RĪGAS TEHNISKĀ UNIVERSITĀTE
Datorzinātnes un informācijas tehnoloģijas fakultāte
Informācijas tehnoloģijas institūts

2.praktiskais darbs
mācību priekšmetā
“Mākslīgā intelekta pamati”

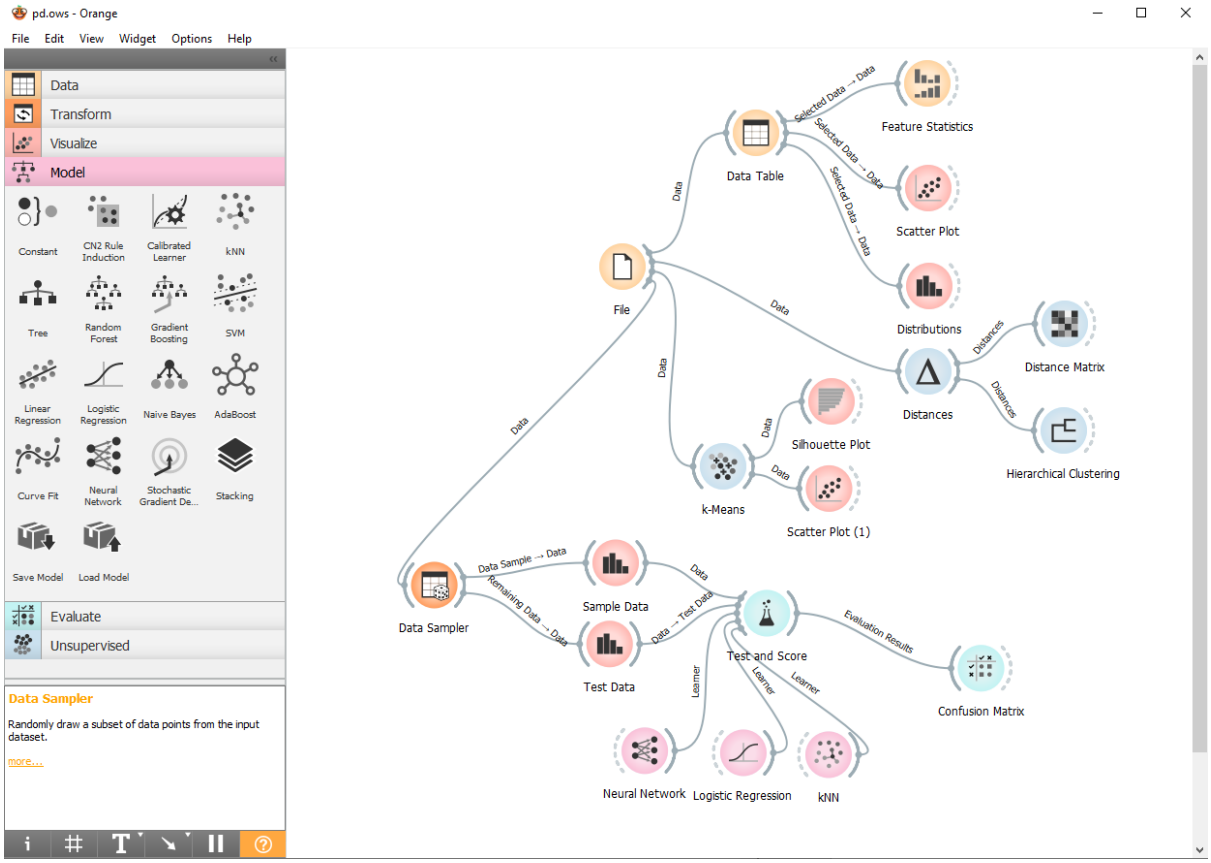
Mašīnmācīšanās algoritmu lietojums

GitHub (projekts un datu kopa):
https://github.com/tarbins/maksligaisintel_pd2

Izstrādāja: Kristaps Lukjanovs
201RDB194

2021./22. māc. Gads

Orange rīka darbplūsma



1. attēls

I. daļa

Datu kopas apraksts

Datu kopas nosaukums ir *Raisin Dataset*, datu kopa ņemta no kaggle.com, oriģinālais avots ir vietnē https://www.muratkoklu.com/datasets/, zem nosaukuma *Raisin Dataset*. Datu kopas izveidotāji un īpašnieki – Ilkay Cinar, Murat Koklu un Sakir Tasdemir.

Datu kopas problēmsfēra aptver divu rozīņu šķirnu (Kecimen un Besni) klasificēšanu pēc to ārējām pazīmēm.

Datu kopa ir zem CC0: Public Domain licences, tā ir brīvi pieejama visiem, kuri to vēlas izmantot.

Tika savāktas 900 rozīnes – 450 no katras šķirnes. Rozīnes tika nofotografētas, katrs attēls izgāja caur dažādiem apstrādes soļiem un tika izgūtas 7 dažādas morfoloģiskas pazīmes par katru rozīni. (atsauce - https://dergipark.org.tr/tr/download/article-file/1227592)

Sākumā datu kopa bija .xlsx faila formātā, taču pārveidoju to uz csv.

Datu kopas satura apraksts

Datu kopā satur 900 objektus.

Kopā ir 2 klases, kuras atbilst rozīnes šķirnei – Kecimen un Besni. Abām klasēm ir vienādas klašu iezīmes, tās var apskatīt 1. tabulā.

Katrai klasei (rozīnes šķirnei) pieder 450 datu objekti.

Datu kopā ir 7 atribūti, kuras raksturo rozīnes ārējās pazīmes[1].

Atribūts	Skaidrojums	Vērtību tips	Vērtību diapazons
Laukums (*Area*)	pikseļu skaits rozīnes robežās	Vesels skaitlis	25387 - 235047
Perimetrs (*Perimeter*)	pikseļu skaits starp rozīnes robežām un apkārtējo vidi	Reāls skaitlis	619.074 - 2697.753
Galvenās ass garums (*MajorAxisLength*)	garākās iespējamās līnijas, ko var novilkt uz rozīnes, garums	Reāls skaitlis	225.63 - 997.292
Īsākās ass garums (*MinorAxisLength*)	īsākās iespējamās līnijas, ko var novilkt uz rozīnes, garums	Reāls skaitlis	143.711 - 492.275
Ekscentriskums (*Eccentricity*)	ellipses ekscentriskums uz rozīnes	Reāls skaitlis	0.34873 – 0.962124
Apmērs (*Extent*)	rozīnes reģiona attiecība pret kopējo pikseļu skaitu ierobežojošajā lodziņā	Reāls skaitlis	0.379856 – 0.835455

1. tabula – datu kopas atribūti

[1] no raksta par datu kopu - https://dergipark.org.tr/tr/download/article-file/1227592

Datu faila struktūras fragments (2. tabula)

Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perimeter	Class
87524	442,2460114	253,291155	0,819738392	90546	0,758650579	1184,04	Kecimen
75166	406,690687	243,0324363	0,801805234	78789	0,68412957	1121,786	Kecimen
90856	442,2670483	266,3283177	0,798353619	93717	0,637612812	1208,575	Kecimen
45928	286,5405586	208,7600423	0,684989217	47336	0,699599385	844,162	Kecimen

2. tabula

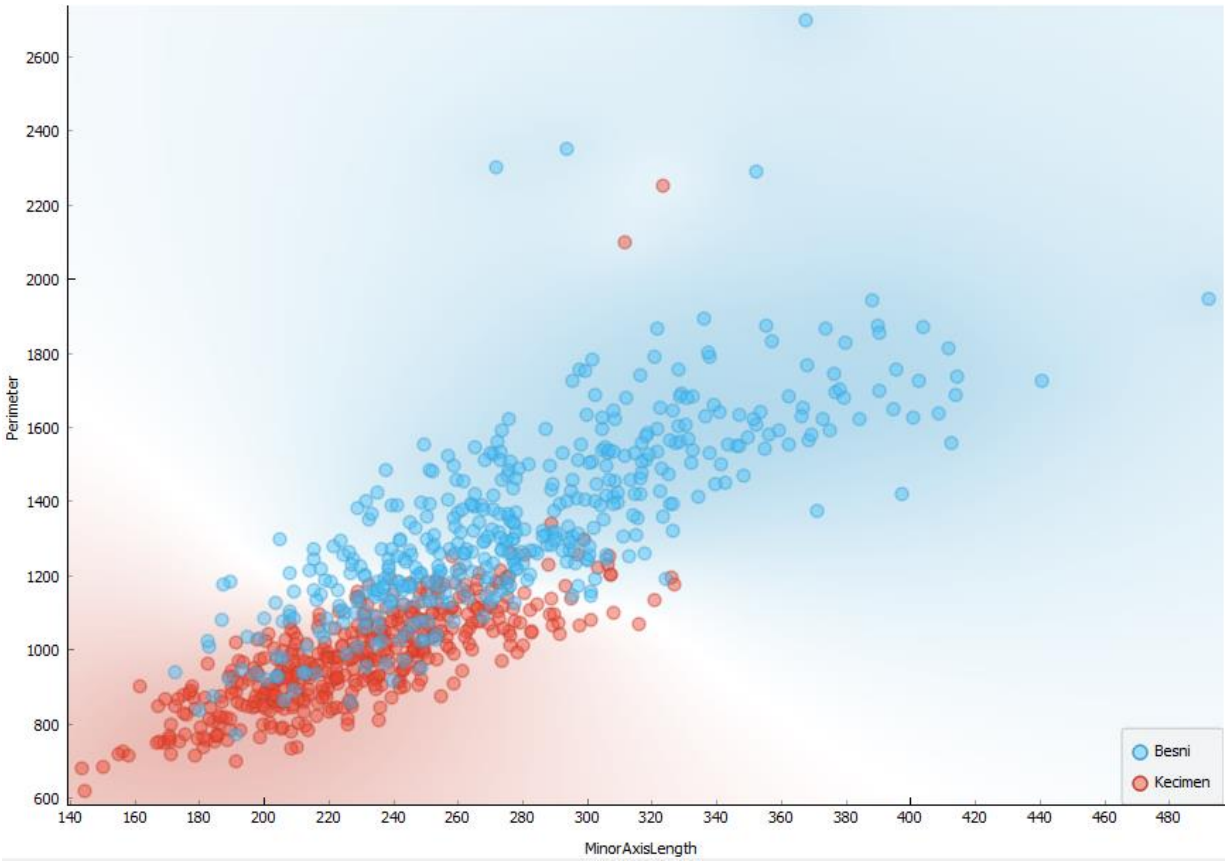
Secinājumi par datu kopas klašu atdalāmību

Klases datu kopā ir līdzsvarotas – abām klasēm ir vienāds skaits datu objektu (450).

