RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Informācijas tehnoloģijas institūts

2. praktiskais darbs studiju kursā

“Mākslīgā intelekta pamati”

Mašīnmācīšanās algoritmu lietojums

Izstrādāja: Deniss Ponomarenko 3. IT grupa 201rdb407

Saite uz darba projekta (resursiem):

2023./23. studiju gads

SATURA RĀDĪTĀJS

DATU KOPAS APRAKSTS

1. **Datu kopas nosaukums**: Wine Quality Dataset;
2. **Datu kopas autors**: M Yasser H;
3. **Datu kopas izdošanas laiks:** 2022. gads;
4. **Licencēšanas nosacījumi**: CC0: Public Domain. Šīs datu kopas izveidotājs ir atteicies no autortiesībām, ikvienam, kas izmanto šo dokumentu, ir tiesības kopēt, izplatīt, rediģēt un izmantot šo informāciju komerciāliem nolūkiem.
5. **Problēmsfēras apraksts**: Šī datu kopa apraksta vīna kvalitāti no 0 līdz 10, un tai ir tikai statistiskie dati par dzēriena sastāvu. Minimālais vērtējums vīns šajā datu kopā ir vērtējums 3, bet maksimālā vērtība ir 8. Šeit nav informācijas par vīnogu ražotāju, cenu, šķirnēm un citiem ārējiem faktoriem, datu kopu veido ķīmisko elementu, piemēram, spirta, cukura, skābes, pH līmeņa utt., novērtēšana;
6. **Veids, kā datu kopa tika savākta**: Nav zinams.

Datu kopas satura apraksts

1. Objektu skaits datu kopā: 1143;
2. Klašu skaits datu kopā: Šajā datu kopā ir tikai viena klase, tā ir vīna kvalitāte. Šīs datu kopas aprakstā teikts, ka kvalitāte tiek novērtēta no 0 līdz 10, taču praksē pastāv ieraksti ar kvalitāti no 3 līdz 8;
3. Objektu skaits, kas pieder katrai klasei: Tā kā klase šajā datu kopā ir tikai viena, tālāk aprakstīts ierakstu skaits katram kvalitātes novērtējumam (no 3 līdz 8):
    3.1. 3 – 6 ieraksti;
    3.2. 4 – 33 ieraksti;
    3.3. 5 – 483 ieraksti;
    3.4. 6 – 462 ieraksti;
    3.5. 7 – 143 ieraksti;
    3.6. 8 – 16 ieraksti.
4. Pazīmes (parametri): Datu kopa sastāv no 13 parametriem, detalizētāka informācija aprakstīta 1.1. tabulā.

1.1. tablula

Parametru apraksts

Nr.	Nosaukums	Datu tips	Vērtību diapazons	Nozīme
1.	quality	Skaitlisks	3-8	vīna kvalitātes novērtējums
2.	fixed acidity		4.6-15-9	fiksētais skābums
3.	volatile acidity		0.120-1.580	gaistošais skābums
4.	citric acid		0-1	citronskābe
5.	residual sugar		0.9-15.5	atlikušais cukurs
6.	chlorides		0.012-0.611	hlorīdi
7.	free sulfur dioxide		1-68	brīvais sēra dioksīds
8.	total sulfur dioxide		6-289	kopējais sēra dioksīds
9.	density		0.99-1.003	blīvums
10.	pH		2.74-4.01	skābuma rādītājs
11.	sulphates		0.33-2	sulfāti
12.	alcohol		8.4-14.9	spirta procents
13.	Id		0-1597	ieraksta numurs

5. **Datu faila struktūras fragments**:

	quality	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	Id
104	4	9.2	0.520	1.00	3.40	0.61	32.0	69.0	0.9996	2.74	2.00	9.4	151
389	6	14.3	0.310	0.74	1.80	0.075	6.0	15.0	1.0008	2.86	0.79	8.4	544
1048	5	10.0	0.690	0.11	1.40	0.084	8.0	24.0	0.99578	2.88	0.47	9.7	1470
311	8	12.6	0.310	0.72	2.20	0.072	6.0	29.0	0.9987	2.88	0.82	9.8	440
715	6	8.0	0.180	0.37	0.90	0.049	36.0	109.0	0.99007	2.89	0.44	12.7	1018
467	5	10.7	0.430	0.39	2.20	0.106	8.0	32.0	0.9986	2.89	0.50	9.6	656
462	5	10.7	0.430	0.39	2.20	0.106	8.0	32.0	0.9986	2.89	0.50	9.6	650
935	6	9.1	0.760	0.68	1.70	0.414	18.0	64.0	0.99652	2.90	1.33	9.1	1319

1.att. Datu kopas parametri

ORANGE RĪKA DARBA GAITA

Šajā nodaļā aprakstīta datu kopa tālākai praktiskai lietošanai, izpētei un analīzei.

1. Lai strādātu ar Orange rīku, bija nepieciešams to lejupielādēt no oficiālās vietnes un instalēt savā datorā.
2. Tālāk jāizveido jauns projekts, sadaļā Menu noklikšķinot uz pogas New (Menu - > New), pēc tam veidojas jauns projekts, kurā var veikt datu un algoritmu darbības analīzi.
3. Lai ielādētu datu kopu Orange instancē, ir jāpievieno File (Data - > File) un jānorāda nepieciešamais fails ar datu kopu. To var izdarīt, izmantojot tīmekļa adresi (URL), kur atrodas datu kopa vai izvēlēties konkrētu failu datorā. Šī darba autors ir izvēlējies datu kopu, kas apraksta vīna kvalitāti pēc vairākiem parametriem, kura autors ir M YASSER H. Saiti uz šo datu kopu var atrast avotu sarakstā darba beigās.
4. Datu kopas aprakstam un analīzei ir nepieciešama datu kopas modifikācija. Autors Quality sleju norādījis kā mērķi, jo vīna dalījuma kvalitāte veidojas no visiem pārējiem ieraksta parametriem. Visu atlasītās datu kopas iestatījumu sarakstu sk. 1.1. attēlā

Info
1143 instances
13 features (no missing values)
Data has no target variable.
0 meta attributes

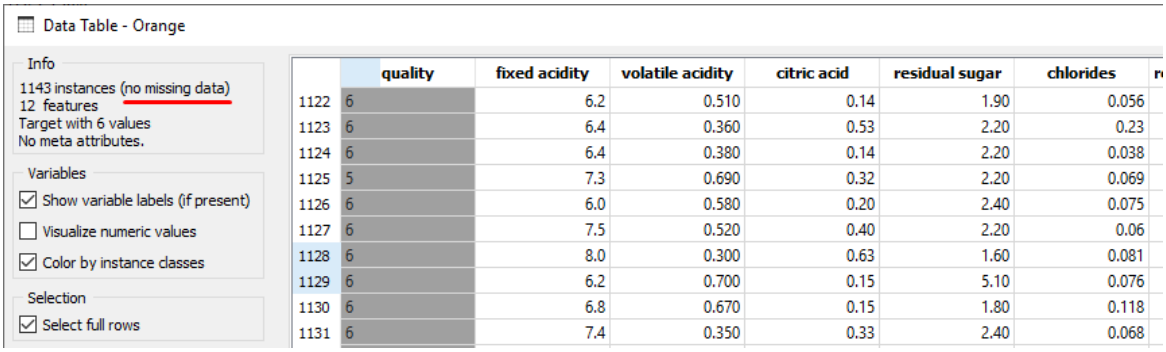
Columns (Double click to edit)

	Name	Type	Role	Values
1	fixed acidity	numeric	feature	
2	volatile acidity	numeric	feature	
3	citric acid	numeric	feature	
4	residual sugar	numeric	feature	
5	chlorides	numeric	feature	
6	free sulfur dioxide	numeric	feature	
7	total sulfur dioxide	numeric	feature	
8	density	numeric	feature	
9	pH	numeric	feature	
10	sulphates	numeric	feature	
11	alcohol	numeric	feature	
12	**quality**	**categorical**	**target**	
13	Id	numeric	feature	

1.att. Datu kopas parametri

5. Tika pievienots Table elements (Data - > Table), lai redzētu datus Orange rīkā. Elements Table tiek savienots ar elementu File un tad var redzēt datus. Skatot

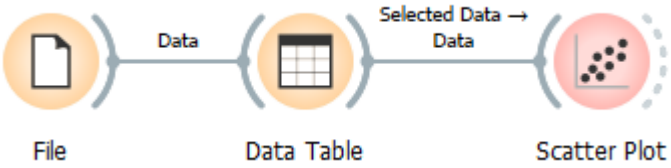
datus, netika atrasti ieraksti ar trūkstošo informāciju, tāpēc manipulācijas ar datu aizpildīšanu nav jāveic (sk. 2. att.), var arī redzēt, ka izvēlētajā datu kopā ir 1143 ieraksti un 12 parametri.



