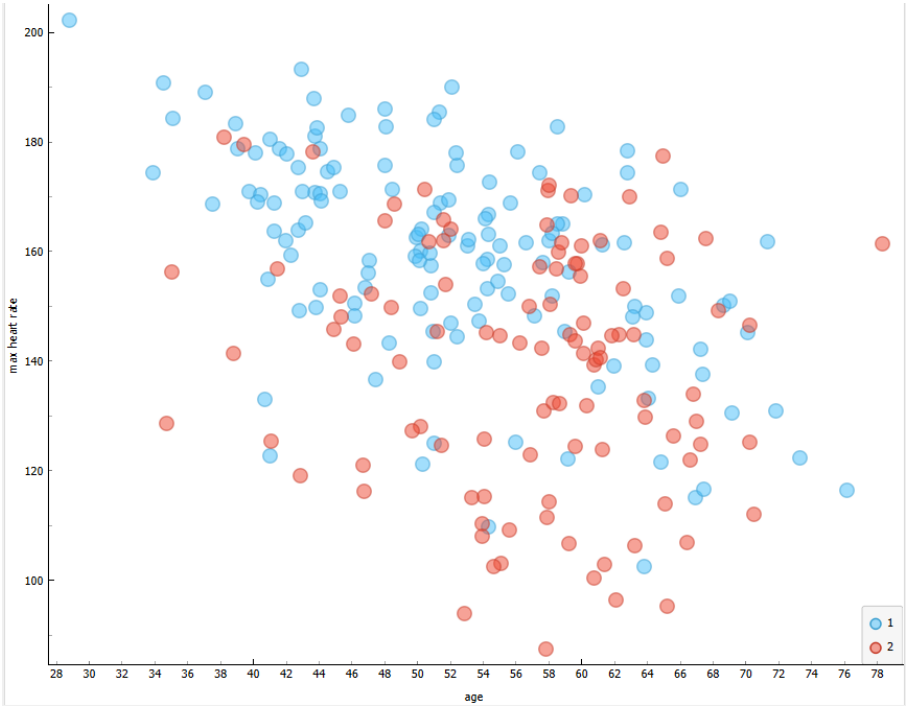
Informācija par trūkstošajām vai izlecošajām vērtībām:

Datu kopai tika noņemtas izlecošās vērtības 10% apmērā no datu kopas (skat. 1.3. att.). Pēc izlecošo vērtību noņemšanas datu kopā palika 243 datu objekti.

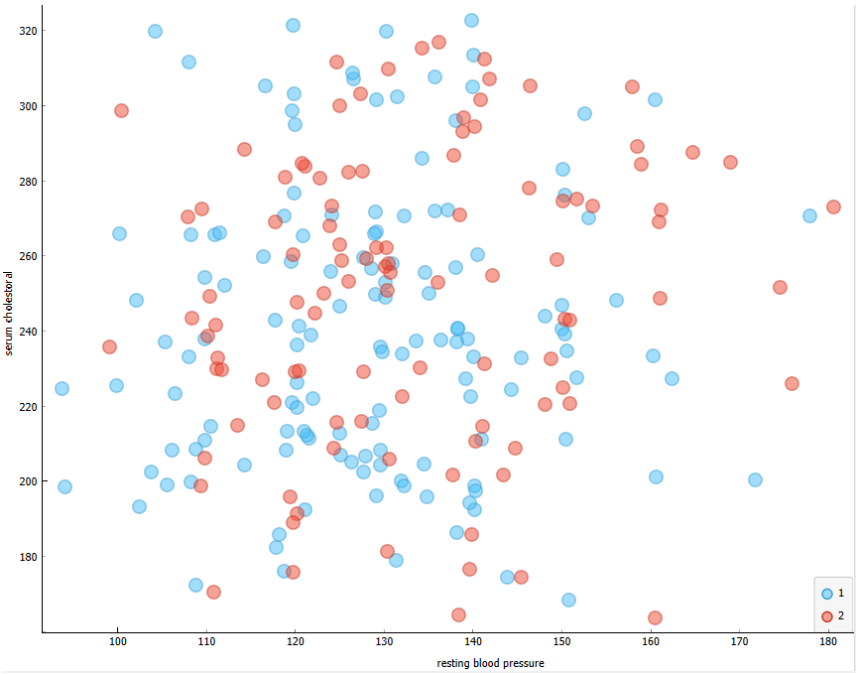


1.3. attēls. Izlecošo vērtību noņemšana

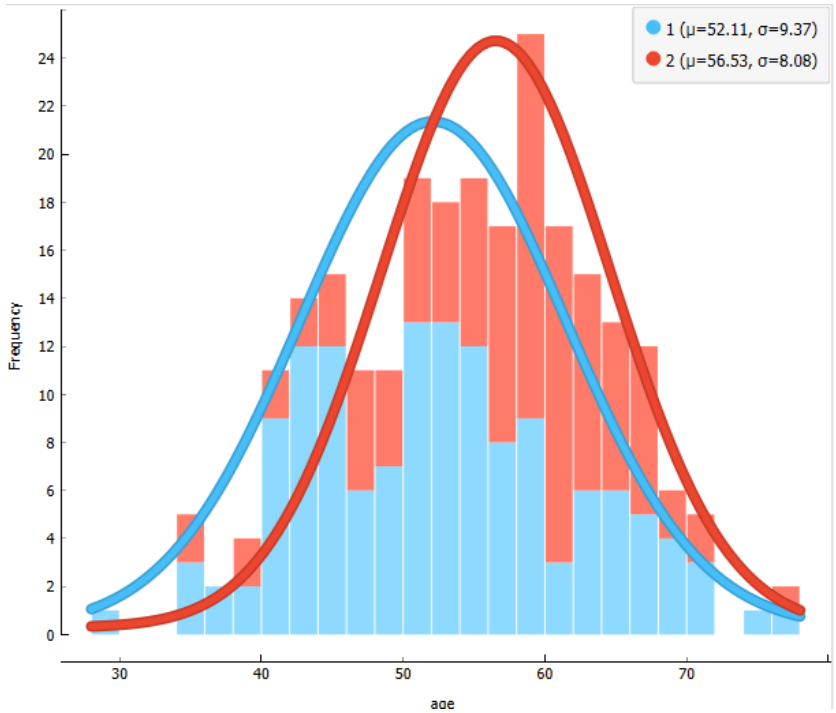
Datu kopas vizuālais un statistiskais atspoguļojums



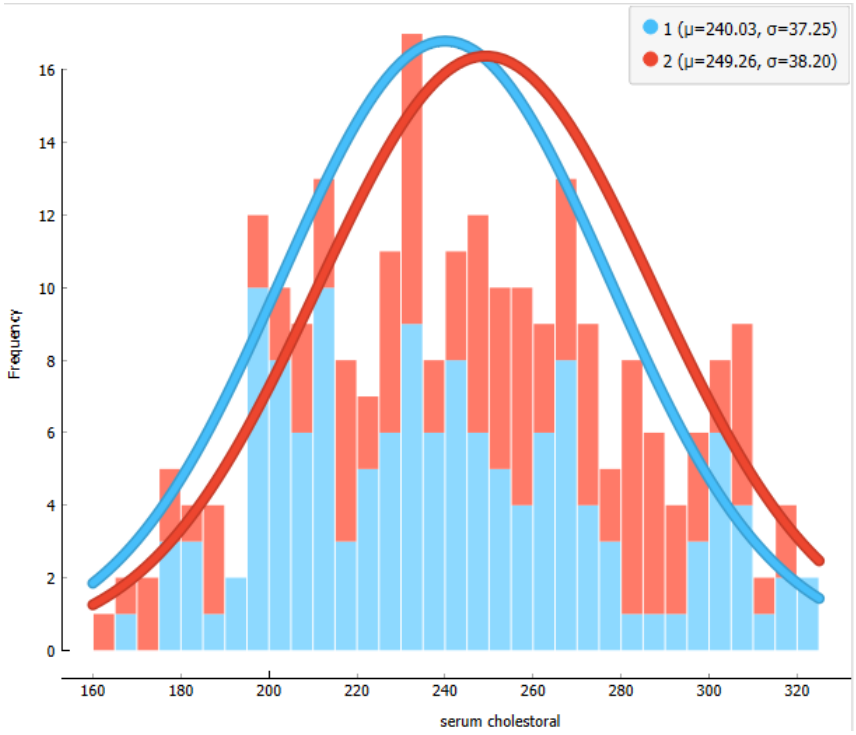
1.4. attēls. “age” un “max heart rate” izkliedes diagramma pēc to piederības konkrētai klasei



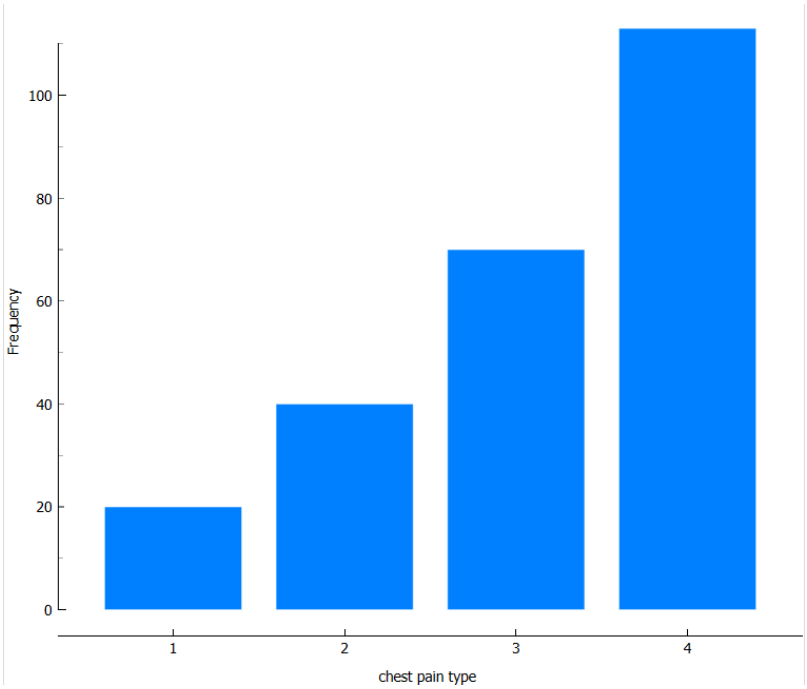
1.5. attēls. “resting blood pressure” un “serum cholestoral” izkliedes diagramma



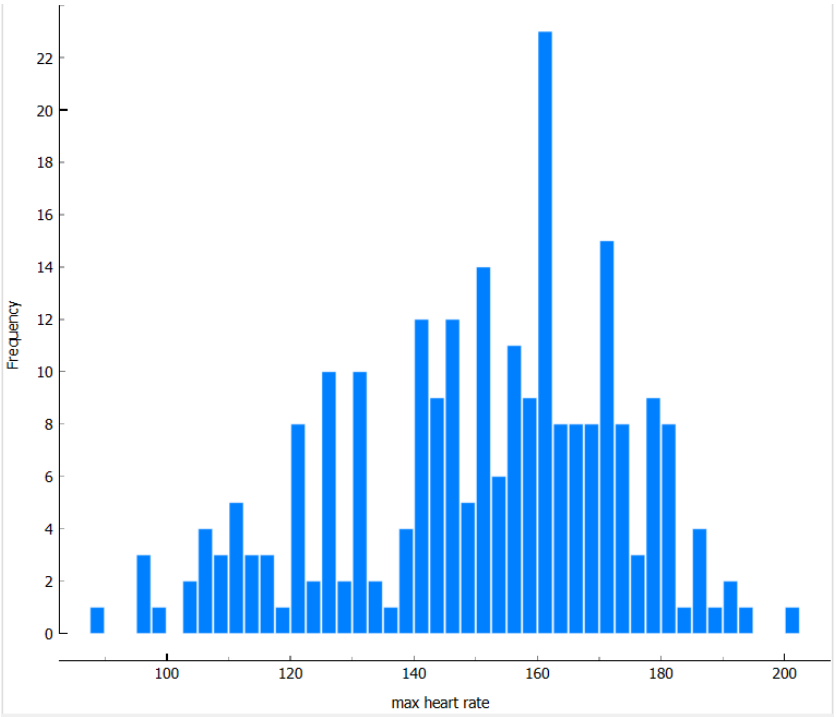
1.6. attēls. “age” histogrammas attēlojums un klases sadalījums



1.7. attēls. “serum cholestoral” histogrammas attēlojums un klases sadalījums



1.8. attēls. Sadalījumi pazīmei “chest pain type”, histogramma



1.9. attēls. Sadalījumi pazīmei “max heart rate”, histogramma

Name	Distribution	Mean	Mode	Median	Dispersion	Min.	Max.	Missing
age		54.43	54	55	0.17	29	77	0 (0 %)
chest pain type			4		1.19			0 (0 %)
exercise induced angina			0		0.634			0 (0 %)
fasting blood sugar			0		0.419			0 (0 %)
heart disease			1		0.687			0 (0 %)
major vessels		0.67	0	0	1.41	0	3	0 (0 %)
max heart rate		149.68	162	153.50	0.15	71	202	0 (0 %)
oldpeak		1.050	0.0	0.8	1.089	0.0	6.2	0 (0 %)
resting blood pressure		131.34	120	130	0.14	94	200	0 (0 %)
resting electrocardiogr... results			2		0.731			0 (0 %)
serum cholestoral		249.66	234	245	0.21	126	564	0 (0 %)

1.10. attēls. Statistiskie rādītāji rādītāji datu kopai

sex			1		0.629			0 (0 %)
ST segment			1		0.891			0 (0 %)
thal			3		0.844			0 (0 %)

1.11. attēls. Statistiskie rādītāji rādītāji datu kopai

Atbildes uz jautājumiem

Vai klases datu kopā ir līdzsvarotas, vai dominē viena klase (vai vairākas klases)?

Klases ir salīdzinoši līdzsvarotas. Varam redzēt, ka 150 pacientiem nav sirds slimību un 120 ir sirds slimības. (Skatīt 1.1. tabulu) Atšķirība starp klasēm piederošo datu objektu skaitu ir 20%.

Vai datu vizuālais atspoguļojums ļauj redzēt datu struktūru?

Datu struktūra ir grūti saskatāma, jo datu objekti no abām klasēm izkliedes grafikos pārsvarā atrodas vienādās pozīcijās, bet kopumā ir novērojams atdalījums, kad skatās uz galējiem datu objektiem, piemēram, 1.4.attēlā ir novērojama sakarība, starp vecumu un to piederību konkrētai klasei (jo vecāks cilvēks, jo lielāka iespējamība sirds slimībai), taču jāņem vērā, ka datu kopā pārsvarā arī atrodas vecāki cilvēki, kas rosina jautājumus par to, kā tika izvēlēti cilvēki šīs datu ievākšanas ietvaros.

Cik datu grupējumus ir iespējams identificēt, pētot datu vizuālo atspoguļojumu?

Datu grupējumi ir grūti atdalāmi, jo tie saplūst kopā un nav īpaši izteikti atribūti, kas tos nodalītu

Vai identificētie datu grupējumi atrodas tuvu viens otram vai tālu viens no otra?

Lielākā daļa datu objektu, atrodas ļoti tuvu viens otram. Izņēmumi ir dažas izlecošās vērtības.

Secinājumi, kas izriet no statistisko rādītāju analīzes

Redzams, ka vidējais vecums ir aptuveni 54 gadi, biežākais vecums ir 54 gadi (Skatīt 1.10. attēlu). Dati aptver pacientus no 29 līdz 77 gadiem, kas norāda uz plašu vecuma diapazonu, bet pārsvarā informācija ievākta par cilvēkiem, kas ir pāri pusmūžam. Tas varētu norādīt uz to ka mūsu iegūtie rezultāti nebūs pielietojami cilvēku grupām, kur lielākā daļa ir jaunāki cilvēki, piemēram, 20-30 gadu veci.

Pie krūts reģiona sāpju veidiem redzams, ka moda ir 4, kas norāda uz to, ka ir liela daļa gadījumu, kuros pacients sāpes nejūt, un redzams, ka tas ir arī plašāk novērojams tieši cilvēkiem kuriem konstatēta kāda sirds slimība.