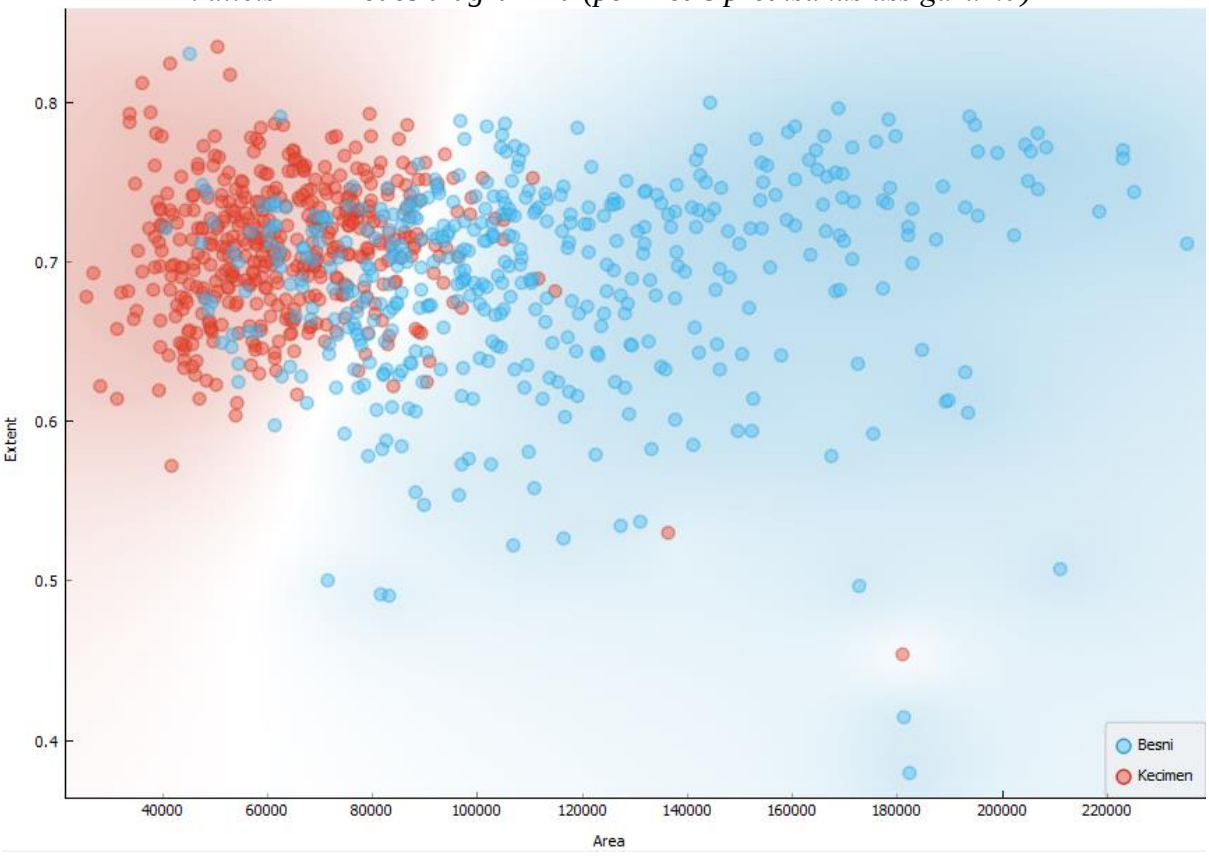
Klašu datu objekti kopumā ir nodalīti viens no otra, taču daži klašu datu objekti pārklājas. Tas tiek atspoguļots 3. un 4. attēlā, kur Kecimen rozīnes šķirnes normālā sadalījuma labā puse (no vidējās vērtības) nedaudz pārklājas ar Besin šķirnes kreiso pusi, taču sadalījumu virsotnes nav tuvu viena otrai, kas liecina par to, ka datu objektus var atdalīt pēc klases.

Izkliedes diagrammās var izdalīt klases pēc datu objektiem. Par to liecina 1.attēls, kur skaidri redzams, ka Besni šķirnes rozīnēm ir lielāks perimetrs un garāka īsā ass nekā Kecimen šķirnei, un 2. attēls, kur Besni ir lielāks laukums par Kecimen, kā arī Besni šķirnes rozīņu datu grupējums ir plašāks par Kecimen.

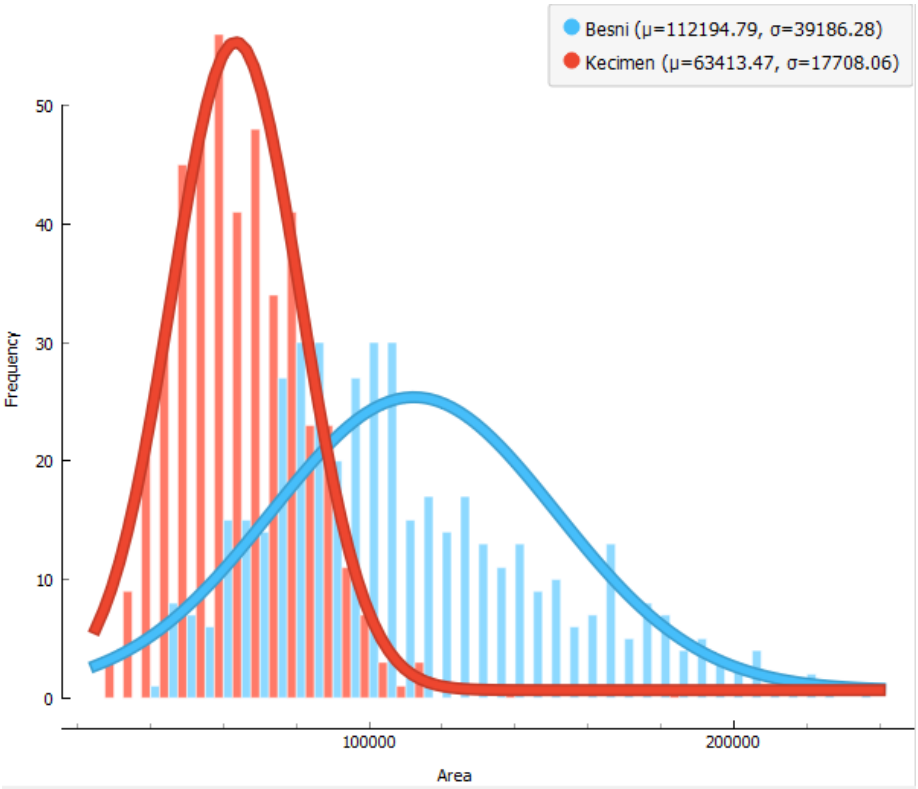
Kopā ir divi datu grupējumi, attiecīgi, katrai klasei ir savs grupējums. Datu grupējumi atrodas tuvu viens otram. Grupējumu galējās vērtības (priekš Besni – zemā sliekšņa vērtības, priekš Kecimen – augstā sliekšņa vērtības) saplūst kopā, taču grupējumi ir pietiekami plaši, lai tos varētu skaidri atšķirt vienu no otra.



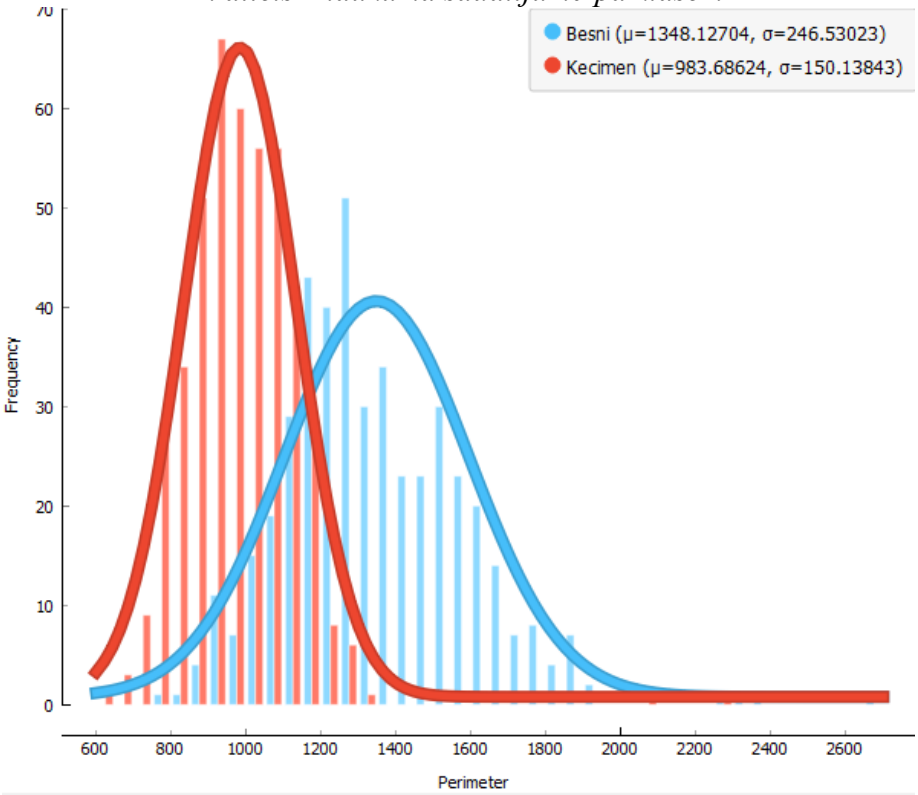
2. attēls – izkliedes diagramma (perimetrs pret īsākās ass garumu)