	quality	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides
1122	6	6.2	0.510	0.14	1.90	0.056
1123	6	6.4	0.360	0.53	2.20	0.23
1124	6	6.4	0.380	0.14	2.20	0.038
1125	5	7.3	0.690	0.32	2.20	0.069
1126	6	6.0	0.580	0.20	2.40	0.075
1127	6	7.5	0.520	0.40	2.20	0.06
1128	6	8.0	0.300	0.63	1.60	0.081
1129	6	6.2	0.700	0.15	5.10	0.076
1130	6	6.8	0.670	0.15	1.80	0.118
1131	6	7.4	0.350	0.33	2.40	0.068

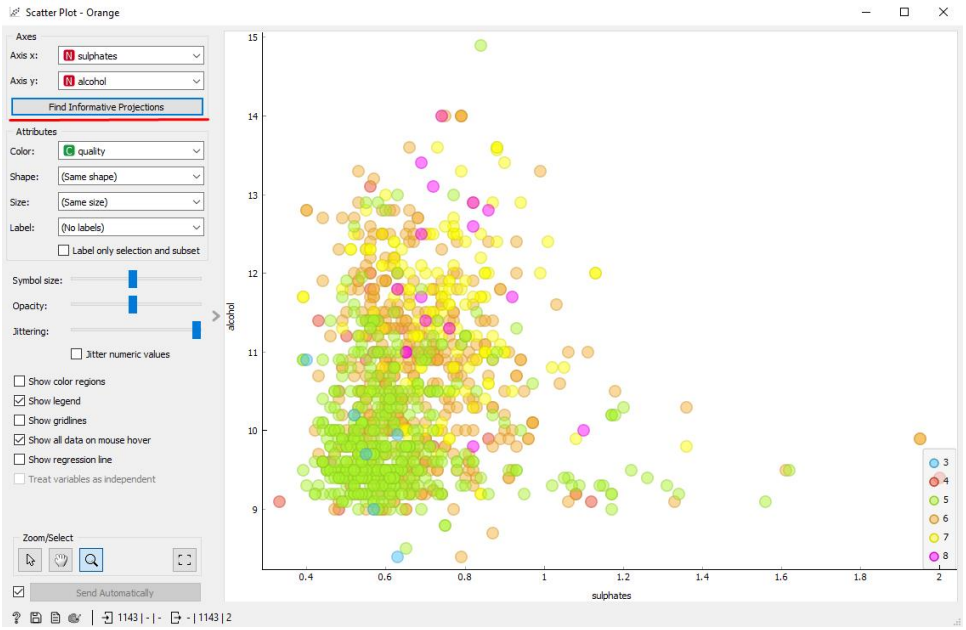
2. att. Table elementā datu kopa

6. Atlasītajai datu kopai ir tikai skaitliska informācija, tāpēc arī šeit nav jāveic datu modifikācija.
7. Tālāk jāsaprot, kā dati var būt savstarpēji saistīti un kādas var būt atkarības. Lai to izdarītu, ir jāpievieno izkliedes grafiks (Visualise - > Scatter plot) un savienot to ar elementu Table, lai augšupielādētu datus grafikā (sk. 3. att.).



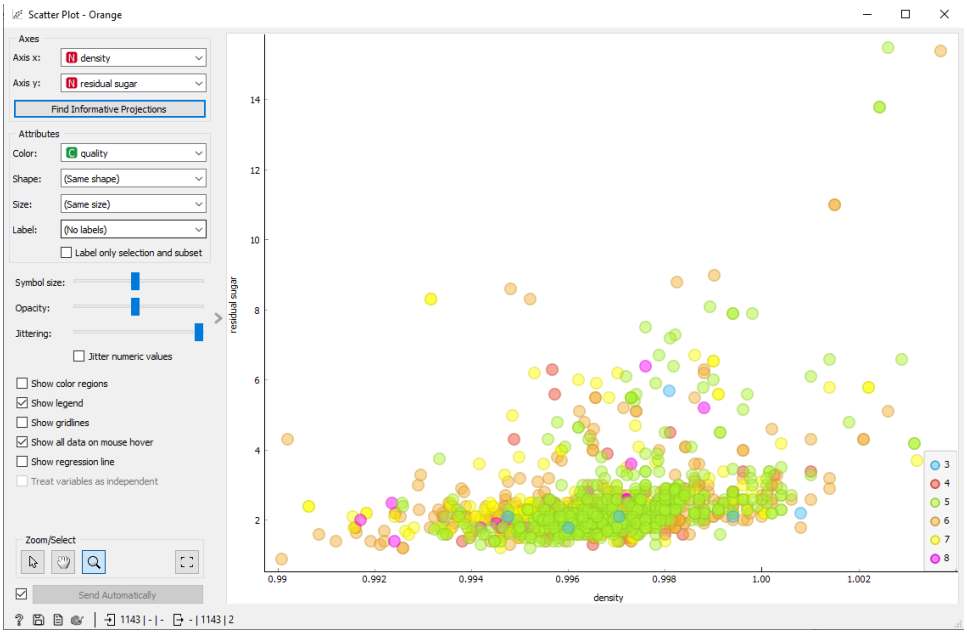
3. att. Projekta diagramma

8. Pirmais tests par iespējamām saiknēm starp hlorīdiem un alkoholu, un tika konstatēts, ka vairumā gadījumu, jo mazāk hlorīdu ir vīnā, jo stiprāks būs dzēriens (sk. 4.att.).



4. att. Pirmais Scatter Plot

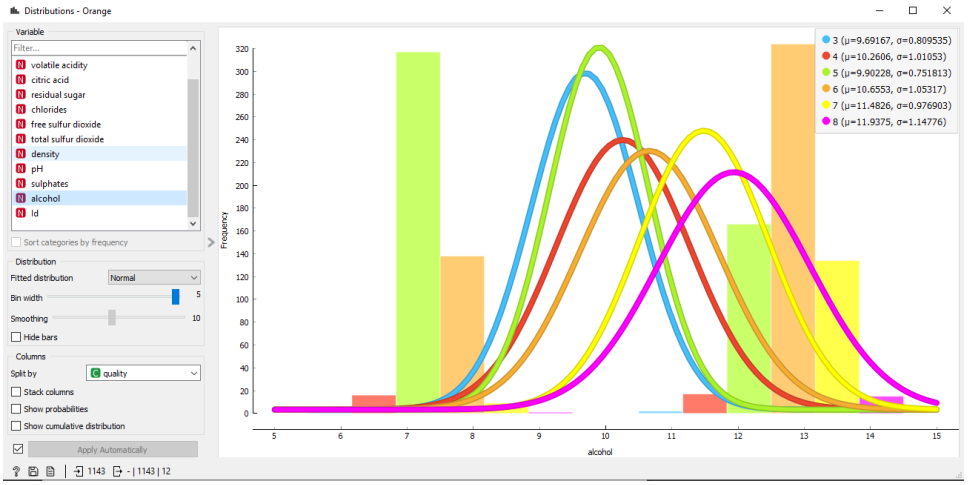
9. Kā otrā pārbaude tika ņemti density un residual Sugar parametri, autors paredz, ka saikne starp šiem abiem parametriem varētu būt šāda: jo vairāk cukura palicis vīnā, jo lielāks ir vīna dzēriena blīvums (sk. 5. att.).



5. att. Otrais Scatter Plot

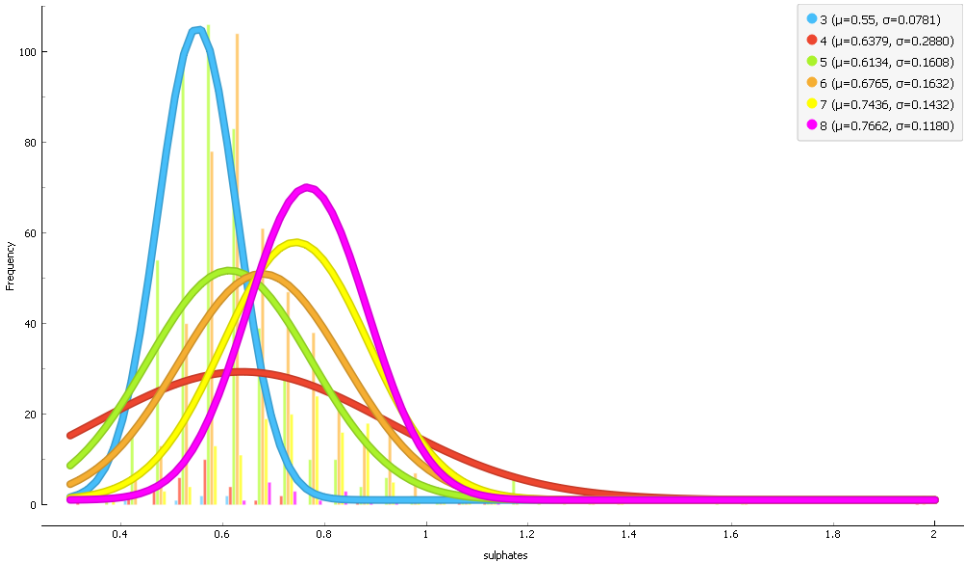
10. Lai parādītu klašu sadalījumu, ir jāpievieno elements Distributions (Visualize - > Distributions) un savienot to ar elementu Table. Split by šūnā tika atlasīta

klase Quality kā rezultējošā vērtība divām pārbaudēm. Pirmajai pārbaudei tika atlasīts parametrs alcohol, klases sadalījums šim parametram ir redzams 6. attēlā. Izvēlēts arī normāls sadalījums (Normal Distribution).



