

2. Praktiskais darbs

Mākslīga Intelektā pamati

“Mašīnmācīšanās algoritmu lietojums”

https://github.com/ArtjomsBreitmanis/otrais_praktiskais_darbs

Rīgas Tehniskā Universitāte

Students: Artjoms Breitmanis

St. apl. nr: 131RMC272

Fakultāte, grupas nr: DITF, 3.g

Rīga, 11.05.2023

Saturs

Contents

Saturs.....	2
Ievāds	3
I daļa - Datu pirmapstrāde/izpēte	4
Darba gaita:	5
Pirmās Daļas Secinājumi.....	12
II daļa – Nepārraudzītā mašīnmācīšanās.....	13
Hierarhiskā klasterizācija: (3 eksperimenti).....	14
K-Means algoritms:	15
III daļa – Pārraudzītā mašīnmācīšanās.....	19
kNN.....	19
“Iss” apraksts.....	19
Informācijas avoti:	19
Orange rīkā kNN pieejamie hiperparametri un to nozīme[21]:	20
kNN peilietošana:	20
SVM algoritms	22
“Iss” apraksts:	22
Datu avoti:	22
Orange rīkā SVM pieejamie hiperparametri un to nozīme:[26]	22
Pielietojums:.....	23
Neural Network	25
Orange rīkā SVM pieejamie hiperparametri un to nozīme:[31]	25
Pielietojums:.....	26
Secinājumi:	29

Ievāds

Šajā darbā es gribu uzzināt vairāk par dažādiem mašīnmācīšanās rīkiem un to darbību es veikšu Orange rīkā. Kura ir domāta mašīnmācīšanās testēšanai un pētīšanai.

Es izvēlējos trīs dažādus algoritmus kNN, SVM un Neural Network, tie būs salīdzināti un testēti, ka arī aprakstīti, lai sapratu labāk ka tie ir būvēti un ka darbojas, balstoties uz kādiem principiem.

Datu kopa kuru es izmantošu ir brīva un to var brīvi lejupielādēt no internēta (links ir "darba gaita" sadaļā), šī kopa tika savāktā fotografējot pupiņas un ar datora palīdzību sūrstēt un datēt tos pa kategorijām. Problemsfēra šim kopām ir tāda: ļaut mākslīgam intelektam mācīties saprast ar fotogrāfiju kādā pupiņas šķirē tas ir.

I daļa- Datu pirmapstrāde/izpēte

Darba uzdevums:

1. Ir jāizvēlas un jāapraksta datu kopa, pamatojoties uz informāciju, kas sniegta krātuvē, kurā datu kopa ir pieejama.
2. Ja no krātuves iegūtā datu kopa nav formātā, ar kuru ir viegli strādāt (piemēram, komatatzīmītas vērtības vai .csv fails), ir jāveic tās transformācija vajadzīgajā formātā.
3. Ja kādu pazīmju (atribūtu) vērtības ir tekstveida vērtības (piemēram, yes/no, positive/neutral/negative, u.c.), tās ir jātransformē skaitliskās vērtībās.

4. Ja kādiem datu objektiem trūkst atsevišķu pazīmju (atribūtu) vērtības, ir jāatrod veids, kā tās iegūt, studējot papildu informācijas avotus.

5. Ir jāatspoguļo datu kopa vizuāli un jāaprēķina statistiskie rādītāji:

- a) ir jāizveido vismaz divas 2- vai 3-dimensiju izkliedes diagrammas (scatter plot), kas ilustrē klases atdalāmību, balstoties uz dažādām pazīmēm (atribūtiem); studentam ir jāizvairās izmantot datu objekta ID vai klases iezīmi kā mainīgo izkliedes diagrammā;
- b) ir jāizveido vismaz 2 histogrammas, kas parāda klašu atdalīšanu, pamatojoties uz interesējošām pazīmēm (atribūtiem);
- c) ir jāatspoguļo 2 interesējošo pazīmju (atribūtu) sadalījums;
- d) ir jāaprēķina statistiskie rādītāji (vismaz vidējās vērtības un dispersiju).

Darba atskaitē ir jāiekļauj šāda informācija:

- datu kopas apraksts (sniedzot arī atsauci uz izmantotajiem informācijas avotiem):
 - o datu kopas nosaukums, avots, izveidotājs un/vai īpašnieks;
 - o datu kopas problēmsfēras apraksts;
 - o datu kopas licencēšanas nosacījumi (ja tādi ir);
 - o veids, kā datu kopa tika savākta;
- datu kopas satura apraksts (sniedzot arī atsauci uz izmantotajiem informācijas avotiem):
 - o datu objektu skaits datu kopā;
 - o datu kopas pazīmju (atribūtu) atspoguļojums kopā ar to lomām Orange rīkā;
 - o klašu skaits datu kopā, katras klases nozīme un klašu atspoguļošanas veids (klasēm atbilstošo iezīmju skaidrojums); ja datu kopa nodrošina vairākas iespējamās datu klasifikācijas, tad atskaitē skaidri ir jāidentificē, kāda tieši klasifikācija tiek apskatīta darbā;
 - o datu objektu skaits, kas pieder katrai klasei;
 - o pazīmju (atribūtu) skaits un nozīme datu kopā, kā arī to vērtību tipi un diapazoni (šī informācija būtu jāatspoguļo tabulā, norādot pazīmes (atribūta) apzīmējumu, skaidrojumu, vērtību tipu un datu kopā pieejamo vērtību diapazonu);
 - o datu faila struktūras fragments, kurā ir redzamas visas datu faila kolonnas un to vērtības vismaz dažiem datu objektiem;
- secinājumi, kas izriet no izkliedes diagrammu, histogrammu un sadalījumu analīzes (sk. I daļas 5. solis) par datu kopas klašu atdalāmību. Studentiem ir jāatbild uz šādiem jautājumiem:
 - o Vai klases datu kopā ir līdzsvarotas, vai dominē viena klase (vai vairākas klases)? Tas tiek noteikts, spriežot pēc tā, cik daudz datu objektu pieder katrai kopai.
 - o Vai datu vizuālais atspoguļojums ļauj redzēt datu struktūru? Runa ir par to, vai datu objekti, kuri pieder dažādām klasēm, ir skaidri atdalāmi.
 - o Cik datu grupējums ir iespējams identificēt, pētot datu vizuālo atspoguļojumu? Runa ir par to, vai ir kaut cik atdalāmi datu grupējumi, ja gadījumā dažādu klašu datu objekti saplūst kopā.
 - o Vai identificētie datu grupējumi atrodas tuvu viens otram vai tālu viens no otra?
- secinājumi, kas izriet no statistisko rādītāju (vidējo vērtību un dispersijas vērtību) analīzes.

Darba gaita:

1. Es izvēlējos datu kopu, kura bija lejupielādētā no šīs saites:
Dry Bean Dataset Data Set

<https://archive.ics.uci.edu/ml/datasets/Dry+Bean+Dataset>

Kopsavilkums: Ar augstas izšķirtspējas kameru tika uzņemti 7 dažādu reģistrētu sausu pupiņu 13 611 graudu attēli. Kopā 16 funkcijas; No graudiem tika iegūti 12 izmēri un 4 formas.

Datu kopas raksturlielumi:	Daudzfaktoru	Gadījumu skaits:	13611	Apgabals:	Dators
Atribūtu raksturojums:	Vesels skaitlis, reāls	Atribūtu skaits:	17	Ziedošanas datums	2020-09-14
Saistītie uzdevumi:	Klasifikācija	Vai trūkst vērtību?	N/A	Tīmekļa trāpījumu skaits:	2216166

Source:

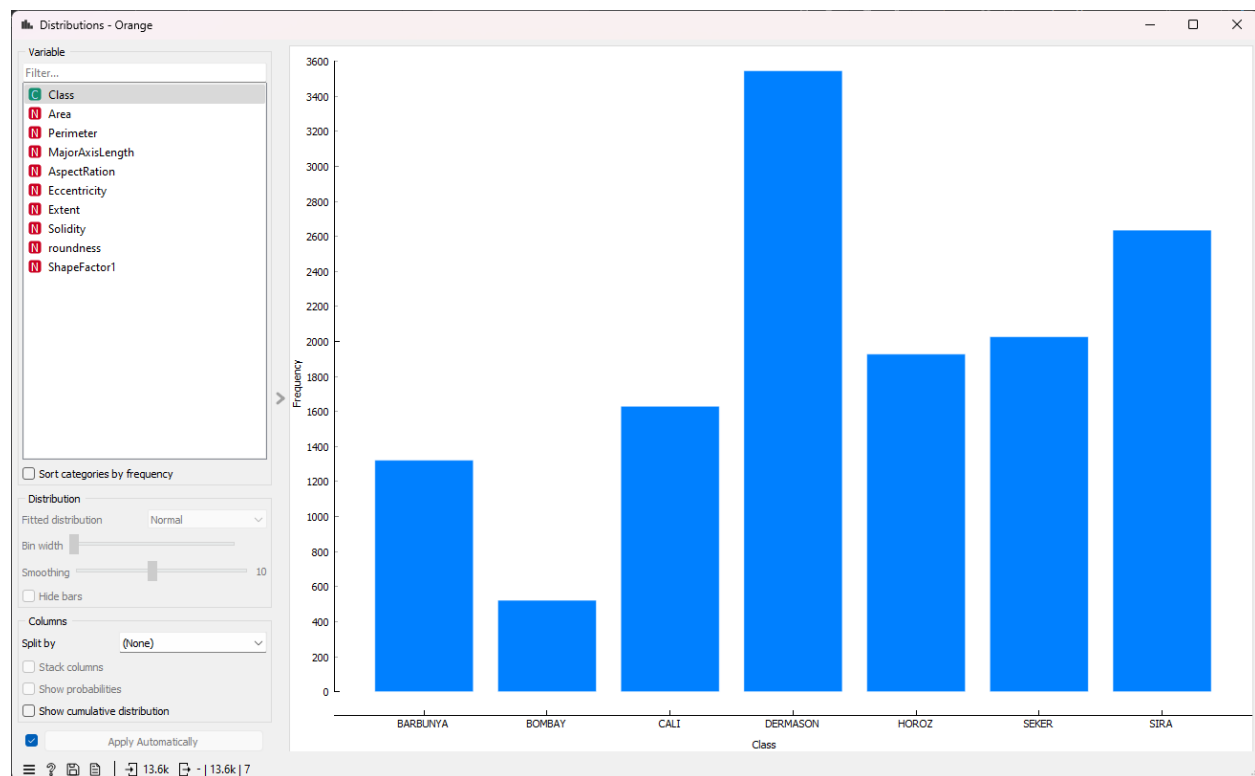
Murat KOKLU
Faculty of Technology,
Selcuk University,
TURKEY.
ORCID : 0000-0002-2737-2360
mkoklu '@' selcuk.edu.tr

Ilker Ali OZKAN
Faculty of Technology,
Selcuk University,
TURKEY.
ORCID : 0000-0002-5715-1040
ilkerozkan '@' selcuk.edu.tr

Informācija par datu kopu:

Šajā pētījumā tika izmantoti septiņi dažādi sauso pupiņu veidi, ņemot vērā tādas pazīmes kā forma, forma, veids un struktūra atbilstoši tirgus situācijai. Tika izstrādāta datorredzes sistēma, lai atšķirtu septiņas dažādas reģistrētas sauso pupiņu šķirnes ar līdzīgām pazīmēm, lai iegūtu vienotu sēklu klasifikāciju. Klasifikācijas modelim ar augstas izšķirtspējas kameru tika uzņemti 7 dažādu reģistrētu sauso pupiņu 13 611 graudu attēli. Pupiņu attēli, kas iegūti, izmantojot datorredzes sistēmu, tika pakļauti segmentācijas un pazīmju iegūšanas posmiem, un kopā 16 pazīmes; No graudiem tika iegūti 12 izmēri un 4 formas.