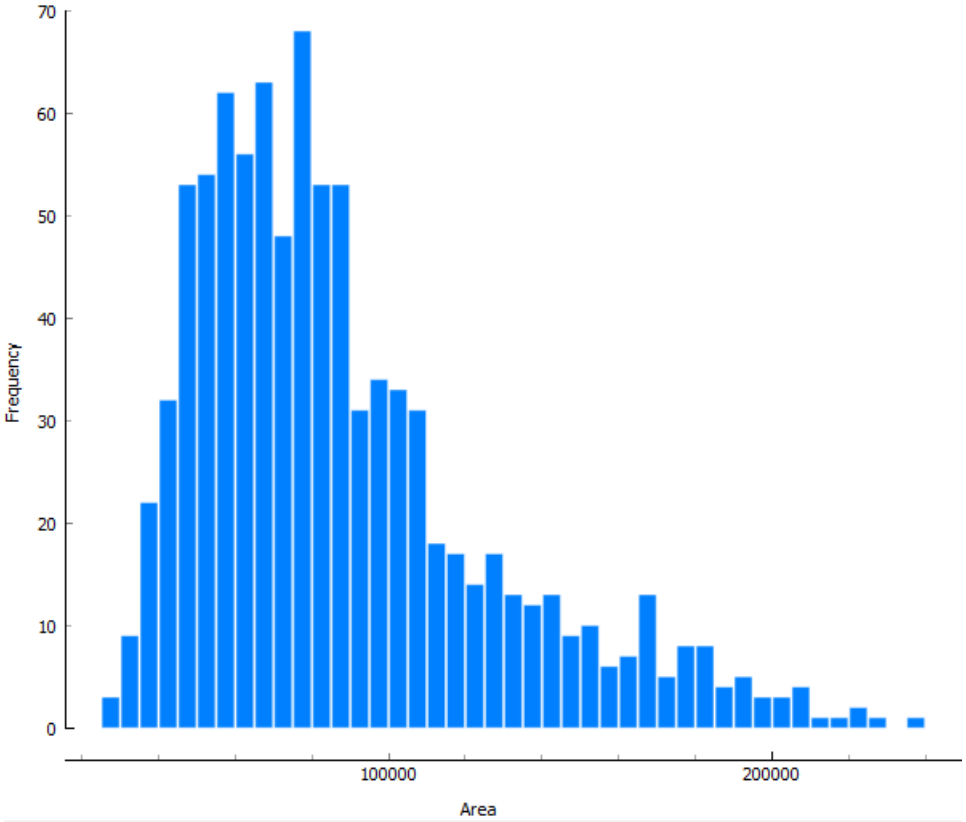
3. attēls – izkliedes diagramma (apmērs pret laukumu)



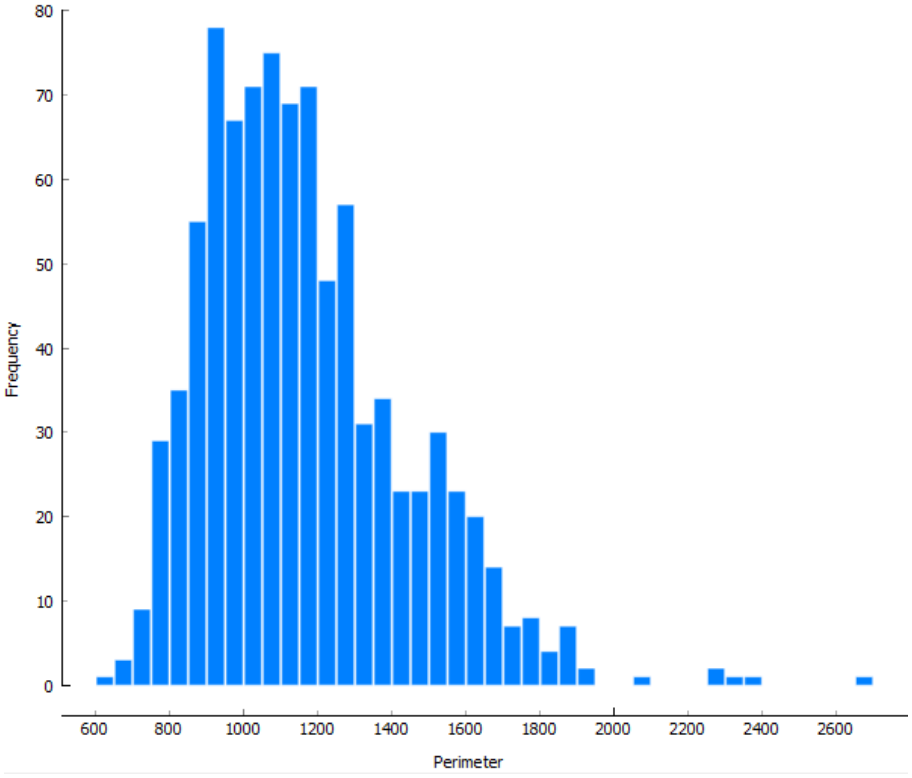
4. attēls – laukuma sadalījuma pa klasēm



5. attēls – perimetra sadalījuma pa klasēm



6. attēls – laukuma sadalījums



7. attēls – perimetra sadalījums

Secinājumi par statistiskiem rādītājiem

Atribūts	Vidējā vērtība	Dispersija
Perimetrs	1119,509	0,23468
Īsākās ass garums (*MinorAxisLength*)	247,848	0,19632
Garākās ass garums (*MajorAxisLength*)	407,804	0,269117
Apmērs (*Extent*)	0,707367	0,0763944
Ekscentriskums (*Eccentricity*)	0,798846	0,1155
Izliekuma laukums (*ConvexArea*)	91186,09	0,45
Laukums	87804,13	0,44

3. tabula – statistiskie rādītāji

Laukums un izliekuma laukums ir atribūti ar vislielāko dispersiju un vidējo vērtību, kas liecina, ka variācija šajos atribūtos potenciāli vislabāk iezīmēs atšķirību starp klasēm (tās labāk nodalīs vienu no otras).

Savukārt apmērs un ekscentriskums ir atribūti ar vismazāko dispersiju. Šie atribūti vāji atdala klases vienu no otras, par cik lielākā daļa vērtību koncentrējas ap vidējo vērtību.

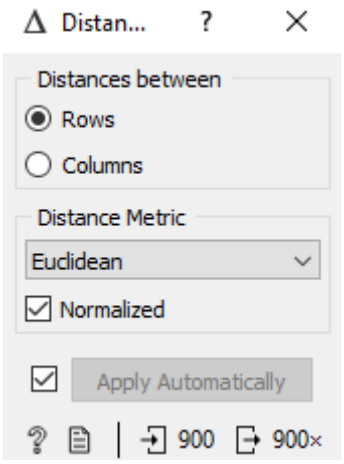
II. daļa

Hierarhiskā klāsterizācija

Hiperparametri – saistīšanas metode[2], tā nosaka klasterus pēc saistības ar datu objektiem tajos[3].

Iestatījumi visiem eksperimentiem:

- **Distances:** between Rows
- **Distance Metric:** Euclidean
- nav Normalized (nav atķeksēts).

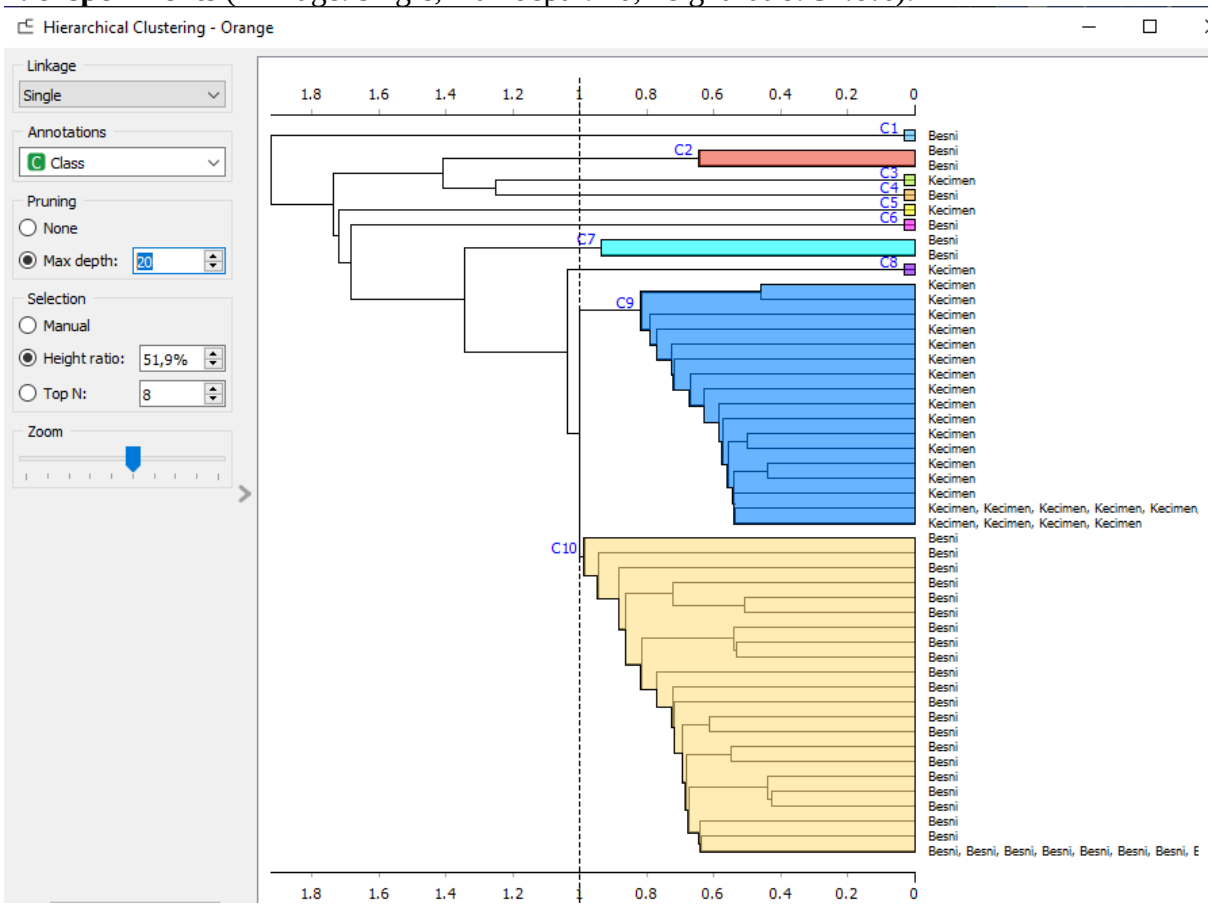


8. attēls

Veicu klasterizāciju ar mērķi iegūt klasterus, kuri pēc iespējas satur tikai vienu klasi (rozīnes šķirni) un klasteru skaits un izmērs lai ir saprāta robežās (pēc iespējas mazāk skaita klasteru un pēc iespējas vairāk vienādu datu objektu vienā klasterī).