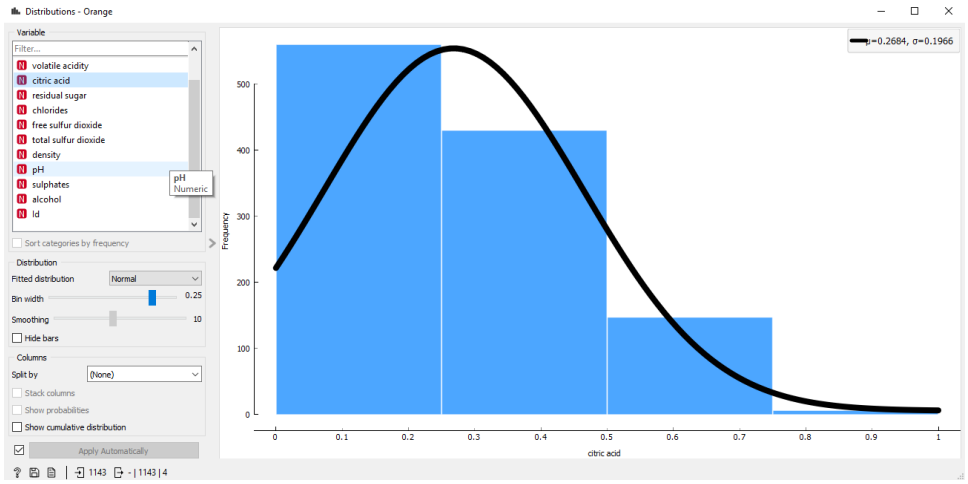
6. att. Pirmā histogramma

11. 7. Attēlā var redzēt klašu sadales atkarība no sulphates parametra.



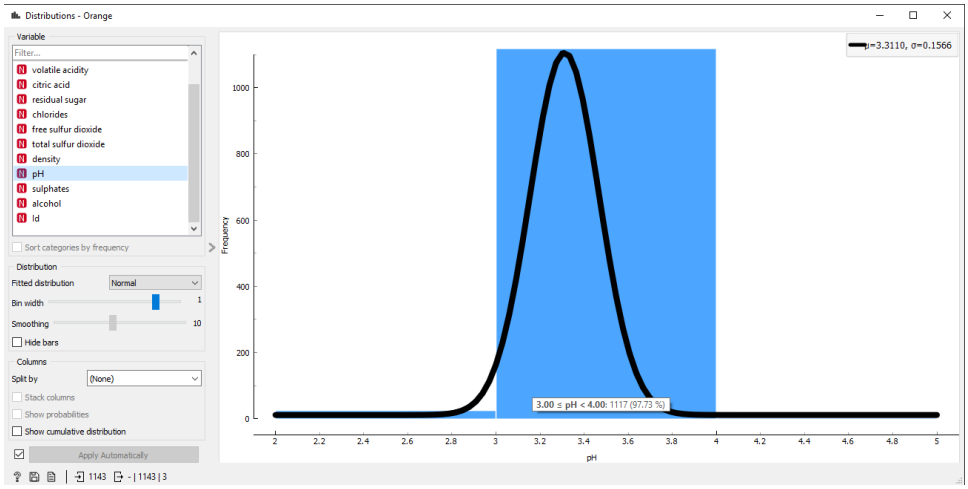
7. att. Otra histogramma

12. Tālāk ir jārāda 2 interesējošo pazīme, lai to izdarītu, šūnā Split by ir jānovāc klase Quality. Pirmā interesējošā pazīme bija Citic acid (sk. 8. att.).



8. att. Citic acid parametra sadalījums

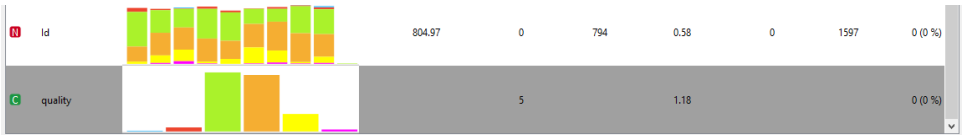
13. Otra interesējošā pazīme bija pH parametrs. 9. attēlā redzams, ka vairāk nekā 97% datu ir pH rādītāji diapazonā no 3 līdz 4 un tikai 2% datu ir pH rādītājs mazāk par 3.



9. att. pH parametra sadalījums

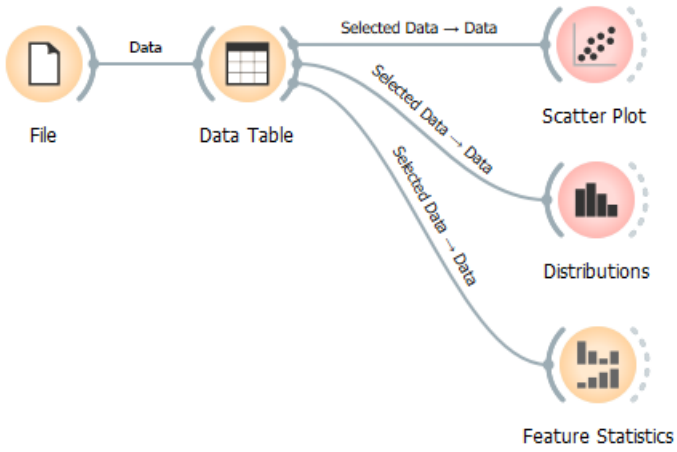
14. Lai analizētu datus, ir jāatrod datu dispersija, vidējā, minimālā un maksimālā vērtība. Lai atrastu šo informāciju, pievienojiet elementu Feature Statistics (Data - > Feature Statistics) un savienot to ar Table elementu. Rezultātā Feature Statistics izvada visu nepieciešamo informāciju katram parametram, modei, mediānai, dispersijai, maksimālajai un minimālajai vērtībai, tukšajiem datiem% un vidējo vērtību (sk. 10. un 11. att.)

10. att. Statistiskie rādītāji (1)



11. att. Statistiskie rādītāji (2)

Pēc šī darba pirmās daļas veikšanas kopējā diagramma ir redzama 12. attēlā.



12. att. Kopējā diagramma I daļa

Secinājumi pēc pirmās darba daļas

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

Izvēlētajā datu kopā var redzēt klašu sadalījumu atkarībā no izvēlētā atribūta, to lieliski parāda histogramma. Piemēram, otrā ekspirementa gaitā ar histogrammu redzams, ka dominē klase Quality ar vērtību 3 (sk. 7. att. Otra histogramma).

2) **Vai datu vizuālais atspoguļojums ļauj redzēt datu struktūru?**

Scatter Plot izmantošana nedod skaidru sadalījumu, ir vērojama neliela loģika par parametru savstarpējo saistību, taču tie ir tikai atsevišķi gadījumi, kas nevar 100% atbildēt uz jautājumu, vai vērtības ir atkarīgas no citiem parametriem. Lielākā daļa datu ir pārāk tuvu viens otram, un gandrīz netiek sadalīti (sk. 5. att. Otrais Scatter Plot).

3) **Cik datu grupējumus ir iespējams identificēt, pētot datu vizuālo atspoguļojumu?**

Ir diezgan grūti identificēt lielu skaitu datu grupējumu, jo grupējumi, rezultējošai vērtībai ir ļoti tuvu cits citam. To var apskatīt 5. un 6. attēlos.

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

Identificētie dati atrodas ļoti tuvu viens otram, to var apskatīt 5. un 6. attēlos.

5) **Secinājumi**:

Pēc izpildītās šī darba pirmās daļas var teikt, ka atlasītajai datu kopai ir visa nepieciešamā informācija, tajā nav tukšu vērtību. Pašas vērtības katram no parametriem ir pirmajā sadaļā norādītā diapazona vidū, piemēram, alcohol vidējā vērtība ir 10.44, kamēr minimālā vērtība ir 8.4, bet maksimālā 14.9.

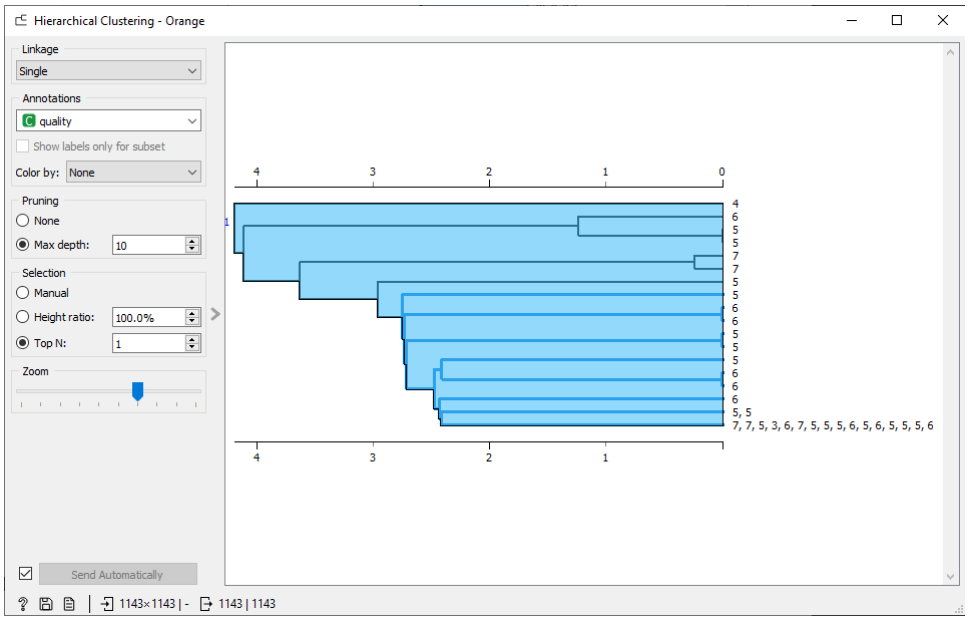
Dispersijas gadījumā tā nav tik viennozīmīga visiem šīs datu kopas parametriem, piemēram, parametram citric acid vidējā vērtība ir 0.2384, taču šī parametra dispersija ir 0.7326, tādējādi var secināt, ka dati ir pietiekami daudz izkliedēti salīdzinājumā ar vidējo vērtību.