- 1.) Area (A): Pupu zonas laukums un pikseļu skaits tās robežās.
- 2.) Perimeter (P): Pupas apkārtmērs tiek definēts kā tās apmales garums.
- 3.) Major axis length (L): Attālums starp garākās līnijas galiem, ko var novilkt no pupiņas.
- 4.) Minor axis length (l): Garākā līnija, ko var novilkt no pupiņas, stāvot perpendikulāri galvenajai asij.
- 5.) Aspect ratio (K): Definē attiecības starp L un l.
- 6.) Eccentricity (Ec): Elipses ekscentriskums ar tādiem pašiem momentiem kā reģionam.
- 7.) Convex area (C): Pikseļu skaits mazākajā izliektajā daudzstūrī, kurā var būt pupiņu sēklas laukums.
- 8.) Equivalent diameter (Ed): Apļa diametrs, kura laukums ir vienāds ar pupiņu sēklu laukumu.
- 9.) Extent (Ex): Ierobežojošā lodziņa pikseļu attiecība pret pupiņu laukumu.
- 10.) Solidity (S): Zināms arī kā izliekums. Pikseļu attiecība izliektajā apvalkā pret tiem, kas atrodami pupiņās.
- 11.) Roundness (R): Aprēķināts ar šādu formulu: $(4\pi A)/(P^2)$
- 12.) Compactness (CO): Mēra objekta apaļumu: Ed/L
- 13.) ShapeFactor1 (SF1)
- 14.) ShapeFactor2 (SF2)
- 15.) ShapeFactor3 (SF3)
- 16.) ShapeFactor4 (SF4)
- 17.) Class (Seker, Barbunya, Bombay, Cali, Dermosan, Horoz and Sira)

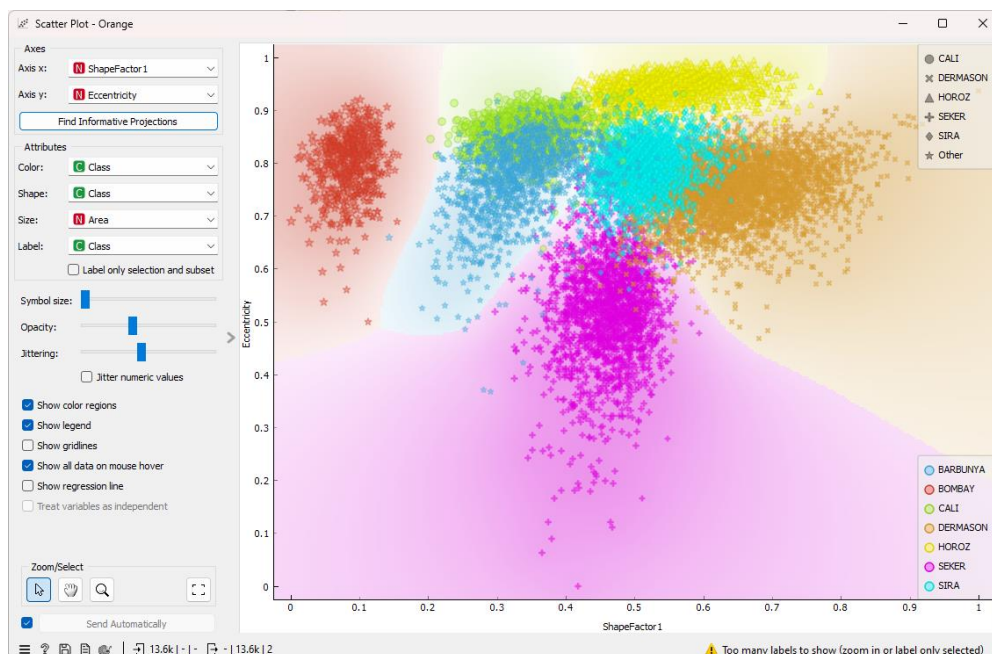


[1]

Barbunya	-	1322	9,71%
Bombay	-	522	3,84%
CALI	-	1630	11,98%
Dermason	-	3546	26,05%
Horo	-	1928	14,17%
Seker	-	2027	14,89%
Sira	-	2636	19,37%

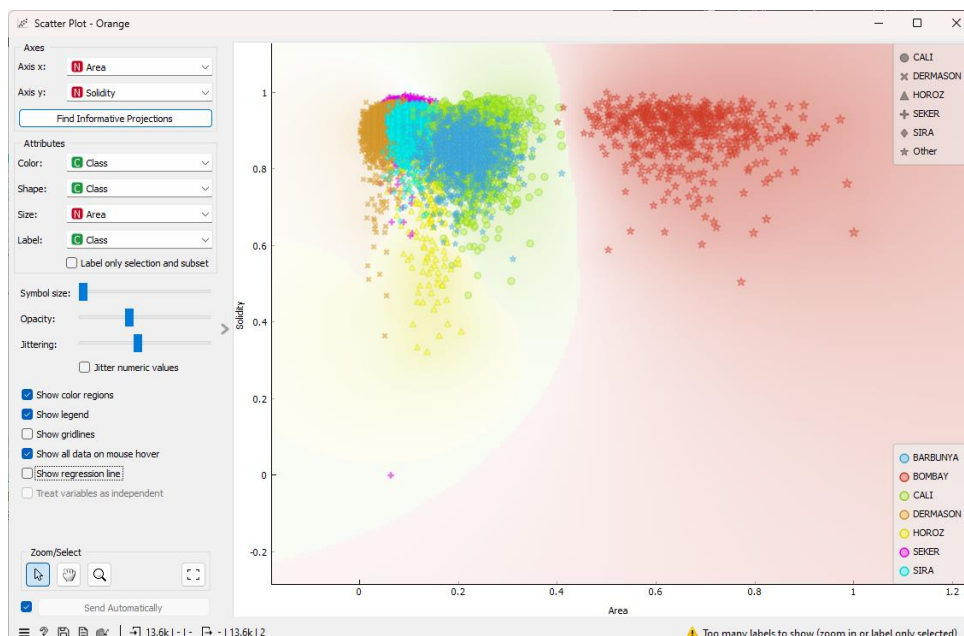
No [1] bildes iegūtie dati.

2. Manā gadījumā man paveicas un visi dati atbilstajā prasībām un man nebija vajadzības tos izmainīt vai papildināt.
3. Visas vērtībās bija dotas pareizā (numeric) formātā, un Bula tipa formāta datu nebija, tādēļ man nekas nebija jāmaina.
4. Ka arī šī punktā manos izvēlētos datos viss bija dots, un man nav bijis vajadzīgs jāatraso veids, ka to iegūt.
5. Ir jāatpoguļo datu kopa vizuāli un jāaprēķina statistiskie rādītāji:



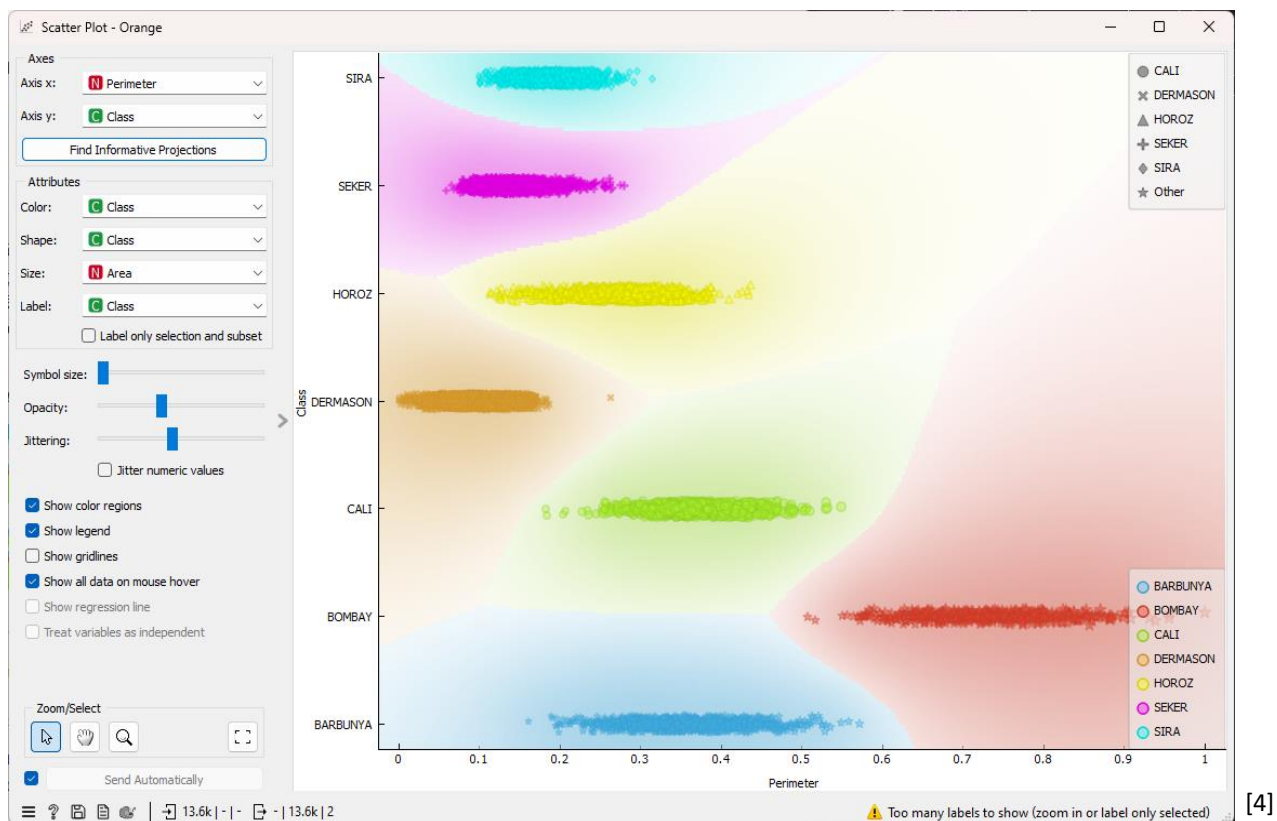
[2]

Pirmo diagrammu [2] es izvēlējos atlasīt pēc tādiem faktoriem kā: Eccentricity un Shape Factor, jo šajā gadījumā var ļoti precīzi redzēt ka dažādas pupiņu veidi atšķirās ar formu un no tā forma ļoti labi ir parādīta caur elipses veida atšķirību (šaurāks, vai plašāks). Un no šīs diagrammas var precīzi redzēt ka: iekš vienai pupiņu šķirnei ir ļoti līdzīgs elipses koeficients.

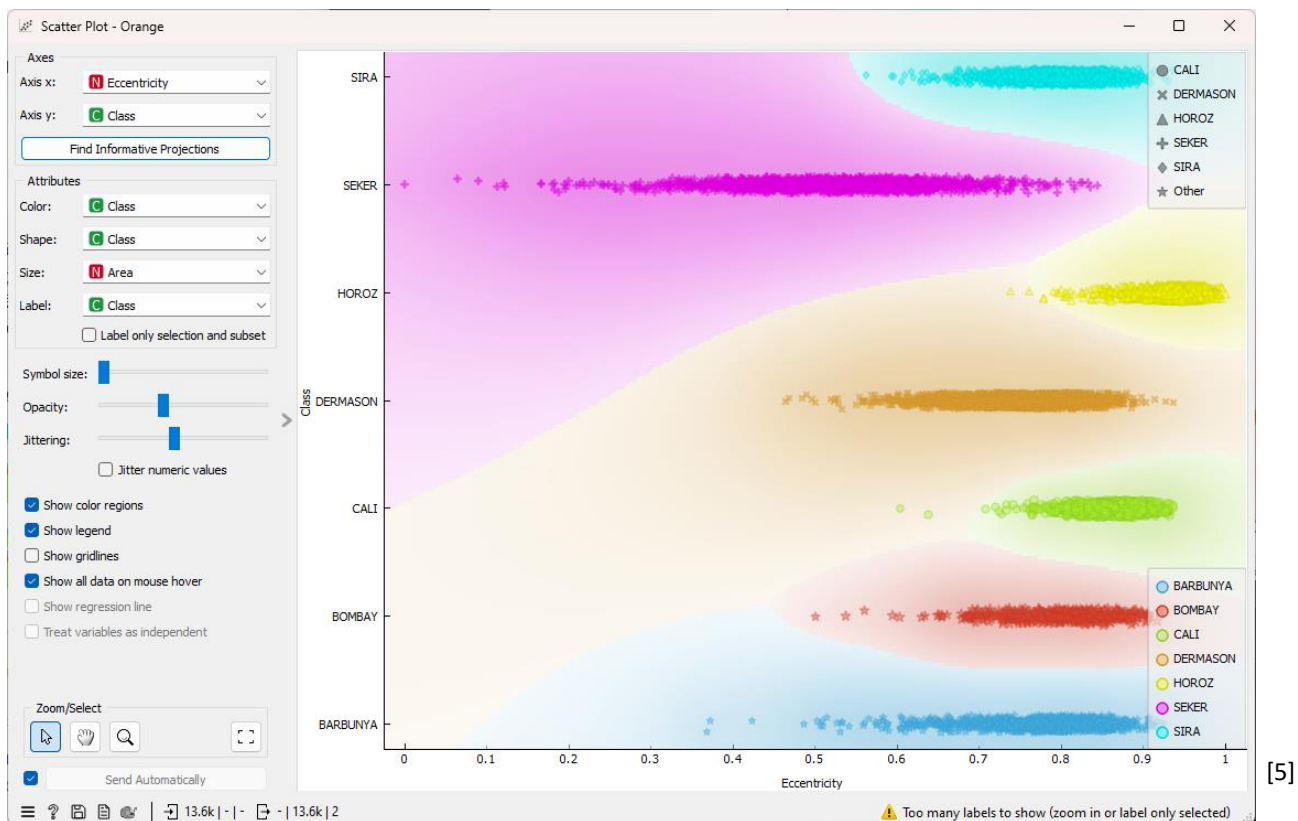


[3]

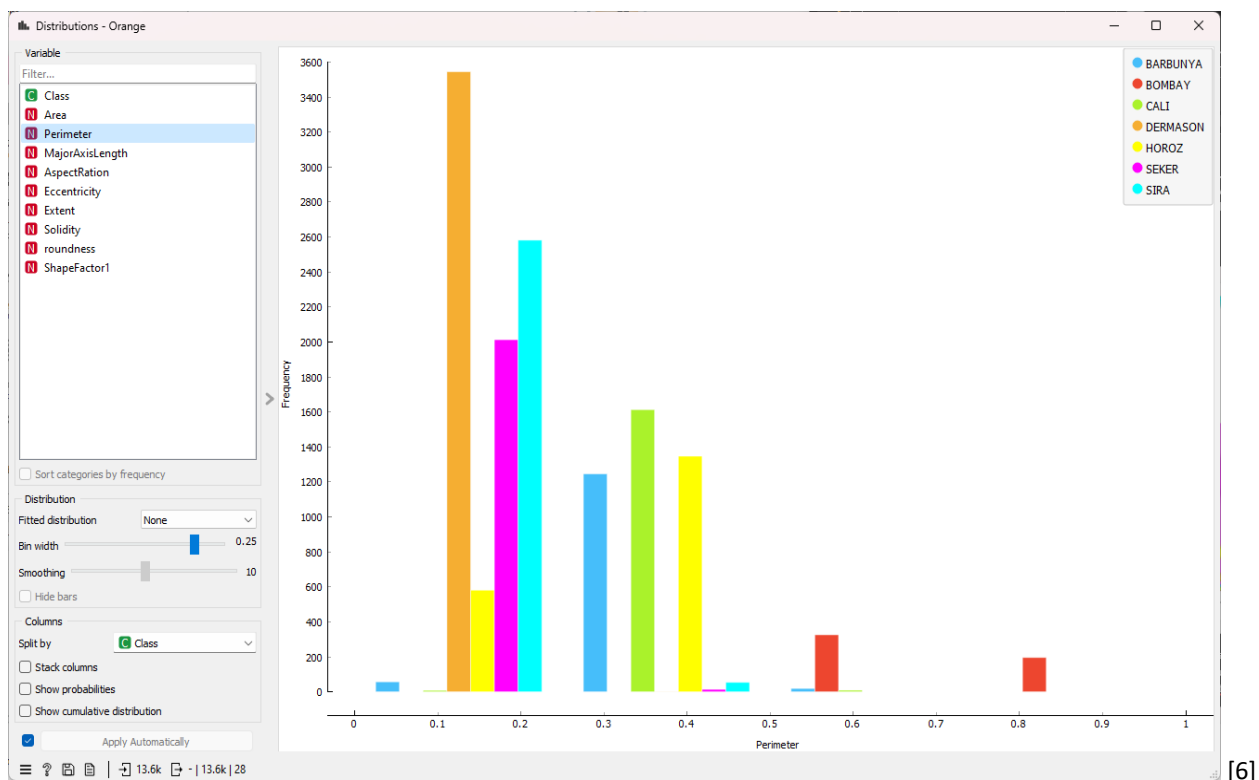
Šeit [3] var redzēt ka Indiešu pupiņas Bombay atšķiras no citiem stingrībā. Pupiņas kuri ir no Indijas ir vislielāk atšķirību no tiem kuri ir no Eirāzijas un no Amerikas kontinenta.