II daļa

Algoritmu veidošanā izlaidām atribūtus ar bināra veida atribūtiem (1/0), jo to klātbūtne ierobežoja hierarhiskās klasterēšanas algoritma spēju sadalīt objektus klasteros. Otrās daļas ietvaros datu kopu pazīmju atspoguļojums (skat. 2.1. att.).

	Name	Type	Role	Values
1	age	N numeric	feature	
2	sex	C categorical	**skip**	0, 1
3	**chest pain type**	C **categorical**	**feature**	
4	resting blood pressure	N numeric	feature	
5	serum cholestoral	N numeric	feature	
6	fasting blood sugar	C categorical	**skip**	0, 1
7	**resting electrocardiog...**	C **categorical**	**feature**	
8	max heart rate	N numeric	feature	
9	exercise induced angina	C categorical	**skip**	0, 1
10	oldpeak	N numeric	feature	
11	**ST segment**	C **categorical**	**feature**	
12	major vessels	N numeric	feature	
13	**thal**	C **categorical**	**feature**	
14	**heart disease**	C **categorical**	**target**	**1, 2**

2.1. attēls. Datu kopas pazīmju (atribūtu) atspoguļojums kopā ar to lomām Orange rīkā.

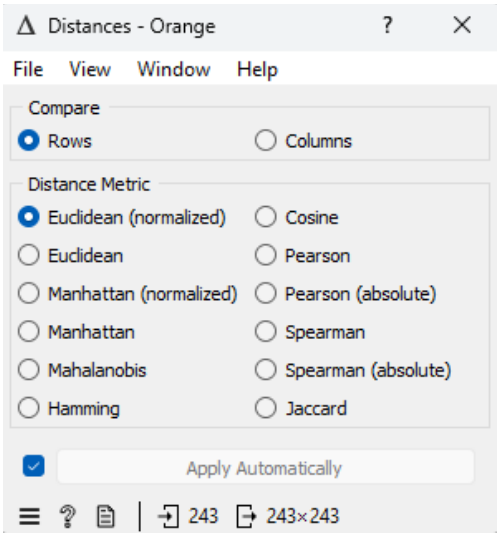
Hierarhiskā klasterēšana

Orange rīkā pieejamo hiperparametru apraksts:

2.1. tabula

Orange rīkā pieejamo hiperparametru apraksts

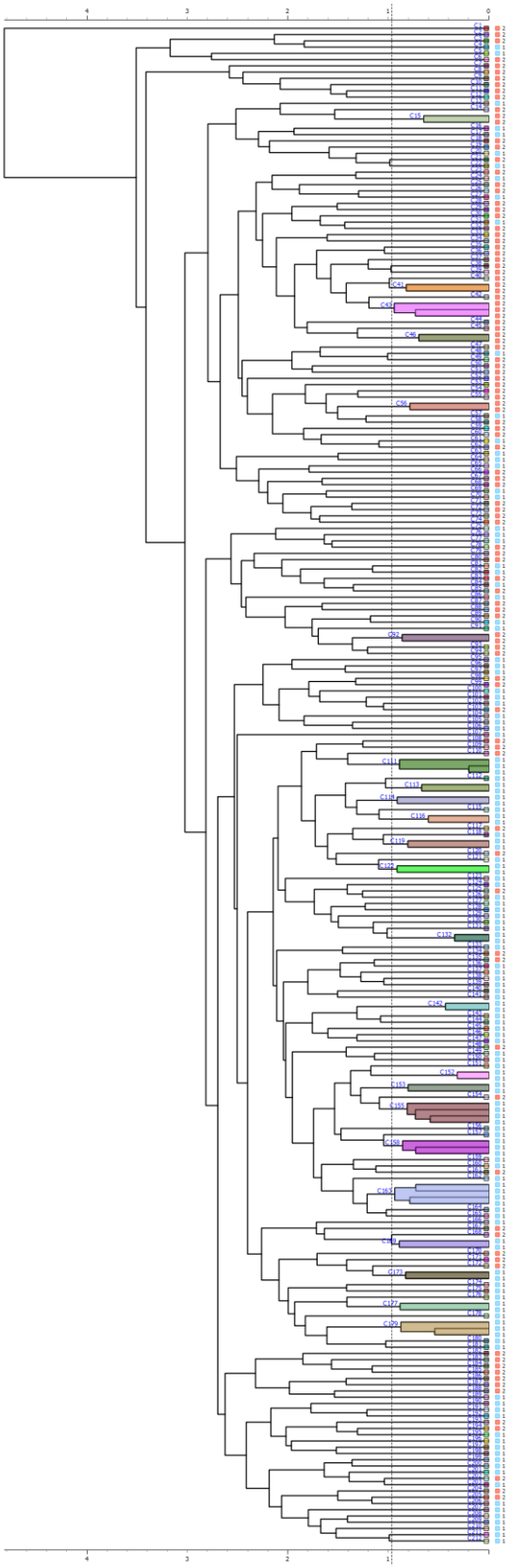
Hiperparametrs	Apraksts
Attāluma metrika (Distance Metric)	Attāluma metrika ir datu struktūra, kas sniedz attāluma vai līdzības mērījumus starp visiem datu punktiem datu kopā [3]. Izvēlēts Eiklīda ar normalizēšanu, jo iepriekš netika veikta normalizēšana
Saites kritērijs (Linkage Criterion)	Nosaka, kā aprēķināt attālumu starp klasteriem [3]. Izvēlēts vidējais (Average), jo izmēģinot dažādus kritērijus tika noskaidrots, ka šim ir labākie rezultāti



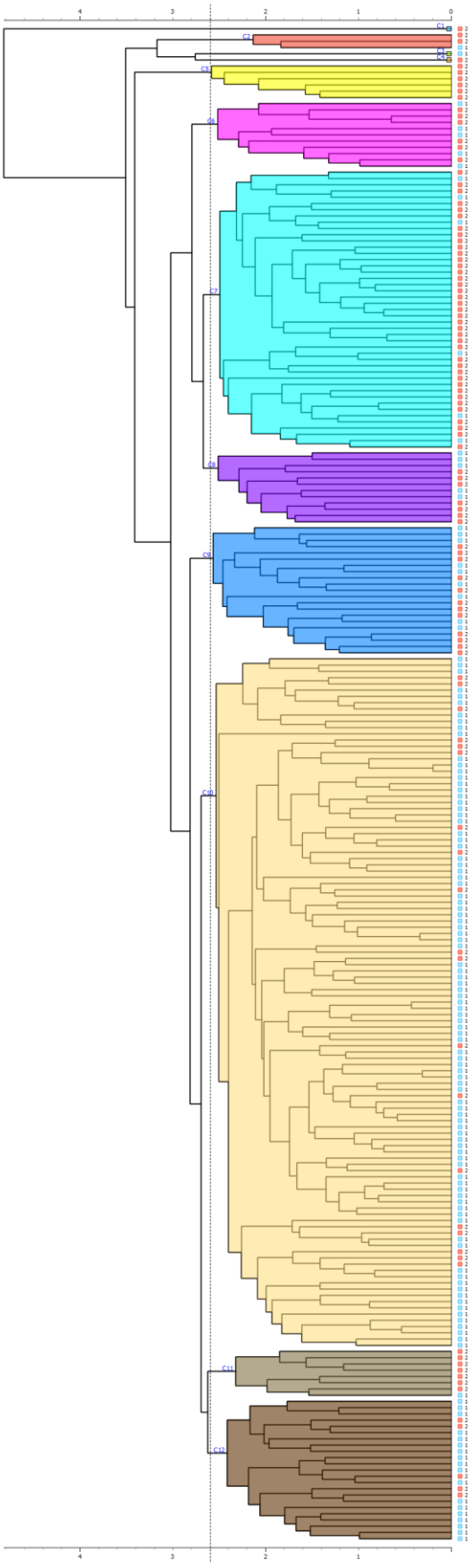
2.2. attēls. Attāluma metrika

2.3. attēls. Saites kritērijs

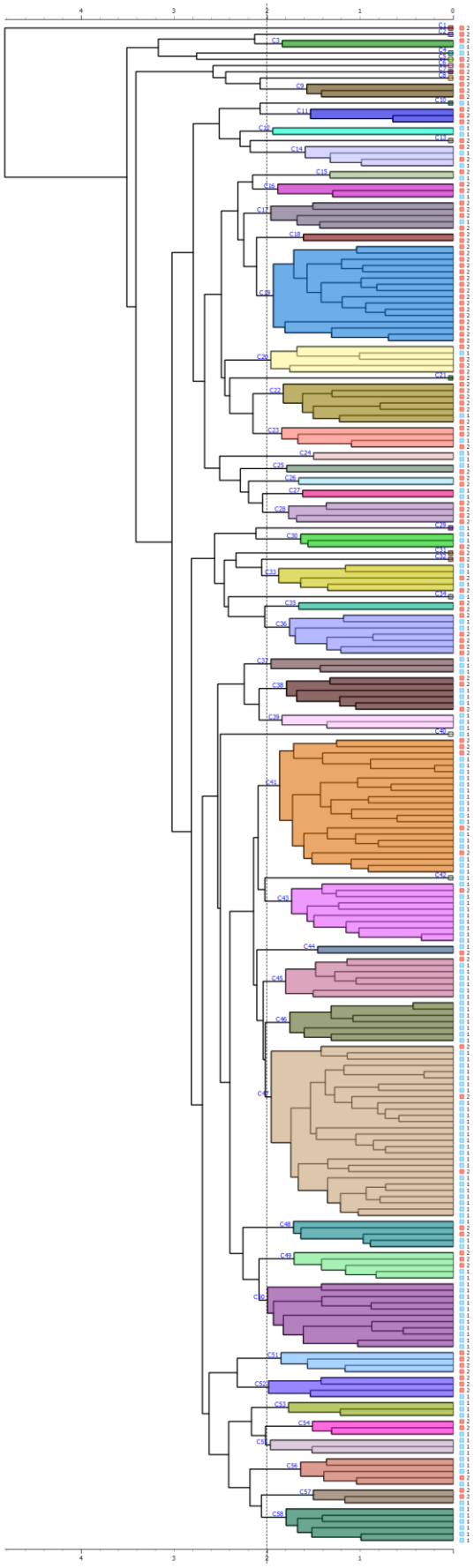
Eksperimentu apraksts



2.4. attēls. Rezultāti 1.eskperimentam



2.5. attēls. Rezultāti 2.eskperimentam



2.6. attēls. Rezultāti 3.eskperimentam

Secinājumi no eksperimentiem:

Pirmajā eksperimentā (skat. 2.4. att.), izdevās sasniegt klašu pilno atdalīšanu, bet ar salīdzinoši lielu klasteru skaitus, kas ir 212 klasteri ņemot vērā ka kopējais ierakstu skaits ir 243.

Otrajā eksperimentā (skat. 2.5. att.), atdalošā līnija nobīdīta uz kreiso pusi ar mērķi samazināt klasteru skaitu un nezaudējot tās atdalāmību. Izdevās samazināt klasteru skaitu līdz pat 12 klasteriem, bet neizdevās izdarīt to tā, lai klasteri būtu atdalīti. Pārsvarā klasteros ir vairākums viena no vērtībām, bet 3 no tiem klasteriem ir sadalījums vienlīdzīgs, kas nav labi.

Trešajā eksperimentā, (skat. 2.6. att.), atdalošā līnija nobīdīta uz labo pusi ar mērķi uzlabot klasteru atdalamību. Rezultātā ir 58 klasteri, tika atdalīti tie klasteri, kuros bija vienlīdzīgs sadalījums. Sanāca panākt rezultātu kurā ir klasteri ar vienu konkrētu vērtību, bet ir arī tādi, kuros ir pārākums vienai no vērtībām.

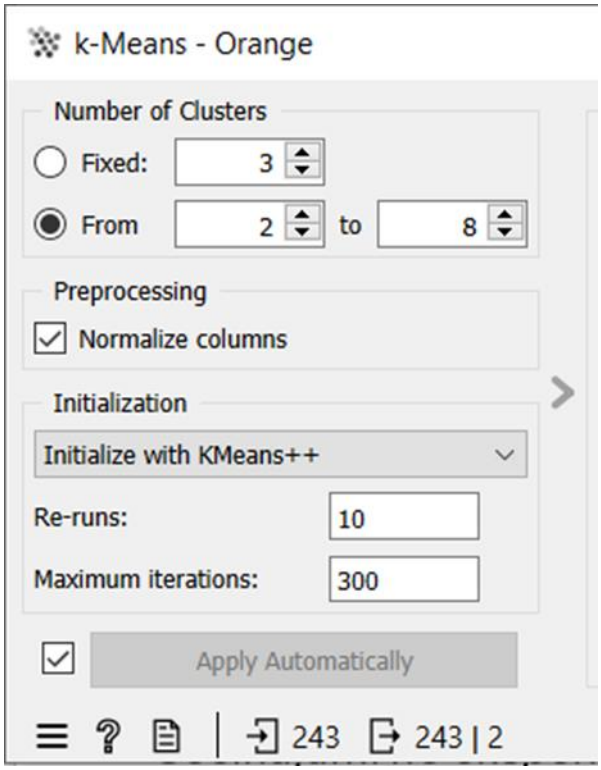
K-vidējo algoritms

Orange rīkā pieejamo hiperparametru apraksts:

2.2. tabula

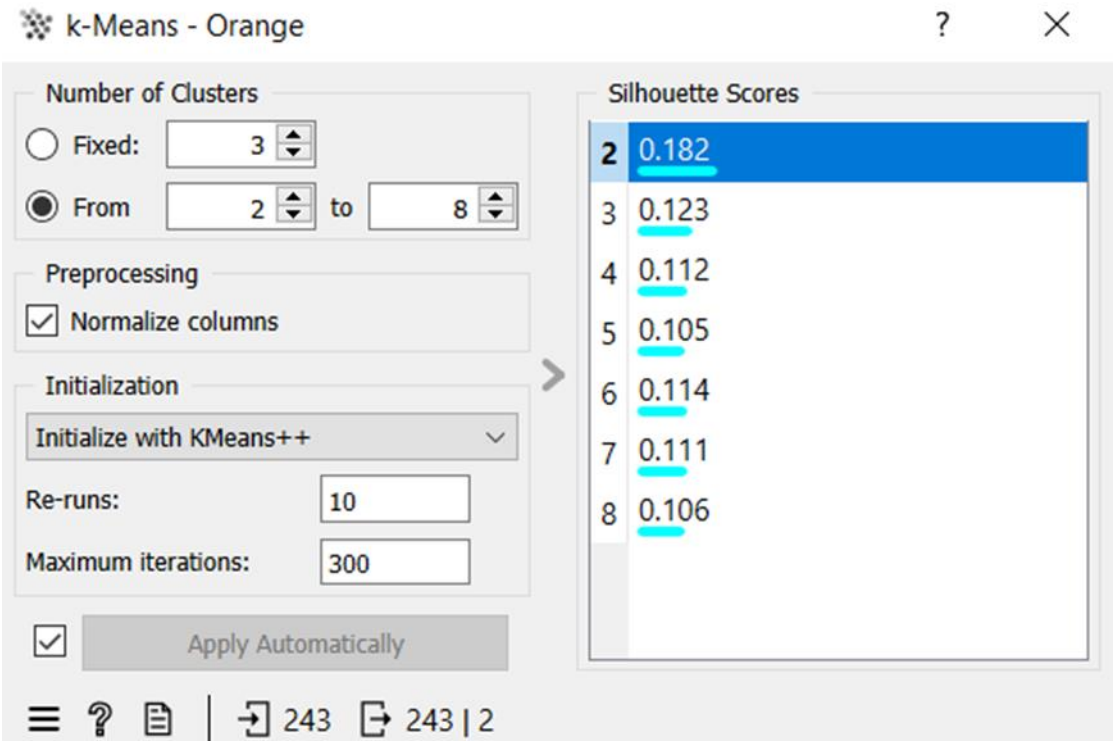
Orange rīkā pieejamo hiperparametru apraksts