[2] https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/hierarchicalclustering.html
[3] https://www.datasciencesmachinelearning.com/2019/10/hierarchical-and-k-means-cluster.html

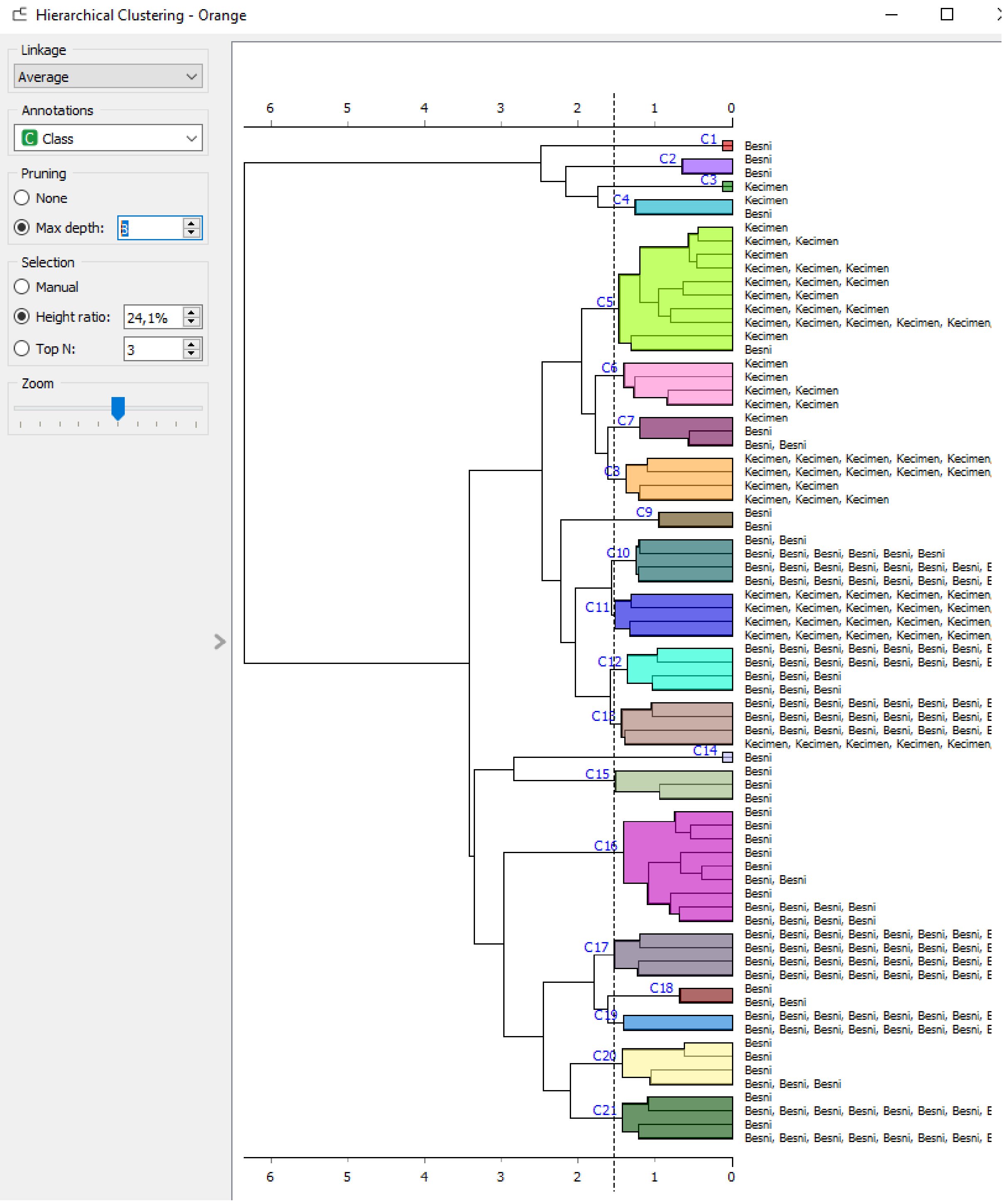
1. eksperiments (Linkage: single, max depth: 20, height ratio: 51.6%):



9. attēls

Klasterizējot ar īsākā (single) attāluma saistīšanas metodi (ar Height Ratio 51.9%) redzams, ka abām rozīņu šķirnēm ir izveidojušies divi lieli klasteri C9 un C10, kā arī mazāki vairāki klasteri. Katrs iekrāsotais klasteris satur tikai vienu šķirni. Kopā ir 10 klasteri.

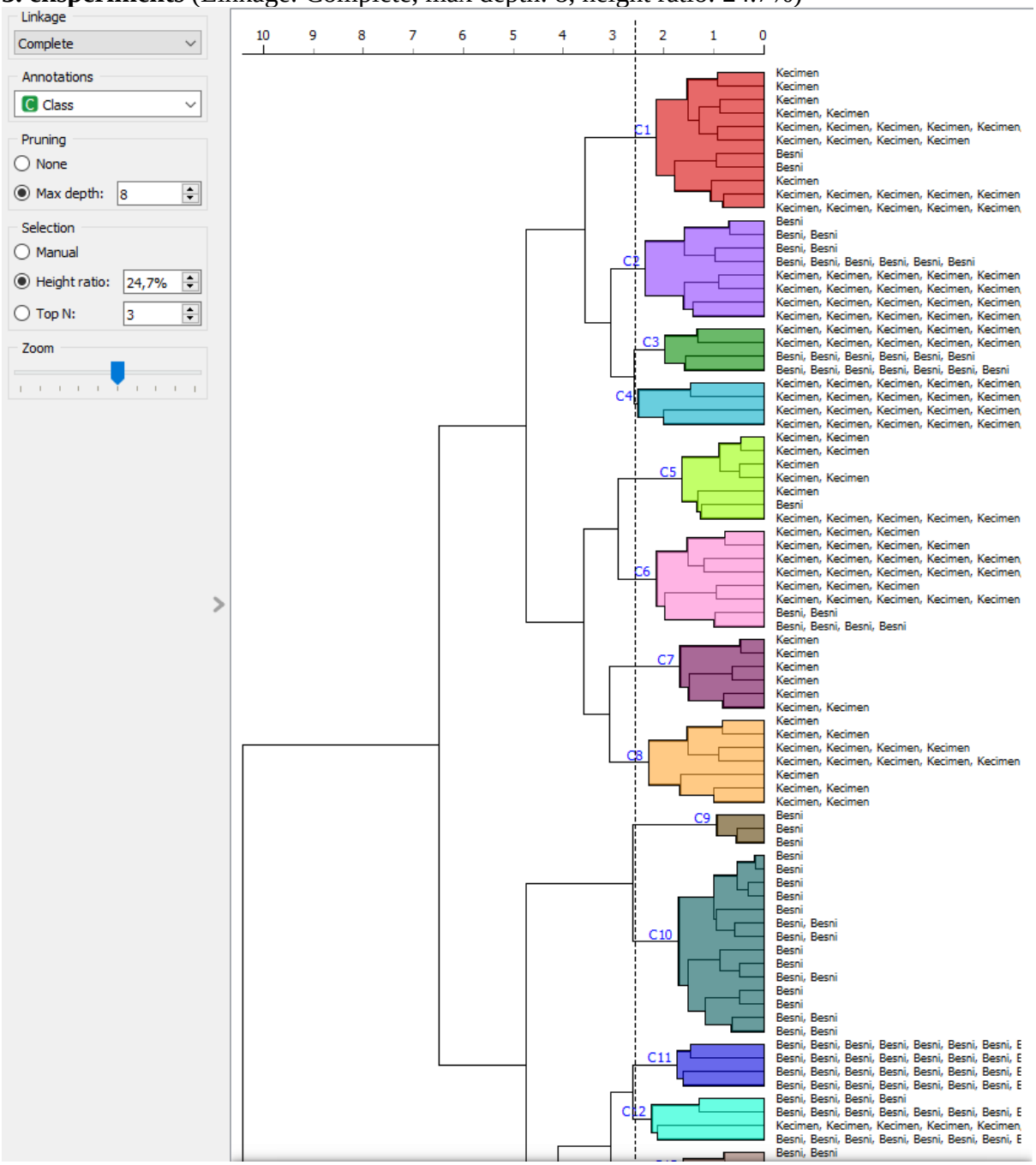
2. eksperiments (Linkage: Average, max depth: 8, Height ratio: 24.1%)**:**