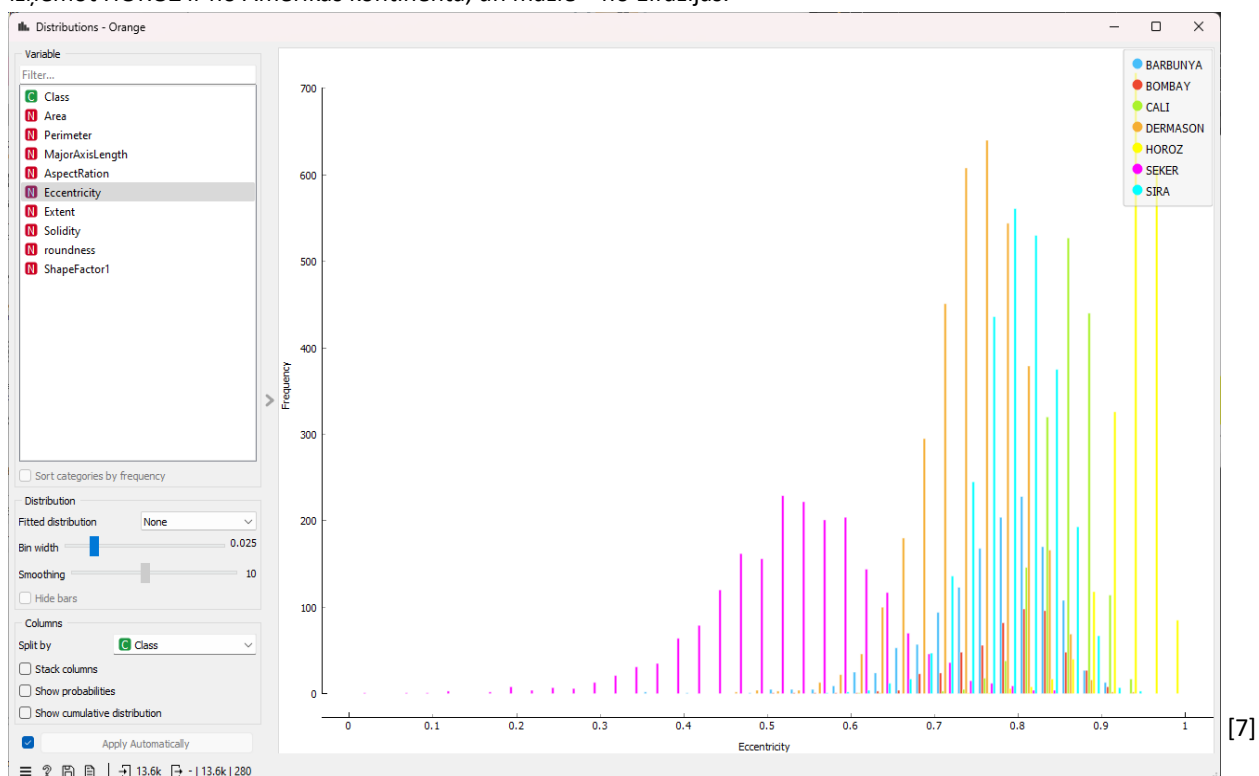
Sadalījumā[4] pēc pupiņas perimetra pa klasēm, var redzēt, ka ir redzami trīs grupas “Mazie”, “Vidējie” un “Lielie”. Sadalījuma pa klasēm ir lieliski redzamas atdalījumā lielums un katra krasa simbolizē atšķirīgo klasi.



Sadalījuma [5] pēc ekscentriskuma pa klasēm, mēs varam ļoti labi redzēt ka gandrīz visas šķires ir vairāk elipsoīda, nevis apaļas, bet izņēmums ir SEKER šķire, tā ir vairāk apaļa, nekā elipsoīdā.



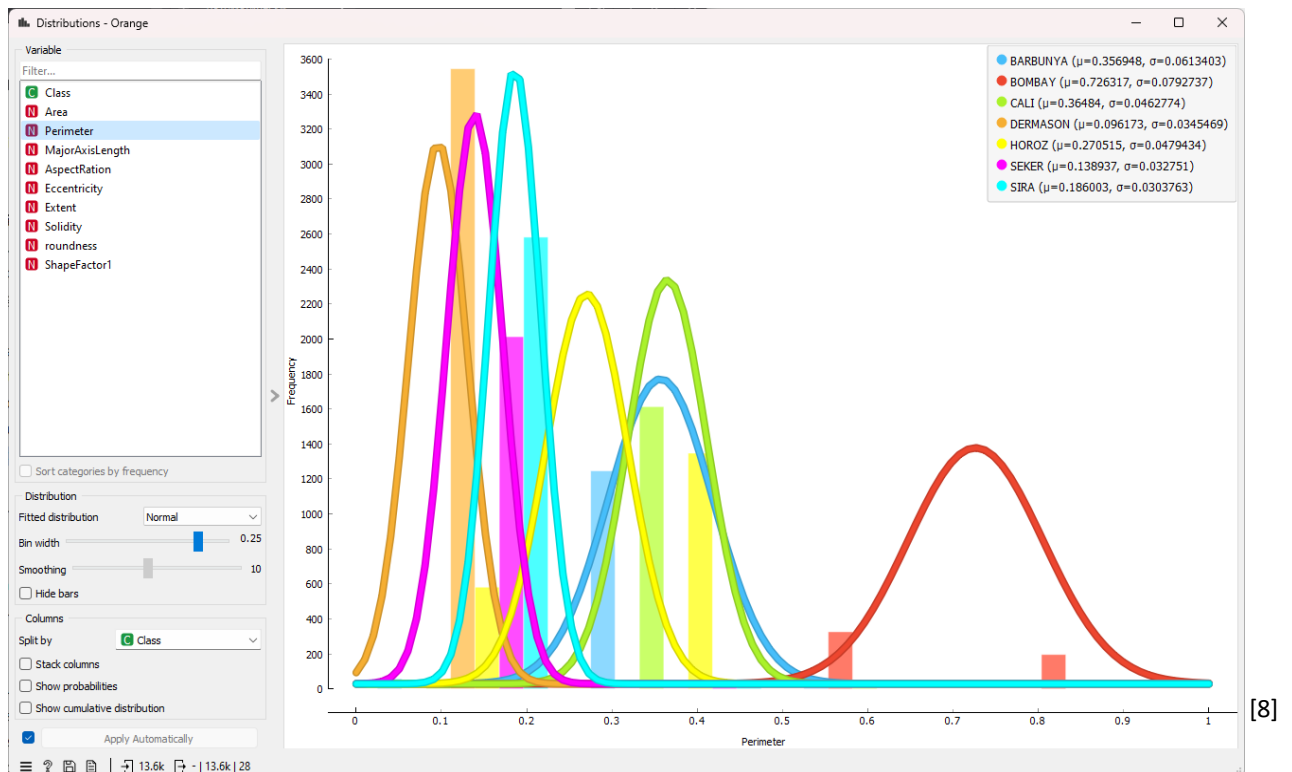
Pēc šī histogrammas [6] var redzēt, ka pupiņas ir sagrupējusies 3 kategorijas, lielle, vidējie, un mazie, tie kuri ir lielle, tā ir viena šķire no pupiņām un ir no Indijas, pārējie ir no Turcijas vai Amerikas kontinenta. Vidējie izņemot HOROZ ir no Amerikas kontinenta, un mazie – no Eirāzijas.



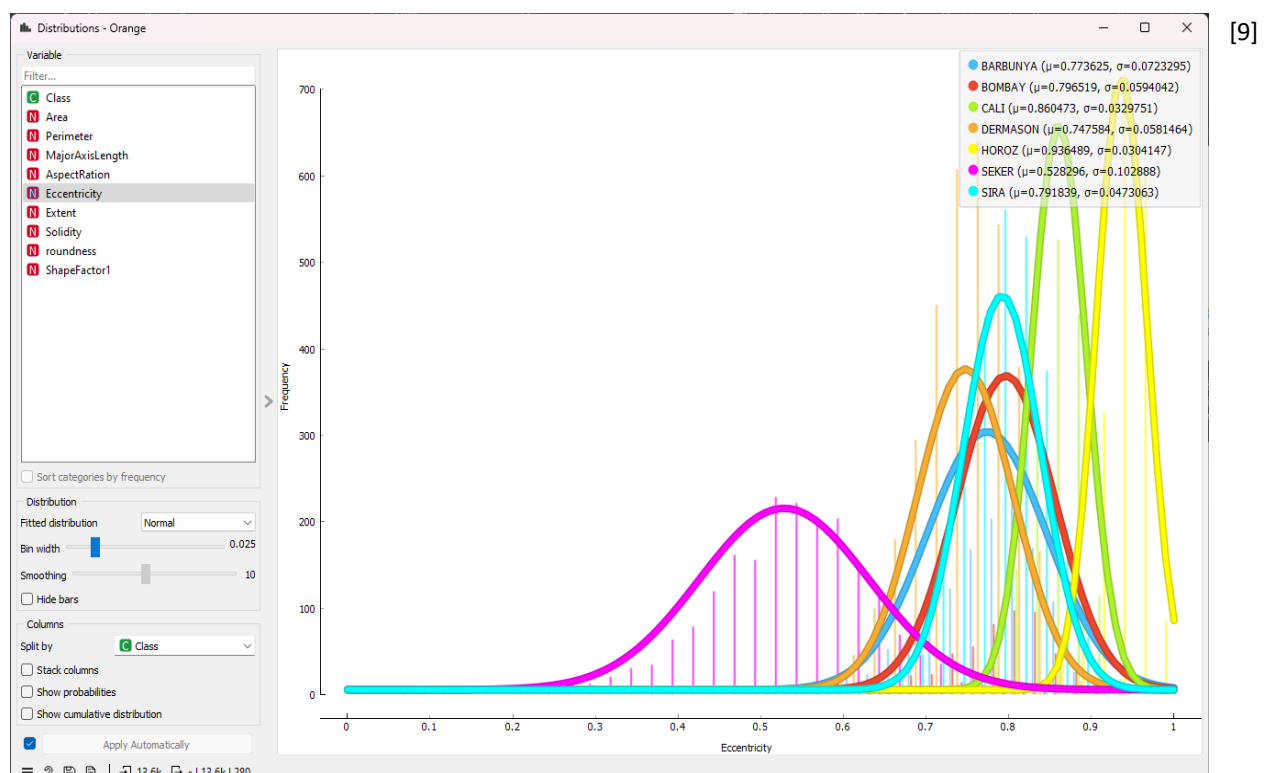
No šīs histogrammas [7] var secināt, ka gandrīz visas populāras pupiņas ir vairāk elipsoīdos nekā apaļas. Vienīgais izņēmums SEKER, tie ir gan apaļi, gan elipsoīdos.

Kas attēcas uz otro šķirošanas veidu, tē var redzēt ka soliditāte un laukums nav isti saistītas ar pupiņu šķirni, jo ka ir redzams izklaidē, viņi gandrīz visi ir vienā laukā, izņemot tādu šķirni kā BOMBAY un HOROZ, tiem var būt diezgan atšķirīgi atribūti iekš savas šķirnes, bet tas ir vairāk izņēmums, jo lielākai daļai no šķirnēm ir uniformāls sadalījums.

Histogrammas.