Hiperparametrs	Apraksts
Klāsteru skaits	Ir atkarīgs no k vērtības, kas ir noteicošās, lai varētu klasificēt grupās. Meklē labāko centroīdu un sadala iegūtos punkuts grupās[7,8.9]
Initialization	Ir veids kā tiks noveitoti centroīdu sākumpunkti, ir divi veidi, nejaušības pēc vai ar kmeans++ algorimu[7,8,9]
Iterāciju skaits	Skaits cik reizes tiks veikts algoritms, lai atrast optimālo centroīdu[7,8,9]

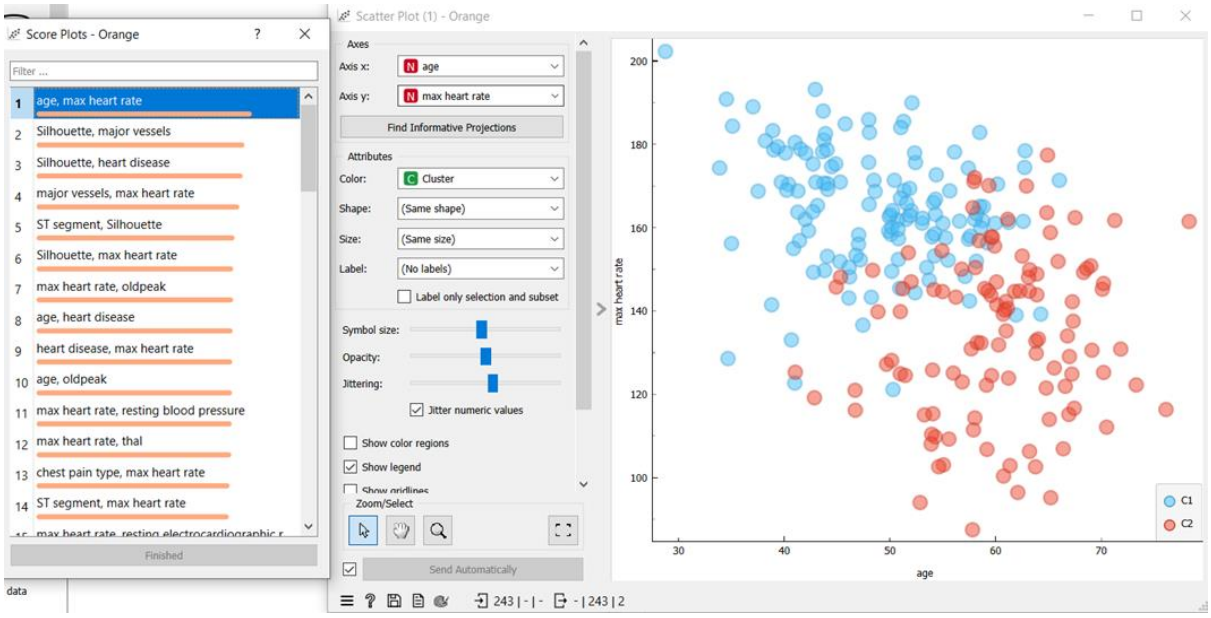


2.7. attēls. k-Means iestatījumi ar hiperparametriem

Eksperimentu apraksts



2.8. attēls. Silueta koeficenti, kas parādīti labajā pusē 8 k vērtībām

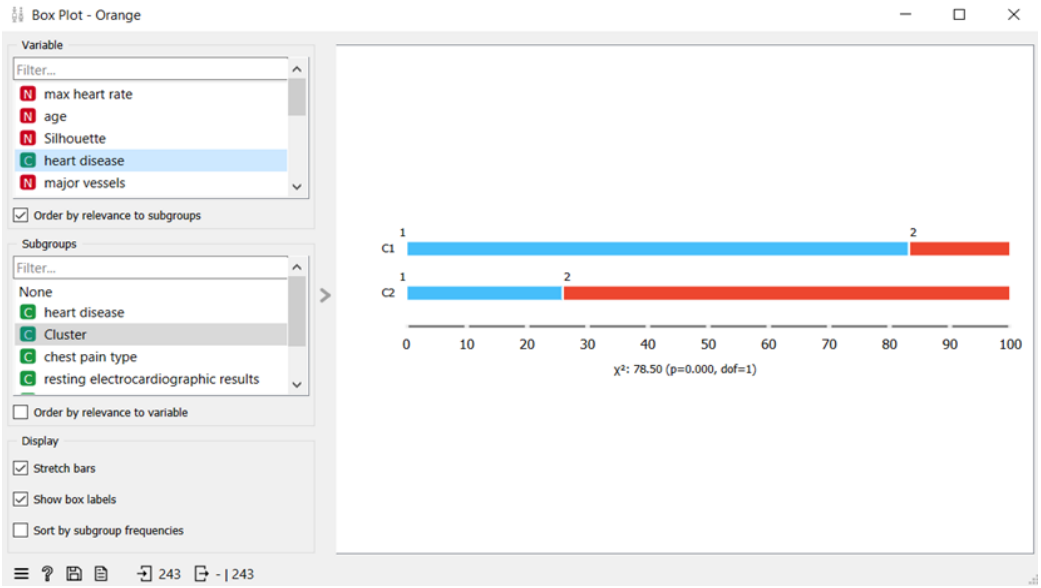


2.9. attēls. Izkliedes i diagrammas uzņēmums 2 k vērtībai, pie labākajiem iespējamiem x un y

Secinājumi no eksperimentiem:

Eksperimentam tika noņemtas kategoriskās vērtības, kuras bija ar 0 vai 1 vērtību variantiem. Jau spriežot pēc silueta varēja secināt, ka visi silueti pie jebkādas k vērtības bija tālu no 1 un tuvu 0, kas liecina par to, ka klāsteru robežas ir pārāk tuvu viena otrai un attēls saplūst īpaši[7], ja likt citas x un y vērtības. Ņemot nost visas kategoriskās vērtības rezultāti palielinās par 0.04, kas nemaina attēlu, ja iet runa par tik zemiem siluetiem.

Klasterējot divos klāsteros pēc target(heart disease), kur C1 nav sirds problēmu un C2 ir ar sirds problēmām, tas jauc tās kopā, pieņemu, ka tas var notikt, ja pie tām pašām apskates vērtībām vienam bija konstatētas problēmas, bet otram nē (skat. 2.10. att.).



2.10. attēls. BoxPlot uzņēmums, attēlots kāds ir klāsteru sadalījums [6]

Noslēguma secinājumi

Pamatojoties uz abu algoritmu darbību secinājumiem var spriest, ka abos rezultāti nav tie labākie un ir grūti atdalīt klasteros. Pēc hierahiskās klasterēšanas var spriest, ka ir iespējams atdalīt ideāli gadījumā, kad klasteru skaits gandrīz sakrīt ar ierakstu skaitu, kas ir slikts rezultāts.

Abos algoritmos diezgan vienādi rezultāti, ja ņemt hierarhiskās klasterēšanas 3. eksperimenta rezultātus, kas ir labākais rezultāts ņemot salīdzinoši mazu daudzumu klasteru. Gan k-vidējā algoritmā, gan hierarhiskajā klasterēšanā, klasteros nav vienlīdzīgs sadalījums, bet gan ir pārākums vienai vai otrai vērtībai. Bet dažos klasteros hierarhiskās klasterēšanā ir vienas vērtības ieraksti. Pēc šī var secināt, ka ar šo datu kopu algoritmi nostrādāja līdzvērtīgi, bet ņemot vērā, ka ar hierarhisko klasterēšanu ir klasteri ar vienu vērtību tad šis algoritms nostrādāja labāk.

III daļa

3. daļā tika turpināta 2. daļā konfigurētās datu kopas izmantošana, lai iegūtie rezultāti būtu salīdzināmi un saglabātu vienotu Orange rīka darba plūsmu gan 2., gan 3. daļai (Skat. 2.1. attēlu).

Izvēlēto algoritmu apraksts

Pirmā algoritma nosaukums: Loģistiskā regresija

Pirmā algoritma apraksts:

Loģistiskā regresija ir statistisks modeļu veids, kas tiek izmantots, lai paredzētu bināru iznākumu vai kategoriju, pamatojoties uz neatkarīgo mainīgo vērtībām. Šis modelis ir piemērots gadījumos, kad atkarīgais mainīgais ir binārs (piemēram, jā/nē, 1/0), un ir vajadzība prognozēt iespējamību, ka šis notikums notiks vai nenotiks.

Loģistiskās regresijas modeļa apmācība sastāv no svaru atrašanas, kuri minimizē klasifikācijas kļūdas visā datu kopā. Šis process notiek, pielietojot maksimālās ticamības metodi vai gradienta nolaidumu metodi.[20]

Otrā algoritma nosaukums: Random Forests

Otrā algoritma apraksts:

Random forests algoritms, ko izmanto gan klasifikācijas, gan regresijas problēmu risināšanai. Īsumā par algoritma darbību: tiek izveidots "bootstrap" datasets, kas sastāv no juceklīgi izvēlētiem ierakstiem no orģinālā dataseta. Parasti tiek izvēlētas tikai 2/3 no kopējiem ierakstiem. Šī procesa gaitā daži individuāli ieraksti, var tikt ierakstīti vairākas reizes, radot dublikātus. Atlikušo 1/3 ar ierakstiem, sauc par "out-of bag-dataset", kas tiks izmantots vēlāk, lai novērtētu šī algoritma precizitāti.

Pēc tam izmantojot "bootstrap" datasetu tiek ģenerēti koki. Priekš katra koka tiek uzģenerēts jauns "bootstrap" datasets. Līdzīgi "decision trees" algoritmam tiek ģenerēti koki, tomēr "Random forest" algoritmā katram kokam attribūti. no kura tas tiks ģenerēts, atšķirsies. Tas no cik atribūtiem tiks ģenerēts koks, ir atkarīgs no izvēlētā hiperprarametra. [17]

Hiperparametru apraksts

3.1. tabula

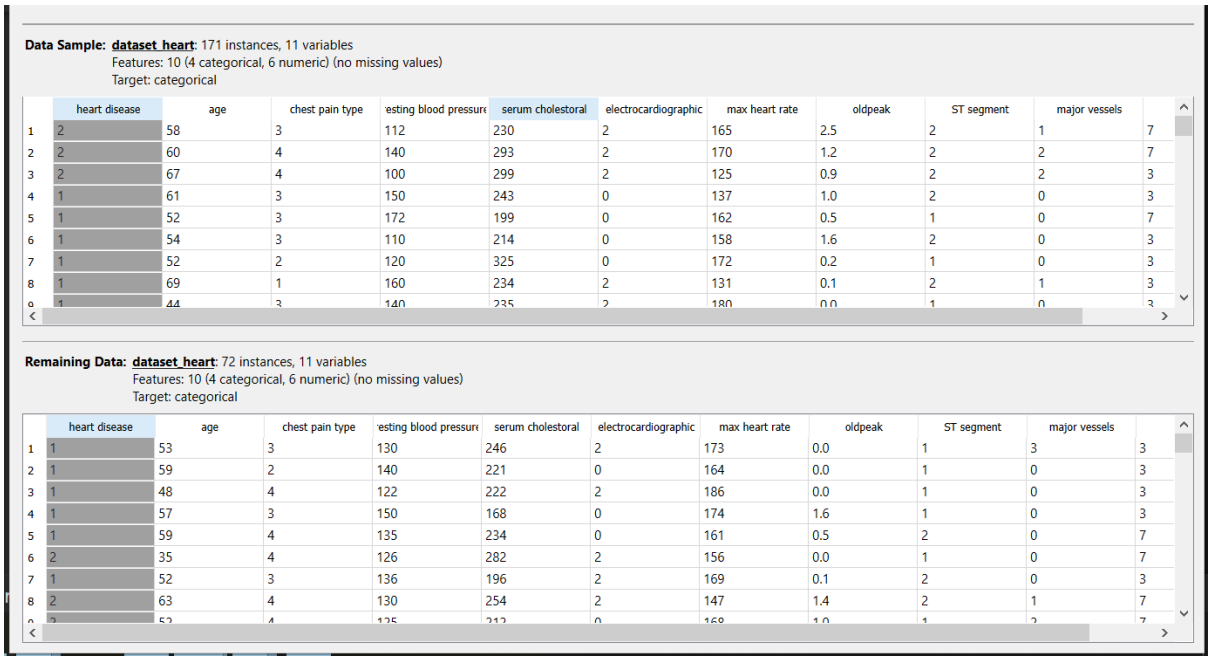
Hiperparametru apraksts

Hiperparameters	Apraksts un vērtības
Mākslīgo neironu tīkli	
Neironu skaits apslēptajos slāņos	Neironu skaits katrā slēptajā slānī (katra ar komatu atdalītā vērtība apzīmē konkrētajā slānī esošo neironu skatu).[10]
Aktivācijas funkcija	Aktivācijas funkcija slēptā slāņa neironiem Pieejamās funkcijas [10]: 1) Identity – noderīgs modeļiem, kuriem ir tādi slāņi, kuru neironu skaits ir krietni mazāks nekā slāņos, kuri atrodas pirms tā. 2) Logistic - loģistiskā funkcija, noderīga gadījumos, kad nepieciešams noteikt varbūtību, kā rezultātu (starp 0 un 1). [11] 3) tanh - Hiperboliskā tangensa funkcija sadala vērtības starp -1 un 1. [11] 4) ReLu - Taisnās lineārās vienības funkcija (Rectified linear unit function). Funkcija visas vērtības, kas ir zem 0, kā 0 un šī funkcija sniedzas līdz bezgalībai.[11]
Svaru optimizācijas algoritms	Nosaka algoritmu svaru optimizācijas veikšanai [10]: 1)L-BFGS-B – kvazi-Ņūtona metodes algoritms, kura izpildei nav nepieciešams liels datora atmiņas apjoms. [13] 2)SGD – stohastisks gradienta algoritms. 3)Adam – stohastisks, uz gradienta algoritma bāzēts optimālā risinājuma meklētājs. Tās mērķis ir minimizēt soda funkcijas vērtību.[12]
Mācīšanās ātrums	Parametrs, ko padod L2 regularizācijas algoritmam, kurš visus svarus samazina proporcionāli un nepieļaj svara samazināšanu līdz nullei. Noved pie tā, ka visiem svariem ir gandrīz vienādi mazas vērtības[10,14]
Maksimālais iterāciju skaits	Nosaka maksimālo iterāciju skaitu, kuru sasniedzot algoritms pārtrauks apmācīšanas procesu.
Loģistiskā regresija	
Regularizācijas tips	Iespēja izvēlēties starp trim vērtībām : L2, L1 un neviens. L1 (LASSO) un L2 (ridge) ir diva veida regularizācijas tipi, kā ieviest papildu ierobežojumus loģistiskajā regresijas modelī, lai uzlabotu tā vispārējo veiktspēju, novēršot pārmērīgu pielāgošanos apmācības datiem. L1 regularizācija veicina svaru saspiešanu un atsevišķu koeficientu samazināšanu

	līdz 0, bet L2 regularizācija cenšas samazināt vispārējo modeļa kompleksitāti, novēršot pārmērīgu jutīgumu pret individuāliem novērojumiem un pārlieku pielāgošanos apmācības datiem.[1,2,4]
C	Tiek izmantots kopā ar L1 vai L2 regularizāciju. Šis hiperparametrs nosaka, cik lielu ietekmi regulārizācija izdara uz modeļa pielāgošanos apmācības datiem. Vertību skala ir no 1000 līdz 0.001.[1]
Līdzsvarot klašu sadalījumu	Opcija: iespējot vai nē. Tiek izmantota, lai samazinātu vai novērstu nevienlīdzību starp dažādām klasēm apmācības datu kopā.[1]
Random Forest	
Koku skaits	Ar šo vērtību ir iespējams mainīt uzģenerēto koku skaitu, daudzveidību. Vērtību skala ir no 1 līdz 9999.[19]
Atribūtu skaits katrā "dalīšanā"	Šī vērtība norāda cik atribūtu(pēcteču) tiks izvēlēts katram uzģenerēto koku mezgliem. Vērtību skala ir no 1-14 mūsu gadījumā, jo mūsu datukopai ir tikai 14 atribūtu.[19]
Līdzsvarot klašu sadalījumu	Opcija: iespējot vai nē. Tiek izmantota, lai samazinātu vai novērstu nevienlīdzību starp dažādām klasēm apmācības datu kopā.[19]
Replicējama apmācība	Opcija: iespējot vai nē. TIek izmantots, lai katrā pie vieniem un tiem pašiem hiperparametriem tiek izmantoti tie paši uzģenerētie koki, nevis citi.[19]
Limitēt dziļumu katram individuālam kokam	Opcijas: iespējot vai nē. Vērtību skala:1-50. Limitē dziļumu uzģenerētajiem kokiem, kas ļauj padarīt apmācības ātrākas.[19]
"Nedalīt" apakškopas mazākas par	Opcijas: iespējot vai nē. Vērtību skala: 2-999. Mūsu gadījumā aktuālā vērtību skala būtu 2-14. Šis hiperparametrs ļauj noteikt to cik apakšdatukopas instancēm nepieciešams būt pirms sadalīt uzģenerētā koka mezglu sīkāk. Tas ļauj izvairīties no pārlieku pielāgošanās datukopai. Tomēr ja vērtība ir pārāk liela tas izraisīt nepietiekamu pielāgošanos, radot sakaribas ar mazu precizitāti.[18, 19]

Informācija par testa un apmācības datu kopām

Apmācības datu kopai tika lietoti 70% no datu kopas datu objektiem, pārējie datu objekti tika lietoti testēšanai. Eksperimentos tiks lietots “Test and Score” logrīks ar “cross validation” opciju.



