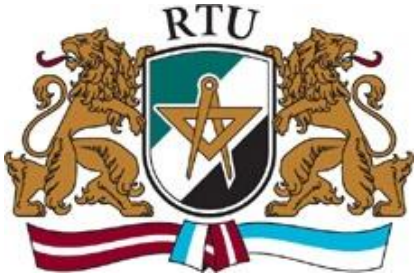
RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte



ATSKAITE PAR 2. PRAKTISKO DARBU

studiju kursā „Mākslīgā intelekta pamati”

Izstrādāja: Artjoms Bogatirjovs 171RDB112

Pārbaudīja: asoc.prof. A.Anohina-Naumeca

2022./2023.māc.g.

SATURS

Ievads

Šajā darbā es izvēlējos viena no populārākajām datu kopu no https://archive.ics.uci.edu/ml/index.php un izmantoju to datu apstrādei, izmantojot gan pārraudzītās, gan nepārraudzītās mašīnmācīšanās algoritmus. Mērķis bija attīstīt savas prasmes darbā ar mašīnmācīšanās algoritmiem un analizēt iegūtos rezultātus. Galarezultāts ir manas sagatavotās atskaite par darba izpildi.

Darba izstrādei tika izmantots Orange rīka. Man bija jāveic dažādi uzdevumi, piemēram, datu ielāde, datu apstrāde, vizualizācija ar grafikiem, statistikas aprēķini, testēšana un rezultātu analīze. Lai sasniegtu šos mērķus, es biju spiests meklēt papildu informācijas avotus, lai atrisinātu darba jautājumus vai sniegtu interpretāciju un analīzi par iegūtajiem rezultātiem.

Datu pirmapstrāde/izpēte

Datu kopas apraksts

Šajā praktiska darba tika izmantota "Raisin Dataset" datu kopu, ko izveidoja Ilkay Cinar, Murat Koklu un Sakir Tasdemir. Šī datu kopa bija publiski pieejama no 2021. gada 1. aprīļa un tika izveidota 2020. gadā ar atvērtā pirmkoda licenci (open source license).

Datu kopā tika izmantota mašīnredzes sistēma, kas bija iepriekš izstrādāta, lai atšķirtu divas dažādas Turcijā audzētas rozīnes - Kecimen un Besni. Kopumā tika izmantoti 900 rozīņu paraugi, no kuriem 450 bija katras šķirnes. Šie rozīņu paraugi tika fotografēti un veikti vairāki pirmsapstrādes soļi. Tālāk veica 7 morfoloģisko pazīmju izguves darbības, izmantojot attēlu apstrādes metodes. Katram objektam tika aprēķinātas minimālās, vidējās, maksimālās un standartnovirzes statistiskās informācijas vērtības.

Tika pētīta abu rozīņu šķirņu dažādība un īpašības, kas tika vizualizētas grafikos. Tālāk tika izveidoti modeļi, izmantojot mašīnmācības tehnikas, piemēram, loģistisko regresiju (LR), daudzslāņu perceptronu (MLP) un mašīnmācības vektoru (SVM). Tika veikti arī veiktspējas mērījumi.

Datu kopas satura apraksts

Datu kopas informācija:

Datu kopā ir attēli ar Turcijā audzētām Kecimen un Besni rozīnēm, kas tika iegūti ar CVS. Kopā tika izmantoti 900 rozīņu graudi, no kuriem 450 bija no katras šķirnes. Šie attēli tika pakļauti dažādiem pirmsapstrādes posmiem, un no tiem tika iegūtas 7 morfoloģiskās pazīmes. Šīs pazīmes tika klasificētas, izmantojot trīs dažādas mākslīgā intelekta tehnoloģijas.

Atribūtu informācija:

1.) Platība: Norāda pikseļu skaitu rozīnes robežās.

2.) Perimetrs: Mēra vides apjomu, aprēķinot attālumu starp rozīnes robežām un pikseļiem apkārt.

3.) Galvenā ass garums: Norāda galvenās ass garumu, kas ir garākā līnija, kas var tikt uzzīmēta rozīnē.

4.) Maza ass garums: Norāda mazās ass garumu, kas ir īsākā līnija, kas var tikt uzzīmēta rozīnē.

5.) Ekscentriskums: Norāda ekscentriskošanas mēru elipsei, kuras momenti ir līdzīgi rozīnei.

6.) Konveksā platība: Norāda pikseļu skaitu mazākajā konvexā apvalkā, kas veidojas no rozīnes reģiona.

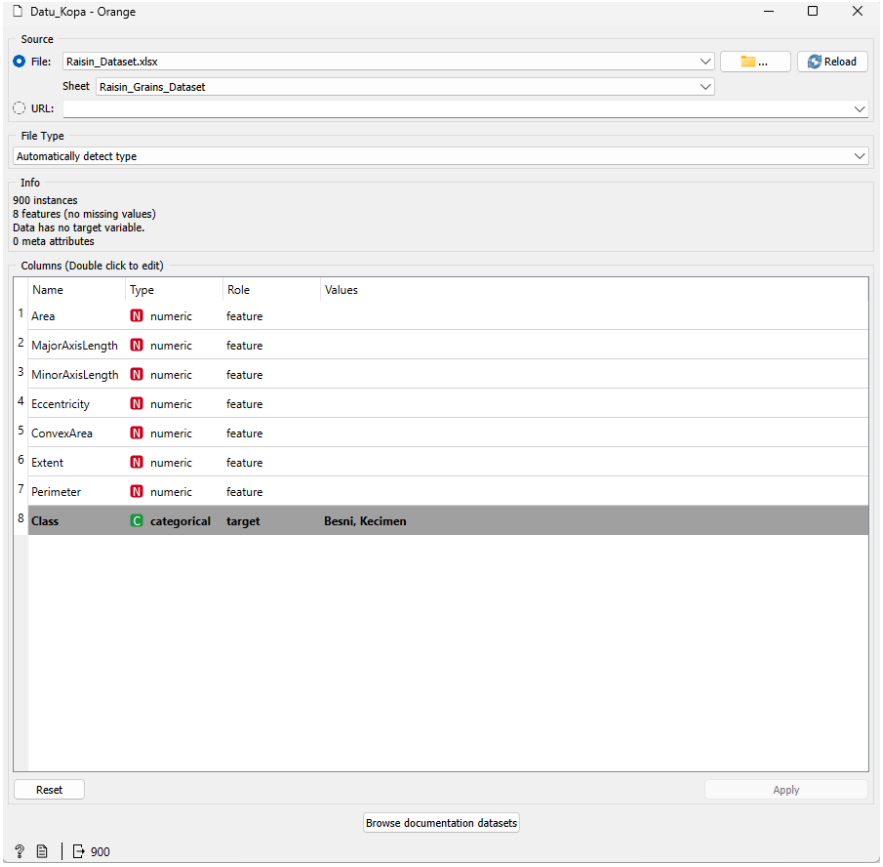
7.) Izplatība: Norāda attiecību starp rozīnes veidotā reģiona platību un kopējo pikseļu skaitu ierobežojošajā lodziņā.

8.) Klase: Kecimen un Besni rozīne.

Statistikas dati par iegūtajām īpašībām

	Area (pikseļi)	MajorAxisLenght	MinorAxisLenght	Eccentricity	ConvexArea	Extent	Perimeter
Minimālā vērtība	25387.000	225.630	143.711	0.349	26139.000	0.380	619.074
Vidējā vērtība	87804.128	430.930	254.488	0.782	91186.090	0.700	1165.907
Maksimālā vērtība	235047.000	997.292	492.275	0.962	278217.000	0.835	2697.753
Standartnovirze	39090.039	116.856	50.675	0.094	40859.720	0.058	276.355