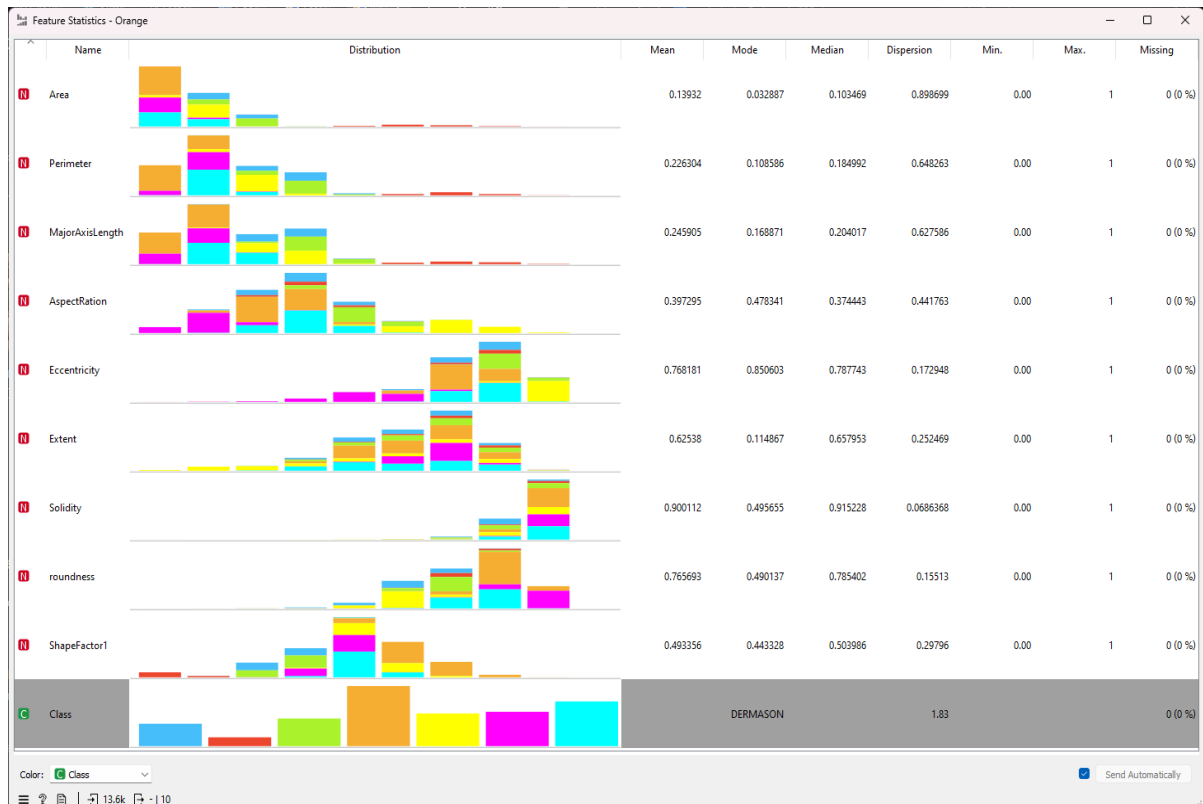


Skatoties uz dotajiem normalsādālijumiem [8], var redzēt ka katrai pupiņu grupai ir krietna atkarība pupiņu perimetra no šķirnes. Tē mēs redzam ka gandrīz visas pupiņas ir vienā izmērā un no šī var secināt ka ir vienadā līmeņi perimetrs, bet Bombay diezgan atšķiras no citām pupiņām.



Attiecīgi ekscentriskuma[9], mēs varam redzēt ka, visi pupiņas veidi ir salīdzināmi ovālas, jo pēc koeficienta var redzēt ka visi koef. > 0,4 -> tādēļ, var secināt ka ja vispopulārākās pupiņas šķiras ir vairāk ovālas nekā apaļi -> mēs varam prognozēt pēc tas atribūta ka ja pupiņu šķiras ir ne tik ovālas, viņas nav garšīgas.

Statistiskie rādītāji.



[10]

Kā ir labi redzams [10] no Feature Statistics, ir diskriptīvi redzami dažādi grupējamie sadalījumi. Area, Perimeter, Major AxisLength, AspectRatio ar vidējo Dispersijas vērtību: 0,654 un Vidējo vērtību: 0,252 ; Eccentricity, Extent, Solidity roundness ar vidējo Dispersijas vērtību: 0,162; un Vidējo vērtību: 0,765; ShapeFactor ar Dispersijas vērtību: 0,298; un Vidējo vērtību: 0,493.

Pirmās Daļas Secinājumi.

Izpētot līdz šim momentam visu to informāciju, kuru es paņemu no brīvas datu bāzes, es varu secināt par Orange rīku un par to ka atšķiras un no ka ir atkarīgas atribūti. Orange rīks ir diezgan lietderīgs rīks, un ar to ir diezgan viegli izpētīt klasificētu informāciju. Personīgi man ļoti patika ka ir organizēta šī programma un ka ātri tā strādā un cik viegli ir saprast kas ar ko ir saistīts, paldies User Friendly grafiskam interfeisam. Kas attiecas uz pupiņu atšķirībām – es pēc veiktajiem darbiem varu secināt, kā forma, izmērs un citi faktori ir atkarīgi no pupiņu šķires, un tā savukārt ir atkarīga no kontinenta uz kurā tā ir audzēta.

II daļa – Nepārraudzītā mašīnmācīšanās

2.daļas uzdevums:

Šajā darba daļā studenti veiks iepriekš izvēlētās datu kopas klasterizāciju. Darba I daļa sniedza studentiem izpratni par to, kādas pazīmes (atribūti) un klases ir datu kopā un cik labi datu objekti sadalās klasēs. Šīs darba daļas mērķis ir, izmantojot klasterizācijas metodes, vēl vairāk izpētīt datu kopu, lai noskaidrotu, vai iepriekš izdarītie secinājumi par datu kopas struktūru ir spēkā.

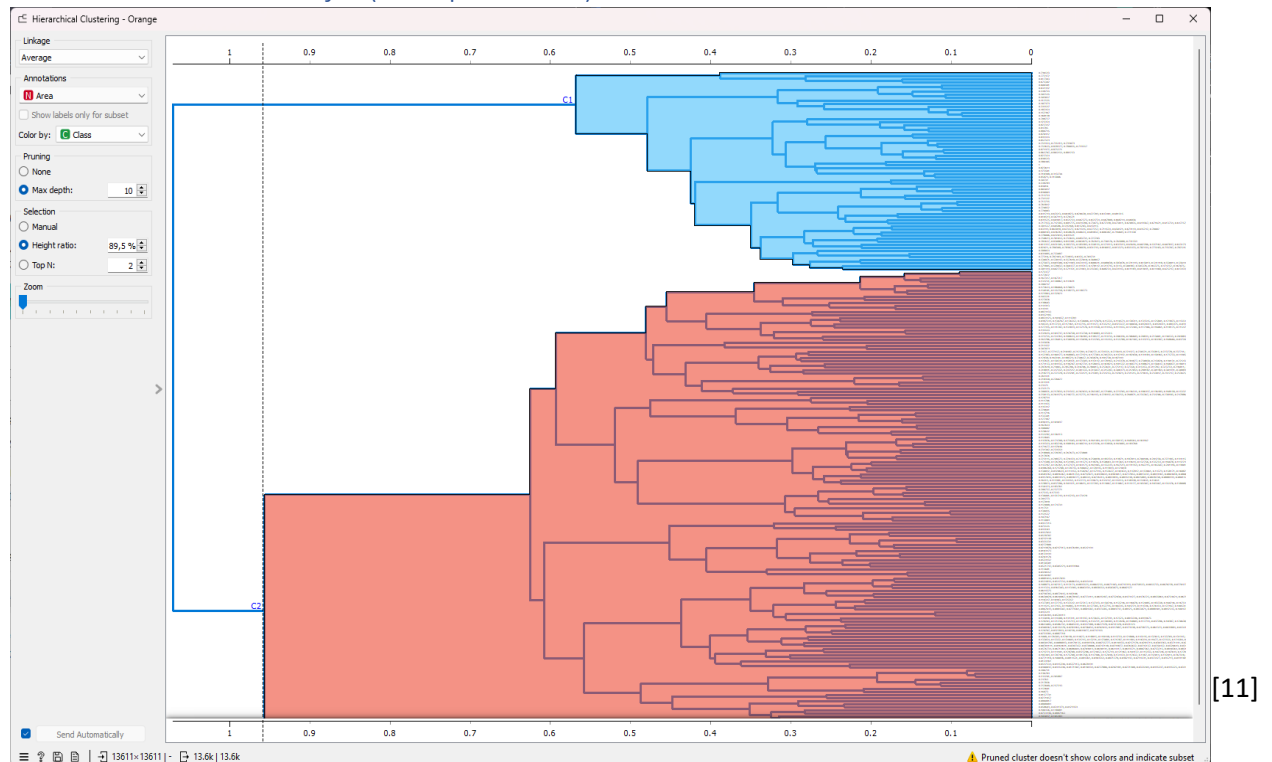
Lai izpildītu šo darba daļu, studentiem ir jāveic šādas darbības:

1. Jāpielieto divi studiju kursā apskatītie nepārraudzītās mašīnmācīšanās algoritmi: (1) hierarhiskā klasterizācija un (2) K-vidējo algoritms.
2. Hierarhiskās klasterizācijas algoritmam ir jāveic vismaz 3 eksperimenti, brīvi pārvietojot atdalošo līniju un analizējot, kā mainās klasteru skaits un saturs;
3. K-vidējo algoritmam ir jāaprēķina Silhouette Score vismaz 5 dažādām k vērtībām, un jāanalizē algoritma darbība.

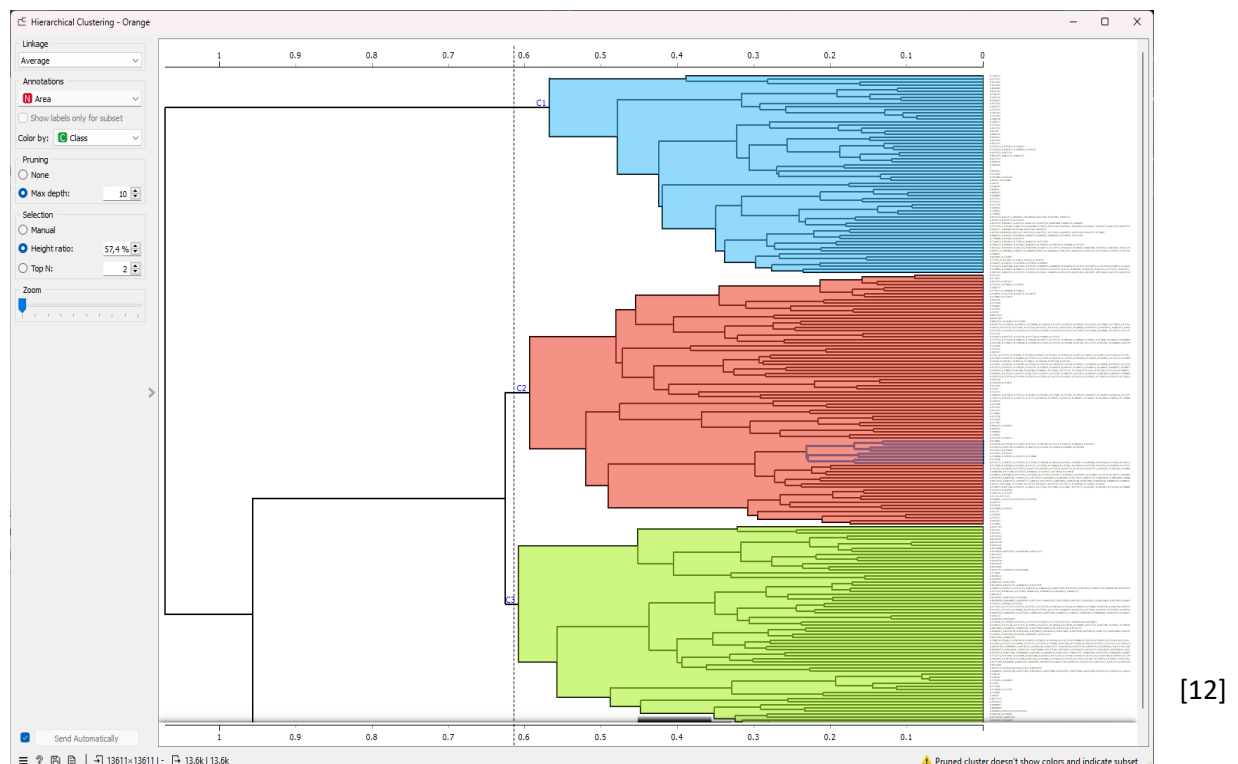
Darba atskaitē ir jāiekļauj šāda informācija par šo darba daļu:

- Katram algoritmam ir jāapraksta Orange rīkā pieejamie hiperparametri un to nozīme.
- Katram algoritmam ir jāapraksta veiktie eksperimenti, skaidri norādot izmantotās hiperparametru vērtības, un sniedzot secinājumus par algoritma darbību no tā viedokļa, cik iegūtie rezultāti atbilst zināmajam klašu skaitam datu kopā.
- Balstoties uz abu algoritmu darbības analīzi, ir jādod studenta secinājumi par to, vai datu kopā esošās klases ir labi vai slikti atdalāmas

Hierarhiskā klasterizācija: (3 eksperimenti)