Data Sample: dataset_heart: 171 instances, 11 variables
Features: 10 (4 categorical, 6 numeric) (no missing values)
Target: categorical

	heart disease	age	chest pain type	esting blood pressure	serum cholestoral	electrocardiographic	max heart rate	oldpeak	ST segment	major vessels	
1	2	58	3	112	230	2	165	2.5	2	1	7
2	2	60	4	140	293	2	170	1.2	2	2	7
3	2	67	4	100	299	2	125	0.9	2	2	3
4	1	61	3	150	243	0	137	1.0	2	0	3
5	1	52	3	172	199	0	162	0.5	1	0	7
6	1	54	3	110	214	0	158	1.6	2	0	3
7	1	52	2	120	325	0	172	0.2	1	0	3
8	1	69	1	160	234	2	131	0.1	2	1	3
9	1	44	3	140	235	2	180	0.0	1	0	3

Remaining Data: dataset_heart: 72 instances, 11 variables
Features: 10 (4 categorical, 6 numeric) (no missing values)
Target: categorical

	heart disease	age	chest pain type	esting blood pressure	serum cholestoral	electrocardiographic	max heart rate	oldpeak	ST segment	major vessels	
1	1	53	3	130	246	2	173	0.0	1	3	3
2	1	59	2	140	221	0	164	0.0	1	0	3
3	1	48	4	122	222	2	186	0.0	1	0	3
4	1	57	3	150	168	0	174	1.6	1	0	3
5	1	59	4	135	234	0	161	0.5	2	0	7
6	2	35	4	126	282	2	156	0.0	1	0	7
7	1	52	3	136	196	2	169	0.1	2	0	3
8	2	63	4	130	254	2	147	1.4	2	1	7
9	2	53	4	125	212	0	168	1.0	1	2	7

3.1. attēls. Informācija par datu kopām

Datu objektu skaits apmācības datu kopā:

172

Datu objektu % proporcija apmācības datu kopā:

3.2. tabula

Datu objektu proporcija datu apmācības datu kopā

Klases iezīme	**Datu objektu skaits apmācības datu kopā**	**Datu objektu % proporcija apmācības datu kopā**
1(nav konstatēta sirds slimība)	95	55%
2(tika konstatē sirds slimība)	76	45%

Datu objektu skaits testa datu kopā:

72

Datu objektu % proporcija testa datu kopā:

3.3. tabula

Datu objektu proporcija datu testa datu kopā

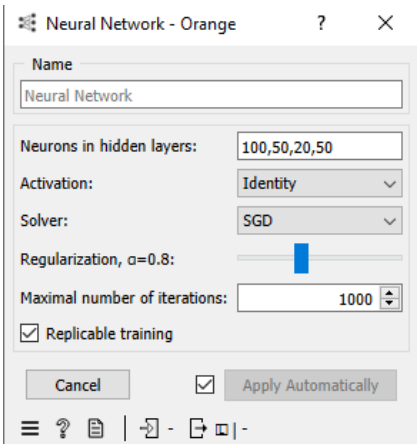
Klases iezīme	Datu objektu skaits testa datu kopā	Datu objektu % proporcija testa datu kopā
1	43	66%
2	28	38%

Eksperimenti ar mākslīgo neironu tīklu

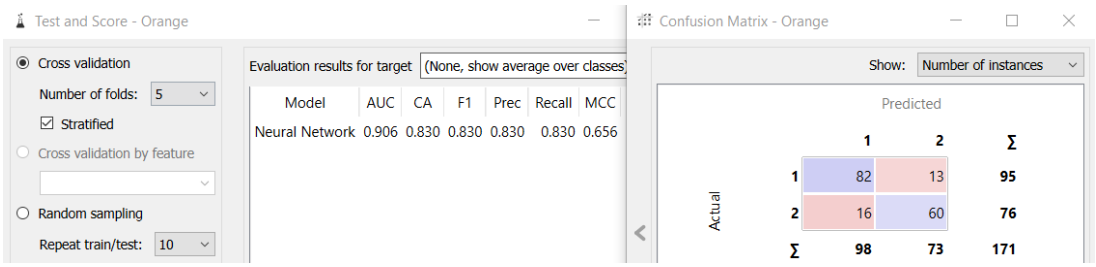
3.4. tabula

Eksperimentos lietotie hiperparametri

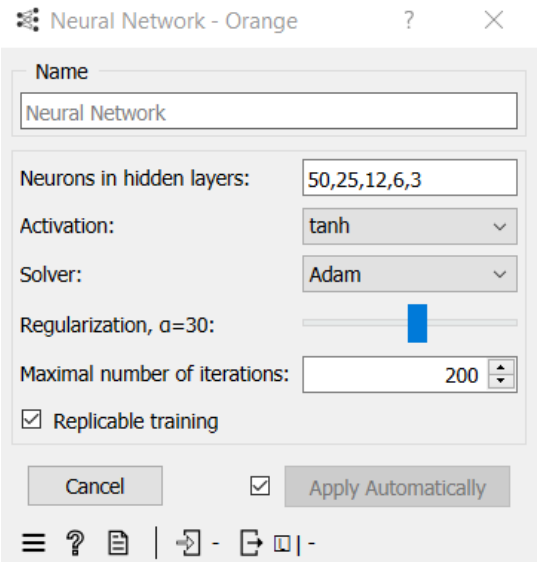
Eksperiments	Hiperparametru vērtības
1.eksperiments	Neurons: (100,50,20,50), Activation: Identity, Solver: SGD, Regularization (0.8), Max iterations: 1000
2.eksperiments	Neurons: (50,25,12,6,3), Activation: tanh, Solver: Adam, Regularization (30), Max iterations: 200
3.eksperiments	Neurons: (100,100,100, 80), Activation: Logistic, Solver: L-BFGS-B, Regularization (3), Max iterations: 100



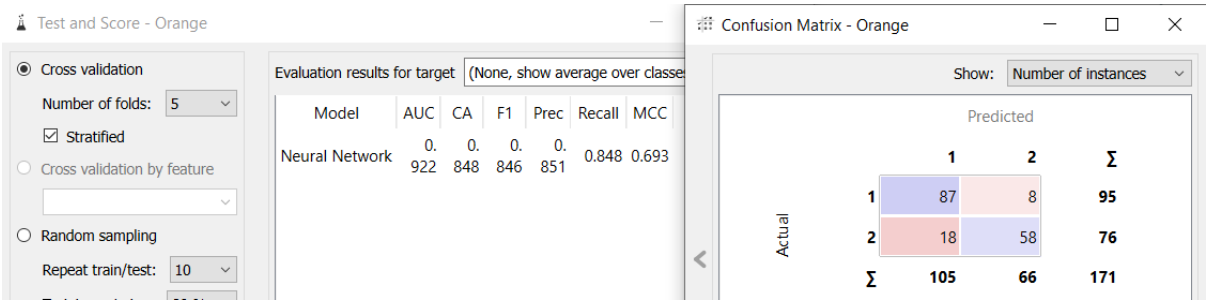
3.2. attēls. Hiperparametri 1.eksperimentam



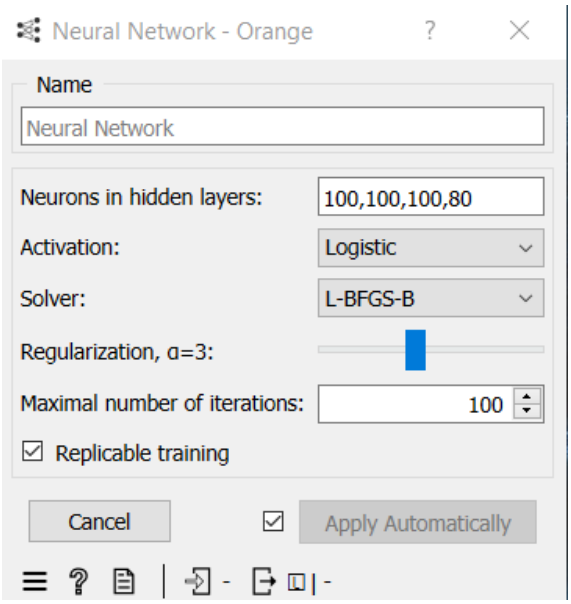
3.3. attēls. Veiktspējas metrikas 1. Eksperimentam



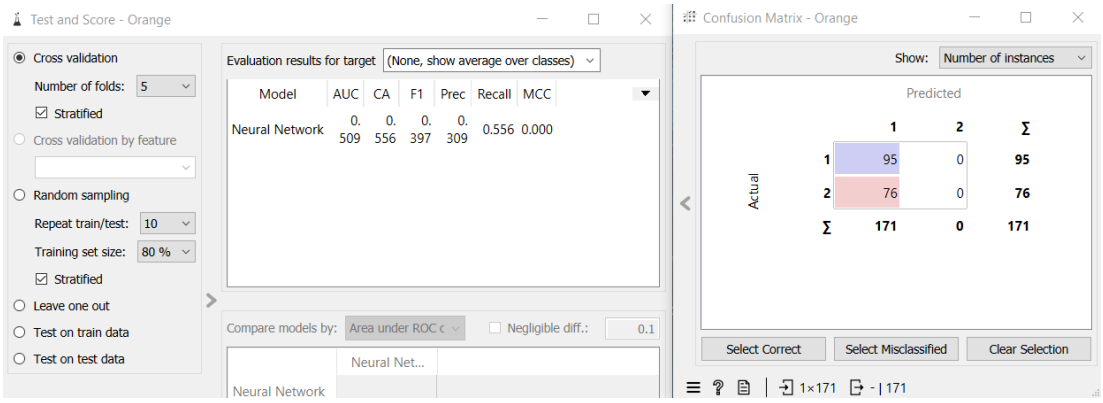
3.4. attēls. Hiperparametri 2. eksperimentam



3.5. attēls. Veiktspējas metrikas 2. eksperimentam



3.6. attēls. Hiperparametri 3. Eksperimentam



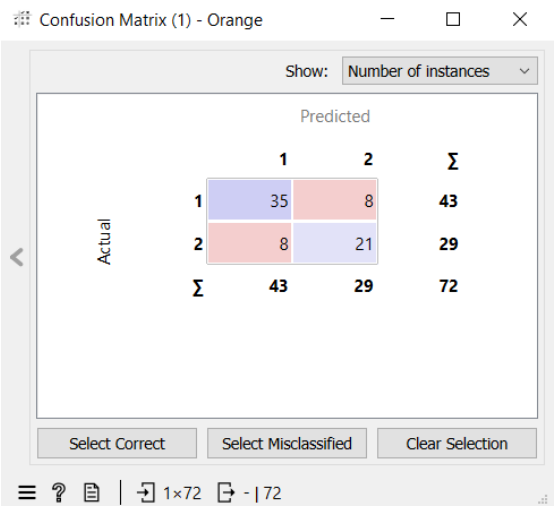
3.7. attēls. Veiktspējas metrikas 3. eksperimentam

Secinājumi no eksperimentiem:

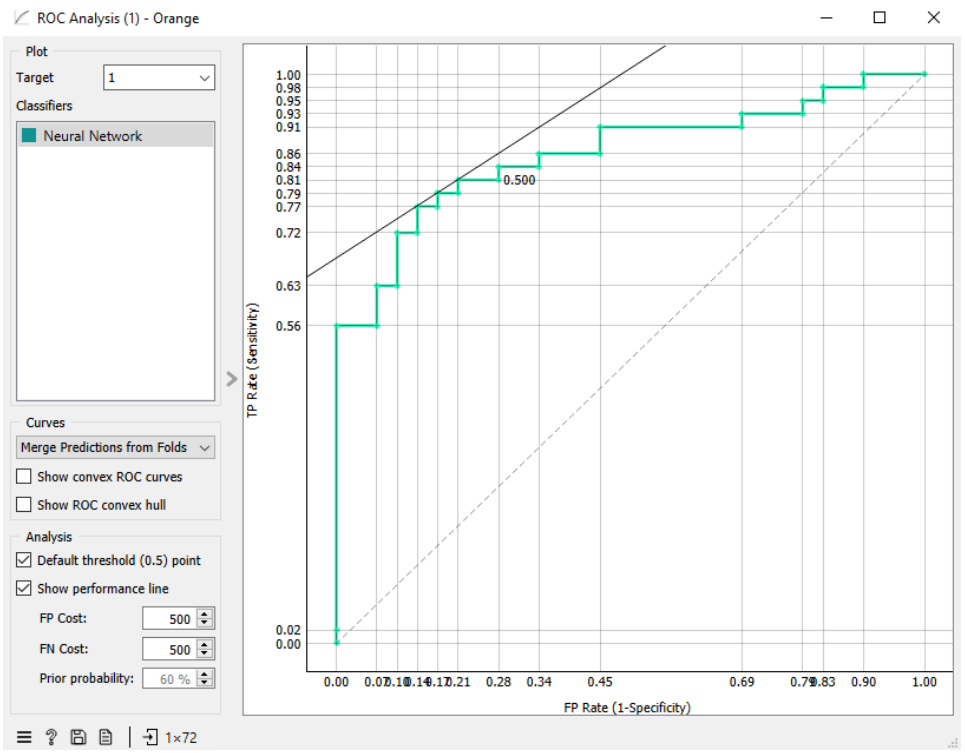
No eksperimentiem, var noteikt, ka lielākajā gadījumā gadījumu, neatkarībā no tā, kā tiek mainīti parametri, tiks sasniegts rezultāts, kur precizitāte ir nedaudz virs 80% (skat. 3.3. un 3.5. att.). Tam iemesls varētu būt mazais nepārtrauktu vērtību apjoms mūsu datu kopā (skat 1.1. att.). Vienīgie izņēmumi, ir gadījumi, kad hiperparametri ir kļūdaini izvēlēti un noved pie situācijas kurā modelis uzskata, ka visi datu objekti pieder klasei 1. (skat. 3.7. att.).