Datu augšupielāde orange riku

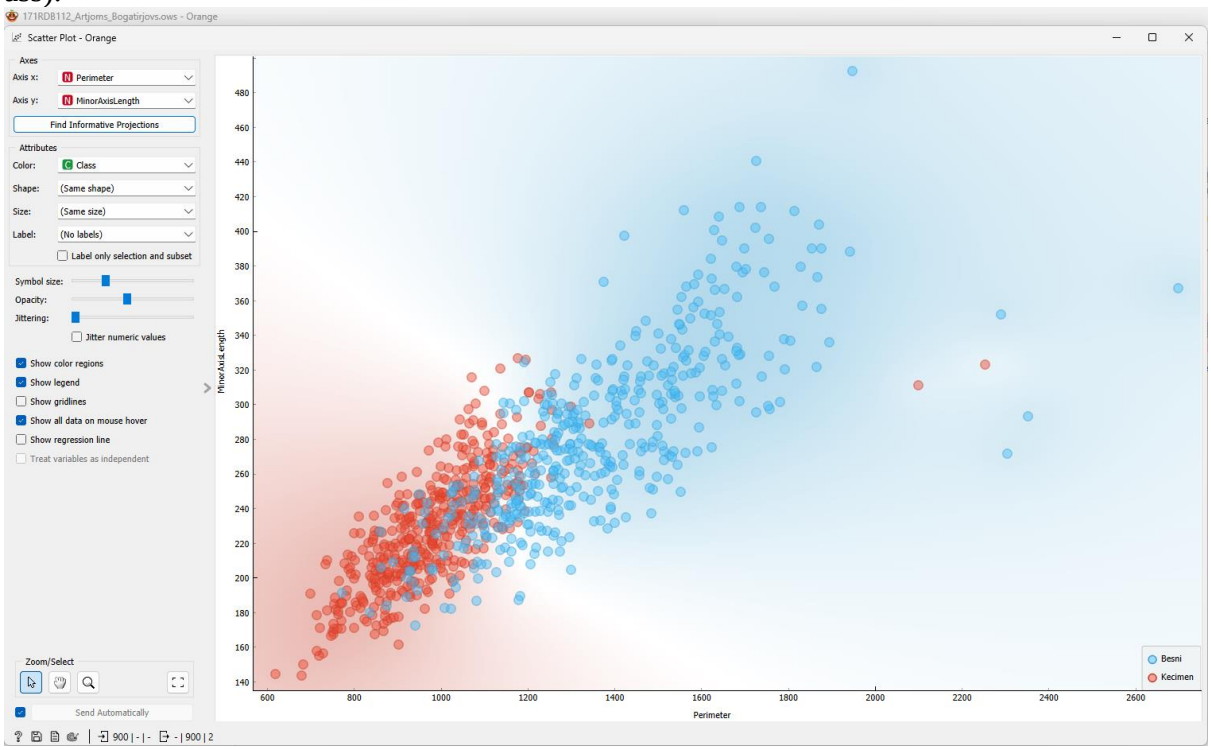


Datu faila struktūras fragments ar visām tā kolonnām un vērtībām

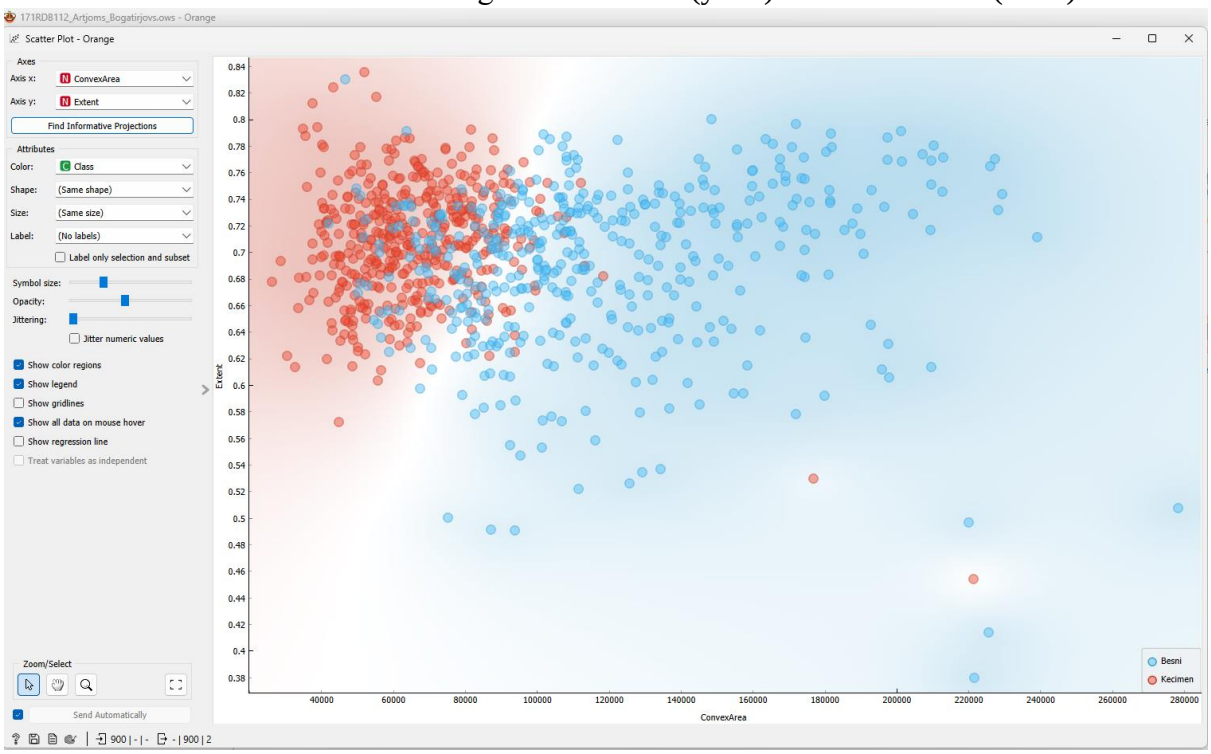
	A	B	C	D	E	F	G	H
1	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perimeter	Class
2	87524	442.2460114	253.291155	0.819738392	90546	0.758650579	1184.04	Kecimen
3	75166	406.690687	243.0324363	0.801805234	78789	0.68412957	1121.786	Kecimen
4	90856	442.2670483	266.3283177	0.798353619	93717	0.637612812	1208.575	Kecimen
5	45928	286.5405586	208.7600423	0.684989217	47336	0.699599385	844.162	Kecimen
6	79408	352.1907699	290.8275329	0.56401133	81463	0.792771926	1073.251	Kecimen
7	49242	318.125407	200.12212	0.777351277	51368	0.658456354	881.836	Kecimen
8	42492	310.1460715	176.1314494	0.823098681	43904	0.665893562	823.796	Kecimen
9	60952	332.4554716	235.429835	0.706057518	62329	0.74359819	933.366	Kecimen
10	42256	323.1896072	172.5759261	0.845498789	44743	0.698030924	849.728	Kecimen
11	64380	366.9648423	227.7716147	0.784055626	66125	0.664375716	981.544	Kecimen
12	80437	449.4545811	232.3255064	0.856042518	84460	0.674235757	1176.305	Kecimen
13	43725	301.3222176	186.9506295	0.784258452	45021	0.697068248	818.873	Kecimen
14	43441	276.6108288	201.8131355	0.683882337	45133	0.690855598	803.748	Kecimen
15	76792	338.8575454	291.3592017	0.510583813	78842	0.772322237	1042.77	Kecimen
16	74167	387.7989307	247.8581228	0.769089738	76807	0.680181585	1084.729	Kecimen
17	33565	261.5543311	167.7084908	0.767374275	35794	0.68155052	751.413	Kecimen
18	64670	403.0839752	206.4846437	0.858829168	66419	0.75677257	1028.445	Kecimen
19	64762	354.2939396	235.7524629	0.746473726	66713	0.694998015	981.509	Kecimen
20	43295	304.2844667	182.8110368	0.799406959	44714	0.713838189	814.68	Kecimen
21	70699	418.6985723	216.5960537	0.855799392	72363	0.728075054	1061.321	Kecimen
22	69726	354.1769124	252.529208	0.701160962	71849	0.734398534	1035.501	Kecimen
23	57346	330.4784385	222.4437485	0.739555027	59365	0.723608833	928.272	Kecimen
24	82028	397.1149759	268.3337727	0.737169367	84427	0.686375085	1106.355	Kecimen
25	61251	301.5077895	273.6599414	0.419753707	64732	0.643595671	971.769	Kecimen
26	96277	447.1345225	275.2161542	0.788128405	97865	0.704057157	1181.921	Kecimen
27	75620	368.2242844	263.4592554	0.698627251	77493	0.726277372	1059.186	Kecimen
28	73167	340.055218	276.0151772	0.58410581	74545	0.778736856	1010.474	Kecimen
29	60847	336.9238696	231.4656959	0.726660229	62492	0.69858783	964.603	Kecimen
30	81021	347.7500583	297.6406265	0.517134931	82552	0.757559607	1063.868	Kecimen
31	59902	358.5919148	222.9020273	0.783331958	63250	0.744124224	982.788	Kecimen
32	88745	429.770355	265.6902361	0.786009488	90715	0.752063524	1162.877	Kecimen
33	41809	307.5327392	175.085568	0.822113695	43838	0.697444367	828.697	Kecimen
34	75329	364.2307798	265.8668635	0.683510499	77541	0.723079729	1075.792	Kecimen
35	61600	350.1827545	225.8427713	0.764243075	63397	0.746829611	972.472	Kecimen
36	46427	253.8420284	235.9068241	0.369212459	48275	0.684219059	844.312	Kecimen
37	40861	249.7402266	213.5732718	0.51832833	43096	0.743089401	784.912	Kecimen
38	55827	305.298843	234.6612245	0.639696077	57724	0.703287982	926.095	Kecimen
39	54182	366.0666742	192.013274	0.851391425	56450	0.611417674	968.729	Kecimen

Raisin_Grains_Dataset | Citation_Request

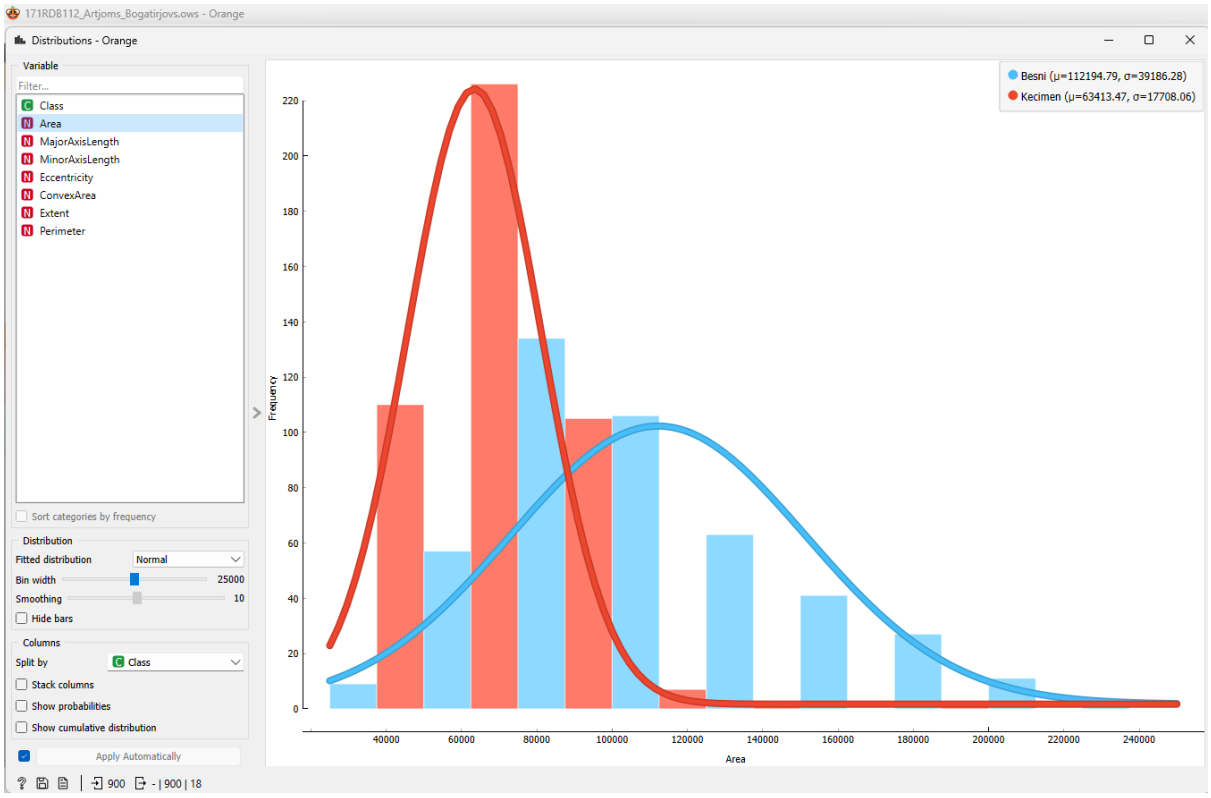
Divdimensiju izkliedes diagrammām (Scatter Plot). MinorAxisLenght (y ass) un Perimeter (x ass).



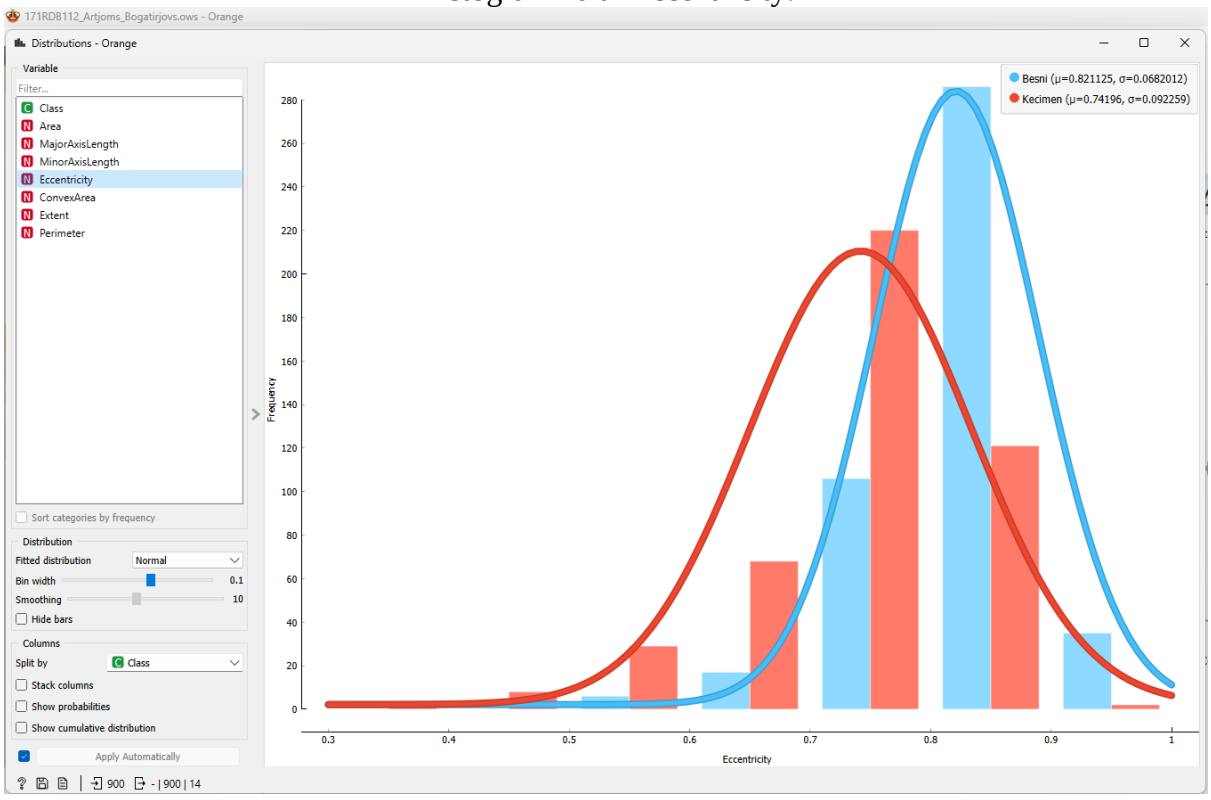
Otrā izveidotā izkliedes diagramma. Extent (y ass) un ConvexArea (x ass).