NEPĀRRAUDZĪTĀ MAŠĪNMĀCĪŠANĀS

Hierarhiska klasterizacija

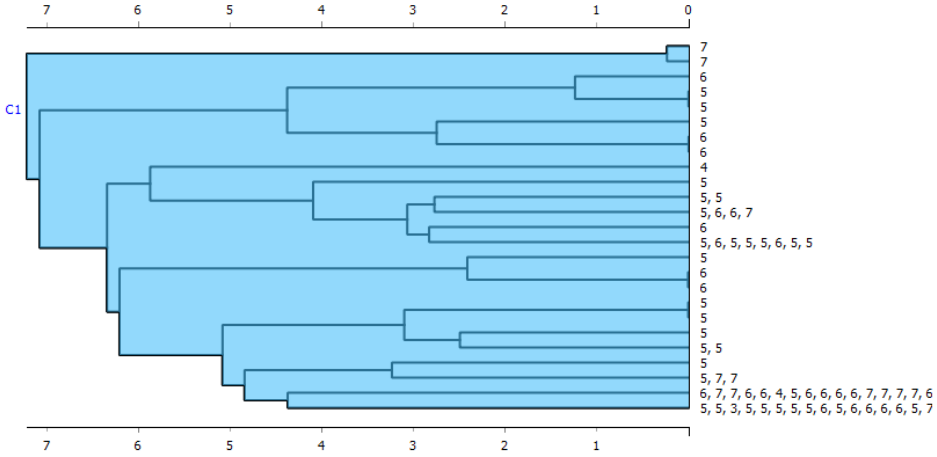
Nepārraudzītās mašīnmācīšanās algoritms ir hierarhiska klasterizacija, kas sadala datu klasterus dažādos līmeņos, lai to izmantotu ar Orange rīku, nepieciešams:

1. Ir jāpievieno divi elementi. Pirmais elements ir distances (Unsupervised - > Distances) un otrais elements Hierarchical Clustering, kas atrodas sadaļā Unsupervised. Nepieciešams arī savienot Distances elementu ar Table un Hierarchical Clustering elementu ar Distances elementu. Bez Distances elementa nevar izmantot Hierarchical Clustering.
2. Hierarchical Clustering elementā pirmajam eksperimentam Linkage šūnā autors izvēlējies vienkāršu sasaisti (Simple linkage). Kā klastera klase tika izvēlēta vienīgā Quality klase (sk. 13. att.). Tika izvēlēta arī atlase pēc Top N = 10.



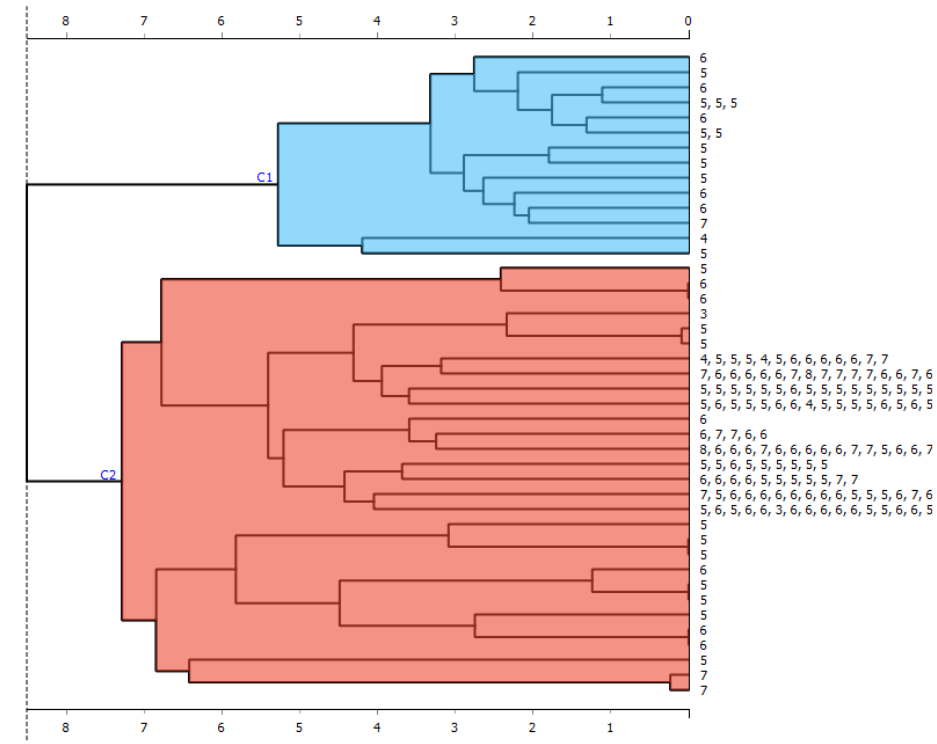
13. att. Single hierarhiskās klasterēšanas eksperiments

3. Hierarchical Clustering elementā otrajam eksperimentam Linkage šūnā autors izvēlējies videjo sasaisti (average linkage). Kā klastera klase tika izvēlēta vienīgā Quality klase (sk. 14. att.). Tika izvēlēta arī atlase pēc Top N = 7.



14. att. Average hierarhiskās klasterēšanas eksperiments

4. Hierarchical Clustering elementā trešajam eksperimentam Linkage šūnā autors izvēlējies svērto sasaisti (weighted linkage). Kā klastera klase tika izvēlēta vienīgā Quality klase (sk. 15. att.). Tika izvēlēta arī atlase pēc augstuma attiecības (height ratio) = 100.0%. Tika izvēlēta arī atlase pēc Top N = 7.



15. att. Weighted hierarhiskās klasterēšanas eksperiments

Algoritmu hiperparametri

Linkage:

- *Single* – jeb vienas saiknes, atrod attālumu starp diviem tuvākajiem klastera elementiem;
- *Average* - aprēķina vidējo attālumu starp divu klasteru elementiem;
- *Weighted* - izmanto WPMA metodi. WPMA ir vienkārša aglomeratīva hierarhiska klasterizācijas metode metode, kas darbojas no apakšas uz augšu.

Annotation - tā parametra izvēle, ar kuru klasteris tiek sadalīts;

Pruning - šī opcija tiek izmantota, lai apgrieztu mērķa diagrammu, tā neietekmē klasteru darbību, bet ir tikai pielāgota funkcija

Selection - ir trīs selekcijas veidi:

- *Manual* - noklikšķinot dendrogrammā, tiks atlasīts klasteris. Vairākus klasterus var atlasīt, turot nospiestu taustiņu Ctrl. Katrs atlasītais klasteris tiek rādīts citā krāsā un izvadē tiek uzskatīts par atsevišķu klasteri;
- *Height ratio* - noklikšķinot uz dendogrammas apakšējā vai augšējā lineāla, grafikā tiek novietota robežlīnija. Tiek atlasīti klasteri pa labi no rinda un katram klasterim būs sava krāsa;
- *Top N* - atlasa augšējo vietņu skaitu.

Informācija ņemta no Orange instrumenta oficiālās dokumentācijas.