Testēšanai izvēlētais modelis:

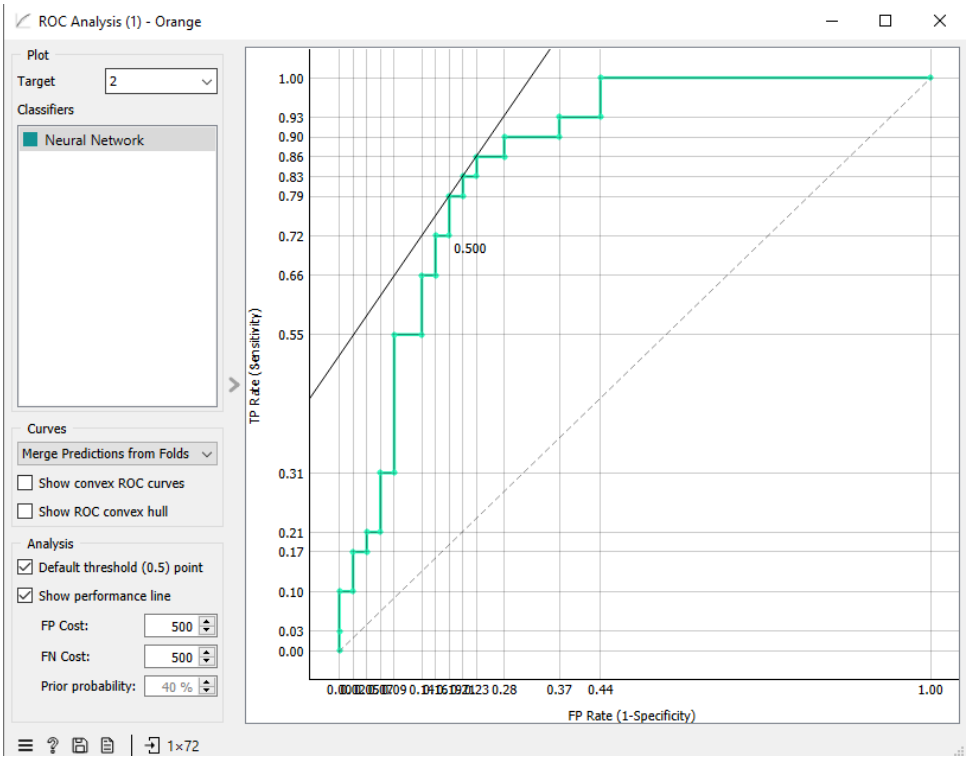
Testēšanai tiks izmantots 2. eksperimenta modelis, jo tas uzrāda vislabākos rezultātus balstoties uz metrikām (skat 3.5. att.).



3.8. attēls. *Confusion matrix* vērtības ar testa kopu



3.9. attēls. *ROC* analīze 1. Klasei



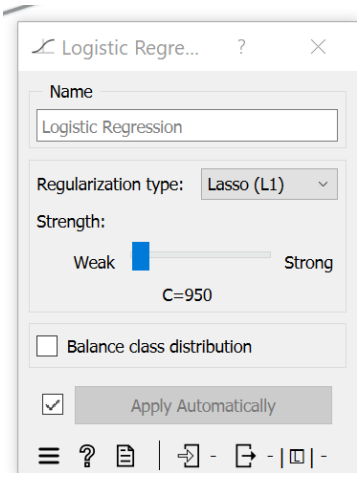
3.10. attēls. *ROC* analīze 2. Klasei

Eksperimenti ar Loģistisko regresiju

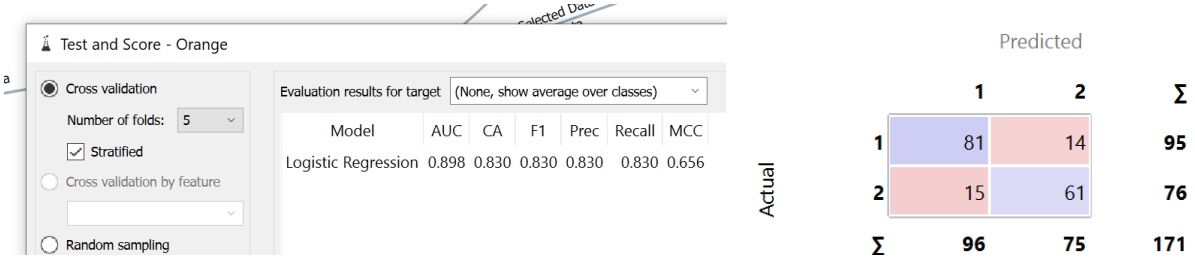
3.5. tabula

Eksperimentos lietotie hiperparametri

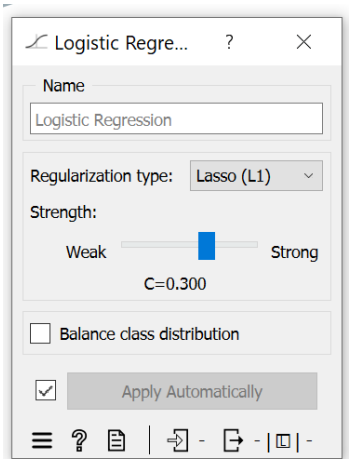
Eksperiments	Hiperparametru vērtības
1.eksperiments	Regulation type: Laso L1, C=950
2.eksperiments	Regulation type: Laso L1, C= 0.300
3.eksperiments	Regulation type: Laso L1 L2,C=0.060
4.eksperminets	Regulation type: Ridge L2, C= 950
5.eksperiments	Regulation type: Rdge L2, C= 0.300
6.ekperiments	Regulation type: Rdge L2, C= 0.300
7.eksperminets	Regulation type: None,



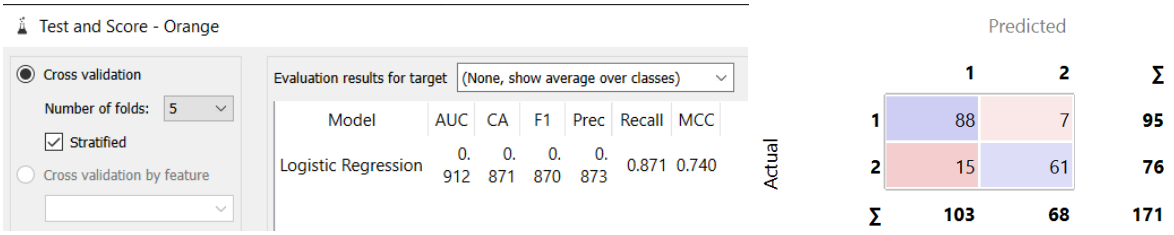
3.11. attēls. Hiperparametri 1. Eksperimentam



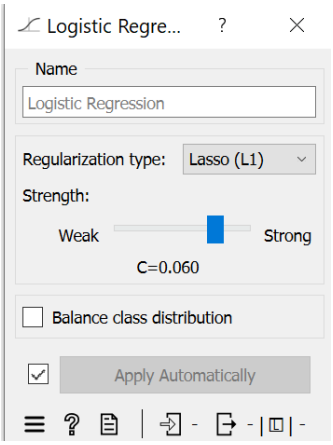
3.12. attēls. Veiktspējas metrikas 1. Eksperimentam



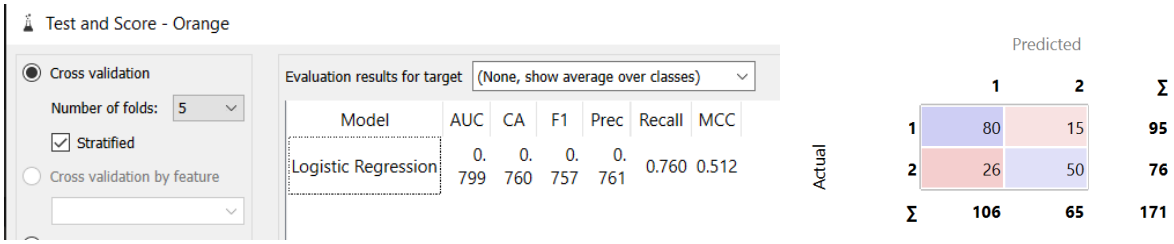
3.13. attēls. Hiperparametri 2. Eksperimentam



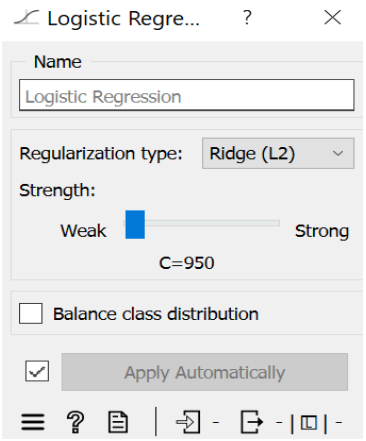
3.14. attēls. Veiktspējas metrikas 2. eksperimentam



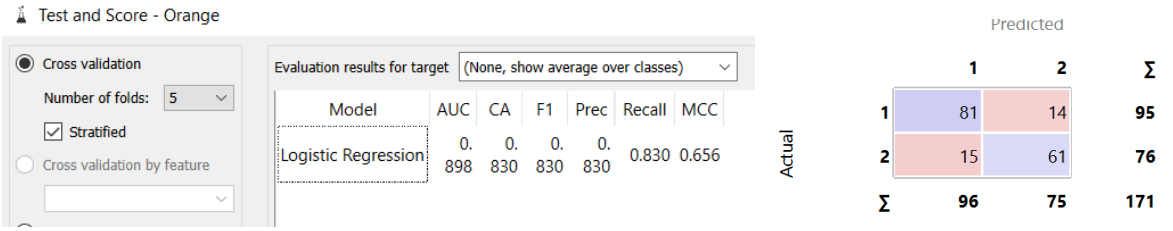
3.15. attēls. Hiperparametri 3. Eksperimentam



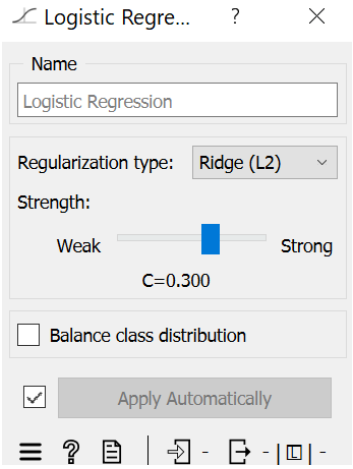
3.16. attēls. Veiktspējas metrikas 3. eksperimentam



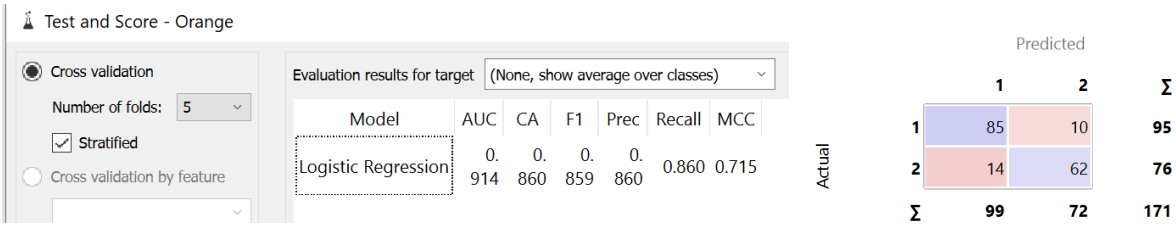
3.17. attēls. Hiperparametri 4. Eksperimentam



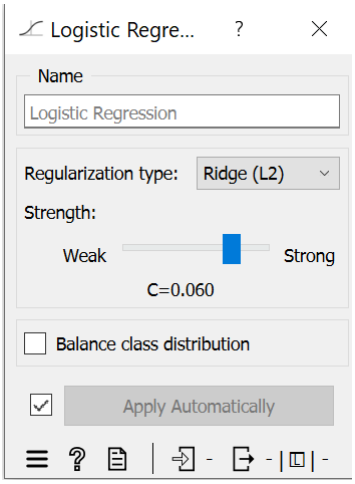
3.18. attēls. Veiktspējas metrikas 4. eksperimentam



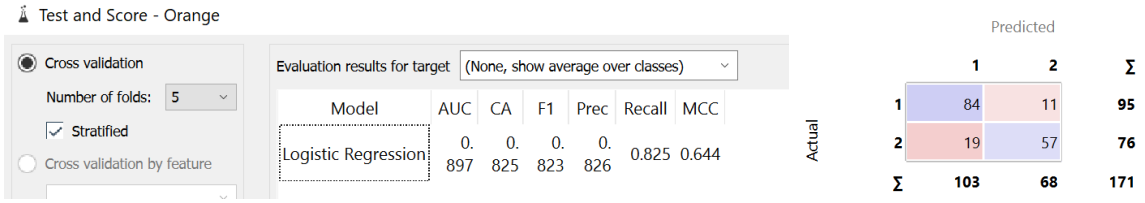
3.19. attēls. Hiperparametri 5. Eksperimentam



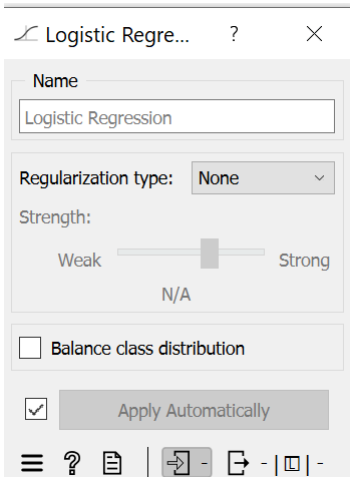
3.20. attēls. Veiktspējas metrikas 5. eksperimentam



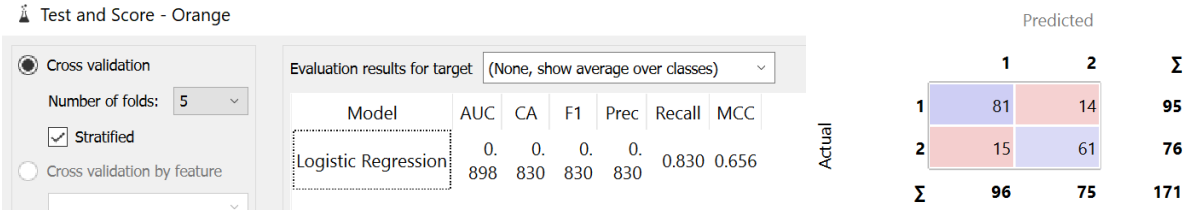
3.21. attēls. Hiperparametri 6. Eksperimentam



3.22. attēls. Veiktspējas metrikas 6. eksperimentam



3.23. attēls. Hiperparametri 7. Eksperimentam



3.24. attēls. Veiktspējas metrikas 7. eksperimentam

Secinājumi no eksperimentiem:

Mainot C hiperparametru varēja secināt, ka visprecīzākais rezultāts bija , kad C= 0.300 abiem regularizācijas tipiem. Rezultāts bija neprecīzāks, kad šo vērtību palielināja vai samazināja. Vislabāko rezultātu bija iespējams iegūt izmantojot L1 tipa regularizāciju.

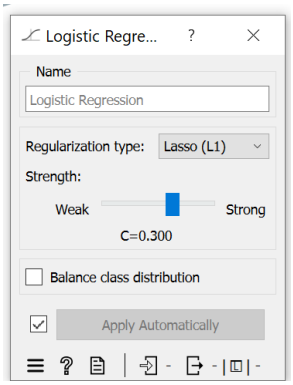
Neizmantojot nevienu regularizāciju(None), rezultāti, salīdzinot ar pārējiem eksperimentiem, bija vāji.

Kā zināms L1 regularizācija veic svaru saspiešanu un var samazināt nevajadzīgos vai mazsvarīgos koeficientus līdz nullei. Ja datu kopā ir daudz mainīgo, no kuriem tikai daži ir būtiskāki un citi mazāk svarīgi vai pat nevajadzīgi, L1 regularizācija veiksmīgi izvēlēsies tikai būtiskākos mainīgos, ignorējot mazsvarīgos. [4]

Visprecīzākie bija L1 rezultāti, tātad tas var iespējams noradīt uz to, ka mūsu datu kopā bija daži nebūtiski mainīgie, kuru koeficienti tika smazināti, tādejādi palielinot precizitāti.

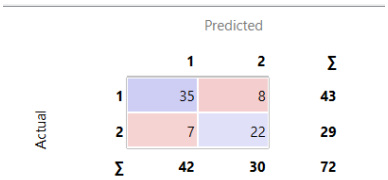
Testēšanai izvēlētais modelis:

Testēšanai tiks izvēlēts šāds modelis: C = 0.300 un regularizācijas tips :L1.