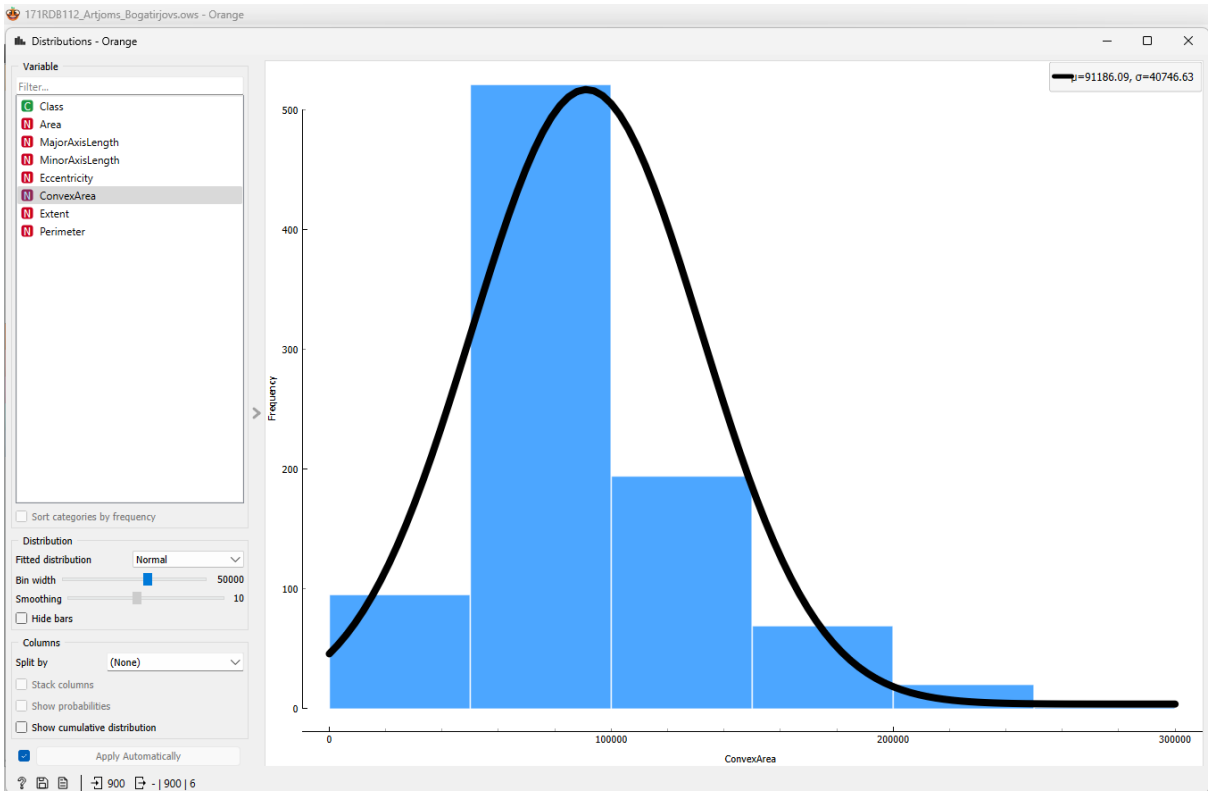
Histogramma, kur klašu atdalīšanai izvēlēta pazīme Area.



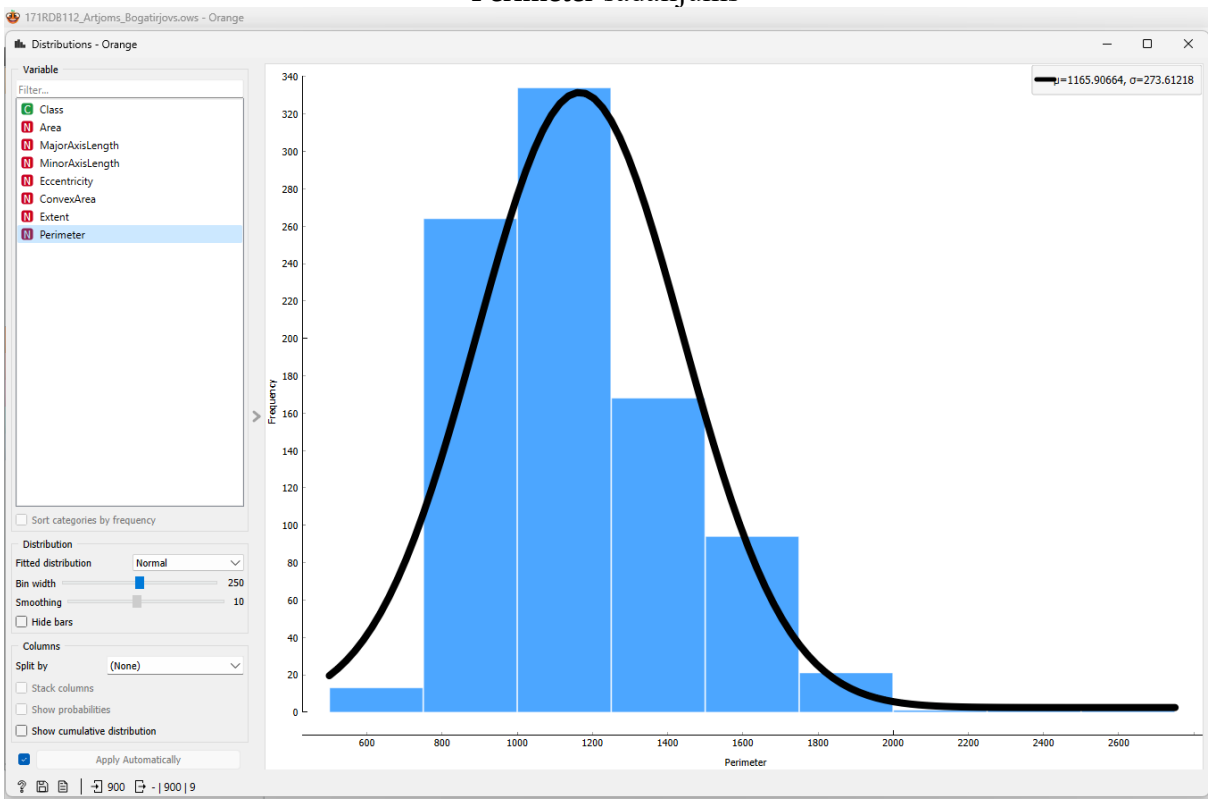
Histogramma ar Eccentricity.



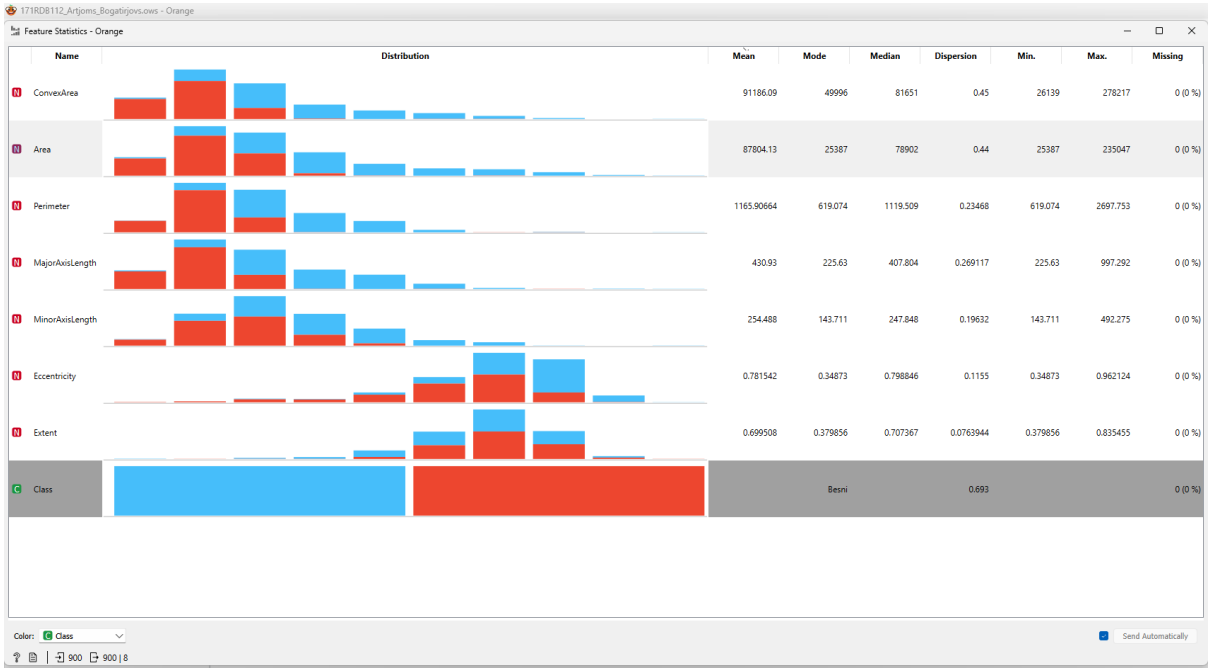
ConvexArea sadalījums



Perimeter sadalījums



Visas statistiskās vērtības: “Mean” - vidējās vērtības un “Dispersion” - dispersija jeb standartnovirze.



Secinājumi

Saskaņā ar datu kopas aprakstu var secināt, ka klases ir līdzsvarotas, jo katrai no tām pieder vienāds skaits objektu.

Analizējot izveidotos diagrammas, var teikt, ka vizuālais attēlojums ļauj redzēt datu struktūru, bet daļa objektu nav skaidri atdalāmi. Konkrētāk runājot par Scatter Plot diagrammām, sarkanā klase ir ciešāk saistīta ar zilo klasi, kas ir izkliedētāka un nedaudz nobīdīta no sarkanās klases. Tādēļ var secināt, ka atdalāmi datu grupējumi ir daļēji novērojami. Šie grupējumi ir tuvu viens otram un daļēji pārklājas.

Pēc statistisko rādītāju analīzes var secināt, ka dispersija ir vismazākā īpašībās Eccentricity un Extent (0.12 un 0.08), bet vislielākā īpašībās ConvexArea un Area (0.45 un 0.44). Līdzīgi, maksimālā un minimālā vērtība atšķiras visvairāk īpašībās ConvexArea un Area (252078 un 209660), kur starpība ir aptuveni 10 reizes, bet vismazākā starpība ir īpašībās Eccentricity un Extent (0.61 un 0.46), kur starpība ir mazāka par 3 reizēm.

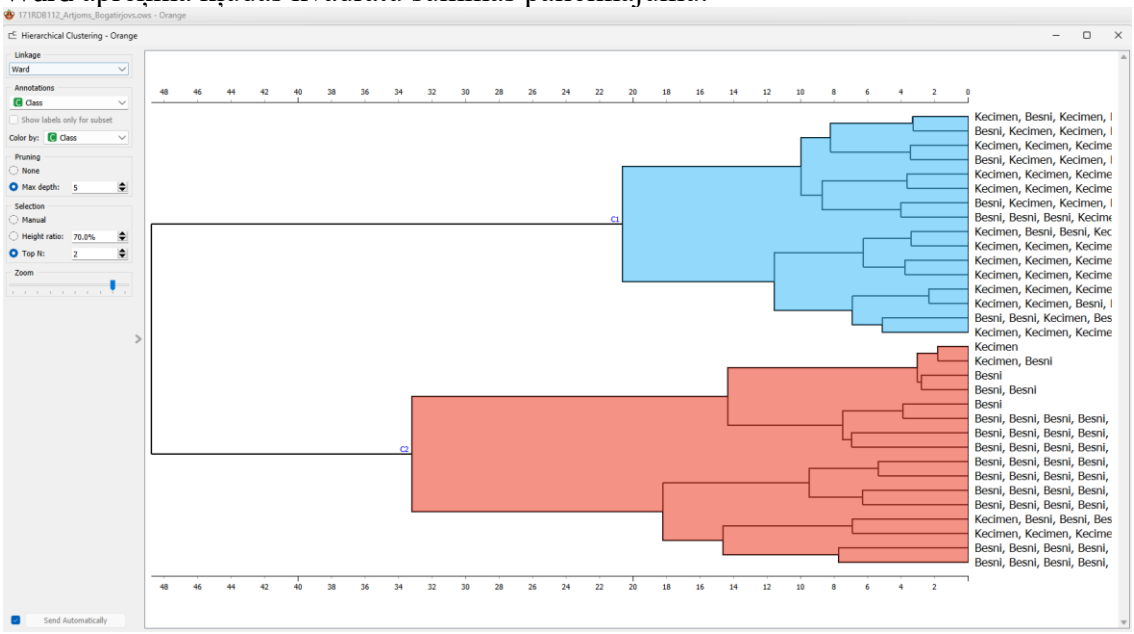
Nepārraudzītā mašīnmācīšanās

Hierarhiskā klasterizācija

Hiperparametri:

- Linkage - Ward;
- Pruning - Max depth: 5;
- Selection - Top N: 2.