Secinājumi

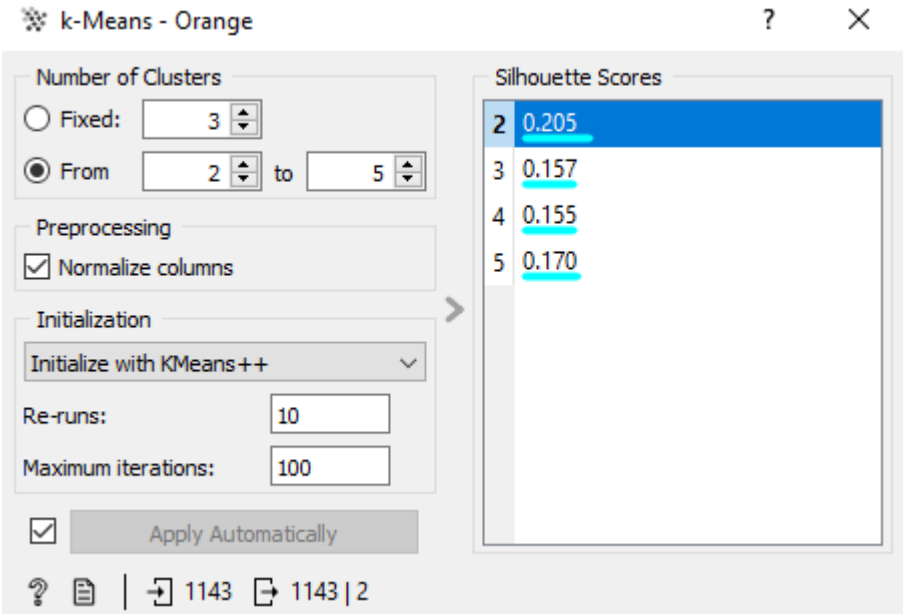
Mainot algoritma hiperparametrus, uzreiz var redzēt izmaiņas klastera dalījumā, izmantojot dažādas izlases, piemēram, Manual vai Height ratio nemaina klastera dalījumu, tās ir realizētas Orange rīkā ērtākai lietošanai. Pats klastera grupējums un mezglu skaits mainās atkarībā no izvēlētā Linkage, lielākais klasteru skaits no visiem 3 ekspirementiem bijis Average Linkage.

K-vidējo algoritms

5. Pēc darba ar Hierarchical Clustering elementu tika pievienots elements k-means (Unsupervised - > k-means), kas palīdzēs veikt eksperementus ar K-vidējo

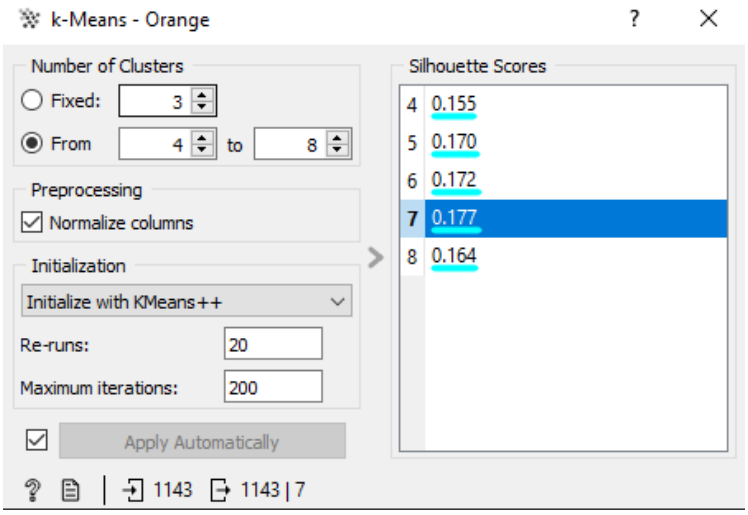
algoritmu. Pēc k-means elementa pievienošanas tas ir jāsaista ar elementu Table.

6. Pirmajam ekspirementam tika izvēlēti šādi nosacījumi: Klasteru skaits atlasīts no 2 līdz 5, aktivizēts preprocesings, inicializācijai izmanto Initialize with KMeans + +, atkārtota palaišana ir 10, maksimālais iterāciju skaits 100.



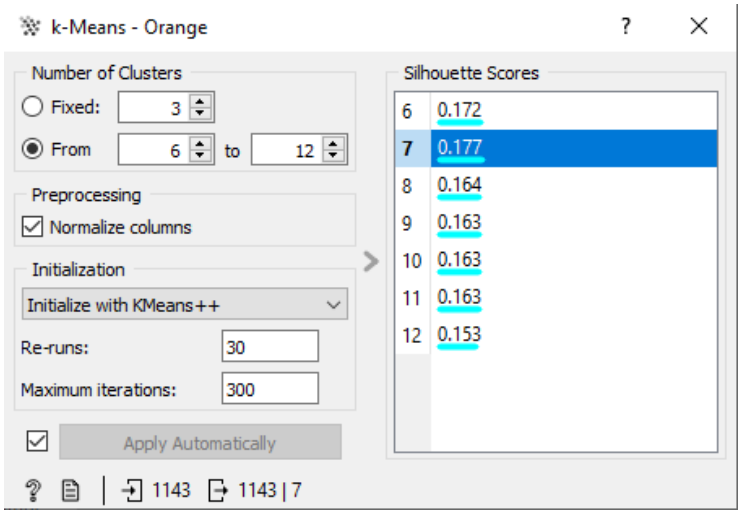
16. att. Pirmais k-means ekspirements

7. Otrajam ekspirementam tika izvēlēti šādi nosacījumi: Klasteru skaits atlasīts no 4 līdz 8, aktivizēts preprocesings, inicializācijai izmanto Initialize with KMeans + +, atkārtota palaišana ir 20, maksimālais iterāciju skaits 200.



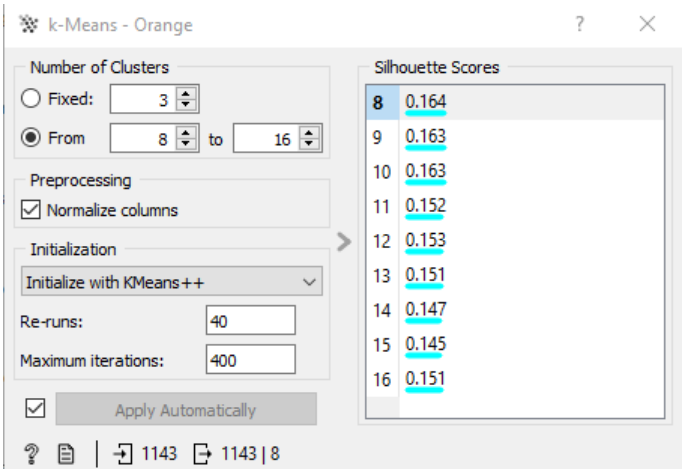
17. att. Otrais k-means ekspirements

8. Trešajam ekspirementam tika izvēlēti šādi nosacījumi: Klasteru skaits atlasīts no 6 līdz 12, aktivizēts preprocesings, inicializācijai izmanto Initialize with KMeans + +, atkārtota palaišana ir 30, maksimālais iterāciju skaits 300.



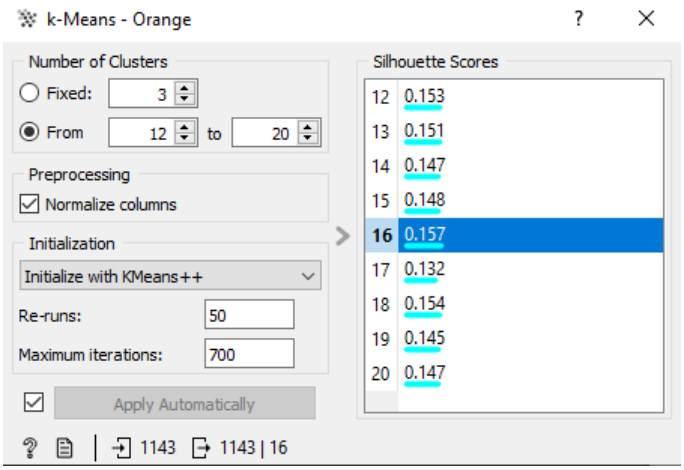
18. att. Trešais k-means ekspirements

9. Ceturtajam ekspirementam tika izvēlēti šādi nosacījumi: Klasteru skaits atlasīts no 8 līdz 16, aktivizēts preprocesings, inicializācijai izmanto Initialize with KMeans + +, atkārtota palaišana ir 40, maksimālais iterāciju skaits 400.



19. att. Ceturtais k-means ekspirements

10. Piektajam ekspirementam tika izvēlēti šādi nosacījumi: Klasteru skaits atlasīts no 10 līdz 20, aktivizēts preprocesings, inicializācijai izmanto Initialize with KMeans + +, atkārtota palaišana ir 50, maksimālais iterāciju skaits 700.



20. att. Piektais k-means ekspirements

Algoritmu hiperparametri

Numbers of Cluster:

- *Fiksēts* - algoritmu klasteri datus noteiktam klasteru skaitam.
- *From X to Y* - algoritms parāda atlasītā klasteru diapazona klasteru rezultātus, izmantojot silueta punktu skaitu (kontrastē vidējo attālumu līdz elementiem tajā pašā klasterī ar vidējo attālumu līdz elementiem citos klasteros).

Preprocessing - Ja opcija ir atlasīta, kolonnas tiek normalizētas (vidējais centrēts uz 0 un standartnovirze mērogota uz 1).

Initialization - veids, kā algoritms sāk klasterēšanu:

- *k-Means++* - pirmais centrs tiek izvēlēts nejauši, nākamie tiek izvēlēti no atlikušajiem punktiem ar varbūtību, kas proporcionāla kvadrātveida attālumam no tuvākā centra.
- *Random initialization* - klasteri sākumā tiek piešķirti nejauši un pēc tam atjaunināti ar turpmākiem atkārtojumiem.

Re-run - cik reižu algoritms tiek izpildīts no nejaušām sākotnējām pozīcijām; tiks izmantots rezultāts ar viszemāko kvadrātu summu klasterī.