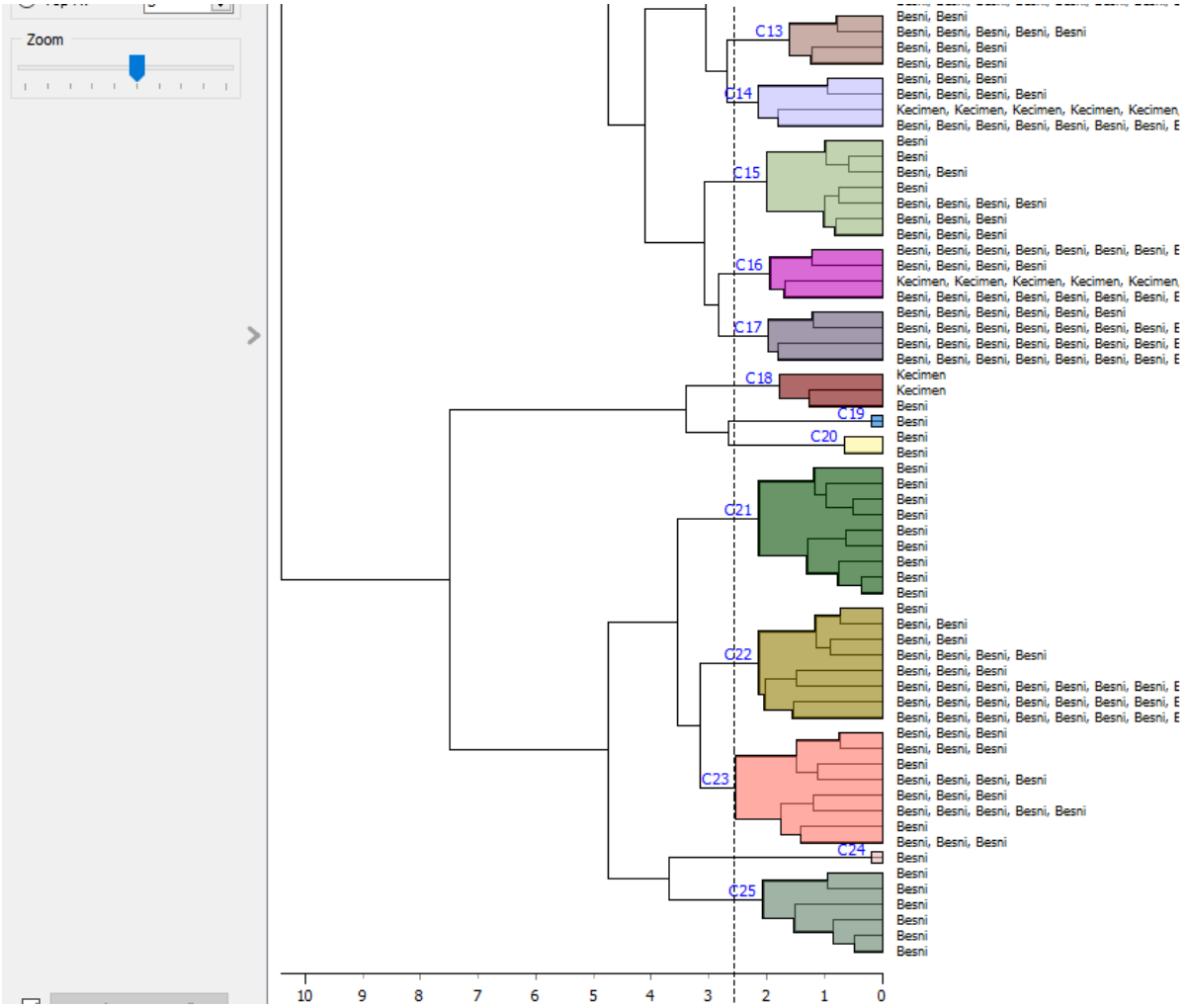
10. attēls

Klasterizējot ar vidēja attāluma saistīšanas metodi (ar Height Ratio 24.1%) redzams, ka ir izveidojušies daudz klāsteru. Trīs klasteri satur abas šķirnes – C5, kur ir viens Besni šķirnes, C13, kur ir Kecimen šķirnes un C7, kur ir viens Kecimen. Kopā ir 21 klasteris.

3. eksperiments (Linkage: Complete, max depth: 8, height ratio: 24.7%)



11. attēls – 3. eksperimenta attēla 1. daļa



12. attēls – 3. eksperimenta attēla 2. daļa

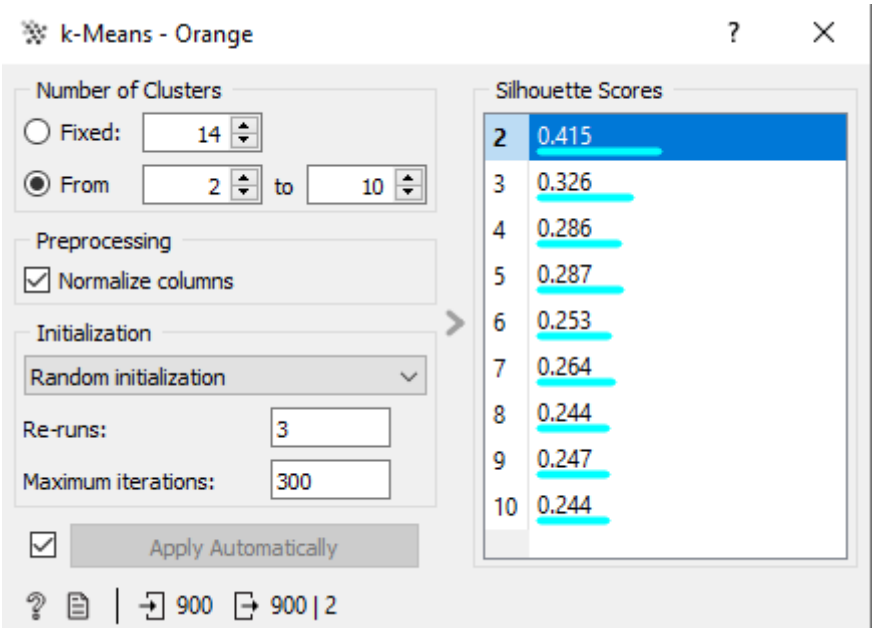
Ar tālāko saistīšanas metodi kopā ir 25 klasteri, daudzos klasteros ir abas klases, paliek arvien grūtāk nodalīt klases pa klasteriem. Visgrūtāk ir izdalīt Kecimen, jo daudzi klases Kecimen datu objekti atrodas klāsteros ar iejauktiem Besni datu objektiem.

Secinājumi (par hierarhisko klasterizāciju)

Visos eksperimentos vislabāk tika klasterizēta Besni šķirne. Kecimen klasteros bieži parādās datu objekti no Besni šķirnes. Vislabākā saistīšanas metode priekš hierarhiskas klasterizācijas šai datu kopai ir īsākā saistīšanas metode – tā izveidoja lielākos vienas šķirnes klasterus, veiksmīgi atdalot klases vienu no otras.

K-vidējo algoritms

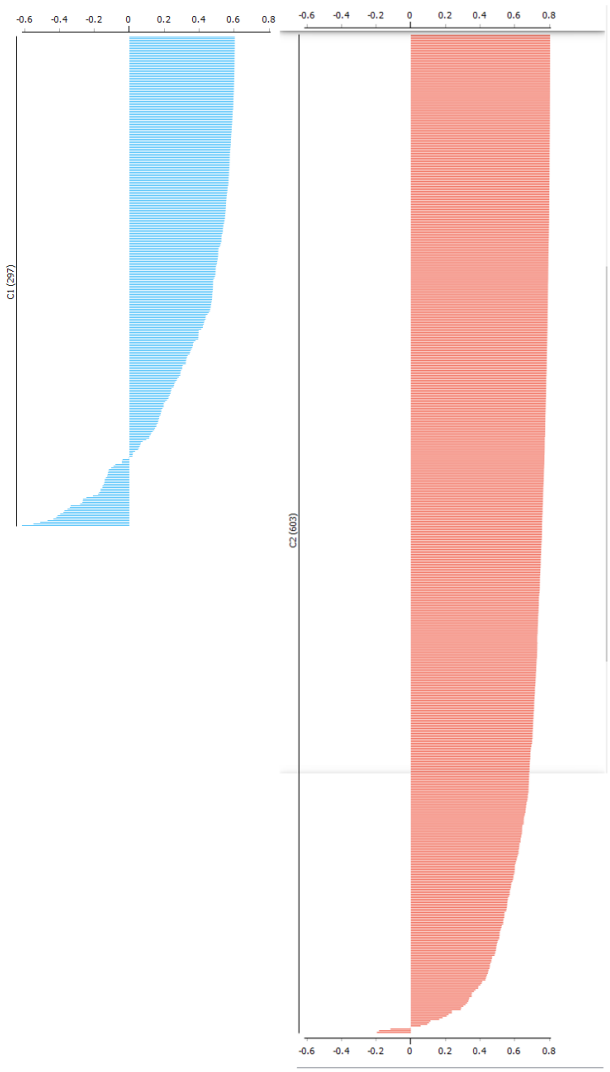
Hiperparametri – klasteru skaits[4], klasteros tiks izdalīti pēc iespējas līdzīgi datu objekti[5].



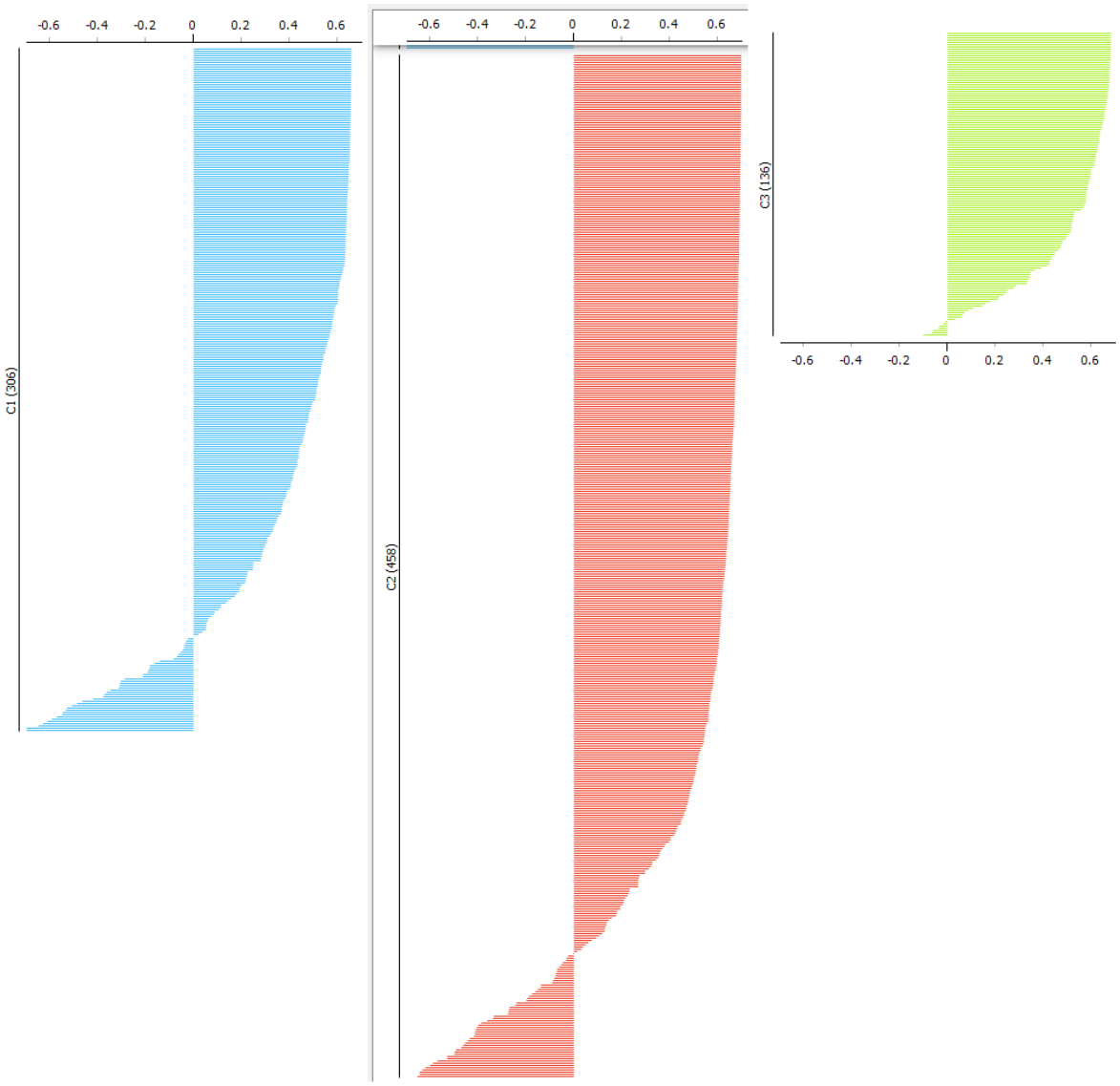
13. attēls

Vislabākais siluetu rezultāts (0.415) ir ar 2 klāsteriem (12. attēls), kas šķiet loģiski, par cik ir 2 klases.