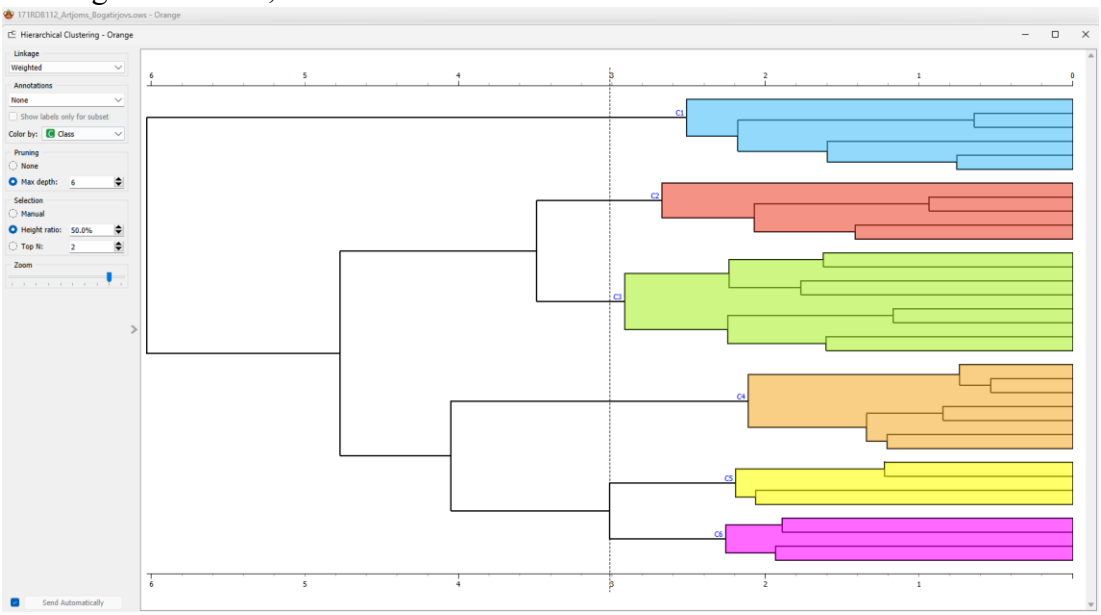
Te redzams, ka iegūti tikai divi klasteri, jo tā norādīts ar opcijas Top N palīdzību. Opcija Ward aprēķina kļūdas kvadrātu summas palielinājumu.



Hiperparametri:

- Annotations - None;
- Selection- Height ratio: 50%.

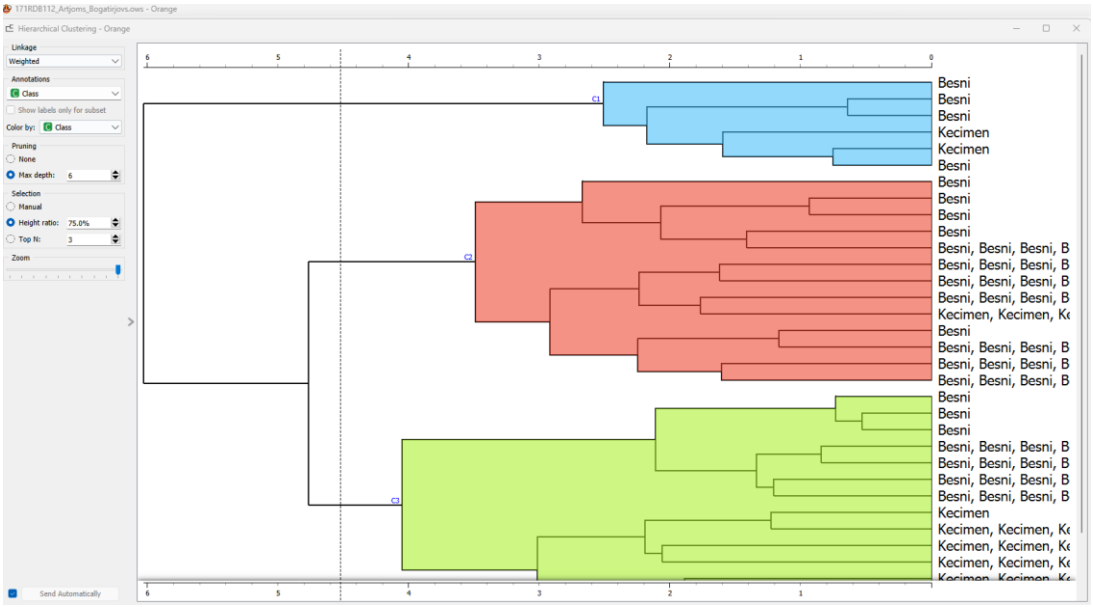
Tika iegūti 6 klasteri, kas iekrāsoti dažādās krāsās.



Hiperparametri:

- Linkage - Weighted;
- Pruning - Max depth: 6;
- Selection- Height ratio: 75%.

Šeit redzami 3 klasteri - zilā, sarkanā un zaļā krāsā.

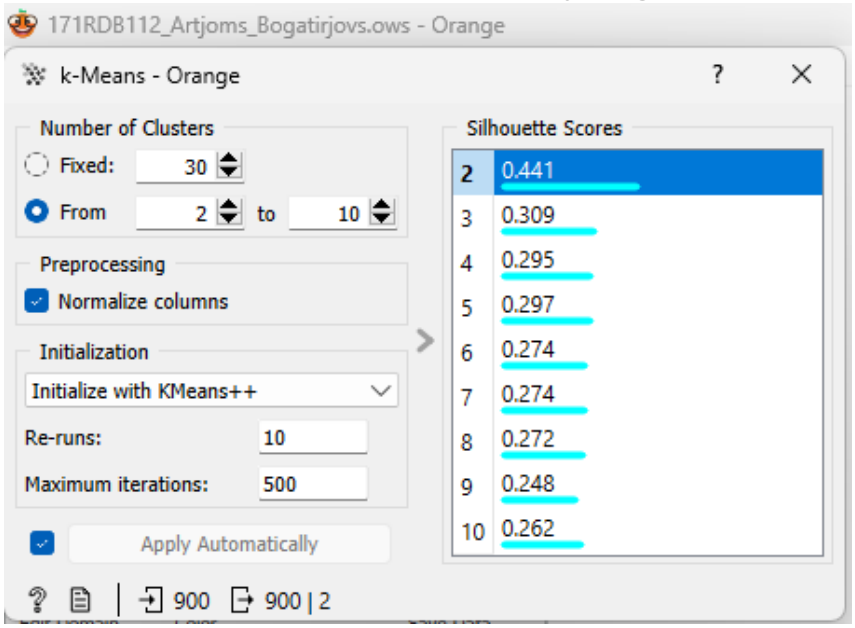


Hierarhiskās klasterizācijas algoritma hiperparametru apraksts

Hierarhiskās klasterizācijas algoritma Orange rīkā ir pieejami pieci hiperparametri, no kuriem trīs ietekmē dendrogrammas un tās vizuālo izskatu.

1. Linkage: Šis hiperparametrs ietekmē attālumu mērīšanas veidus starp klasteriem. Pieejamie veidi ir "Single" (vienkāršais), "Average" (vidējais), "Weighted" (svinātais), "Complete" (pilnais) un "Ward" (Varda).
2. Pruning: Šis hiperparametrs ļauj izvēlēties maksimālo dendrogrammas dziļumu. Tas galvenokārt ietekmē dendrogrammas izskatu, nevis pašu klasterizācijas procesu.
3. Selection: Šis hiperparametrs piedāvā trīs opcijas:
    - "Manual" (manuāla) ļauj lietotājam atlasīt vienu vai vairākus klasterus iekš dendogrammas.
    - "Height ratio" (augstuma attiecība) automātiski atlasa dendrogrammas elementus, pamatojoties uz lietotāja noklikšķinājumiem uz augšējās vai apakšējās mērjoslas.
    - "Top N" (augšējie N) atlasa noteiktu augšējo mezglu skaitu (klasteru skaitu) dendrogrammā.
4. Annotations: Šis hiperparametrs ļauj lietotājam mainīt dendrogrammas etiķetes vai labelus, kas attēlo objektu vai klasteru nosaukumus.
5. Zoom: Šis hiperparametrs ļauj mainīt attēla un dendrogrammas tuvumu, ļaujot lietotājam izvēlēties vēlamo skatīšanās līmeni.

K-vidējo algoritms



Attēlā ir redzamas izvēlētās k vērtības no 2 līdz 10, kā arī katras k vērtības Silhouette Score. Silhouette Score vērtības ir mērītājs, kas norāda klasteru atdalāmību. Jo tuvāk šī vērtība ir vieniniekam, jo labāka ir klasteru atdalāmība. No otras puses, jo tuvāk šī vērtība ir nullei, jo maznozīmīgāks ir klasteru sadalījums.
Konkrētajā gadījumā var redzēt, ka vislabākais sadalījums ir starp diviem klasteriem, kuram ir Silhouette Score vērtība 0.441. Katru papildu klasteru pievienošana samazina Silhouette Score vērtību, kas norāda uz sliktāku un maznozīmīgāku datu sadalījumu.

K-vidējo algoritma hiperparametru apraksts

Šim algoritmam ir trīs hiperparametri: Number of Clusters, Preprocessing un Initialization.

1. Number of Clusters: Šis hiperparametrs ļauj izvēlēties klasteru skaitu. Ir pieejamas divas opcijas:
   - "Fixed" (fiksēts) grupē datus līdz noteiktam klasteru skaitam.
   - "From X to Y" (no X līdz Y) parāda rezultātus grupēšanai ar klasteru skaita diapazonu (Silhouette Score).
2. Preprocessing: Šī opcija, ja tā ir atlasīta, veic priekšapstrādi (preprocessing) datus. Tas ietver kolonnu normalizāciju, kur vidējā vērtība tiek centrēta ap nulles vērtību, un standartnovirze tiek mērogojama uz vienīnieku.

3. Initialization: Šis hiperparametrs ļauj izvēlēties, kā algoritms sāks klasterēšanas procesu.
   - "Re-runs" (atkārtoti palaist) norāda, cik reizes algoritms tiks izpildīts, sākot no nejaušām sākotnējām pozīcijām.
   - "Maximum iterations" (maksimālais iterāciju skaits) nosaka maksimālo iterāciju skaitu, kāds tiks veikts algoritma izpildes laikā.

Šie hiperparametri ļauj pielāgot klasterizācijas procesu atbilstoši konkrētajiem dati un vajadzībām.

Secinājumi

Pēc abu algoritmu darbības analīzes var secināt, ka datu kopā esošās klases ir atdalāmas, taču šī atdalāmība nav ļoti laba. Tas ir norādīts gan ar iegūtajiem Silhouette Score rezultātiem, gan ar dendrogrammu, kas iegūta hierarhiskajā klasterizācijā.
Silhouette Score vērtības norāda uz klasteru atdalāmību, un ja šī vērtība ir tuvu vieniniekam, tas norāda uz labu klasteru atdalījumu. Tomēr, ja Silhouette Score vērtība ir tuvu nullei, tas norāda uz maznozīmīgu klasteru sadalījumu.
Tāpat arī dendrogramma hierarhiskajā klasterizācijā sniedz vizuālu attēlu par klasteru sadalījumu. Analizējot iegūto dendrogrammu, var novērot, ka klases ir daļēji atdalāmas, bet nav pārāk skaidri atšķiramas viena no otras.
Tas liecina par to, ka datu kopā esošās klases ir atdalāmas, bet to atdalījums nav ideāls vai pilnīgi skaidrs. Iespējams, ka dati satur dažas pārklājošās vai sajauktas īpašības, kas sarežģī klasterizācijas procesu un samazina klasteru atdalāmību.