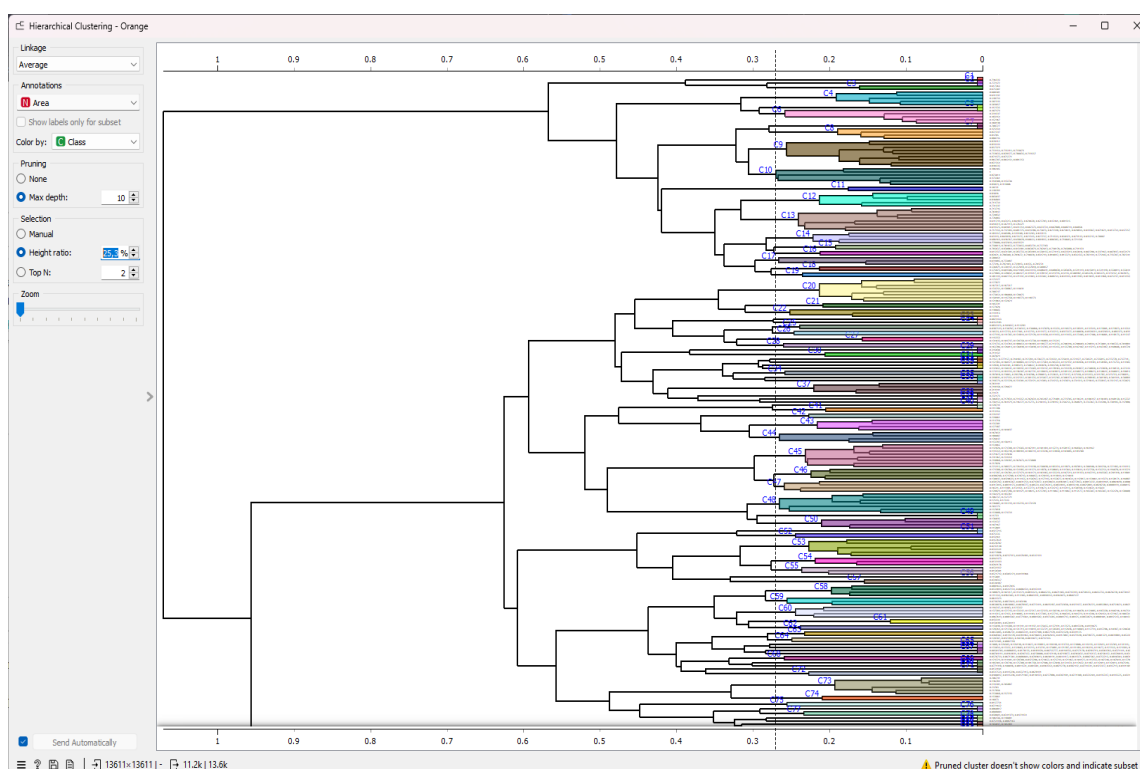


Ja Height Ratio [11] atdalošo līniju pārvietot uz 89,5% marku, tad mēs varam redzēt tādu sadalījumu kur ir divas grupas, un algoritms iekļauj iekša šitos divos klasteros.



Pārvietojot [12] līmeņlīniju uz 57,4% marku, jau ir izveidoti 3 lieli klasteri, lielākais no klasteriem sadalījās divos, un katru līmeni lielākais dalīsies uz diviem, bet katru nākamo līmeni tas tiek darīts

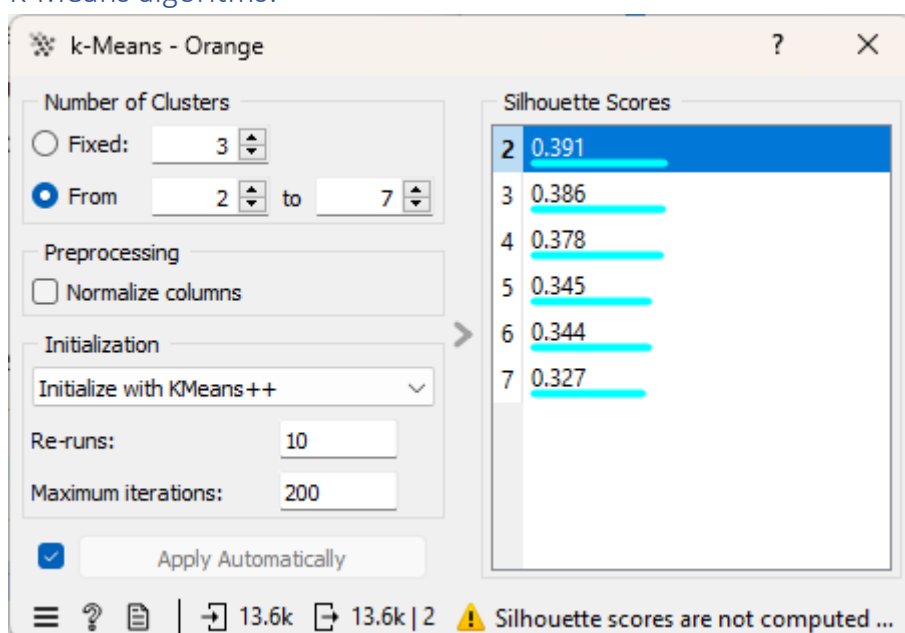
pārvietojot līmeņlīniju uz mazāko vērtību. Piem: 89,5-57,4=32,1% pārvietojums. Tādēļ no 2 uz 3 klasteriem. Bet ja pārvietosim vēlreiz uz 32,1% būs lielāks skaits.



[14]

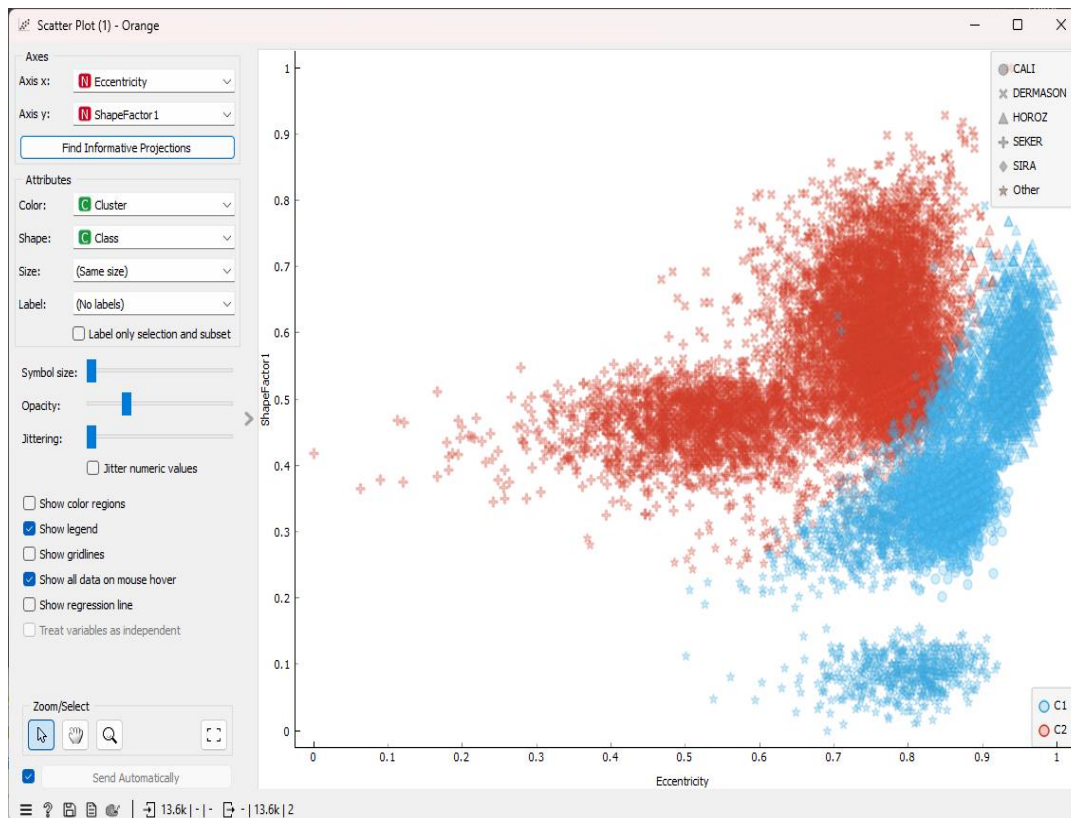
Un ka bija teikts, pēc pārvietojumā uz tādu pašu vienību[14] dabūjam 25,3% marku, un te mēs redzam ka klasteru izveidojas diezgan daudz.

K-Means algoritms:



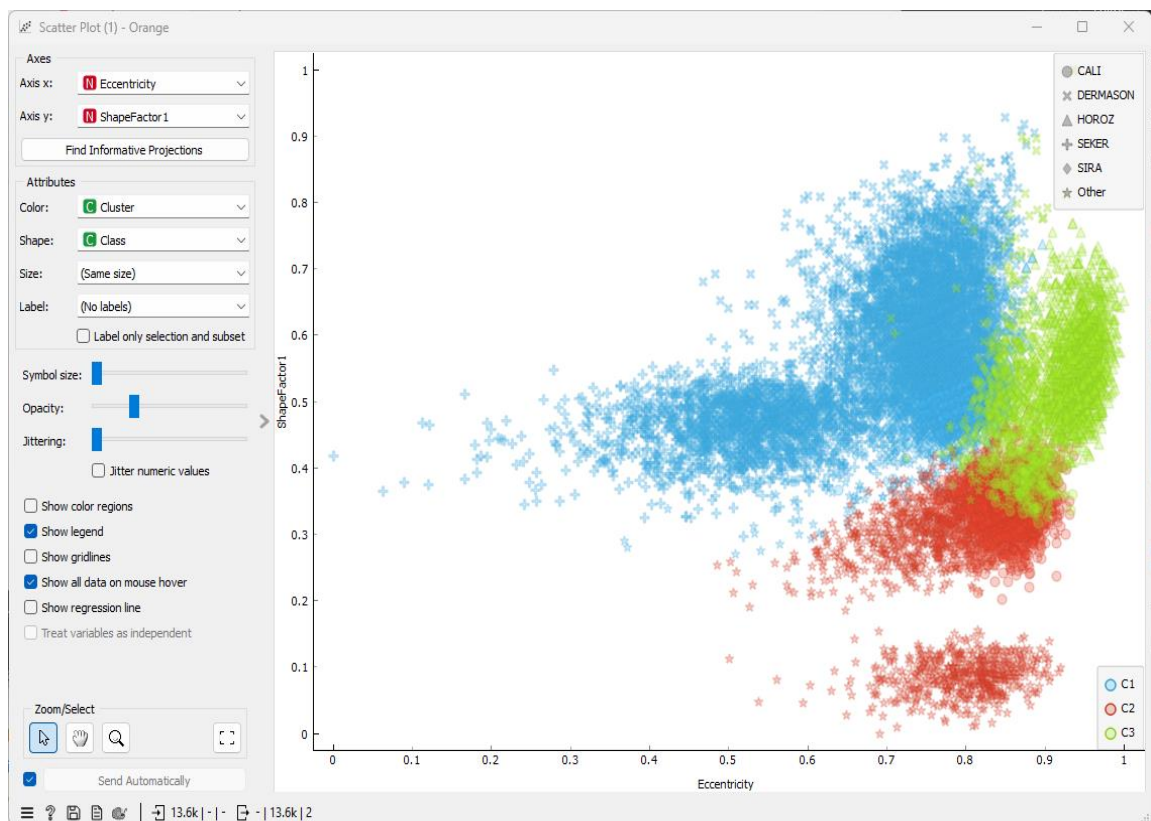
[15]

Silhouette Score [15] ir sadalīta no 2 līdz 7, un ir atkarība no klasteru skaitu, jo lielā skaits, jo mazāks Score.



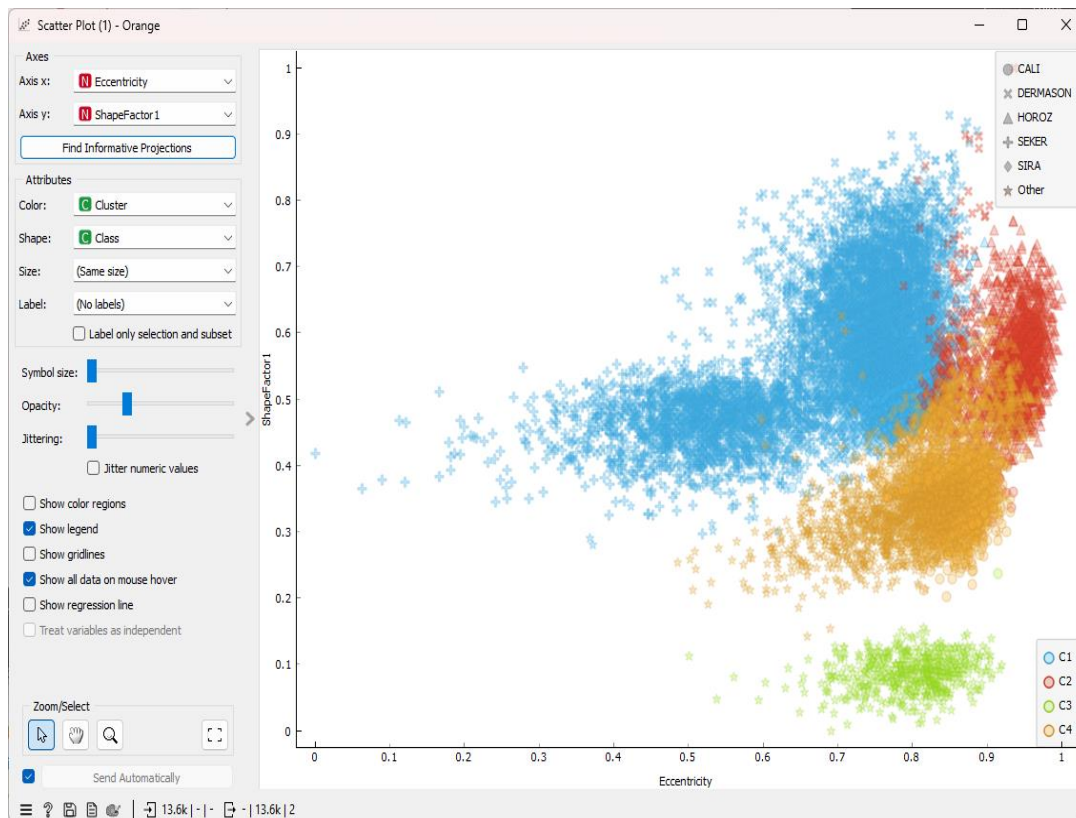
[16]

Divi klasteri [16] ir vislabākā vizualizacija, jo klasteri pārklājas savu starp ļoti maz.