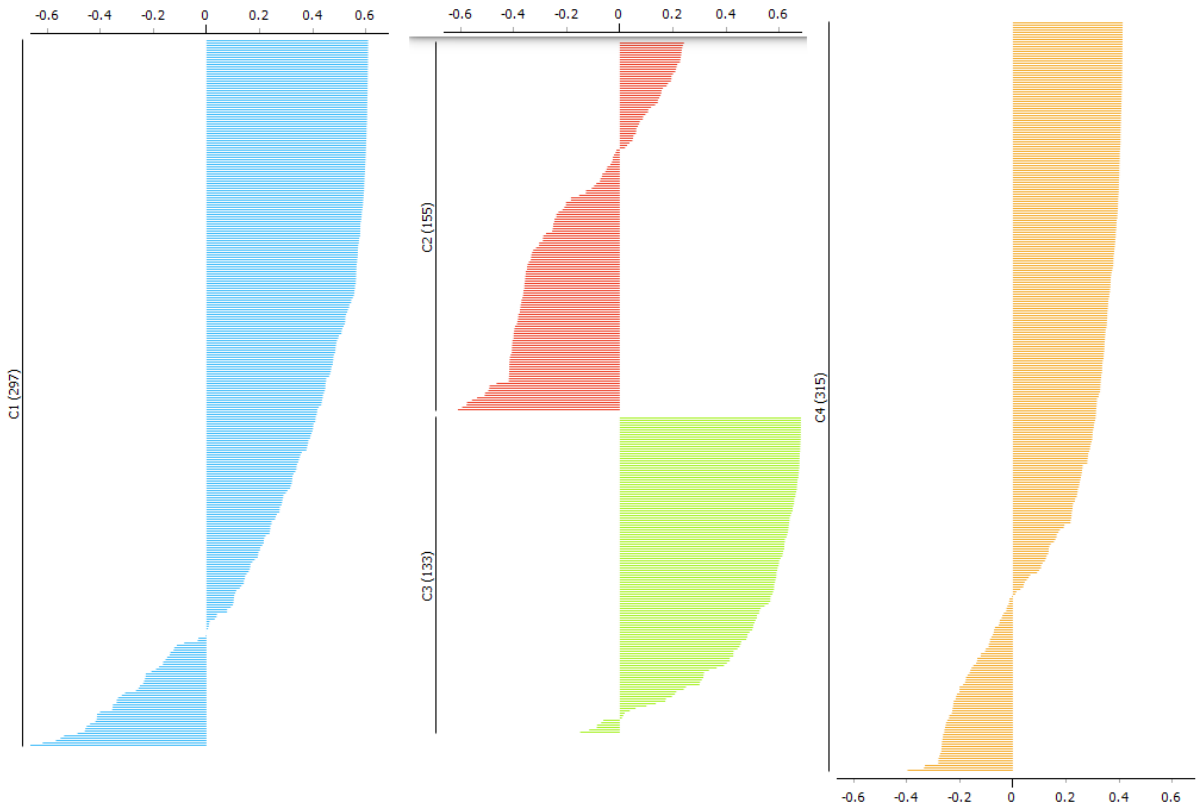
[4] https://orangedatamining.com/widget-catalog/unsupervised/kmeans/
[5] https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1



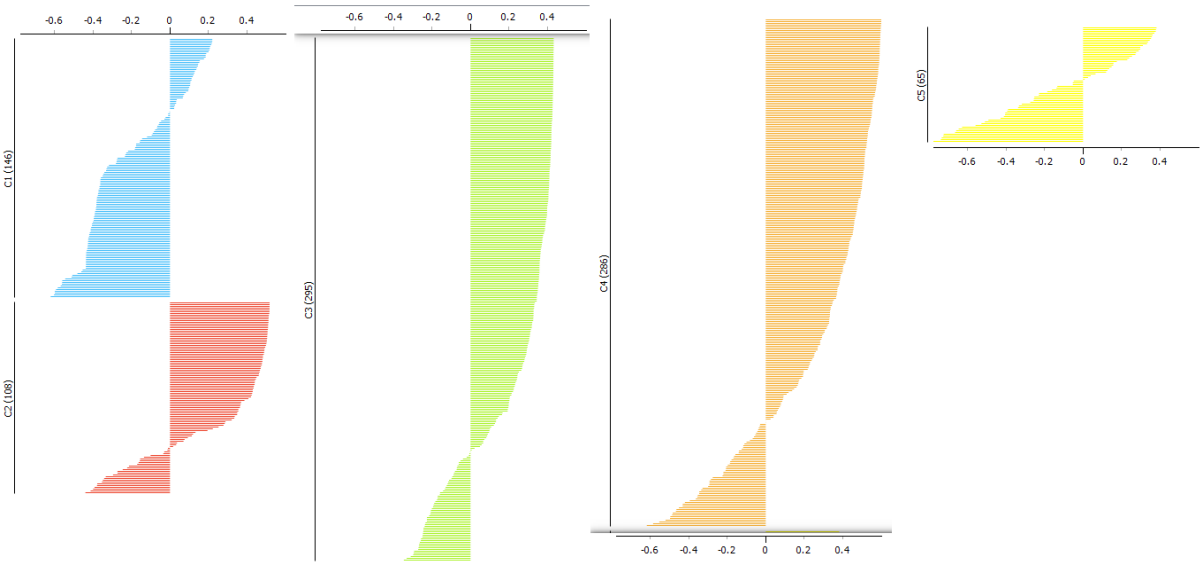
14. attēls - silueta sadalījums pie 2 klasteriem



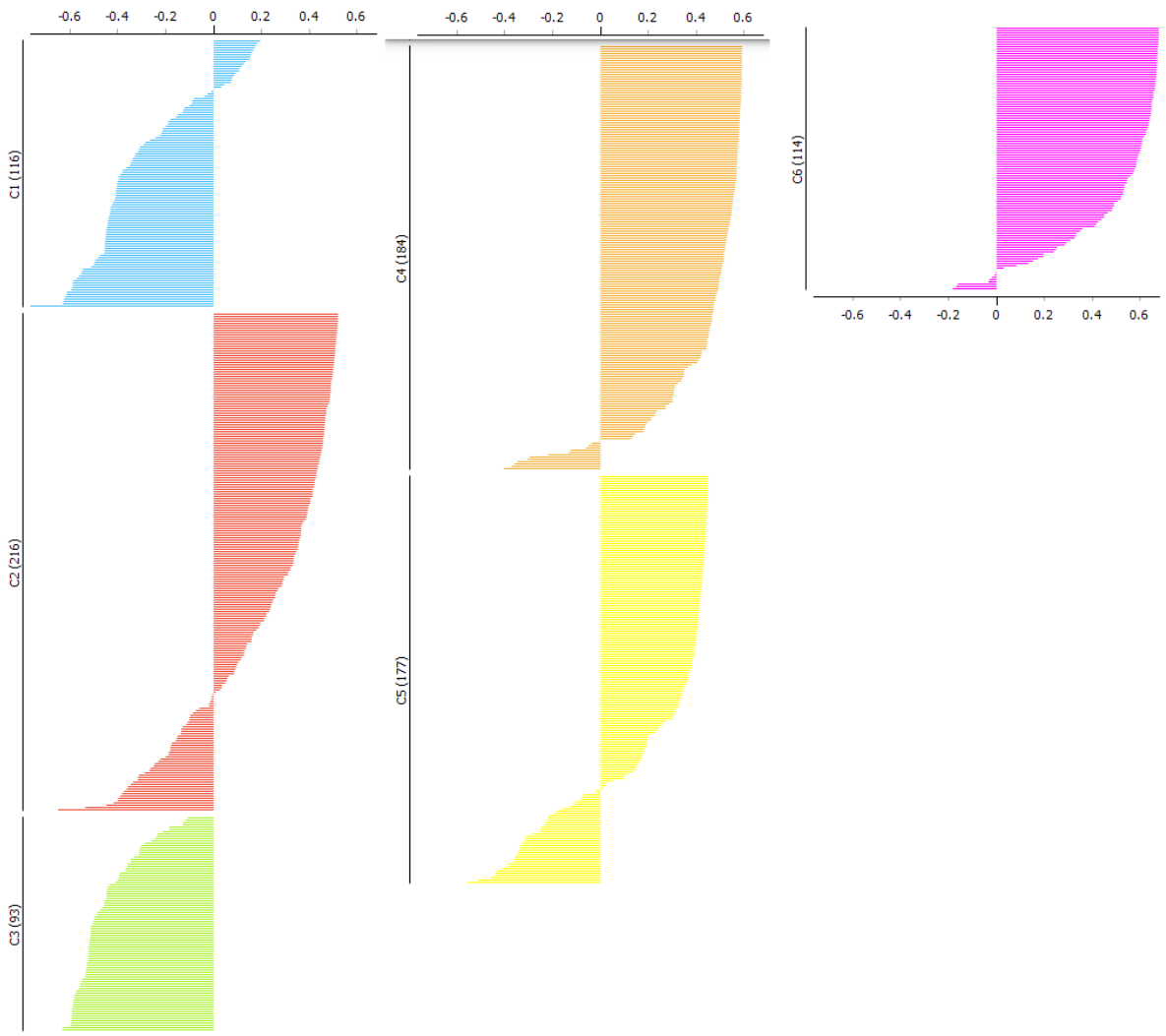
15. attēls - silueta sadalījums pie 3. klasteriem



16. attēls - silueta sadalījums pie 4 klasteriem



17. attēls - silueta sadalījums pie 5 klasteriem



18. attēls - silueta sadalījums pie 6 klasteriem

17. attēlā - zaļais klasteris ir vispār nederīgs – tas ir neatdalāms.