Parraudzītā mašīnmācīšanās

Šajā daļā tika brīvi izvēlēti “Naive Bayes” un “kNN” pārraudzītās mašīnmācīšanās algoritmi.

- Naive Bayes ir klasifikators no vienkāršu varbūtējo klasifikatoru saimes un tā pamatā ir Bayes teorēma, kas īsteno pieņēmumu par līdzekļu neatkarību.
- kNN ir algoritms, kas meklē k tuvākos apmācības piemērus līdzekļu telpā un izmanto to vidējo vērtību kā paredzējumu (prediction).

kNN algoritma pieejamie hiperparametri

kNN (k-tuvāko kaimiņu) algoritms piedāvā trīs hiperparametrus: Neighbors (kaimiņu skaits), Metric (attāluma mērīšanas veids) un Weight (svēršanas veids).

1. Neighbors: Šis hiperparametrs ļauj iestatīt tuvāko kaimiņu skaitu, kas tiks izmantots kNN algoritmā. Tas nosaka, cik tuvus kaimiņus ņems vērā, lai veiktu klasifikāciju vai prognozi.
2. Metric: Šis hiperparametrs nosaka attāluma mērīšanas veidu, kas tiek izmantots, lai noteiktu kaimiņu attālumu no dotā punkta. Ir pieejami trīs iespējamie veidi:
    - Eiklīda attālums (Euclidean): Mēra attālumu starp diviem punktiem taisnā līnijā.
    - Menhetenes attālums (Manhattan): Aprēķina attālumu kā visu atribūtu absolūto atšķirību summu.
    - Mahalnobisa attālums (Mahalanobis): Izmanto attālumu starp punktu un izkliedi, ņemot vērā kovariāciju starp atribūtiem.
3. Weight: Šis hiperparametrs nosaka svēršanas veidu, kas tiek izmantots, lai piešķirtu nozīmi kaimiņiem. Ir pieejamas divas opcijas:
    - Vienāds svars (Uniform): Visiem punktiem katrā apgabalā tiek piešķirts vienāds svars vai nozīme.
    - Attāluma svars (Distance): Tuvinieku punktiem ir lielāka ietekme nekā tālākiem kaimiņiem. Tātad, attālums starp kaimiņiem tiek ņemts vērā, lai nosvērtu to ietekmi algoritmā.

Šie hiperparametri ļauj pielāgot kNN algoritmu atbilstoši specifiskajiem datiem un prasībām, nodrošinot elastību un kontroli pār klasifikācijas vai prognozēšanas procesu.

Neural Network algoritma pieejamie hiperparametri

Neiro tīkla algoritmam ir pieejami pieci parametri:

1. Neironi slēptajā slānī (Neurons per hidden layer): Šis parametrs norāda, cik daudz neironu būs iekļauti slēptajā slānī. Slēptais slānis ir starp ievades un izvades slāņiem un palīdz tīklam apgūt sarežģītākas ievades datu raksturīgās iezīmes.
2. Aktivizācijas funkcija (Activation): Šis parametrs ļauj izvēlēties vienu no četrām aktivizācijas funkcijām, kas tiek izmantotas, lai aktivizētu neironu izvades vērtības slēptajos un izvades slāņos. Pieejamās opcijas ir "Identity" (bez aktivizācijas), "Logistic" (loģistiskā sigmoīda funkcija), "tanh" (hiperboliskā tangensa funkcija) un "ReLu" (rektificētas lineārās vienības funkcija).
3. Optimizācijas algoritms (Solver): Šis parametrs nosaka, kāds optimizācijas algoritms tiks izmantots, lai trenētu neiro tīklu. Pieejamās opcijas ir "L-BFGS-B" (optimizētājs kvazi-Ņūtona metožu saimē), "SGD" (stohastiska gradienta nolaišanās) un "Adam" (stohastisks gradienta optimizētājs).
4. L2 soda parametrs (Alpha): Šis parametrs norāda L2 soda parametra vērtību, kas palīdz kontrolēt neiro tīkla pārmērīgu pielāgošanos datiem. L2 soda palīdz novērst pārmērīgu svaru izmantošanu, kas var novest pie pārmācības.
5. Maksimālais iterāciju skaits (Max iterations): Šis parametrs nosaka maksimālo iterāciju skaitu, ko algoritms veiks, lai trenētu neiro tīklu. Tas nodrošina, ka algoritms pietiekami ilgi mācās no datiem, bet arī novērš pārmērīgu treniņu, kas var novest pie pārmācības.

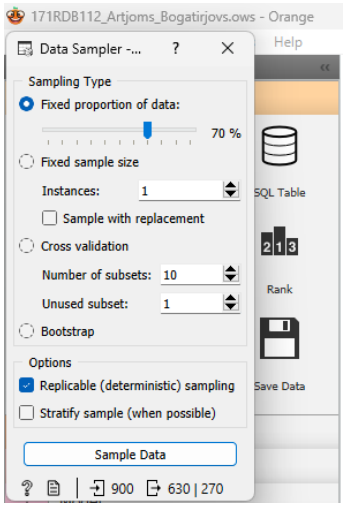
Šie parametri nodrošina iespēju pielāgot neiro tīklu atbilstoši konkrētajiem datiem un problēmai, kas tiek risināta. Tādējādi tiek panākta lielāka efektivitāte un precizitāte trenēšanas procesā.

Naive Bayes algoritma pieejamie hiperparametri

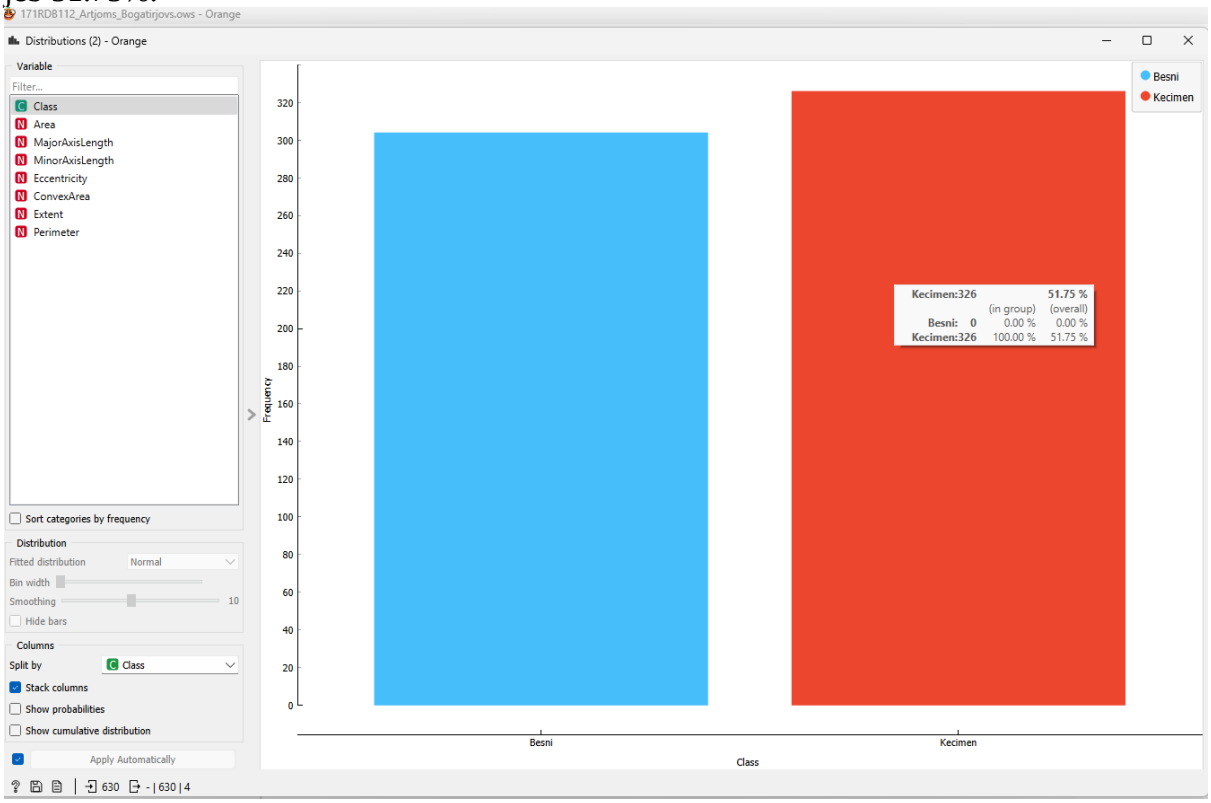
Šim algoritmam Orange rīkā nav pieejamu hiperparametru, kas ietekmētu pašu algoritmu. Iespējams izmainīt tikai tā nosaukumu.

Informācija par testu un apmācību datu kopām

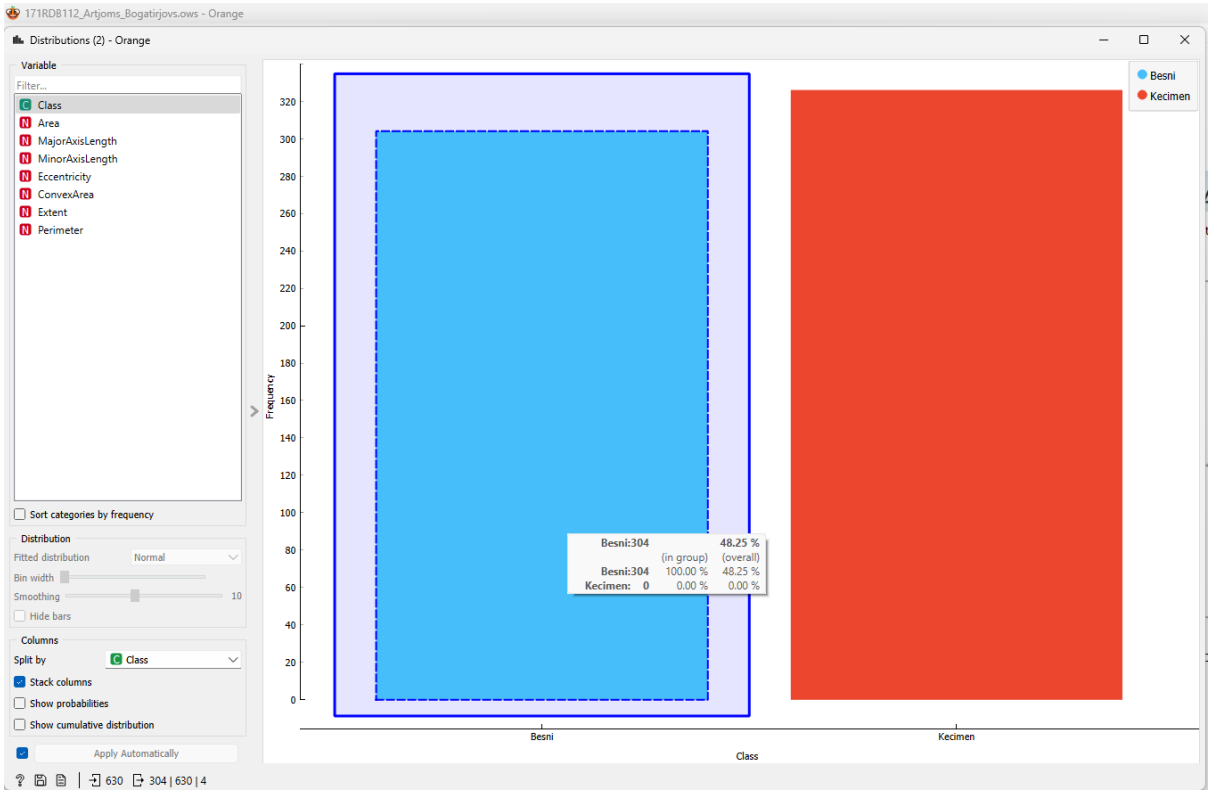
Apmācību datu kopām pievienoto datu objektu skaits. Šeit: 70% jeb 630 objekti.



Redzams Kecimen klases datu objektu skaits, kas iekļauts apmācību un testa datu kopās: 326 jeb 51.75%.