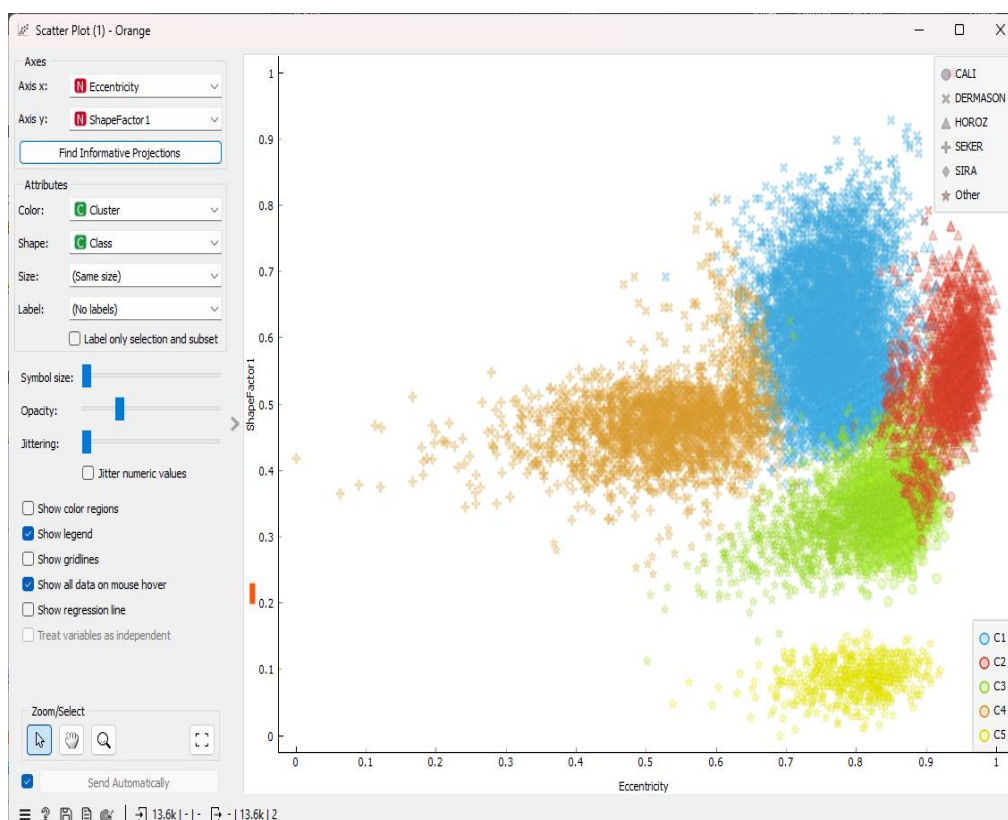
[17]

Trīs klasteri [17] jau pārklājas savu starpā jau lielāk.



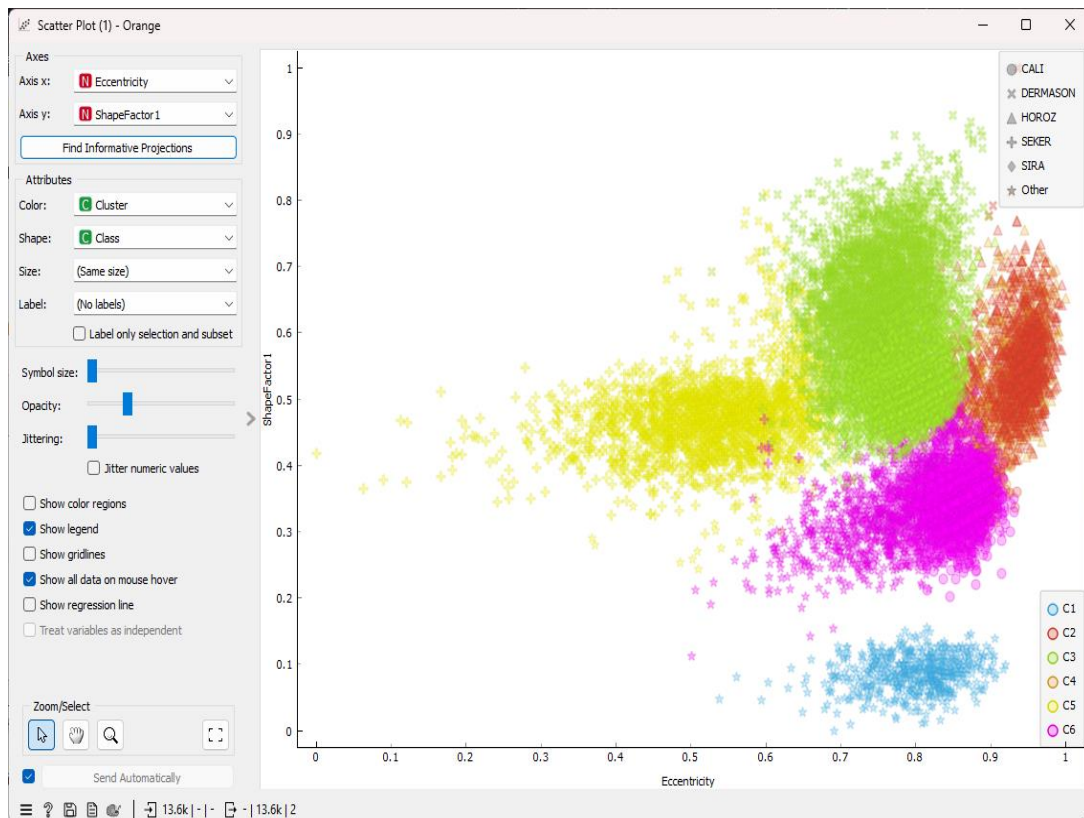
[18]

Ar četriem [18] nekas neizmainījās pārklāšanas jomā.



[19]

Tapāt ar pieciem [19] klasteriem atšķirības no četriem gandrīz nav.



[20]

Un ar šiem klasteriem rezultāts [20] ir vēl sliktāks, jo pārklāšanas ir vislielākās, no kurienes var secināt ka vislabākais sadalījums ir pēc diviem klasteriem, un Silhouette Score arī rada ka pie 2 klasterim ir vislielākais Score, vistuvākais vieniniekam.

III daļa – Pārraudzītā mašīnmācīšanās

Trīs algoritmi kuri teiks aplūkoti:

kNN	-	K-Nearest Neighbour Algorithms
Neural Network	-	Neirona tīklu algoritms (obligāts)
SVM	-	Support Vector Machine algoritms

kNN

“Iss” apraksts.

k-tuvākajiem kaimiņiem ir pārraudzītā mašīnmācīšanās algoritms, kas var būt izmantots kā klasifikācijā, tā un regresijas pētīšanai. kNN ir viens no vienkāršākajiem algoritmiem, bet tam ir vairākas lietderīgas lietojumprogrammas, kā piemēram, preču katalogu ieteikšana vai nu zemes izmantošanas kartēšana.

Svarīgi, ka no lielākajām kNN priekšrocībām ir viņa vienkāršība un minimāla priekš apmācības nepieciešamība. Bet kNN ir diezgan jutīgs pret datu skaņām un dēļ veiktspēju vajadzībām tas nav piemērots lieliem datu kopumiem. Regresijas gadījumā, ja ir nepieciešams paredzēt skaitlisku iznākumu, tādu kā, jebkādu cenu, algoritms darbojas līdzīgi. Tas atrod k tuvākos kaimiņus un vidējo iznākumu starp šiem kaimiņiem piešķirot kā prognozi iegūtas vērtības.

kNN izvēlēs motivācijā ir vienkāršā ka pats algoritms: algoritms ir vienkāršs, bet laika trūkst, un algoritms ir pazīstams, jo mēs tādu apjukām lekcijas laikā.

Informācijas avoti:

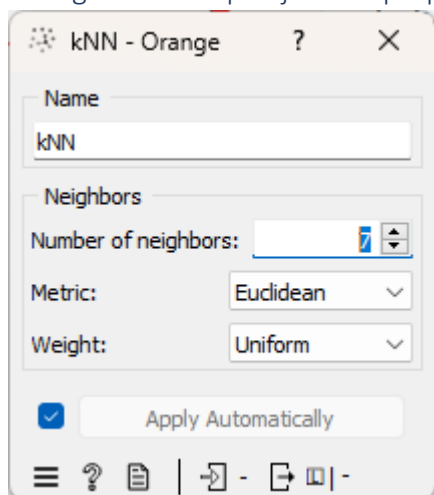
<https://www.ibm.com/topics/knn>

<https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>

<https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

Mākslīga intelekta pamati lekcijas

Orange rīkā kNN pieejamie hiperparametri un to nozīme[21]:



[21]

Name – to ka output būs atspoguļots saistītas tabulas.

Number of neighbors: atspoguļo cik kaimiņi būs ņemti vērā.

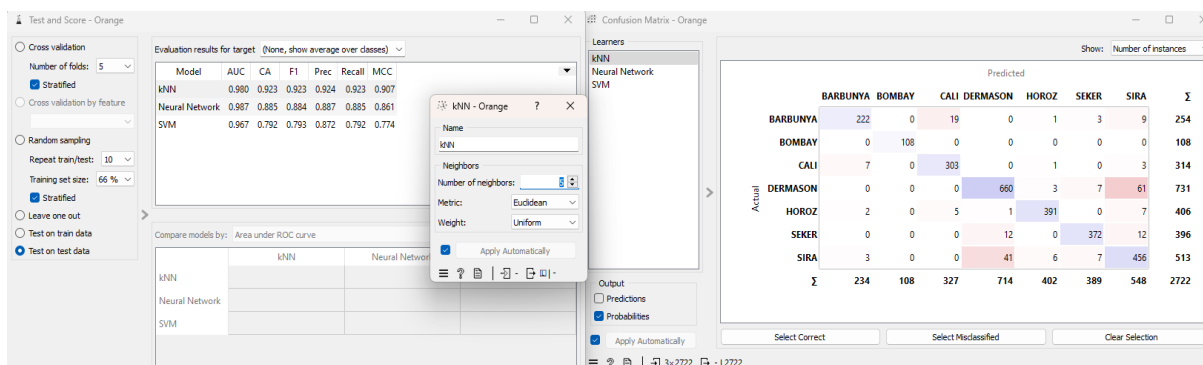
Metric:

- Euclidean: ņemts attālums starp diviem punktiem.
- Manhattan: absolūto atšķirības visiem atribūtiem summā.
- Chebyshev: noderīgs ja telpā ir vairākas dimensijas.
- Mahalanobis: attālums starp sadalījumu un to punktu.

Weight:

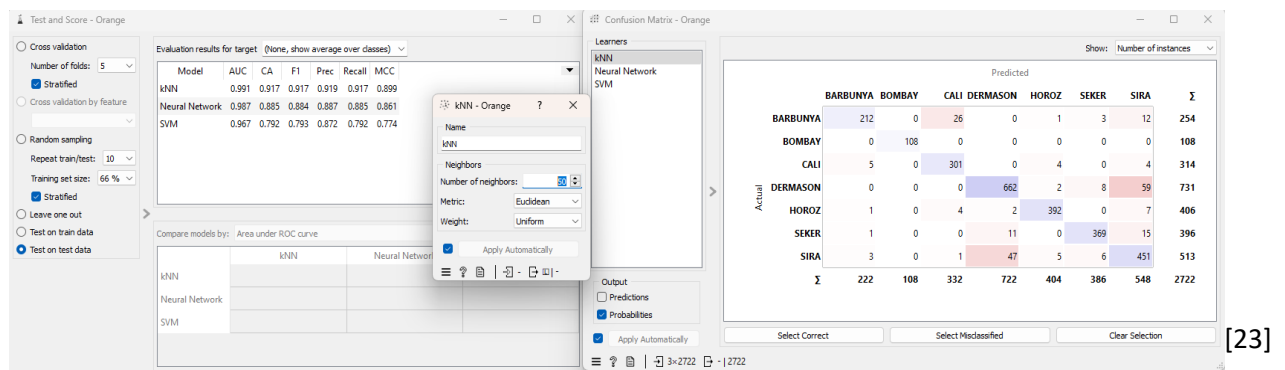
- Uniform: katram datu punktam ir vienādā pieejā nu nosvērums.
- By distances: ietekmē un nosvērums starp punktiem samazinās ar attālumu.

kNN peilietošana:

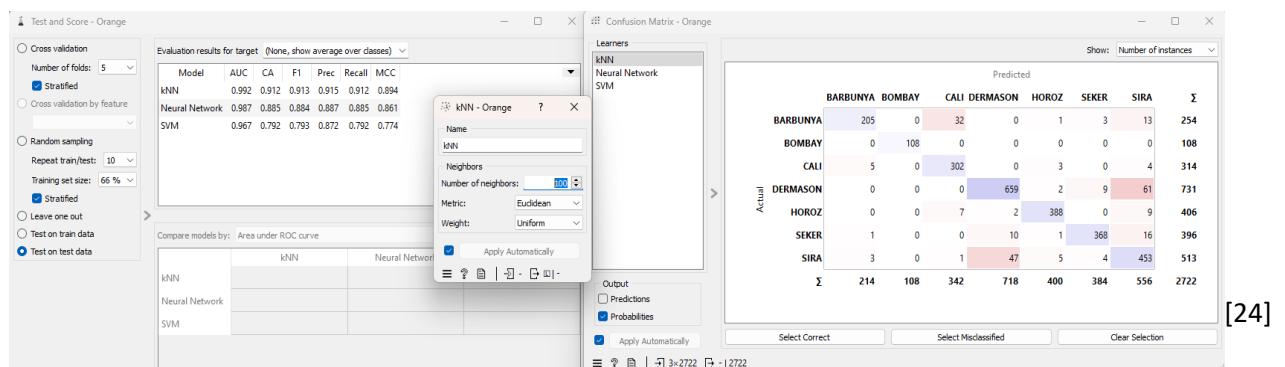


[22]

Ar kNN algoritmu [22] un lielo datu apjomu, ka bija minēts var būt grūti dabūt īsto vērtības precizitāti, bet galvenā ideja strādā, ka palielinot kaimiņu skaitu precizitātē samazinās. Piem. pie 5 kaimiņiem prec ir 0,924.