Maximum iterations - maksimālais iterāciju skaits katrā algoritma izpildes reizē.

Informācija ņemta no Orange instrumenta oficiālās dokumentācijas.

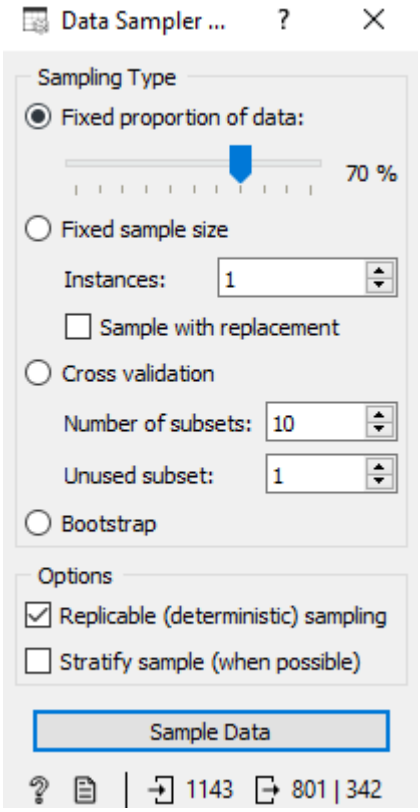
Secinājumi

Ar K-vidējo algoritmu situācija ir ļoti līdzīga, mainot vērtības algoritma kiberparametrus pastāvīgi mainās. Vislabākos rezultātus uzrādīja pirmais ekspirements, kad tika norādīts neliels klasteru skaits. No veiktajiem espirementiem var secināt, ka vislabākais objektu sadalījums ir atkarīgs no klasteru skaita, kā arī Initialization hiberparametru bloks praktiski neietekmē algoritma darbības rezultātu, taču sasāpējušais iterāciju skaits iespējams dod precīzāku rezultātu.

III DAĻA – PĀRRAUDZĪTĀ MAŠĪNMĀCĪŠANĀS

Šī darba trešajā daļā aprakstīti un analizēti trīs pārraudzītās mašīnmācīšanās darba rezultāti. Viens no tiem, kas obligāts saucas mākslīgi neironu tīkli, tālāk autors izvēlējies KNN algoritmu, jo šis algoritms jau iepriekš aprakstīts šajā kursā un SVN, kas tiek izmantots klasificēšanai un regresijas analīzei.

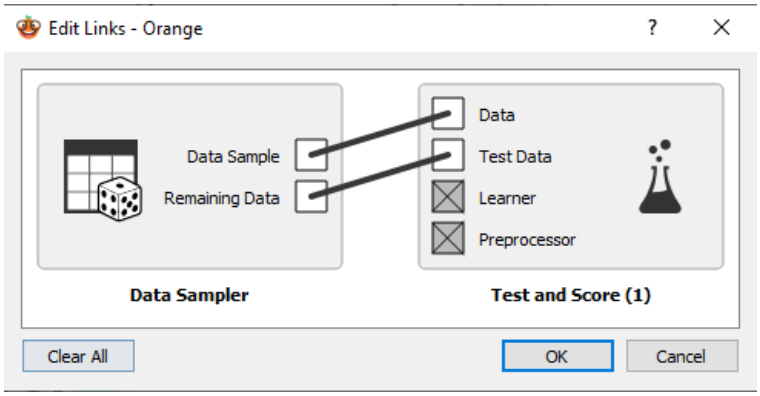
1. Lai strādātu ar šiem algoritmiem, ir jāpievieno elements Data Sampler (Transform- > Data Sampler), lai pievienotu testa un apmācības datus. Šis elements ir savienots ar Table elementu. Tālāk jānorāda procentuāli, cik datu nepieciešams testēšanai un cik apmācībai. Autors izvēlējies proporciju no 70% apmācību datu un 30% testa datu (sk. 21. attēlu).



21. att. Datu Sampler elementa iestatījumi

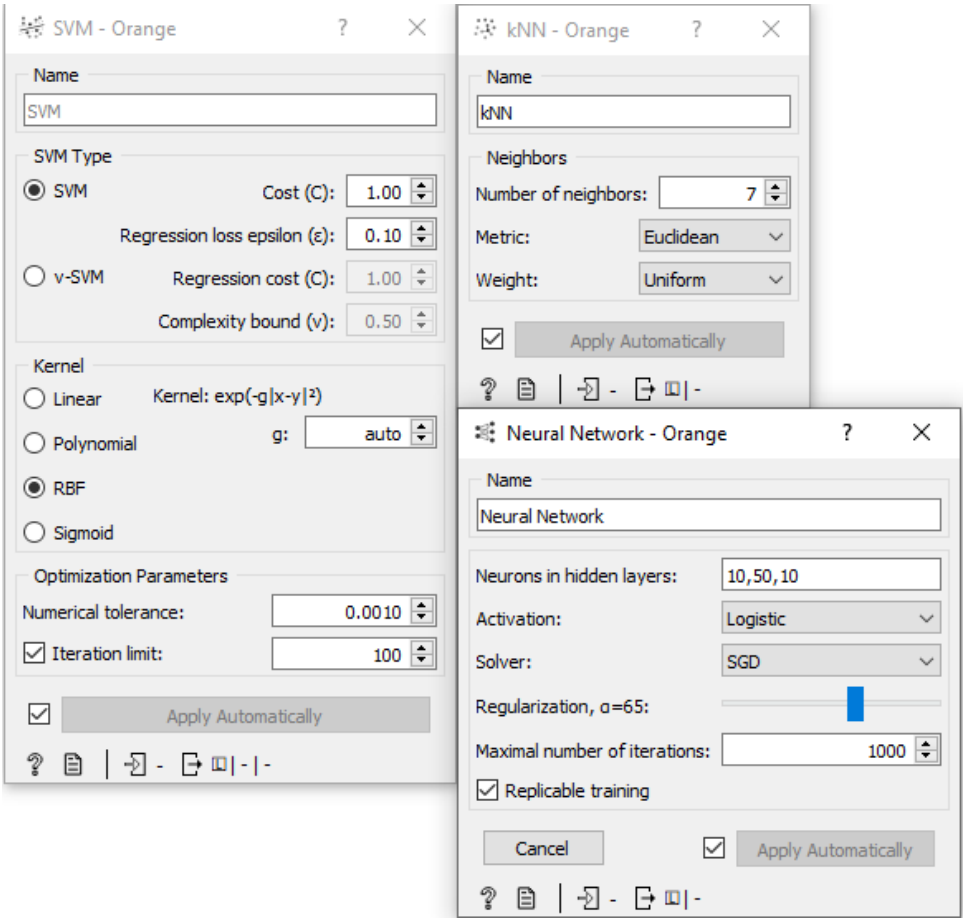
2. Pēc tam jāpievieno elementi pārraudzītajam mašīnmācīšanās algoritmam. Lai to izdarītu, ir jāpievieno kNN elementu (Model - > kNN) un Neural Network elementu (Model - > Neural Network).
3. Tālāk jāpievieno elements Test and Score (Evalute - > Test and Score), šis vienums jāsaista ar Data Sampler un visiem algoritmiem, lai pārbaudītu algoritmus, izmantojot testa datus, nepieciešams savienot Test and Score ar

Data Sampler elementu vēlreiz un norādīt saiti starp Remaining Data un Test data (skatīt 22. attēlu).



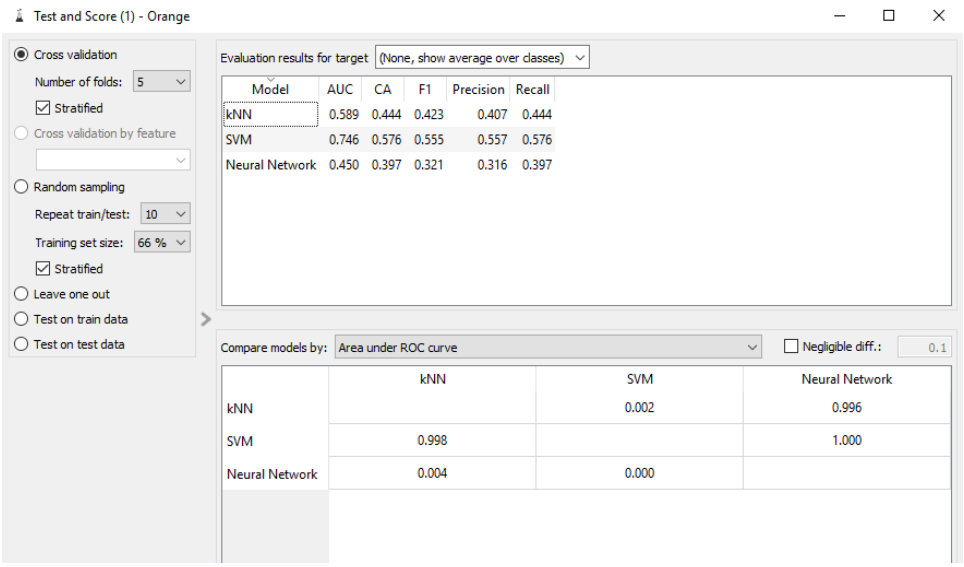
22. att. Saite starp Data Sampler un Test and Score

4. Tālāk jāveic eksperimenti ar visiem trim algoritmiem vienlaikus un jāsalīdzina rezultāti. Lai būtu vieglāk lasīt, attēli ar algoritmu iestatījumiem tiks sagrupēti.
5. Pirmais ekspirements. Šajā scenārijā visi algoritmi tiks konfigurēti pēc noklusējuma.
    - **kNN** algoritms meklēs 7 kaimiņos, par metriku izvēlēts Euclidean, bet par masu izvēlēts Uniform.
    - **Neural Network** algoritmam ir trīs neironiski slāņi (10,50,10), loģiskā aktivizācijas funkcija (Logistic Activation), mācības ātrums ir 65 (Regularization) un maksimālais iterāciju skaits 1000 (max. NUM. of iterations).
    - **SVM** algoritms darbosies ar izmaksām (Cost) 1, regresijas zaudējumus (reg. Loss Epsilon) 0.10, sadaļā Kernel izvēlēts RBF ar auto vērtību g, Numerical Tolerance ir 0.0010 un iterāciju limits ir 100.