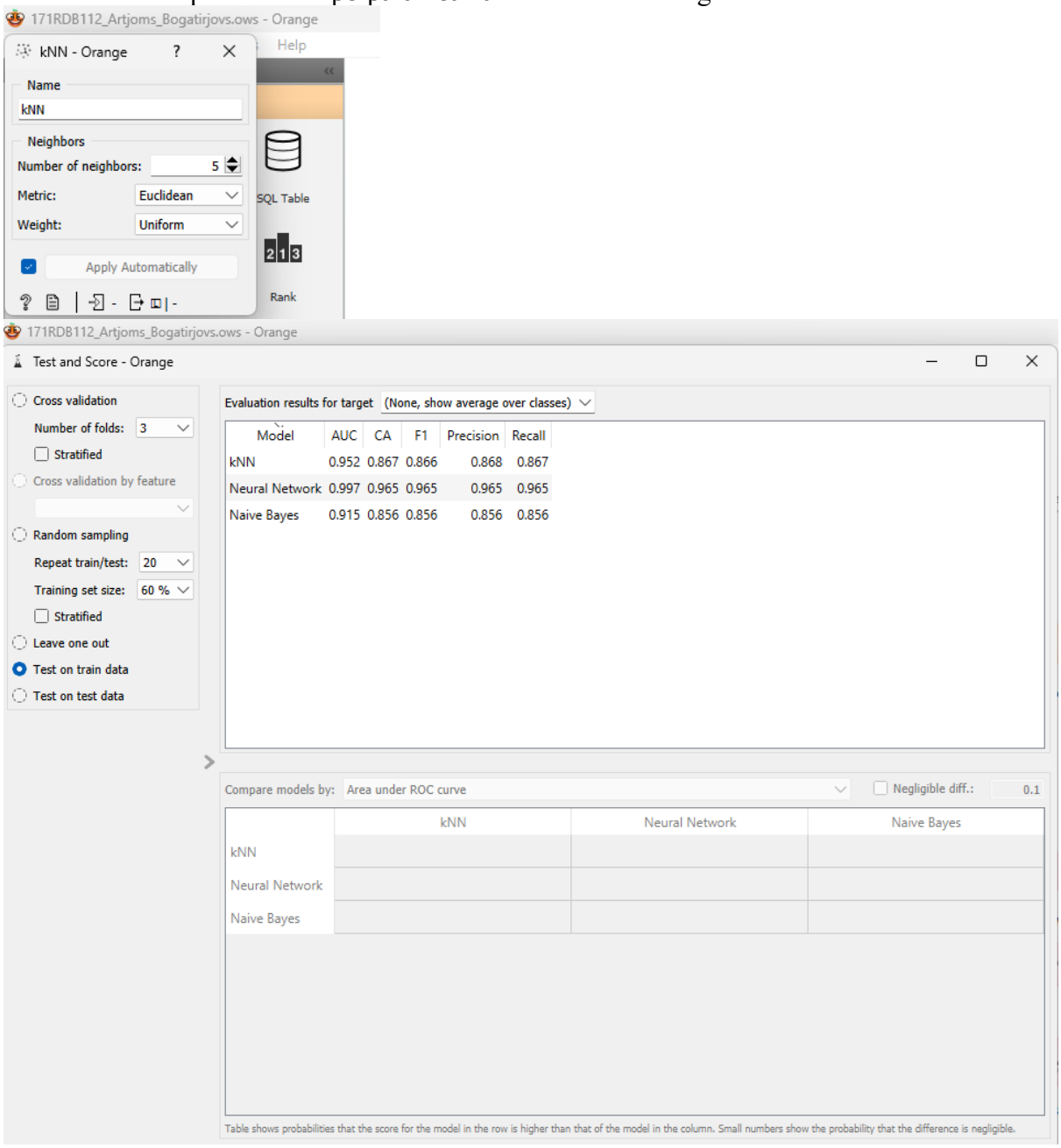


Redzams Besni klases datu objektu skaits, kas iekļauts apmācību un testa datu kopās: 304 jeb 48.25%.

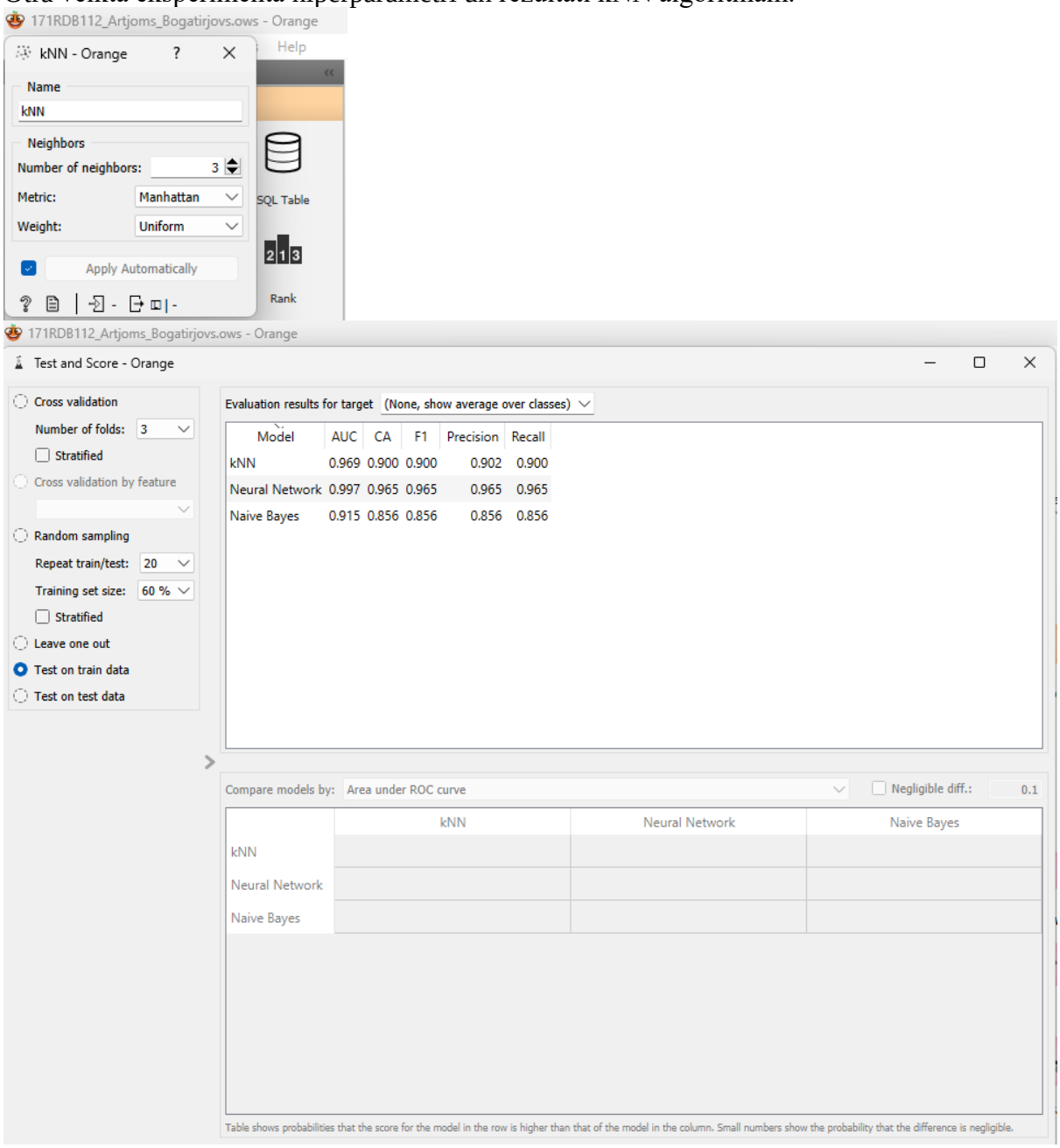


Datu kopas eksperimenti kNN algoritmam

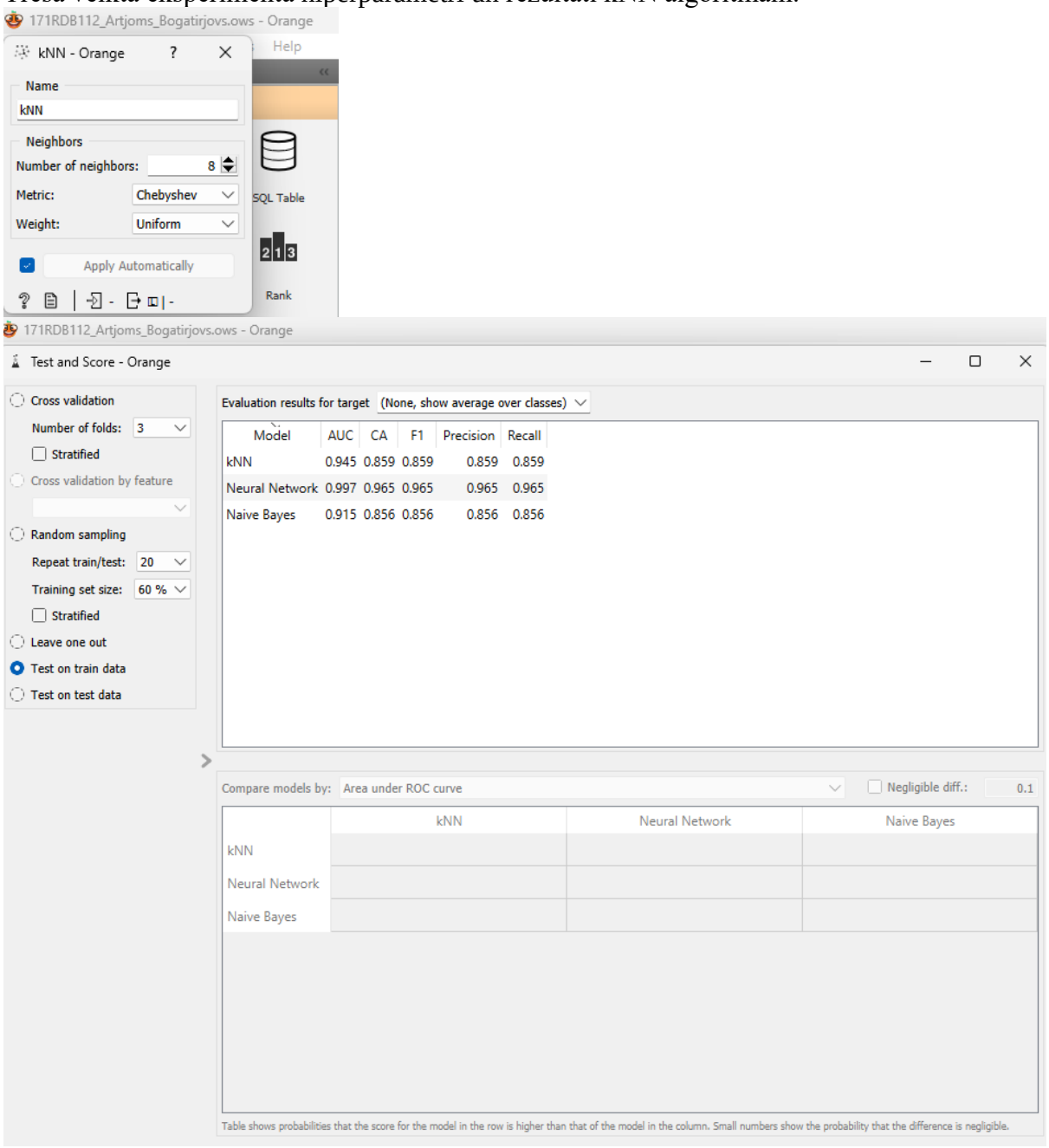
Pirmā veiktā eksperimenta hiperparametri un rezultāti kNN algoritmam.



Otra veiktā eksperimenta hiperparametri un rezultāti kNN algoritmam.