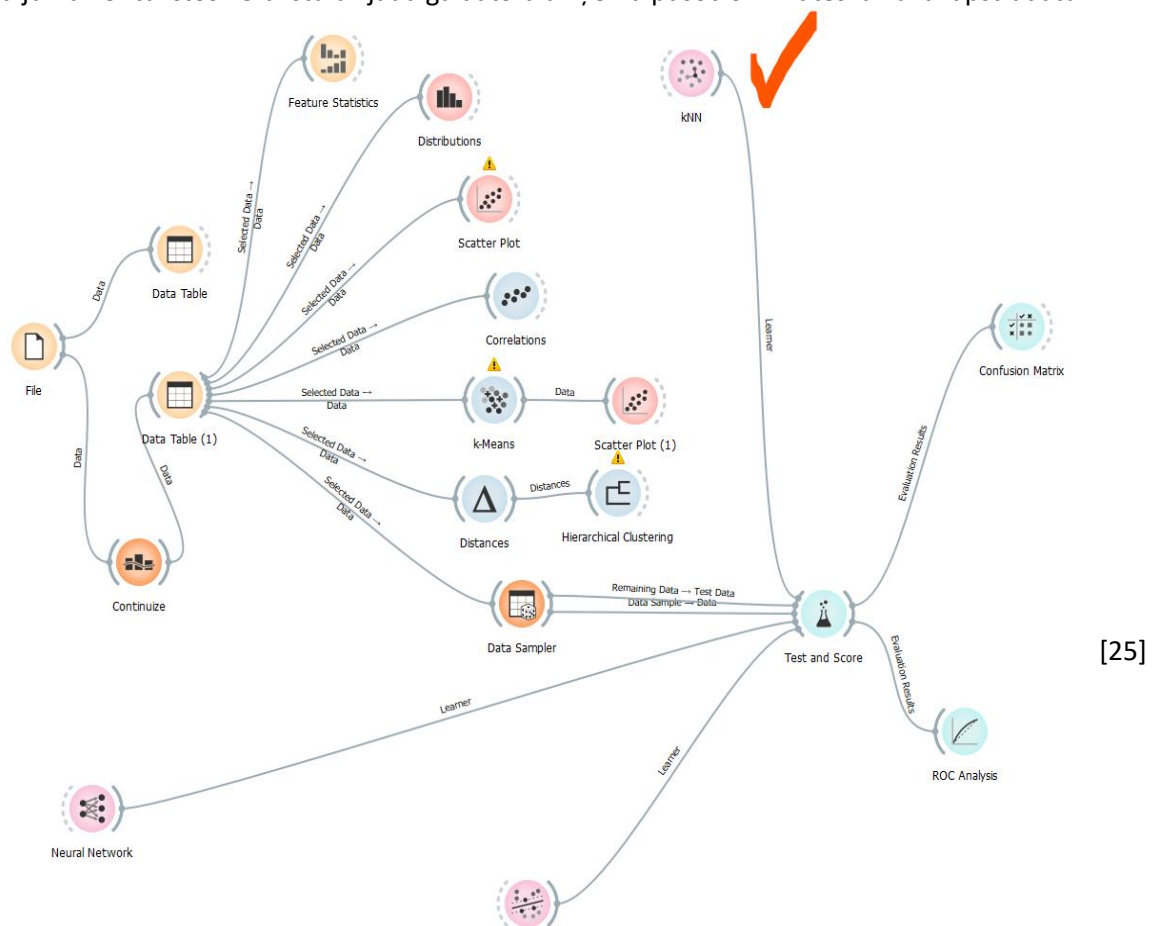


Ka mēs varam redzēt[23] palielinot kaimiņu skaitu no 5 līdz 50 precizitāte samazinājās līdz 0,919



Un ka bija gaidīts, kad kaimiņu skaits tika palielināts līdz 100 kaimiņiem[24], precizitāte samazinājās līdz 0,915.

Un secinot par kNN algoritmu, ka jau bija teikts, tas nav visātrākais algoritms lai strādātu ar lielo kopu, manā gadījumā 13+tukstoši ierakstu uz jaudīgā datora aizņēma pusotro minūtes laiku lai apstrādātu.



SVM algoritms

Kāpēc SVM: es mēģināju izmantot Linearo Regresiju, bet tai vajag lai Target bija numeric tipa, tāpēc man tas nebija derīga, un SVM bija vel vien trīs burtu algoritms.

“Iss” apraksts:

Support Vector Machine izmanto tā saucamo atbalsta vektoriem, kuri ir izmantoti klasifikācijas veikšanai vai regresijas pētīšanai. Klasifikācijā SVM algoritms mēģina sadalīt datu kopu divās klasēs, izveidojot lēmumu robežu telpā, kas sadala datu punktus klasēs, tālāk izmantojot to, kā tie atrodas attiecībā pret šo robežu. SVM algoritms meklē atbalsta vektorus, (datu punktus), kuri atrodas tuvāk robežai telpā un palīdz noteikt lēmuma robežu. SVM ir spēcīgs mašīnmācīšanās algoritms, kas ir noderīgs dažādās problēmās, tostarp attēlu klasifikācijā, teksta klasifikācijā, finanšu prognozēšanā un medicīnas diagnostikā.

SVM var izmantot dažādas funkcijas, kas nosaka, kā tiek veidots lēmumu robežu. Parasti tiek izmantotas lineāras un nelīnijas funkcijas, piemēram, polinomiālās un radiālās funkcijas. SVM algoritms meklē optimālu lēmumu robežu, kas ir tāda robeža, kas sadala datu punktus klasēs ar maksimālo atstatumu starp robežu un tuvākajiem atbalsta vektoriem.

Datu avoti:

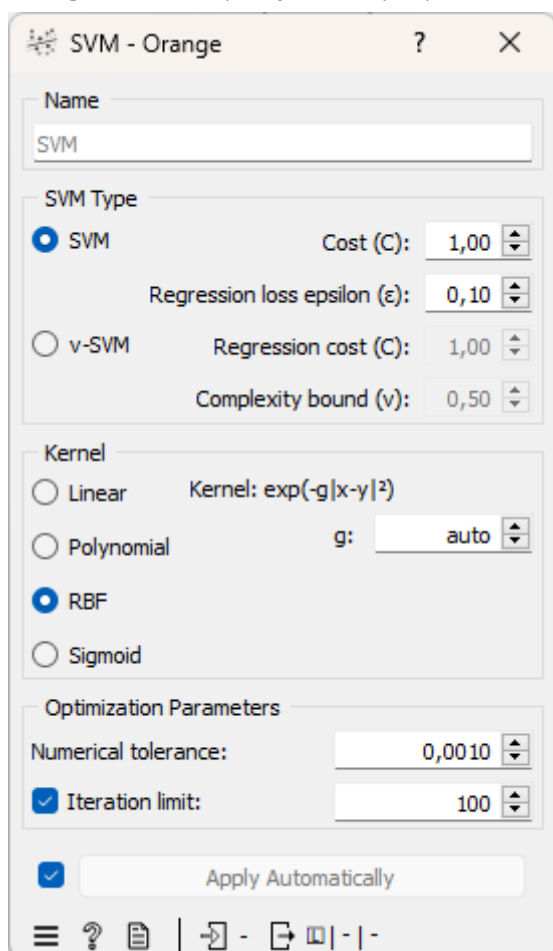
<https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>

https://en.wikipedia.org/wiki/Support_vector_machine

<https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/>

<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>

Orange rīkā SVM pieejamie hiperparametri un to nozīme:[26]



The screenshot shows the 'SVM - Orange' dialog box with the following settings:

- Name:** SVM
- SVM Type:** SVM (selected), Cost (C): 1,00, Regression loss epsilon (ϵ): 0,10
- v-SVM:** v-SVM (unselected), Regression cost (C): 1,00, Complexity bound (v): 0,50
- Kernel:** RBF (selected), Kernel: $\exp(-g|x-y|^2)$, g: auto
- Optimization Parameters:** Numerical tolerance: 0,0010, Iteration limit: 100 (checked)
- Apply Automatically:** checked

[26]

Name: teksts, kurš būs atspoguļots testos

SVM Type:

- Cost(C): Zudumu maksa priekš regressijas un klasifikācijas testiem
- Regression loss epsilon: mainīgais parametrs priekš EpsilonSV modelim, kurš attiecas uz taisnības tuvinātiem vērtībām attālumu, sods nav prognozēta vērtība.
- Regression cost: tas pats ka Cost, bet domāts regresijas uzdevumiem.
- Complexity bound: veica apmācības kļūdu daļu augšējās robežas un noteica atbalsta vektorus apakšējās robežas.

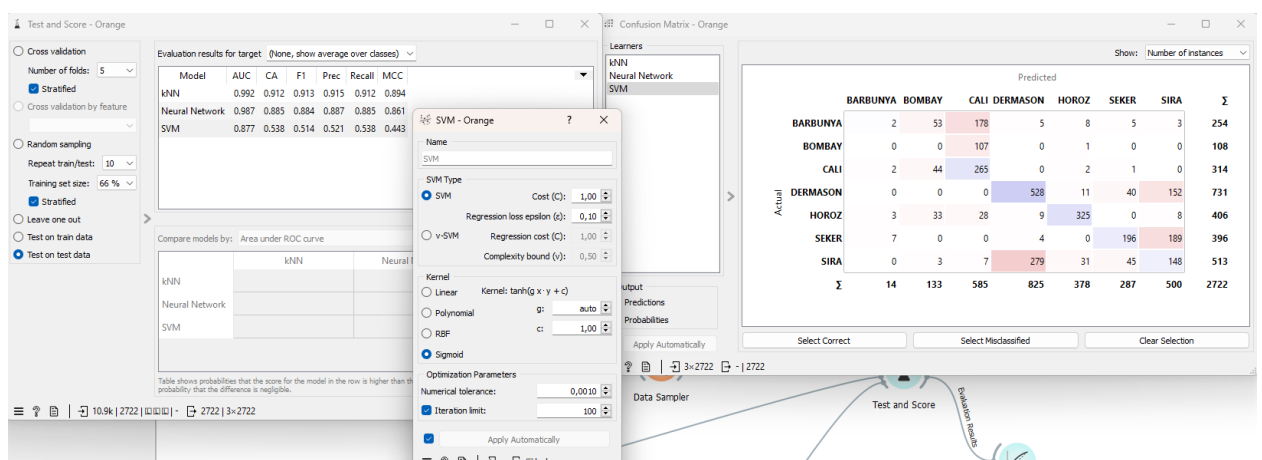
Kernel:

- Linear: tiek izmantots, ja datu punkti ir labi sadalīti ar vienkāršu līniju un klasifikācija jāveic telpā ar mazām dimensijām.
- Polinomial: šis kernelis ir noderīgs, ja dati nav lineāri atdalāmi un klasifikācijas robežas jāveido ar polinomu funkciju. Polinomu pakāpe jāizvēlas kā hiperparametrs, un, ja pakāpe ir pārāk liela, algoritms var tikt pārēķināts.
- RBF: tas ir bieži izmantots kernelis, kas ļauj izveidot nelīnijas lēmumu robežas. RBF kernelis darbojas, ievērojot katru punktu attālumu no centra punkta, un kā attālums tiek izmantots Gauss funkcijas vērtība. RBF kernelis tiek uzskatīts par vienu no vispārējākajiem SVM kerneliem.
- Sigmoid: ja dati nav lineāri atdalāmi, un lēmuma robežas jāveido ar hiperbolisko tangensu.

Numeric tolerance: Numeriskā tolerance ir skaitliska vērtība, kas nosaka, cik tuvu divi skaitļi var atrasties viens otram, lai tos uzskatītu par vienādiem.