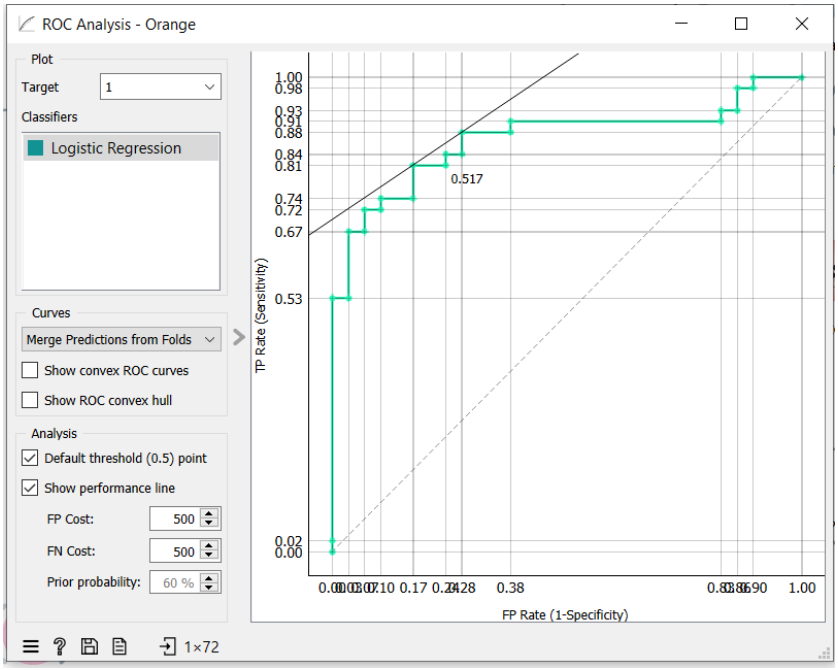


3.25. attēls. Apmācību modelis, kas tiks lietots testa datu kopai

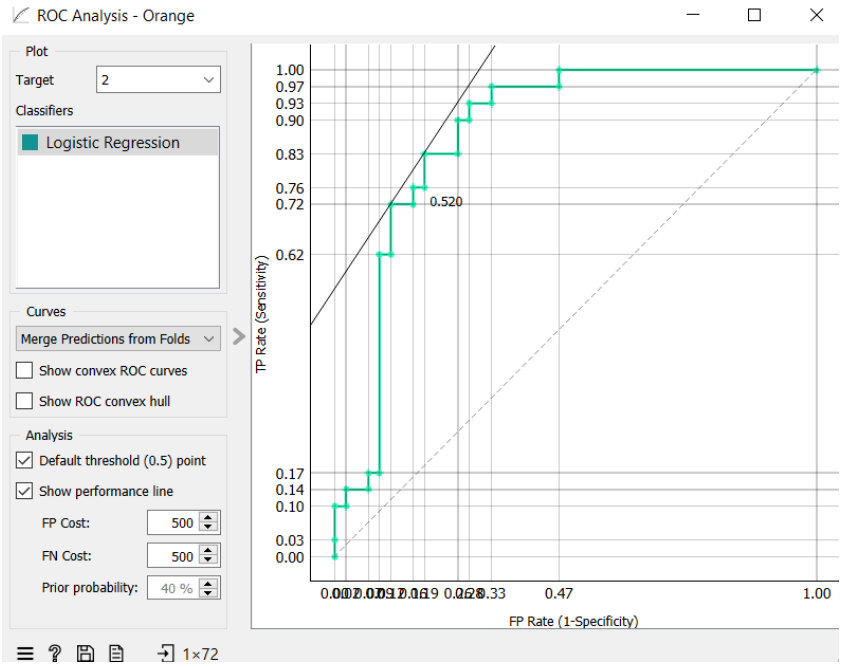
Testēšanas datukopas rezultāti:



3.26. attēls. Testa datukopas rezultāti ,*Confusion Matrix*



3.27. attēls. Testa datukopas rezultāti, *ROC Analysis,* klase : 1



3.28. attēls. . Testa datukopas rezultāti, *ROC Analysis,* klase : 2

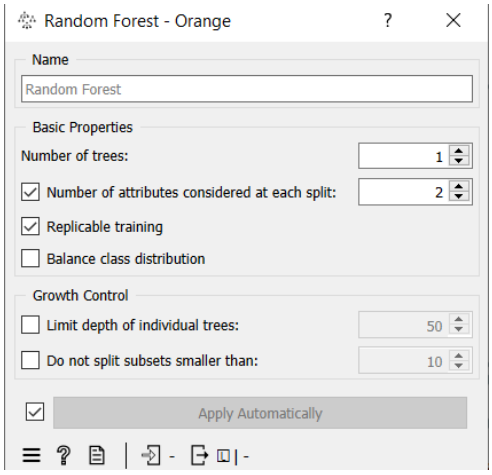
Eksperimenti ar Random Forest algoritmu

"Limit the depth of individual trees" opcija netiks izvēlēta eksperimentos, jo tā ir paradzēta galvenokārt koku ģenerēšanas paātrināšanai, kas nav šo eksperimentu mērķis. Tāpat "Replicable training" opcija būs vienmēr iespējota, lai būtu iespējams salalīdzināt ekperimentu rezultātus.

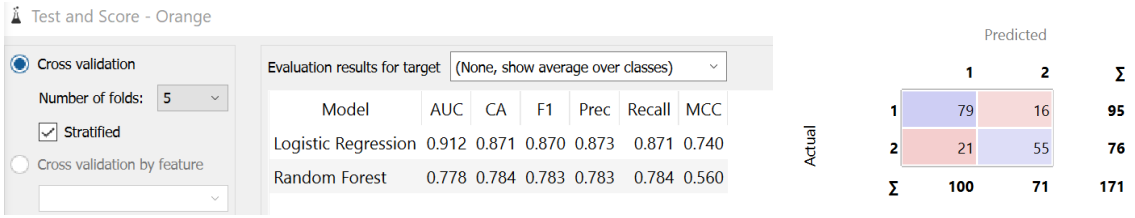
3.6. tabula

Eksperimentos lietotie hiperparametri

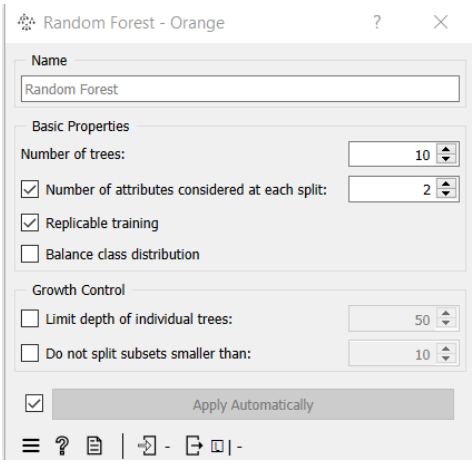
Eksperiments	Hiperparametru vērtības
1.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 2. Number of trees: 1.
2.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 2. Number of trees: 10.
3.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 2. Number of trees: 100.
4.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 2. Number of trees: 625.
5.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 4. Number of trees: 625.
6.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 6. Number of trees: 625.
7.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 6. Number of trees: 625. Do not split smaller than: 2
8.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 6. Number of trees: 625. Do not split smaller than: 6
9.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 6. Number of trees: 625. Do not split smaller than: 20
10.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 6. Number of trees: 625. Do not split smaller than: 35
11.eksperiments	Number of attributes considered at each split: iespējots, vērtība: 6. Number of trees: 625. Do not split smaller than: 100



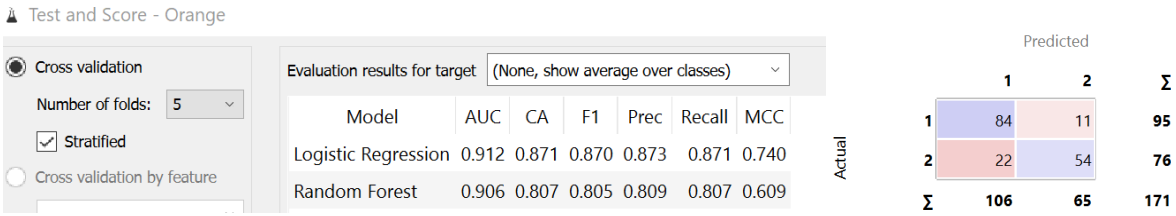
3.29. attēls. Hiperparametri 1. eksperimentam



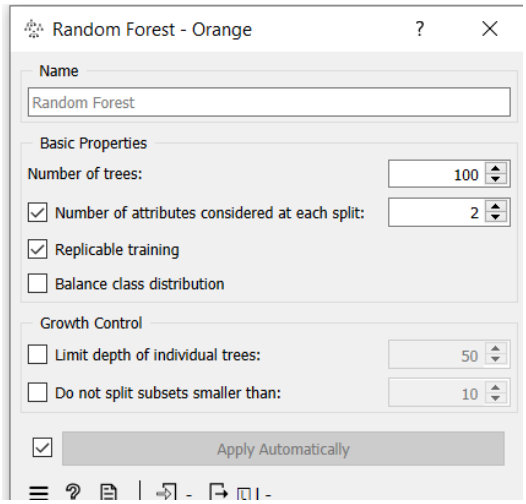
3.30. attēls. Veiktspējas metrikas 1. Eksperimentam



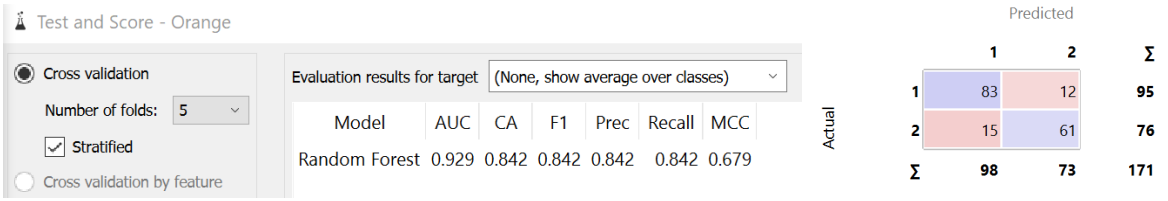
3.31. attēls. Hiperparametri 2. eksperimentam



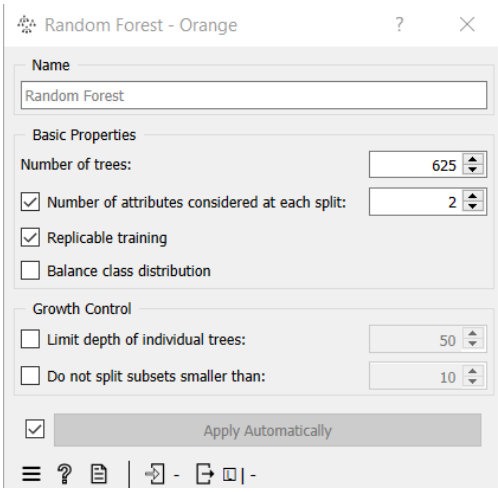
3.32. attēls. Veiktspējas metrikas 2. eksperimentam



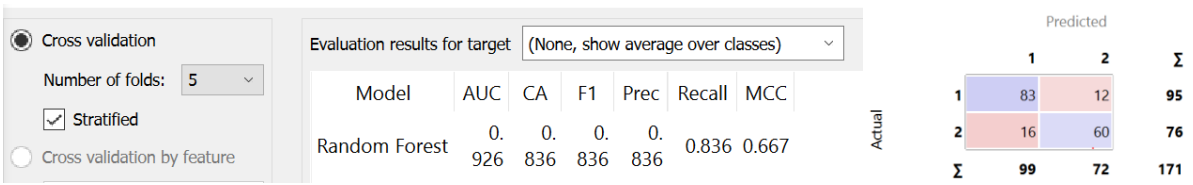
3.33. attēls. Hiperparametri 3. Eksperimentam



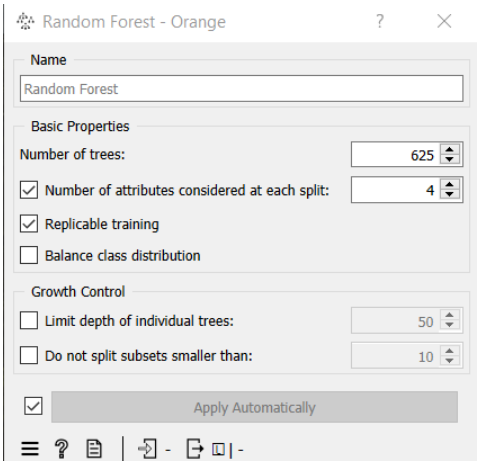
3.34. attēls. Veiktspējas metrikas 3. eksperimentam



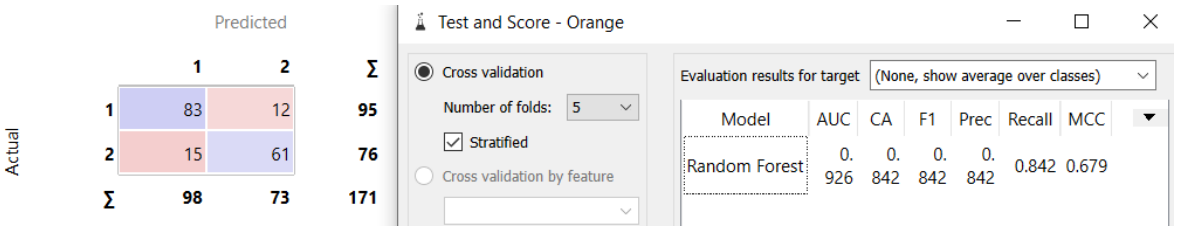
3.35. attēls.Hiperparametri 4. eksperimentam



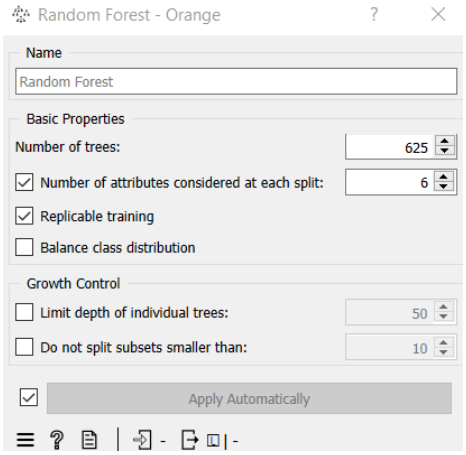
3.36. attēls. Veiktspējas metrikas 4. eksperimentam



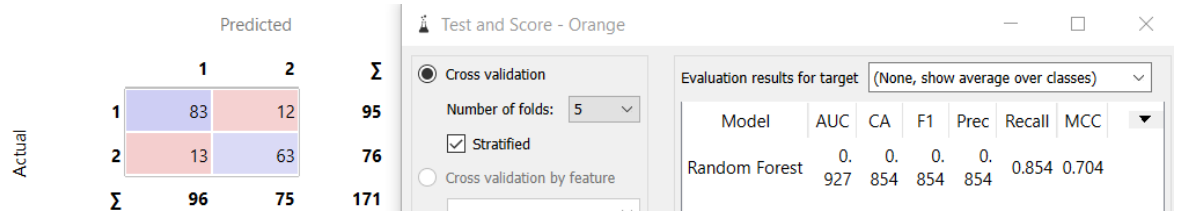
3.37. attēls. Hiperparametri 5. eksperimentam



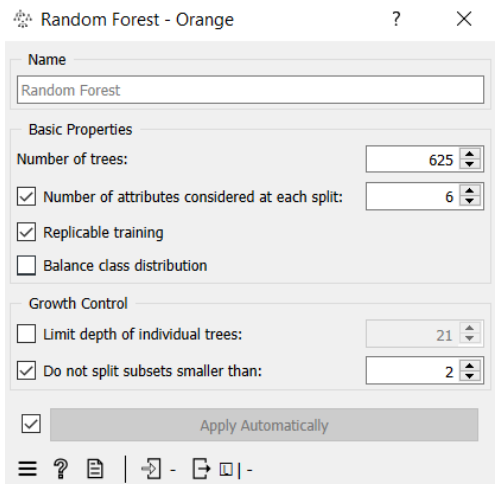
3.38. attēls. Veiktspējas metrikas 5. eksperimentam