Skatoties siluetu sadalījumus (13. – 17. attēls) redzams, ka, jo vairāk klasteru, jo biežāk klasteros veidojas izņēmumi (outliers) un klasteri kļūst grūtāk atdalāmi. 17. attēlā redzams, ka visu datu objektu siluetu vērtības ir negatīvas, tā klastera datu objekti ir ļoti grūti atdalāmi, savukārt tajā pašā silueta sadalījumā, klasteris C6 ir ļoti labi atdalāms. 13. attēlā, pie klasteru skaita 2, ir labs sadalījums un ir maz negatīvo vērtību.

2. daļas secinājumi:

Datu kopā esošās klases ir dāļēji atdalāmas. Daudzus Kecimen rozīnes šķirnes datu objektus ir grūti atšķirt no Besni, kamēr Besni šķirne ir labāk izdalāma. Kopumā klases nedaudz pārklājas, taču tik un tā ir iespējams tās izdalīt vienu no otras.

III. daļa

Izmantotie pārraudzītās mašīnmācīšanās algoritmi: mākslīgie neironu tīkli (*Neural Network*), loģistiskā regresija un kNN.

Loģistiskā regresija – novērtē notikuma iespējamību balstoties uz neatkarīgiem mainīgajiem[6]. Visbiežāk izmanto, lai modelētu bināru iznākumu[7].

Izvēlējos, jo datu kopā ir tikai divas klases un loģistiskā regresija labi strādā ar bināriem klasifikāciju uzdevumiem.

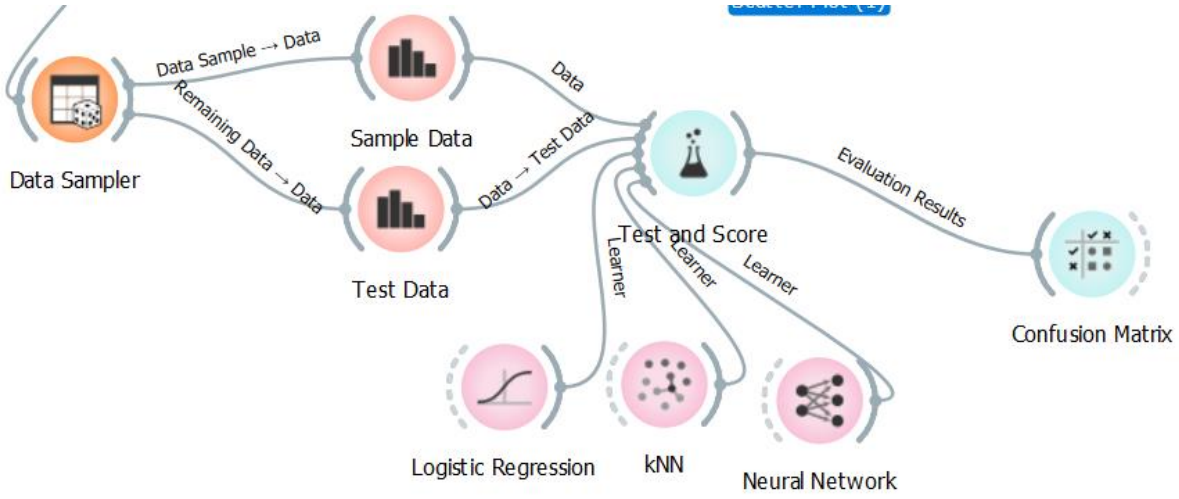
Hiperparametru nav.

kNN – izmanto tuvumu, lai veiktu klasifikāciju. Var izmantot gan regresijas, gan klasifikācijas uzdevumiem, taču visbiežāk tiek izmantots kā klasifikācijas algoritms. Balstās uz domu, ka līdzīgi dati būs tuvu viens otram[8].

Izvēlējos, jo datu kopā ir daudz datu objektu, kurus var atdalīt pēc klases un datu objekti ir tuvu viens otram.

Hiperparametrs – kaimiņu, kuru ir jāapskata, skaits[9].

Neironu tīklu hiperparametri – neironu skaits apslēptajos slāņos un aktivizācijas funkcija. Neironu skaits apzīmē to, cik daudz neironu ir apslēptajā slānī un aktivizācijas funkcija nosaka, vai neirons tiks aktivizēts[10].



19. attēls

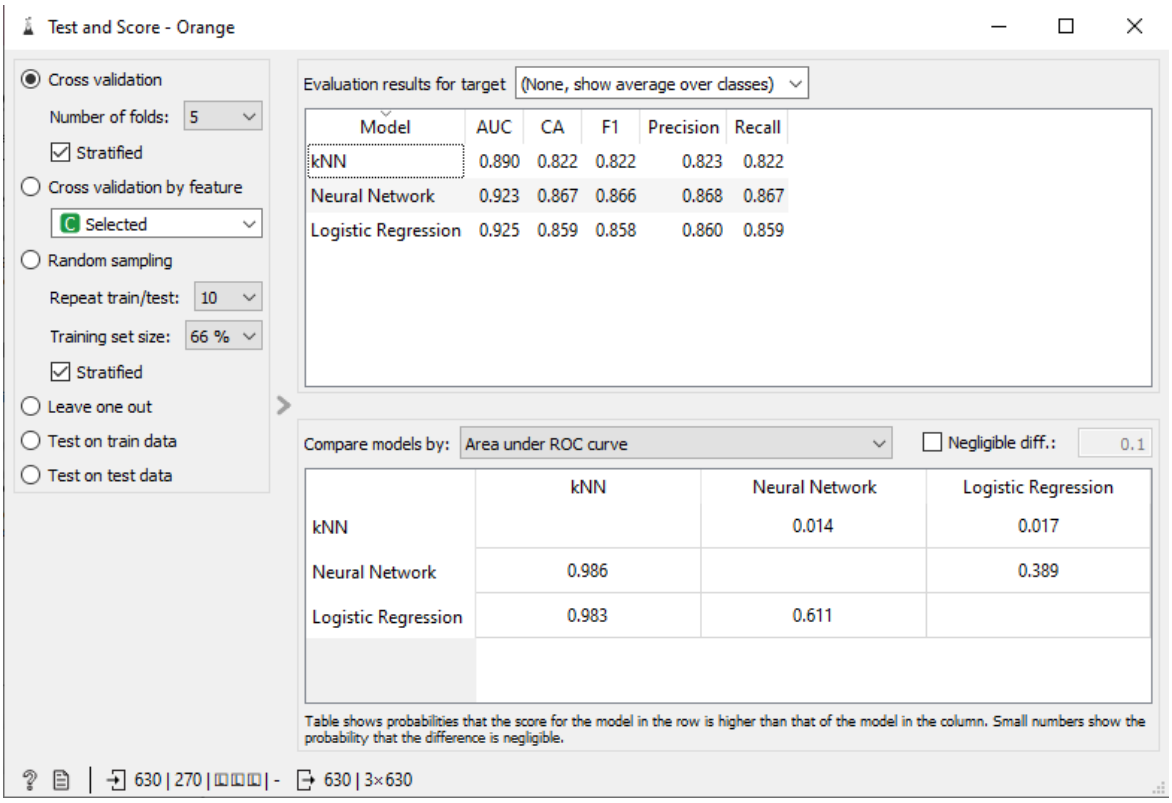
18. attēlā redzams, ka no Data Sampler izvēlos datus, kuri būs apmācību datu kopā un kuri būs testa datu kopā. Sadalījums ir 70-30, t.i. 70% datu no data sampler būs apmācību datu kopā, un 30% būs testa datu kopā.

---

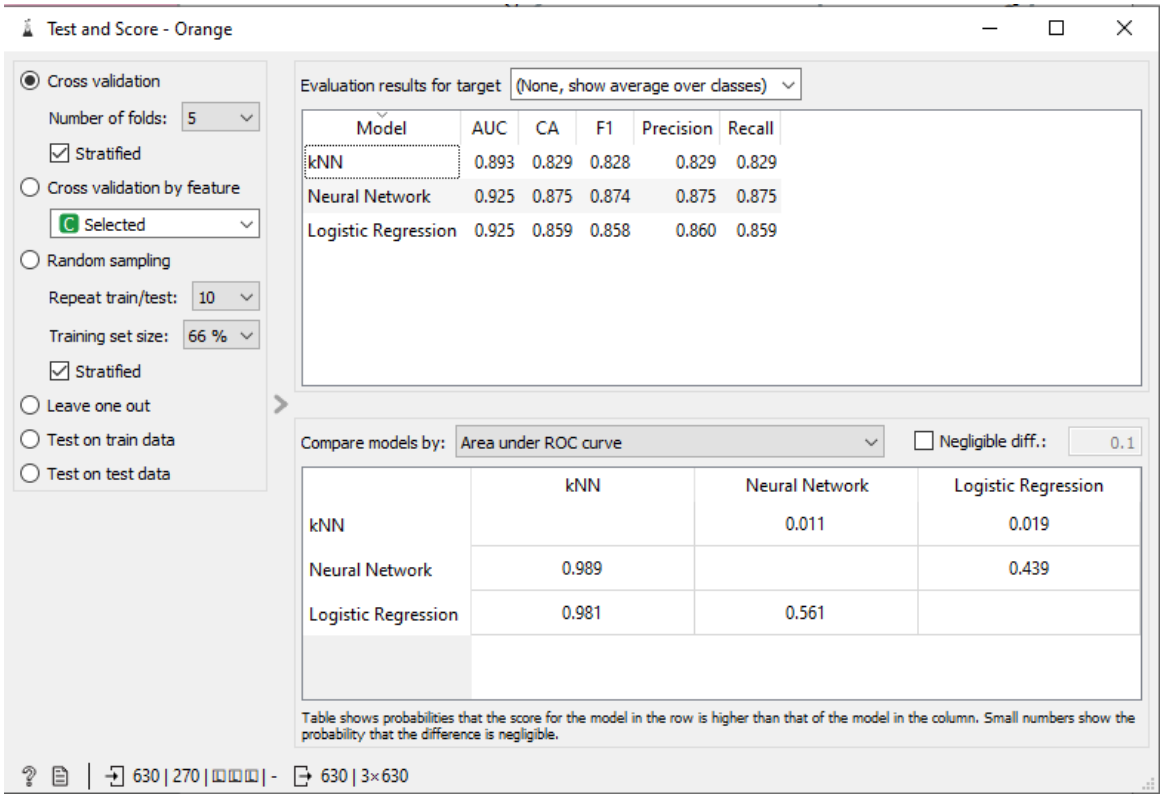
[6] https://www.ibm.com/topics/logistic-regression
[7] https://www.sciencedirect.com/topics/computer-science/logistic-regression#:~:text=Logistic%20regression%20is%20a%20process,%2Fno%2C%20and%20so%20on.
[8] https://www.ibm.com/topics/knn
[9] https://openclassrooms.com/en/courses/6401081-improve-the-performance-of-a-machine-learning-model/6559796-tune-your-hyperparameters
[10] https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/neuralnetwork.html