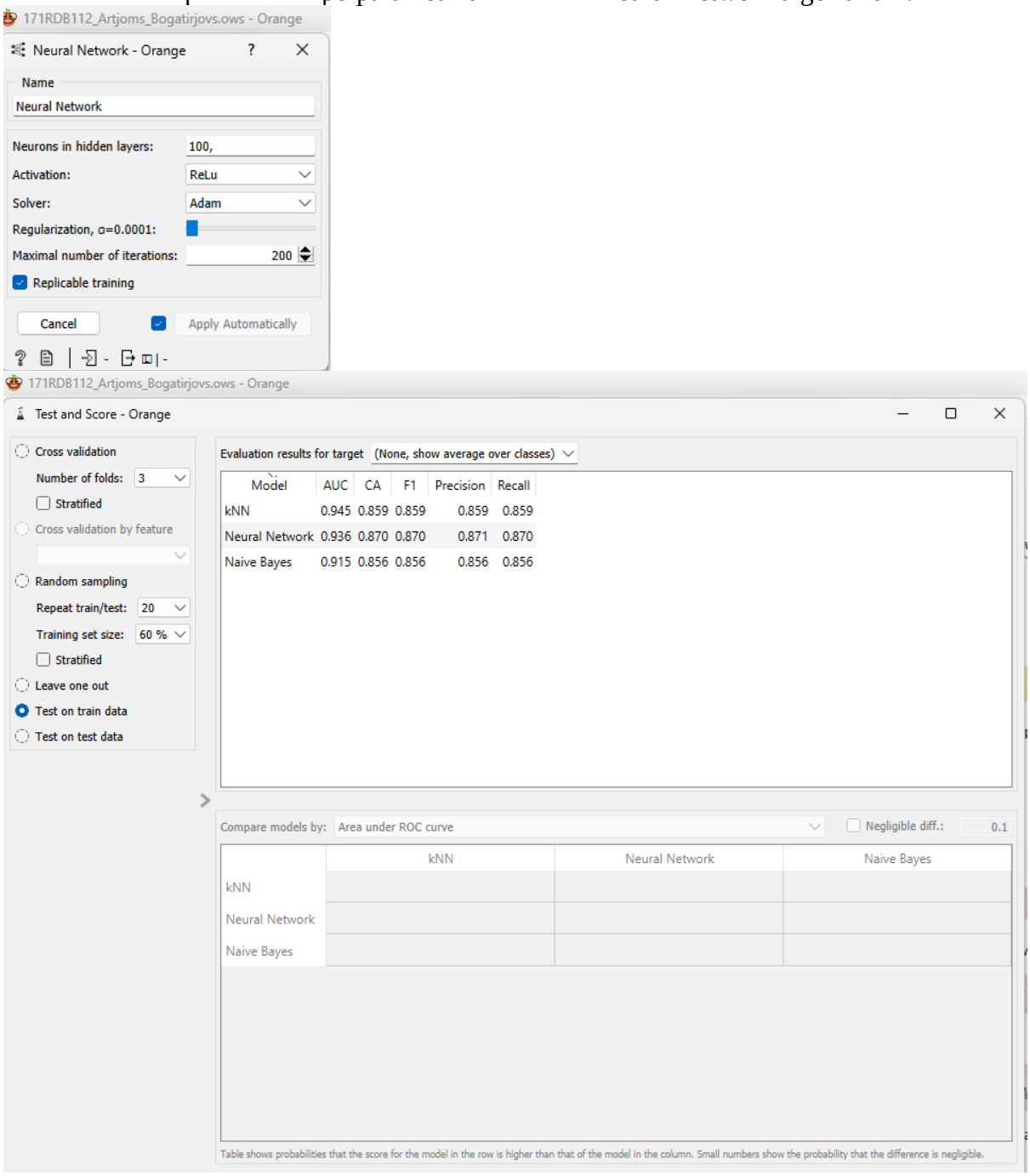
Trešā veiktā eksperimenta hiperparametri un rezultāti kNN algoritmam.



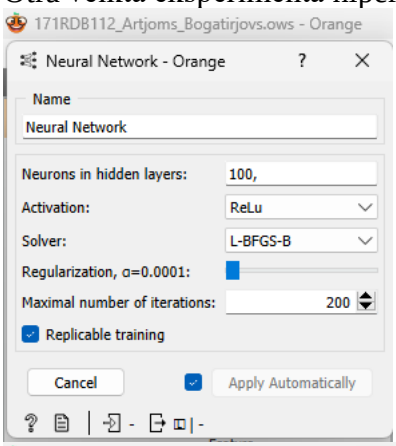
Otra veiktā eksperimenta izvēlētie hiperparametri nodrošināja vislabāko algoritma veiktspēju.

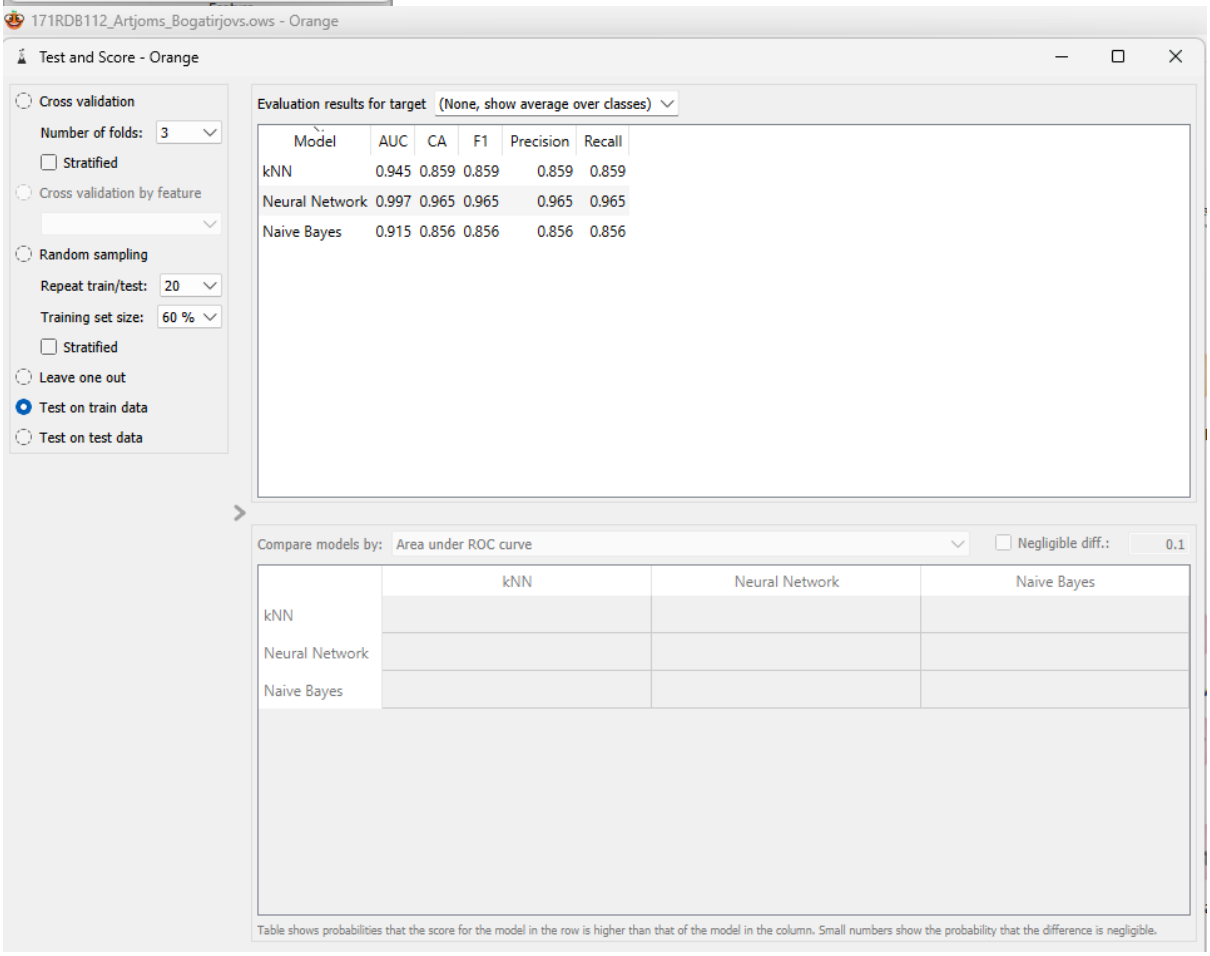
Datu kopas eksperimenti Neural Network algoritmam

Pirmā veiktā eksperimenta hiperparametri un rezultāti Neural Network algoritmam.

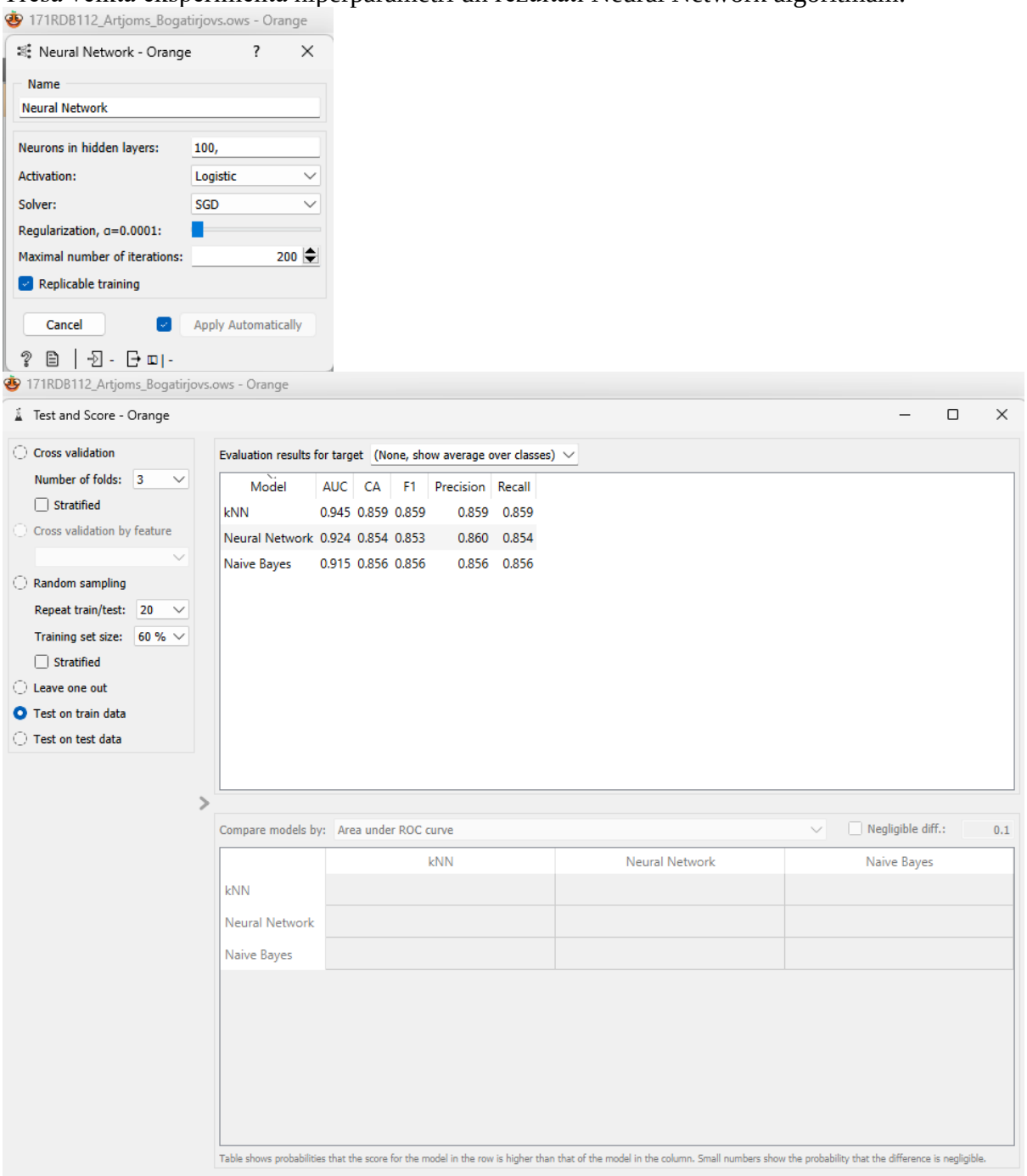


Otra veiktā eksperimenta hiperparametri un rezultāti Neural Network algoritmam.