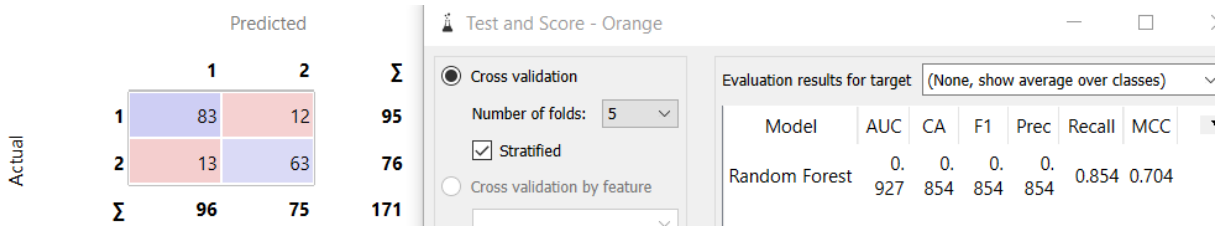
3.39. attēls. Hiperparametri 6. eksperimentam



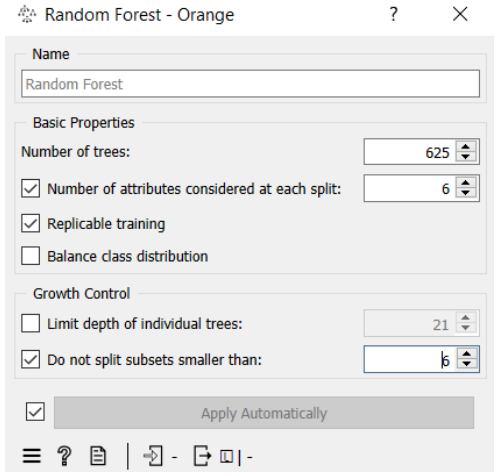
3.40. attēls. Veiktspējas metrikas 6. eksperimentam



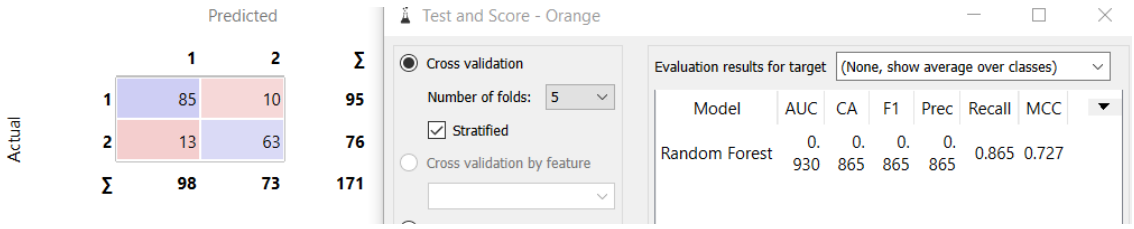
3.41. attēls. Hiperparametri 7.eksperimentam



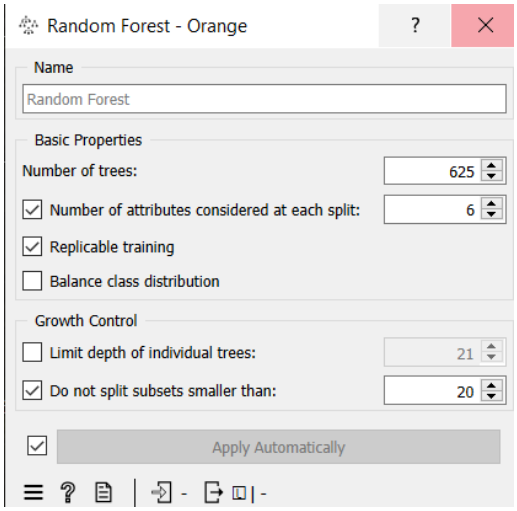
3.42. attēls. Veiktspējas metrikas 7. eksperimentam



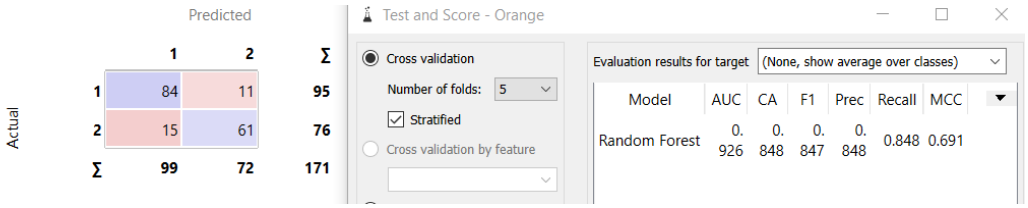
3.43. attēls. Hiperparametri 8. eksperimentam



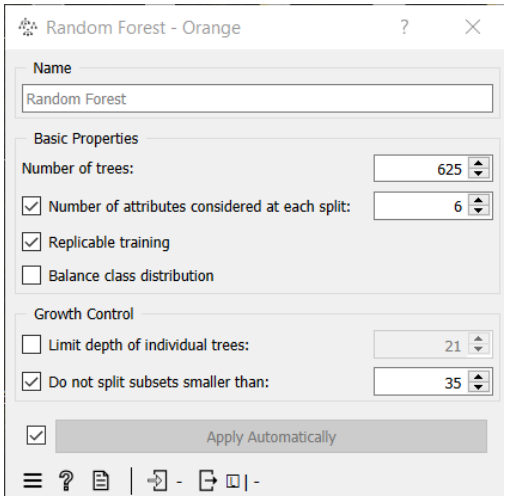
3.44. attēls. Veiktspējas metrikas 8. eksperimentam



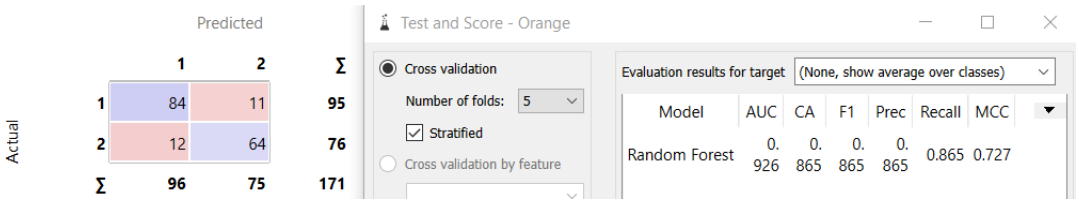
3.45. attēls. Hiperparametri 9. Eksperimentam



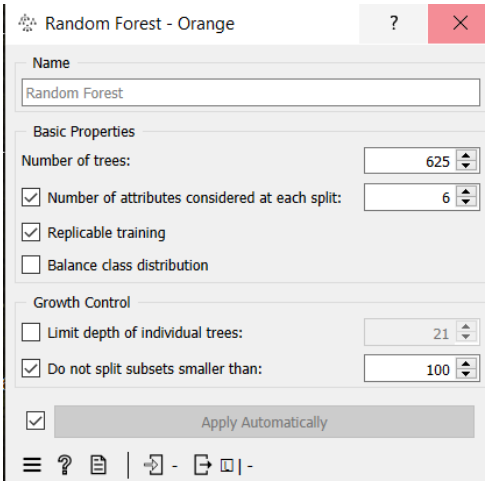
3.46. attēls. Veiktspējas metrikas 9. eksperimentam



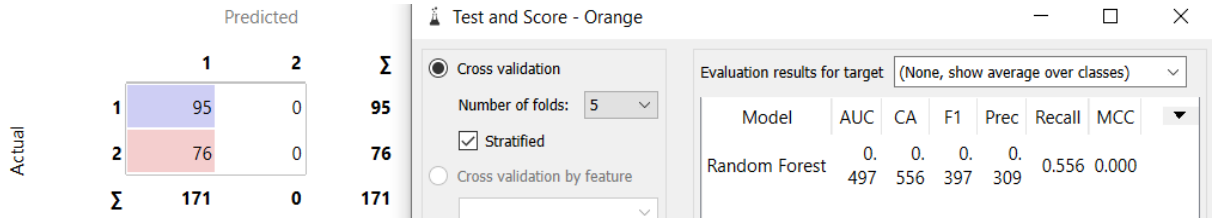
3.49. attēls. Hiperparametri 10. Eksperimentam



3.50. attēls. Veiktspējas metrikas 10. eksperimentam



3.51. attēls. Hiperparametri 11. eksperimentam



3.52. attēls. Veiktspējas metrikas 11. eksperimentam

Secinājumi no eksperimentiem:

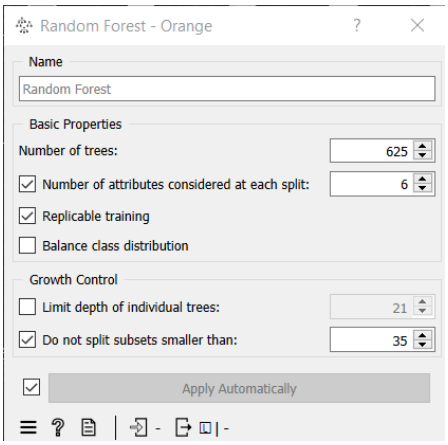
Mainot ģenerēto koku skaitu, tika novērots precizitātes pieaugums. Tas varētu būt saistīts ar to, ka mazāks ģenerēto koku skaits nozīmē arī mazāk "bootstrap" datu kopu. Šāda situācija rada koku daudzveidības trūkumu, jo ne visi ieraksti un atribūti tiek izmantoti vienlīdz bieži. Tādēļ, pārbaudot šos kokus ar ierakstiem no "out-of-bag" datu kopas, kas pieder specifiskai klasei citu atribūtu dēļ(kurus koki neizmanto pietiekami bieži), rodas nepreciztātes.

Palielinot izvēlēto atribūtu skaitu, lai ģenerētu katru koku rezultāti bija līdzīgi, proti precizitāte palielinājās. Var apsvērt, ka tas ir saistīts ar to, ka palielinot atrībūtu skaitu algoritmam apskatīja vairāk no visām iespējamām atribūtu kombinācijām(jo tās bija mazāk), kas radīja koku kopu, kura ir reprezentablāka datukopai.

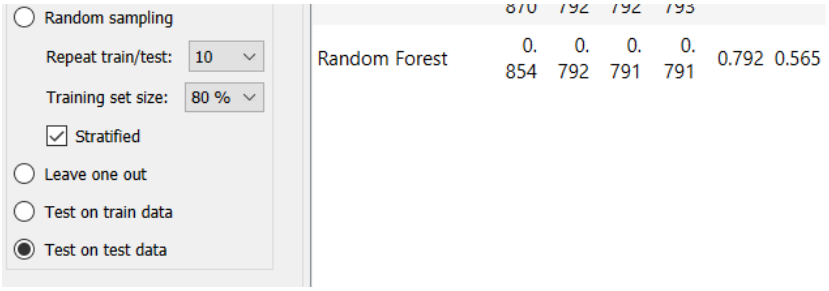
Palielinot vērtību, kas nosaka to cik lielu apakškopu ir iespējams sadalīt sīkāk, precizitāte pieauga, tomēr tikai līdz bīrīdim, kad vērtība nebija lielāka par 35. Tikko tā pārsniedza šo lielumu, precizitāte samazinājas. Tas ir saistīts ar to, ka lai gan šī vērtība palīdz ierobežot pārlieku pielāgošanos specifiskajam datusetam, ja tā ir pārāk liela tā var izraisīt to, ka apmācībās modelis nespēj atrast pietiekami daudz sakarības datasetā padarot to neprecīzu .[19]

Testēšanai izvēlētais modelis:

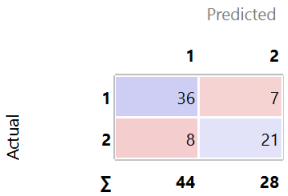
Balstoties uz rezultātiem un secinājumiem par testēšanas modeli tiks izvēlēts 10 eksperimenta hiperparametri, kuriem bija visprecīzākie rezultāti:



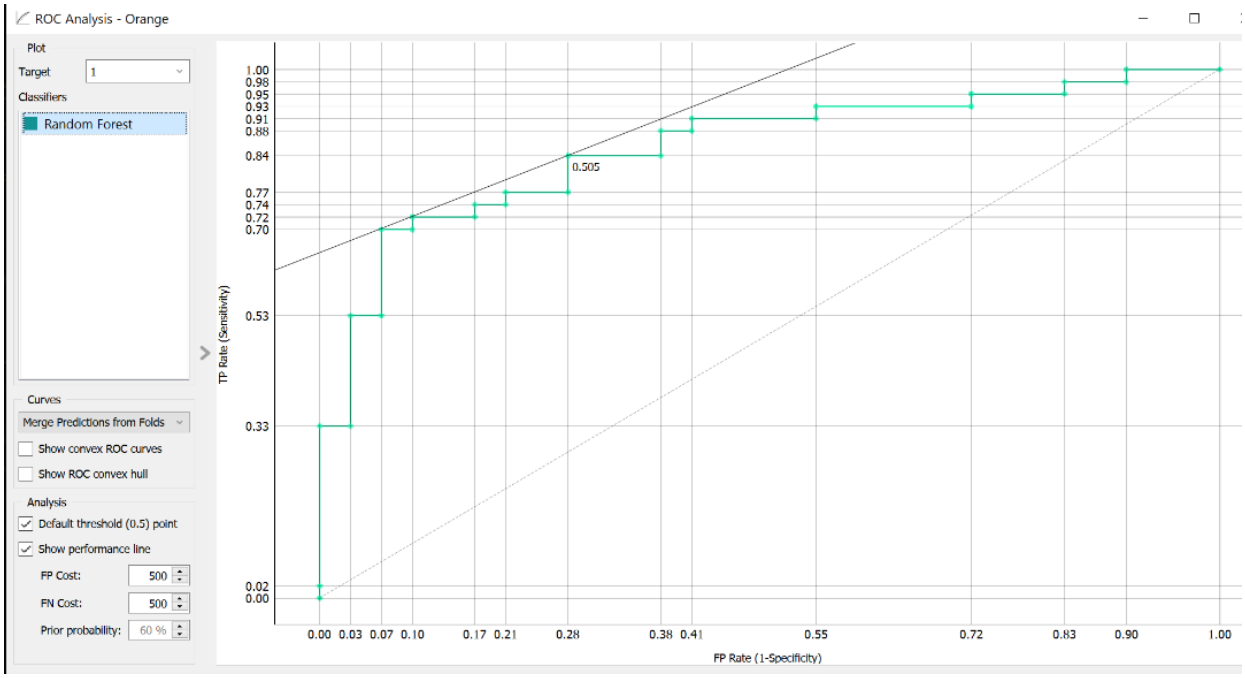
3.53. attēls. Apmacības modelis testa datukopai



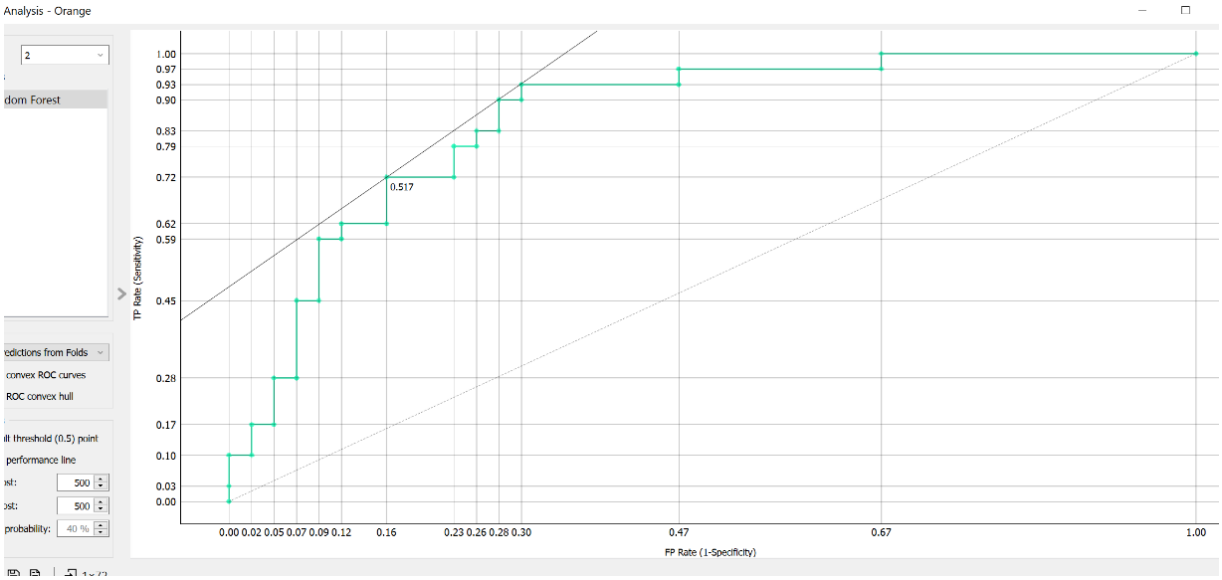
3.54. attēls. Veiktspējas metrikas testa datukopai, *Test and Score* logarīkā