Apmācību datu kopā ir 630 datu objekti – 304 Besni klasei un 326 Kecimen klasei.

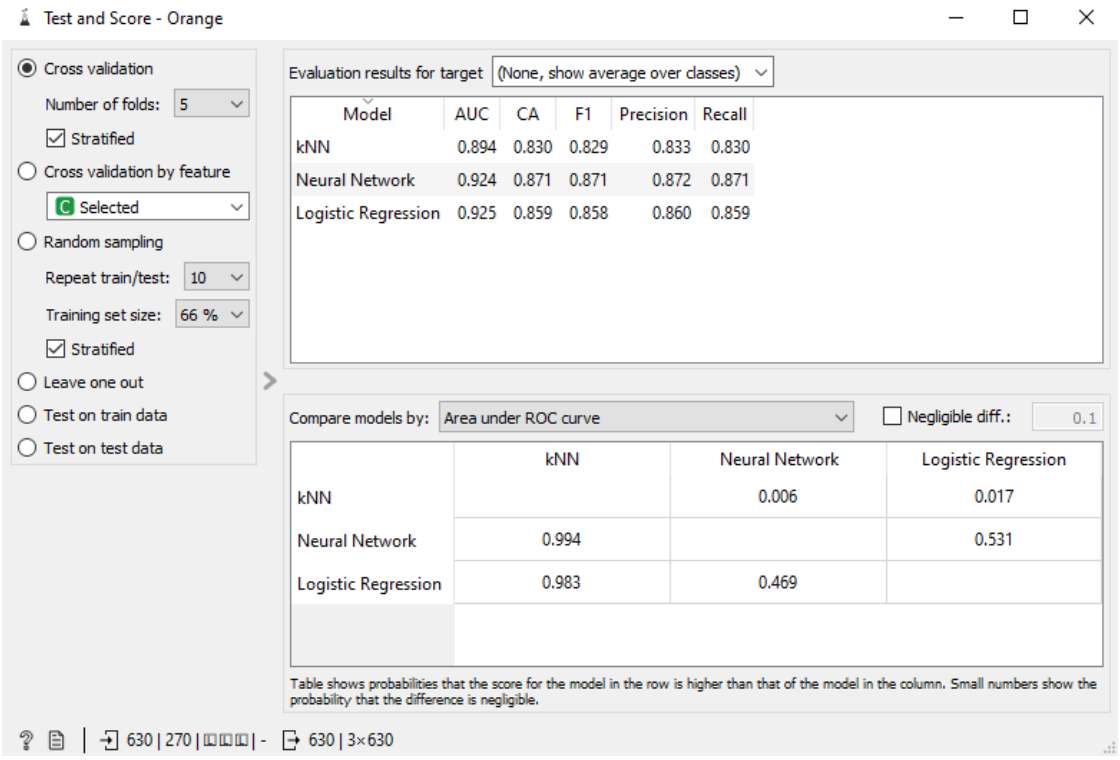
Testa datu kopā ir 270 datu objekti – 146 Besni klasei un 124 Kecimen klasei.



20. attēls – 1. eksperiments



21. attēls – 2. eksperiments



22. attēls – 3. eksperiments

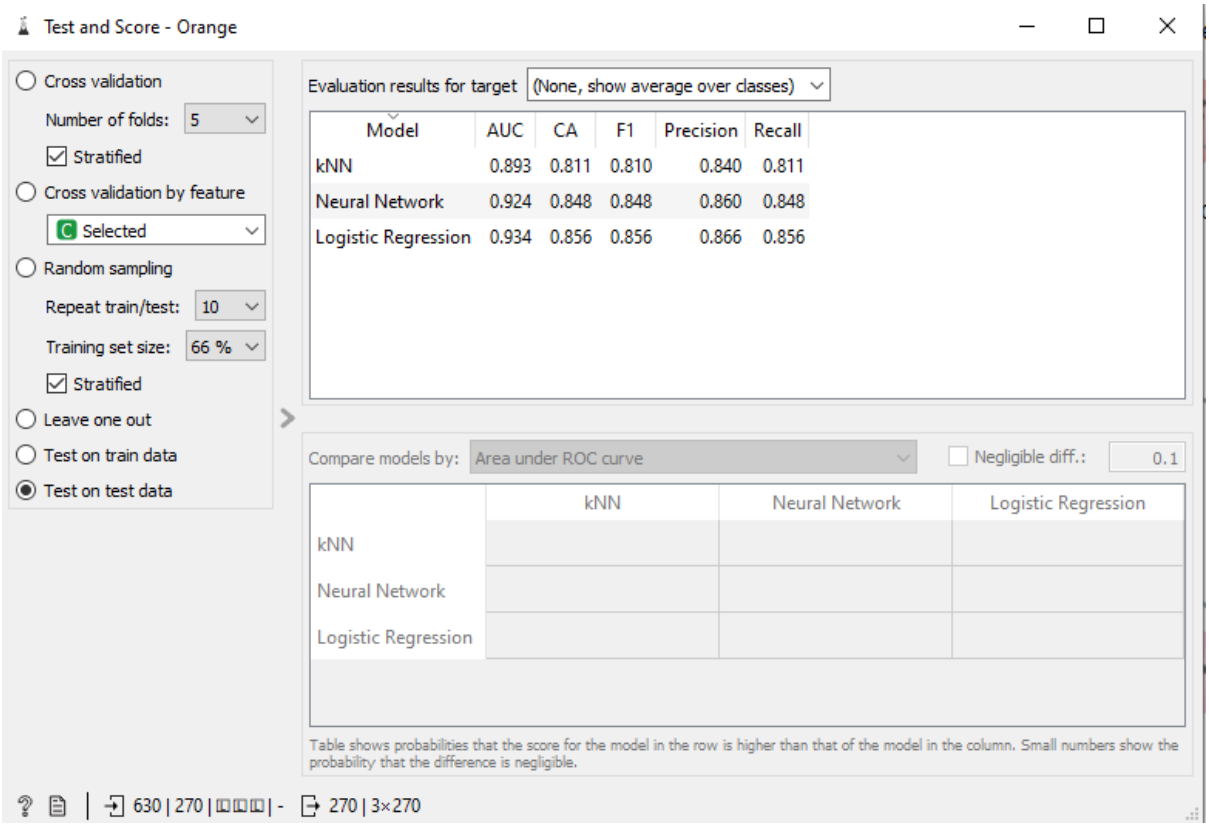
Visos eksperimentos parametra "neironu skaits apslēptajā slānī" vērtība ir 100.

Algoritms	Eksperimenta nr.	Hiperparametrs	Hiperparametra vērtība	Klasifikācijas precizitāte (CA)
Neironu tīkli	1.	Aktivizācijas funkcija	ReLu	0.867
	2.		Identity	0.875
	3.		tanh	0.871
Loģistiskā regresija	1.	-	-	0.859
kNN	1.	Apskatāmie kaimiņi	5	0.822
	2.		10	0.829
	3.		15	0.830

4. tabula – eksperimentu rezultāti ar algoritmu hiperparametriem

Labākais modelis katram algoritmam, kas tiks izmantots testēšanā:

- Neironu tīkli – aktivizācijas funkcija: Identity
- Loģistiskā regresija – nemainās
- kNN – apskatāmo kaimiņu skaits: 15



23. attēls – testēšanas rezultāti

Veicot testēšanu, visiem algoritmiem klasifikācijas precizitāte (CA) samazinājās. Vistuvāk precizitātei, ko ieguva ar apmācības datu kopu, bija loģistiskā regresija.

Neirona tīklu precizitāte samazinājās par 0.027, kNN par 0.019 un loģistiskā regresija – par 0.003. Apmācot datus, labākais (precīzākais) algoritms bija neironu tīkli, taču veicot testēšanu, loģistiskā regresija kļuva par labāko algoritmu, kā arī loģistiskā regresija testa rezultāti, salīdzinot ar apmācības rezultātiem, zaudēja vismazāk punktus, kas liecina par algoritma konsistenci. Iespējams, ka kNN un neirona tīkli labāk strādā ar lielāku skaitu datu objektu.

Izmantota literatūra

1. https://dergipark.org.tr/tr/download/article-file/1227592

2. https://www.kaggle.com/datasets/muratkokludataset/raisin-dataset

3. https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/hierarchicalclustering.html

4. https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/neuralnetwork.html

5. https://www.ibm.com/topics/logistic-regression

6. https://www.sciencedirect.com/topics/computer-science/logistic-regression#:~:text=Logistic%20regression%20is%20a%20process,%2Fno%2C%20and%20so%20on

7. https://www.ibm.com/topics/knn

8. https://openclassrooms.com/en/courses/6401081-improve-the-performance-of-a-machine-learning-model/6559796-tune-your-hyperparameters

9. https://www.datasciencesmachinelearning.com/2019/10/hierarchical-and-k-means-cluster.html

10. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1