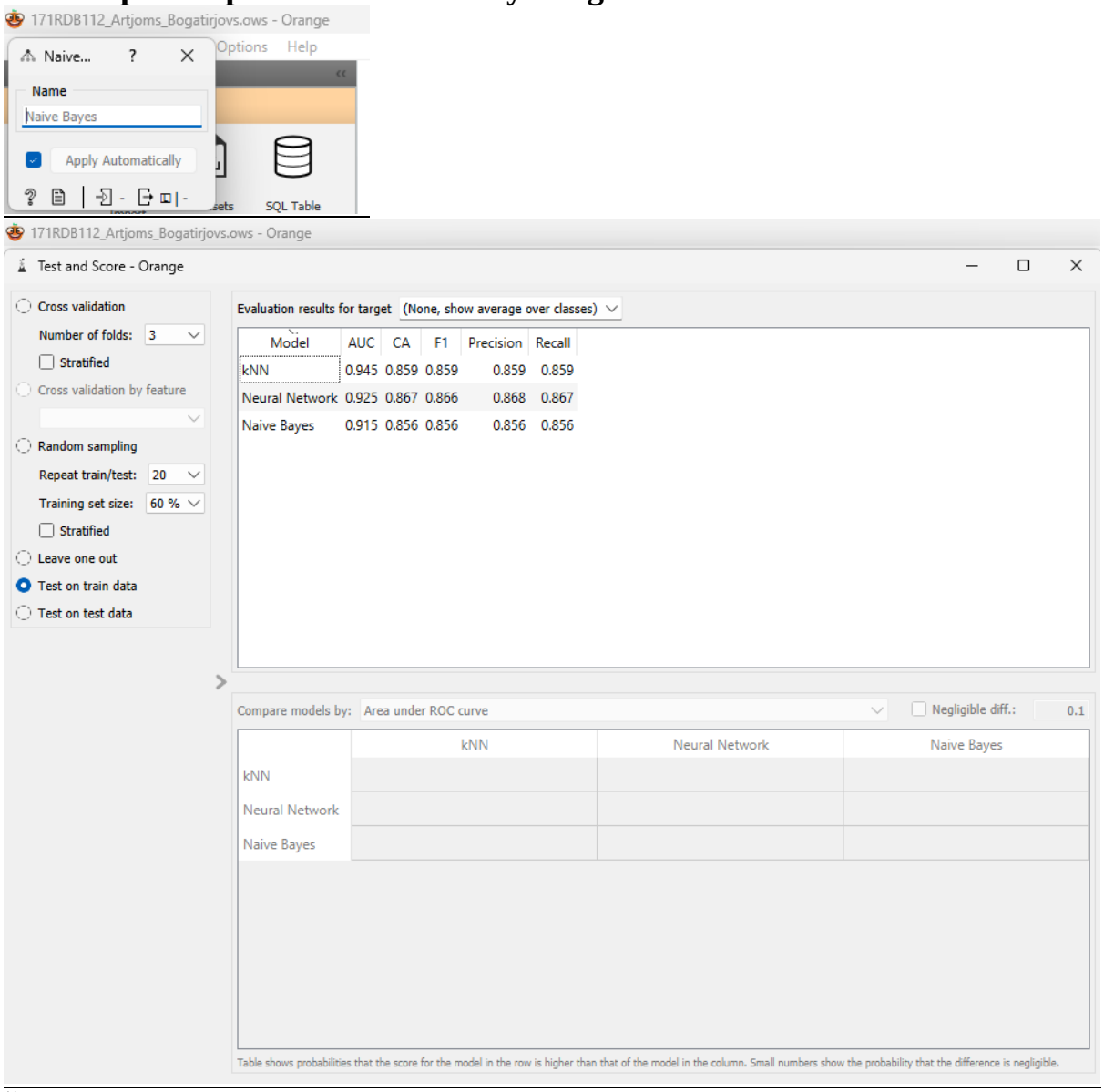


Trešā veiktā eksperimenta hiperparametri un rezultāti Neural Network algoritmam.



Otra veiktā eksperimenta izvēlētie hiperparametri nodrošināja vislabāko algoritma veiktspēju.

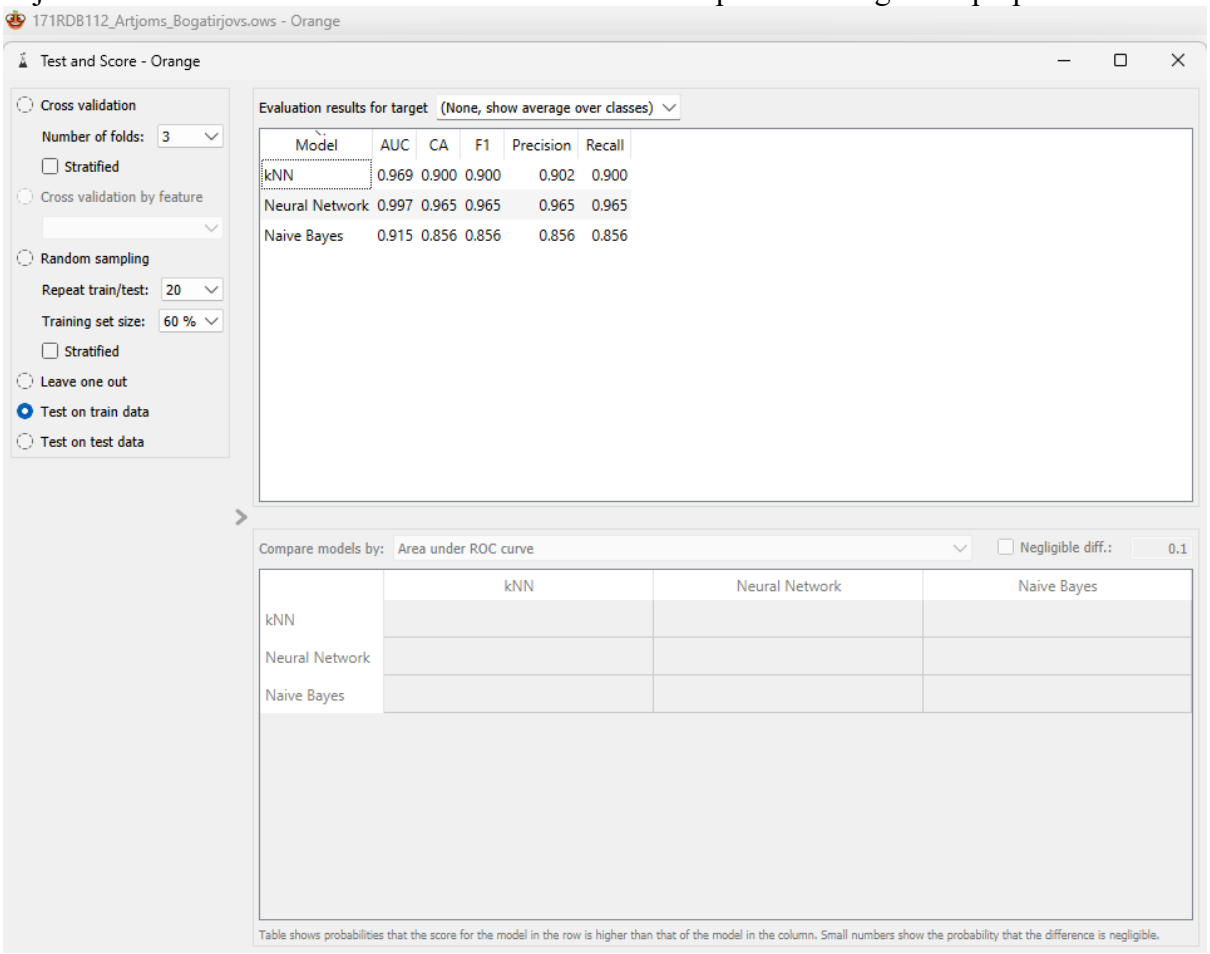
Datu kopas eksperimenti Naive Bayes algoritmam



Šajā gadījumā nav norādīti hiperparametri, jo tādu šim algoritmam nav, līdz ar to rezultāts ir nemainīgs.

Test datu kopas testēšana, izmantojot labākos hiperparametrus

Šajā testā tiks izmantoti kNN un Neural Network 2. eksperimentā iegūtie hiperparametri.

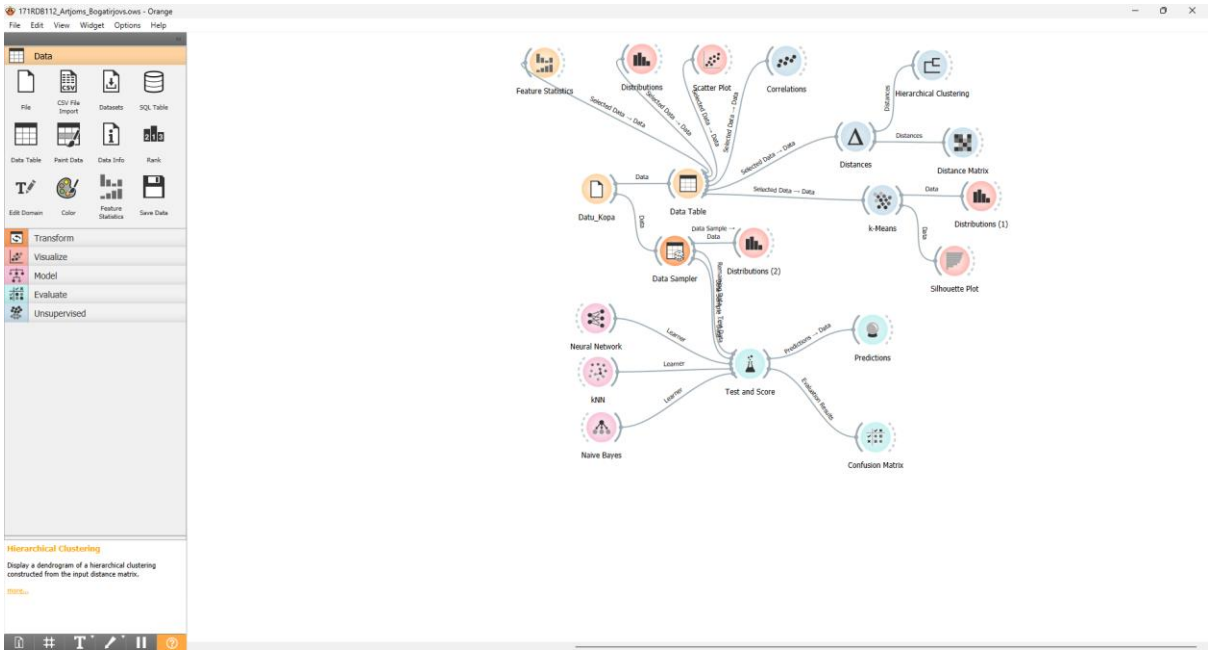


Secinājumi

Pētījuma rezultāti liecina, ka Neural Network algoritms parādīja vislabāko veiktspēju, ar vidējo precizitāti 0.997 un vidējo atgriezenisko saiti (recall) 0.965. To seko Naive Bayes algoritms ar vidējo precizitāti 0.915 un vidējo atgriezenisko saiti 0.855. kNN algoritms rādīja vidējo precizitāti 0.968 un vidējo atgriezenisko saiti 0.9.

Kopumā var secināt, ka Neural Network ir visefektīvākais algoritms šajā konkrētajā pētījumā, nodrošinot augstu precizitāti un atgriezenisko saiti. Tomēr ir jāņem vērā, ka veiktspēja var atšķirties atkarībā no datu kopas un problēmas rakstura. Tādēļ ir svarīgi turpināt pētīt un eksperimentēt ar citiem algoritmiem, lai iegūtu plašāku priekšstatu par to veiktspēju un atbilstību konkrētām situācijām.

Orange rīka darbplūsma



Saites uz avotiem

Datu kopa:

- https://archive.ics.uci.edu/ml/datasets/Raisin+Dataset

GitHub:

- https://github.com/ArtjomsBogatirjovs/AI_2_course_work

Orange:

- https://orangedatamining.com/

Other sources:

- https://estudijas.rtu.lv/course/view.php?id=252548
- https://www.youtube.com/@OrangeDataMining/videos
- https://orangedatamining.com/docs/
    - https://orangedatamining.com/widget-catalog/
- https://www.kaggle.com/learn
- https://archive.ics.uci.edu/ml/index.php
- https://dergipark.org.tr/tr/download/article-file/1227592