3.55. attēls. Veiktspējas metrikas testa datukopai, *Confusion Matrix*

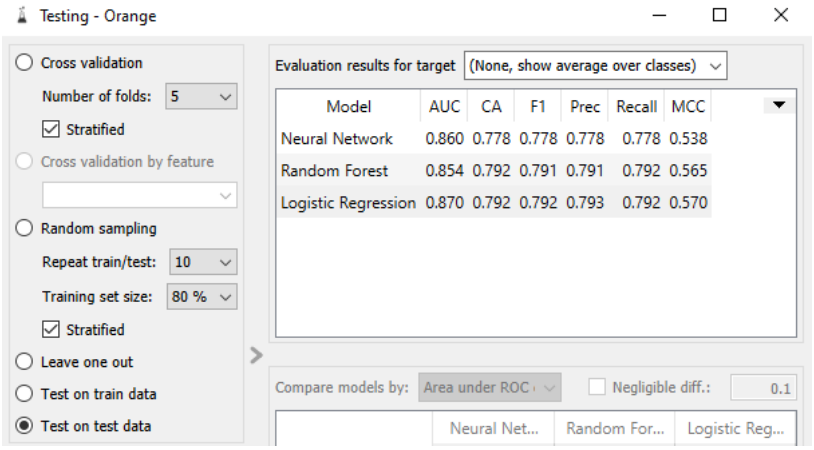


3.56. attēls. Veiktspējas metrikas testa datukopai, *ROC Analysis*, klase :1

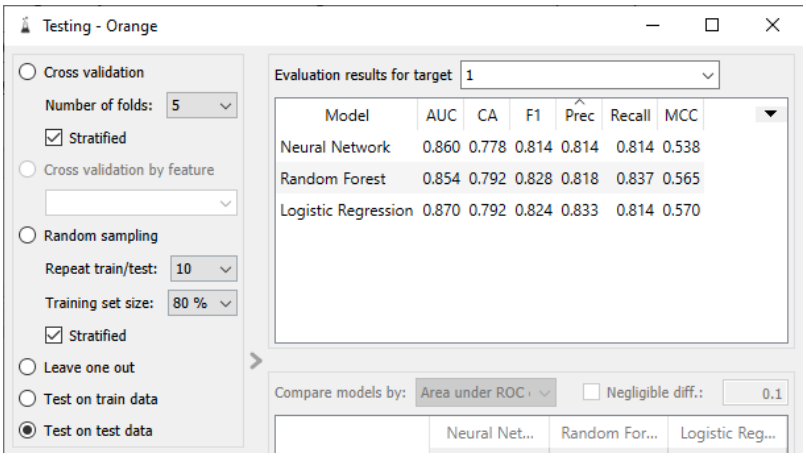


3.57. attēls. Veiktspējas metrikas testa datukopai, *ROC Analysis*, klase :2

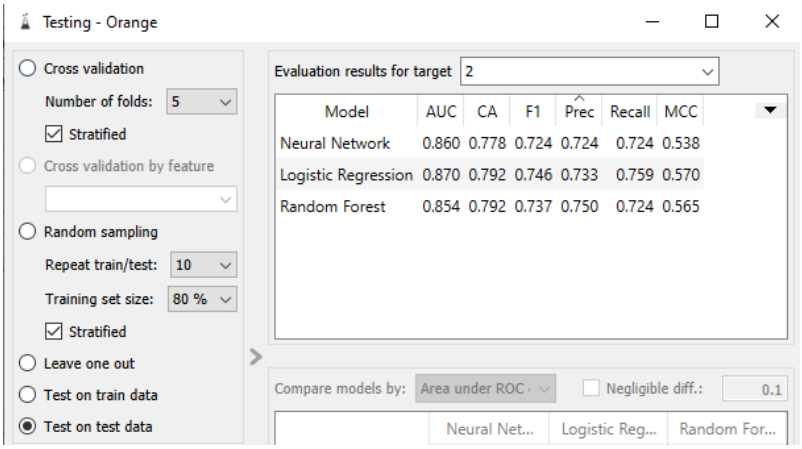
Apmācīto modeļu testēšanas rezultāti



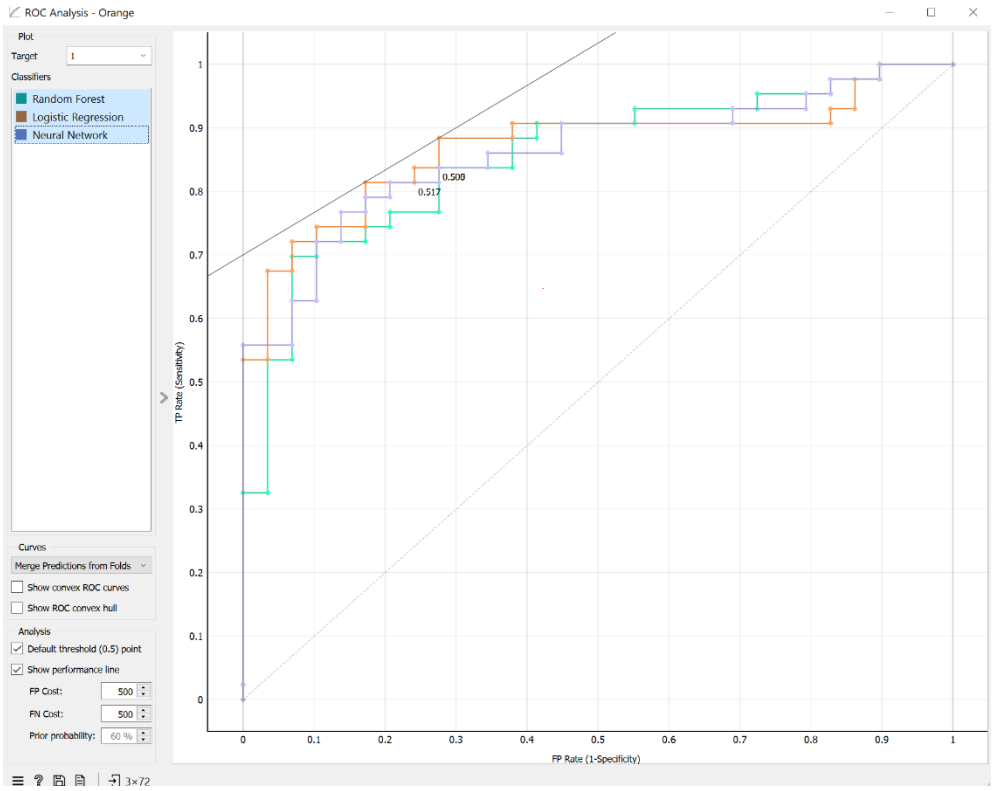
3.58. attēls. Veiktspējas metrikas visiem modeļiem(testa datukopa), visas klases



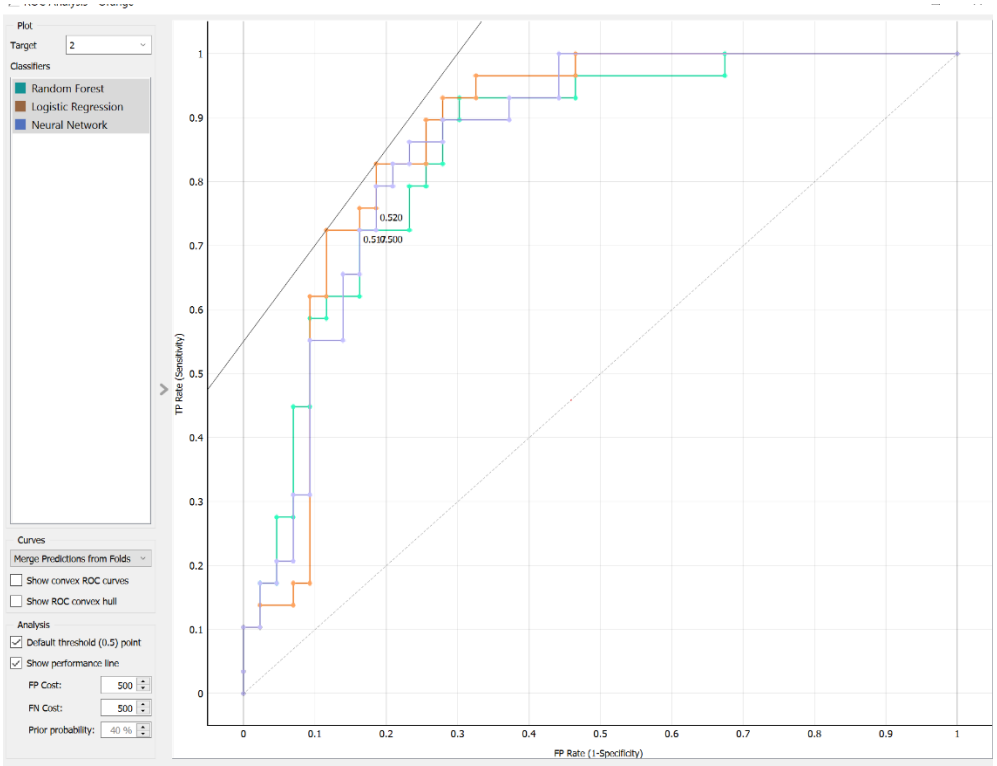
3.59. attēls. Veiktspējas metrikas visiem modeļiem(testa datukopa), klase : 1



3.60. attēls. Veiktspējas metrikas visiem modeļiem(testa datukopa), klase : 2



3.61. attēls. Veiktspējas metrikas testa datukopai, *ROC Analysis*, klase :1



3.62. attēls. Veiktspējas metrikas testa datukopai, *ROC Analysis*, klase :2

Secinājumi pēc testēšanas:

3.7. tabula

Algoritmu precizitātes (Precision) salīdzinājums

Algoritms	**Apmācību(Prec.)**	**Testa(Prec.)**	**Procentuālā starpība, %**
Random Forest	0.865	0.791	0.856
Logistic Regression	0.873	0.793	0.916
Neural Network	0.851	0.778	0.858

Klasificējot testa datukopu, algoritmu veikstpējas metrikas "Test un Score" logarīkā, kā arī ROC analīzes diagrammās bija līdzīgas, tomēr visprecīzākās bija loģistiskās regresijas algoritmam, savukārt vismāzāko precizitāti uzrādija neironu tīklu algoritms. Tas varētu būt izskaidrojams ar to, ka loģistiskajai regresijai jau no paša sākuma, klasificējot apmācību datukopu bija labākie rezultāti, un neironu tīkliem bija sliktākie. Pastāv iespēja, ka neironu tīkliem netika atrasta vislabākā hiperparametru konfigurācija.

Tomēr, salīdzinot algoritmus, ir svarīgi pievērst uzmanību arī procentuālajai starpībai starp algoritmu precizitātēm, izmantojot testa un apmācību datu kopas. Jo lielāka ir šī starpība, jo vairāk algoritms mēdz pielāgoties apmācības datu kopai, un šo parādību mašīnmācīšanās kontekstā sauc par "overfitting" . Tā var padarīt algoritmu mazāk precīzu, lietojot to uz citām datu kopām, mūsu gadījumā testa datukopu.

Salīdzinot šīs procentuālās starpības neironu tīkliem un "random forest" algoritmiem, tās bija gandrīz vienādas, proti 0.858 % un 0.856 % attiecīgi. Tomēr loģistiskajai regresijai tā bija vislielākā - 0.916%. Tas norāda uz to, ka loģistiskā regresijai ir vislielākā nosliece uz pārlieku pielāgošanos jeb "overfitting" apmācību datukopai.

Papildus apstiprinājumam varam redzēt, ka skatoties veiktspējas metrikas AUC jeb laukuma zem ROC līknes vērtību varam noteikt, ka algoritmi spēj datu objektus sadalīt pa klasēm apmēram vienādi, taču loģistiskā regresija to dara nedaudz precīzāk (Skatīt 3.58, 3.59, 3.60 attēlus).

Balsoties uz analīzi, ieteicamais algoritms šīs datu kopas apstrādei ir loģistiskā regresija vai “random forest”. Ņemot vērā, ka neironu tīklu veidošanai ir nepieciešami vairāk resursu nekā abiem pārējiem algoritmiem un tā salīdzinoši zemās metrikas testēšanā, to būtu ieteicams nelietot apstrādājot šo datu kopu vai arī veikt tā hiperparametru tālāku optimizāciju.

Izmantotie informācijas avoti

1. *Logistic Regression*. Orange3. Pieejams: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/logisticregression.html

2. *What is logistic regression*. IBM. Pieejams: https://www.ibm.com/topics/logistic-regression

3. Muratgulcan. *Hierarchical Clustering with Python: Basic Concepts and Application*. Medium, Jūlijs 25, 2023. Pieejams: https://medium.com/@muratgulcan/hierarchical-clustering-with-python-basic-concepts-and-application-cd5f5dc95b1f

4. Vincent, F. *Regularization in Logistic Regression*. Medium, Maijs 30, 2023. Pieejams: https://medium.com/@vincefav/regularization-in-logistic-regression-14b50d7cc31

5. *k-Means*. Orange3. Pieejams: k-Means — Orange Visual Programming 3 documentation (orange3.readthedocs.io)

6. Orange Data Mining. *Explaining k-Means Clusters*. Youtube, Augusts 21, 2023. Pieejams: https://www.youtube.com/watch?v=gJX2jR0-PTY

7. Orange Data Mining. *Initial Centroids for k-Means*. Youtube, Augusts 21, 2023. Pieejams: https://www.youtube.com/watch?v=uCfKLEaNYIc

8. Orange Data Mining. *Explaining k-Means Clusters*. Youtube, Augusts 21, 2023. Pieejams: How to choose k for k-Means? (youtube.com)

9. Orange Data Mining. *k-Means Clustering*. Youtube, Augusts 21, 2023. Pieejams: https://www.youtube.com/watch?v=okqObNqpP2o

10. Orange Data Mining. Neural Network. Pieejams: https://orangedatamining.com/widget-catalog/model/neuralnetwork/

11. Activation Functions in Neural Networks. TowardDataScience. Pieejams: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

12. What is Adam Optimizer? GeeksForGeeks. Pieejams: https://www.geeksforgeeks.org/adam-optimizer/

13. Numerical optimization based on the L-BFGS method. TowardsDataScience. Pieejams: https://towardsdatascience.com/numerical-optimization-based-on-the-l-bfgs-method-f6582135b0ca

14. Coding Neural Network – Regularization. TowardsDataScience. Pieejams: https://towardsdatascience.com/coding-neural-network-regularization-43d26655982d

15. Heart Disease. Kaggle Datasets. Pieejams: https://www.kaggle.com/datasets/utkarshx27/heart-disease-diagnosis-dataset/data

16. Heart Disease. UCI. Pieejams: https://archive.ics.uci.edu/dataset/145/statlog+heart

17. Random Forests, Leo Breiman and Adele Cutlerm, Introduction. Pieejams:

https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#intro

18. Breiman, L. (2001). Random forests. Machine Learning Pieejams: https://link.springer.com/article/10.1023/a:1010933404324

19. Orange data mining: Random Forest. Pieejams: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/randomforest.html

20. LaMorte, W. W. (n.d.). Logistic Regression. Boston University School of Public Health. Pieejams: https://sphweb.bumc.bu.edu/otlt/MPH-Modules/PH717-QuantCore/PH717-Module12-MultipleRegression/PH717-Module12-MultipleRegression7.html