22. att. Pirmais ekspirements

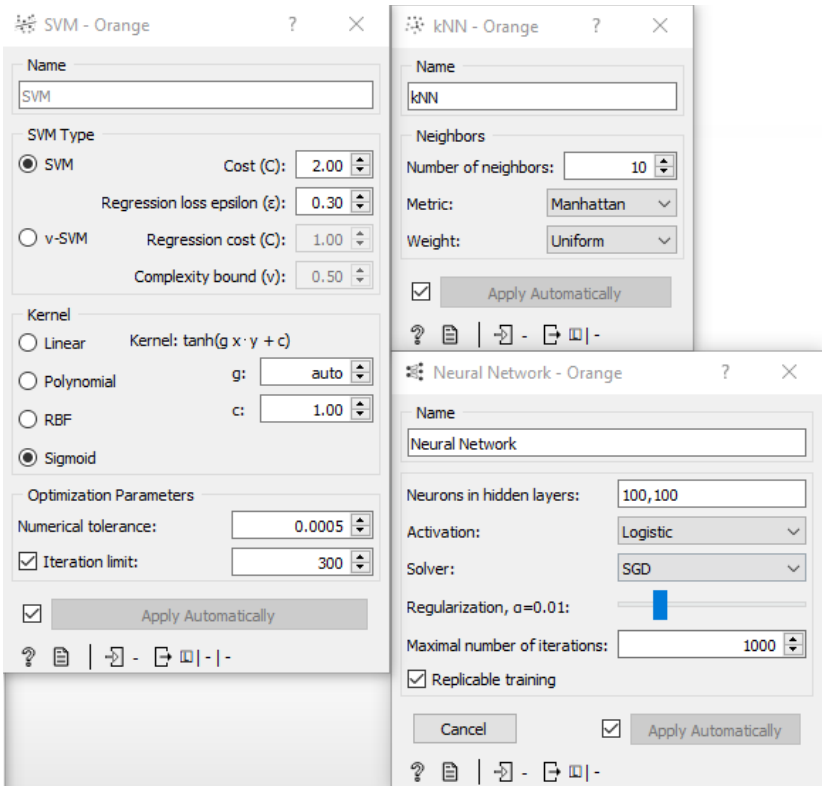
6. Pirmā ekspirementa rezultātu skatīt 23. attēlu.



23. att. Pirmā ekspirementa rezultāts

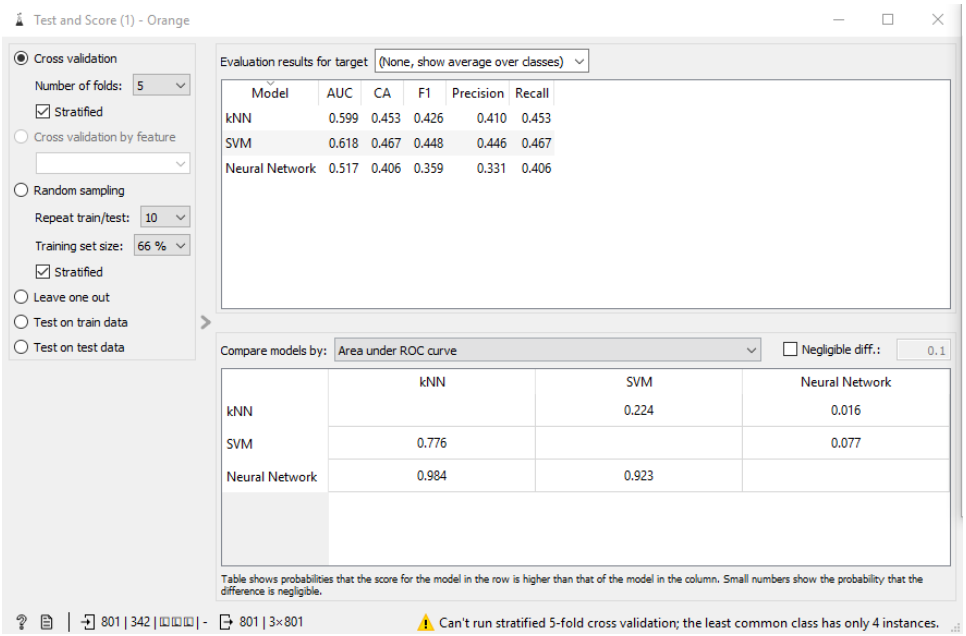
7. Otrais ekspirements. Šajā scenārijā algoritmi ir konfigurēti šādi:

- **kNN** algoritms meklēs 10 kaimiņos, par metriku izvēlēts Manhattan, bet par masu izvēlēts Uniform.
- **Neural Network** algoritmam ir trīs neironiski slāņi (100,100), loģiskā aktivizācijas funkcija (Logistic Activation), mācības ātrums ir 0.1 (Regularization) un maksimālais iterāciju skaits 1000 (max. NUM. of iterations).
- **SVM** algoritms darbosies ar izmaksām (Cost) 2, regresijas zaudējumus (reg. Loss Epsilon) 0.30, sadaļā Kernel izvēlēts Sigmoid ar auto vērtību g un c ir 1, Numerical Tolerance ir 0.0005 un iterāciju limits ir 300.



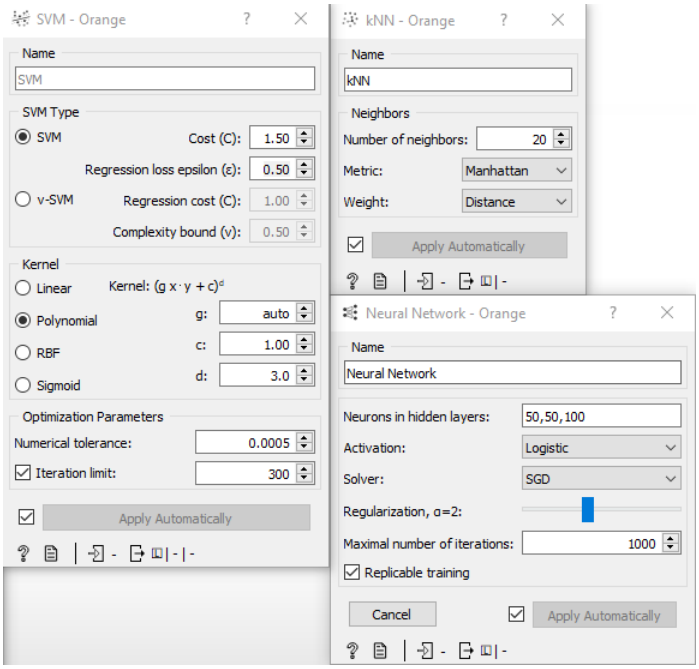
23. att. Otrais ekspirements

8. Otrā ekspirementa rezultāti izskatās šādi:



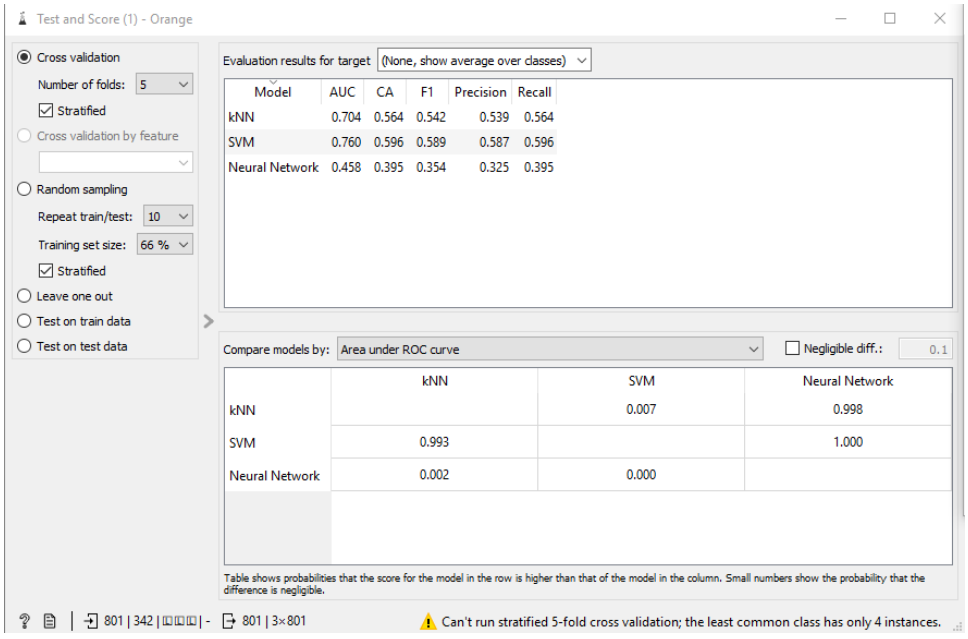
23. att. Otrā ekspirementa rezultāts

9. Trešais ekspirements. Šajā scenārijā algoritmi ir konfigurēti šādi:
   - **kNN** algoritms meklēs 20 kaimiņos, par metriku izvēlēts Manhattan, bet par masu izvēlēts Distance.
   - **Neural Network** algoritmam ir trīs neironiski slāņi (50,50,100), loģiskā aktivizācijas funkcija (Logistic Activation), mācības ātrums ir 2 (Regularization) un maksimālais iterāciju skaits 1000 (max. NUM. of iterations).
   - **SVM** algoritms darbosies ar izmaksām (Cost) 1.5, regresijas zaudējumus (reg. Loss Epsilon) 0.50, sadaļā Kernel izvēlēts Polynominal ar auto vērtību g, d ir 3 un c ir 1, Numerical Tolerance ir 0.0005 un iterāciju limits ir 300.



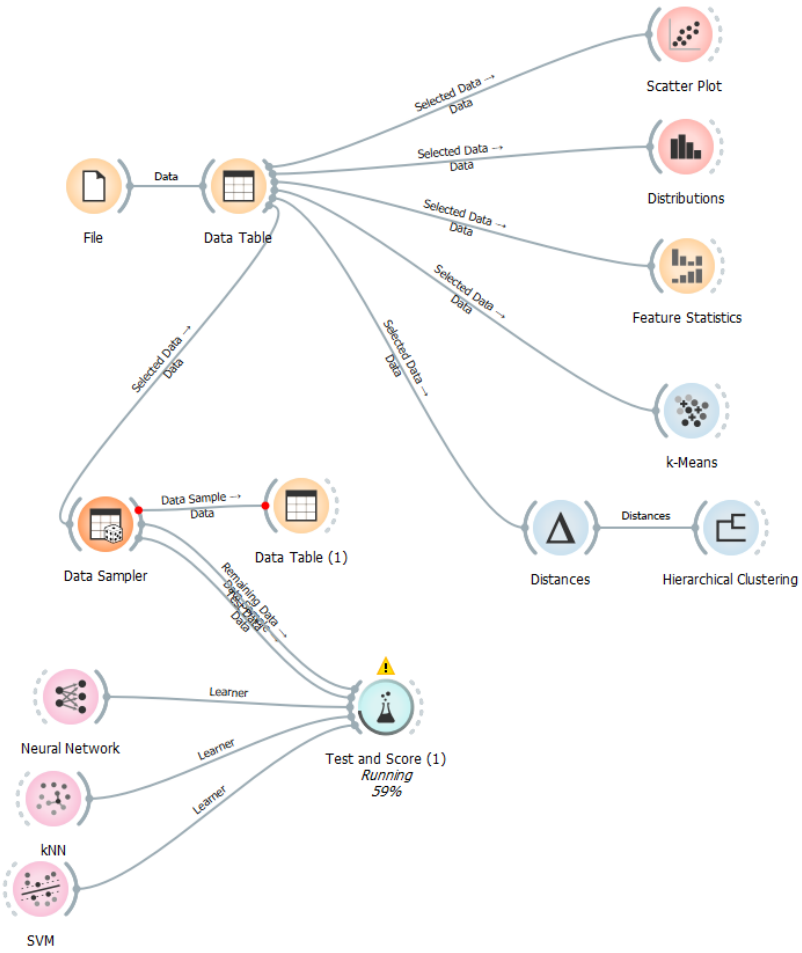
22. att. Trešais ekspirements

10. Trešā ekspresmenta rezultātu skaties 23. attēlā.



23. att. Trešā ekspirementa rezultāts

Galīgā darbvirsma redzama 24. attēlā.



24. att. Fināla darbvirsma

Algoritmi un u hiperparametri

KNN - populārs klasifikācijas algoritms, kas tiek izmantots dažādos mašīnmācīšanās uzdevumu tipos. Līdzīgi risinājumu kokam, tā ir viena no jēdzīgākajām klasifikācijas pieejām. Autors izvēlējās šo algoritmu, jo viņam jau bija neliela pieredze ar to šajā kursā.

- Neighbors - tuvāko kaimiņu skaits;
- Metric
  - Euclidean - attālums starp diviem atribūtiem;
  - Manhattan - visu atribūtu atšķirību summa;
  - Maximal - lielākā no absolūtajām atšķirībām starp atribūtiem;
  - Mahalanobis - attālums starp punktu un sadalījumu.

SVM - tas ir lineārs algoritms, ko izmanto klasifikācijas un regresijas uzdevumos. Šim algoritmam ir plašs pielietojums praksē un tas var risināt gan

lineārus, gan nelineārus uzdevumus. Balsta vektoru "mašīnu" darba būtība ir vienkārša: algoritms veido līniju vai hiperplakni, kas datus sadala klasēs. Autors izvēlējās šo algoritmu, jo to var izmantot klasificēšanai.

- SVM Type
  - SVM – Cost: soda termiņš par zaudējumu. Lieto klasifikācijas un regresijas uzdevumiem. ε: Epsilon-SVR modeļa parametrs, attiecas uz regresijas uzdevumiem. Nosaka attālumu no patiesajām vērtībām, kurā ar paredzētajām vērtībām nav saistīti sodi.
  - ν-SVM – Cost: soda termiņš par zaudējumu. Lieto klasifikācijas un regresijas uzdevumiem. ν: modeļu parametrs xDrive-SVR, attiecas uz klasifikācijas un regresijas uzdevumiem. Apmācības kļūdu daļas augšējā robeža un atbalsta vektoru daļas apakšējā robeža.
- Kernel – tā ir funkcija, kas pārvērš atribūtu telpu jaunā pazīmju telpā, lai tā atbilstu hiperplaknei ar maksimālu rezervi, kas ļauj algoritmam veidot modeli ar Linear, Polynomial, RBF un Sigmoid kodoliem.
  - g ir gamma konstantei kodola funkcijā (ieteicamā vērtība ir 1/k, kur k ir atribūtu skaits, bet, tā kā vidžetam var nebūt treniņu kopas, noklusējuma vērtība ir 0).
  - c - konstante, pec noklusējuma 0.
  - d kodola pakāpei (pēc noklusējuma 3).
- Numerical tolerance - pieļaujamā novirze no paredzamās skaitliskās pielaides vērtības.
- Iteration Limit - iterāciju limits.

Informācija ņemta no Orange instrumenta oficiālās dokumentācijas.

Secinājumi

Visos trijos gadījumos labākā CA vērtība bija algoritma SVM. Vislabākā vērtība iegūta trešā ekspromta laikā, proti, 0.596, tāpat SVM uzvar pēc F1 vērtībām, kas ir 0.589. Tomēr SVM algoritms darbojas nedaudz labāk par kNN algoritmu, iespējams, ka šādas vērtības nav sanācis pārāk abjektīvas un, lai veiktu kvalitatīvāku novērtēšanu, nepieciešams iedarbināt vairāk ekspirementu. Visi trīs algoritmi rāda

bleziski vienādus rezultātus, bet manā skatījumā algoritms mākslīgi neironu tīkli ir ticis galā sliktāk par pārējiem. Iespējams, ka, veicot ekspirementu bleznojami vienādos apstākļos un hiberparametru nozīmēs, visu trīs algoritmu rezultāti būs vēl tuvāki viens otram.

IZMANTOTĀ LITERATŪRA

1. M. Yasser H. (2022) Wine Quality Dataset. Pieejams: https://www.kaggle.com/datasets/yasserh/wine-quality-dataset
2. Wikipedia, Гистограмма, Pieejams: https://ru.wikipedia.org/wiki/Гистограмма
3. Orange Visual Programming, k-Means. Pieejams: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/kmeans.html
4. Orange Visual Programming, Scatter Plot. Pieejams: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/visualize/scatterplot.html
5. Orange Visual Programming, Distributions. Pieejams: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/visualize/distributions.html
6. Orange Visual Programming, Distances. Pieejams: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/distances.html
7. Orange Visual Programming, Hierarchical Clustering. Pieejams: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/hierarchicalclustering.html
8. HandWiki, WPGMA. Pieejams: https://handwiki.org/wiki/WPGMA
9. Orange Visual Programming, k-Means, Pieejams: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/kmeans.html
10. Wikipedia, Метод опорных векторов. Pieejams: https://ru.wikipedia.org/wiki/Метод_опорных_векторов
11. Proglib, Метод k-ближайших соседей. Pieejams: https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19