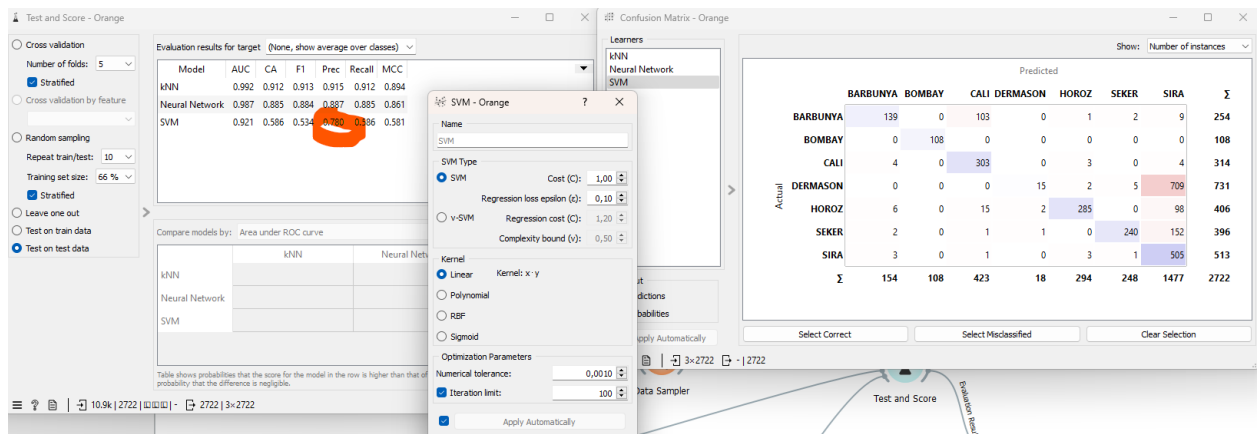
Iteration Limit: cik maksimāli iterāciju var veikt.

Pielietojums:



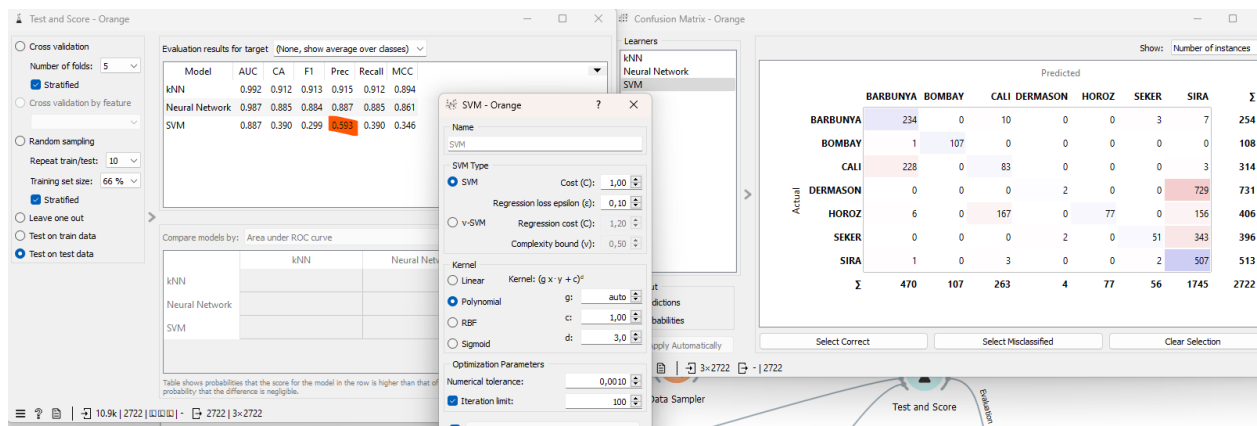
[27]

Ka ir redzams [27] bildē, kNN dod lielāko precizitāti, nekā SVM ar Sigmoid kerneli



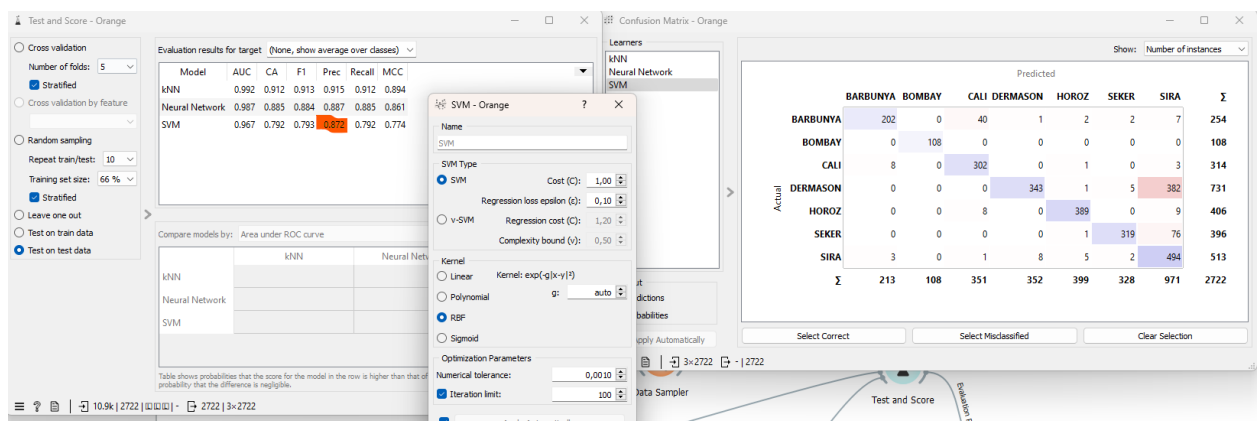
[28]

Ka ir redzams [28] bildē, no SVM ir gandrīz labākais rezultāts ar Lineāro kerneli, bet mazāks par kNN.



[29]

Ka redzams[29] bildē, Polinominal arī nav labākais no variantiem

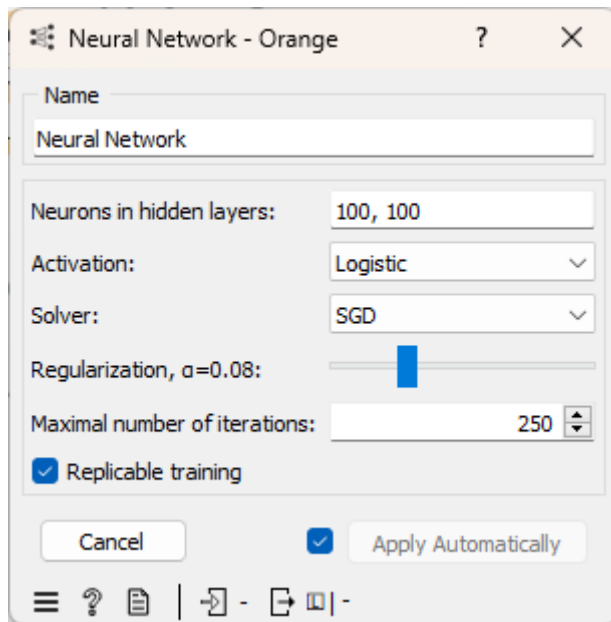


[30]

Ka redzams[30] bildē, vislabākais no SVM variantiem ir RBF kernelis, jo tas dod vislielāko precizitāti: 0,872, bet mazāk nekā kNN.

Neural Network

Orange rīkā SVM pieejamie hiperparametri un to nozīme:[31]



[31]

Name: tas tiks atspoguļots testos un atskaitē.

Neurons in hidden layers: Neironu skaits slēptajos slāņos.

Activation:

- Identity: linearo sašaurinājumu īstenošana ja aktivēts bez operācijām.
- Logistic: sigmoid f-ja.
- tanh: hiperboliska pieskares f-ja.
- ReLU: Rektificētas Lineārās vienības f-ja.

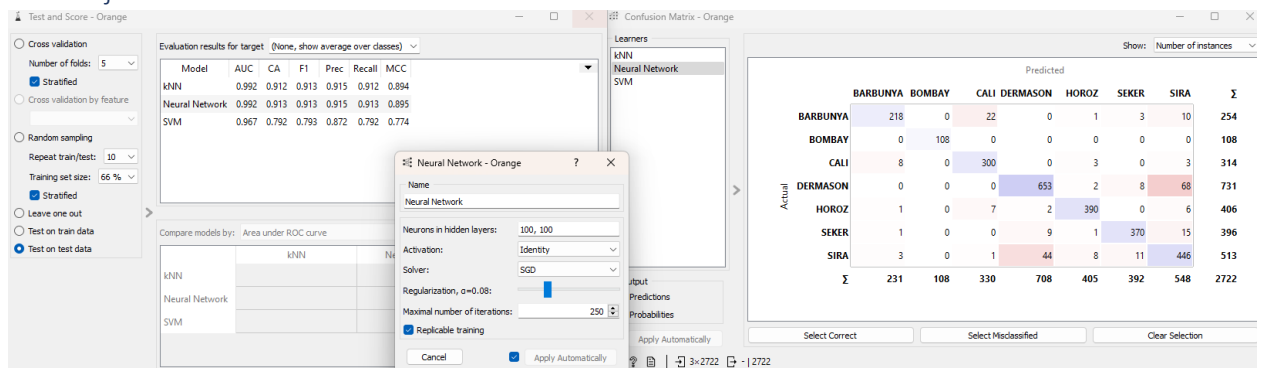
Solver:

- L-BFGS-B: kvaziņjūtona metodes optimizācijā
- SGD: stohastiskā gradienta nolaišanas metode.
- Adam: optimizācija balstītā uz stohastiskas gradienta.
- Alpha: termiņa regulējošs parametrs.

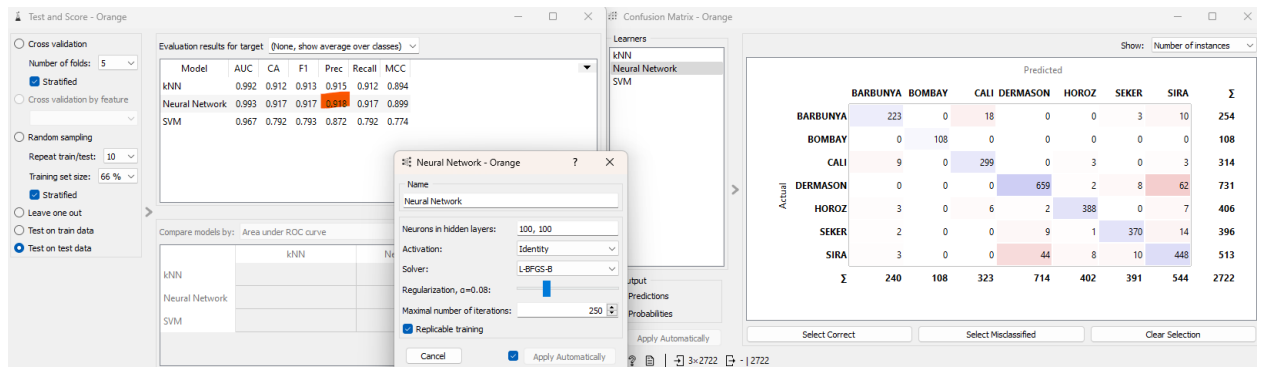
Maximal number of iteration: maksimālo iterāciju līmenis.

Replicable training: nodrošina mašīnmācīšanās modeļu uzticamību un efektivitāti.

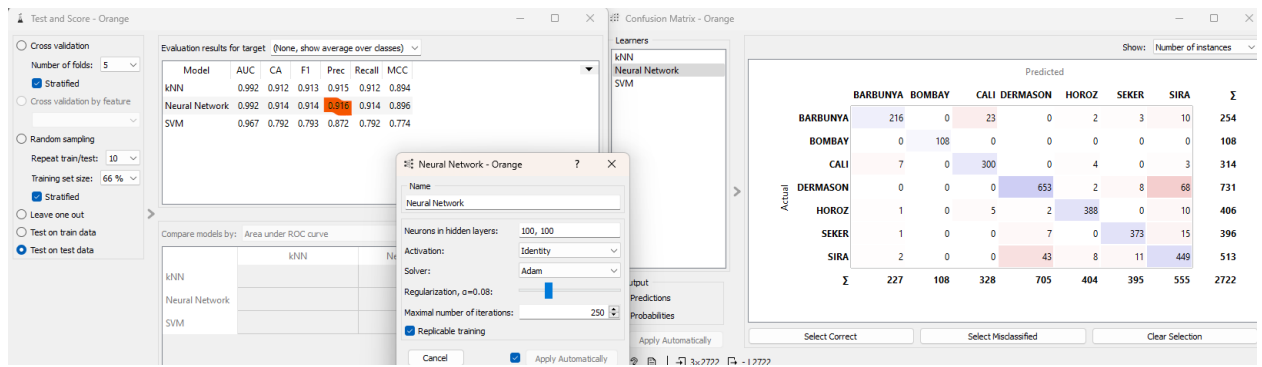
Pielietojums:



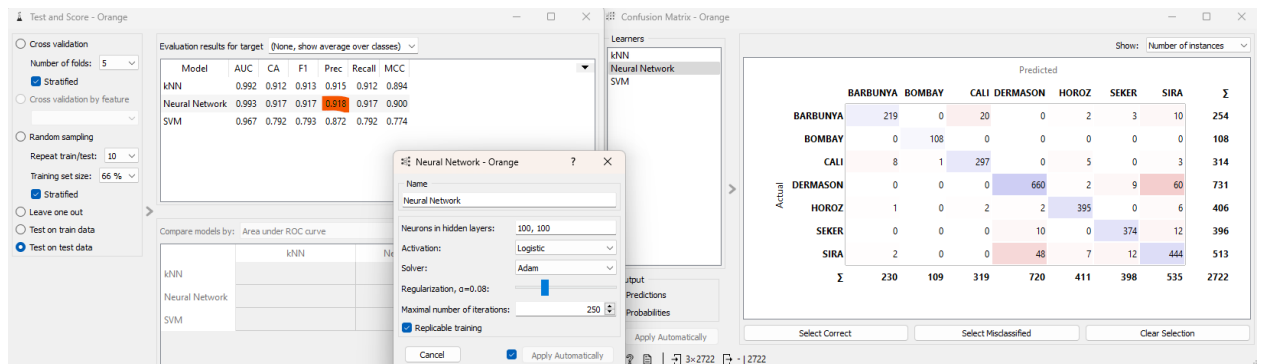
[32]



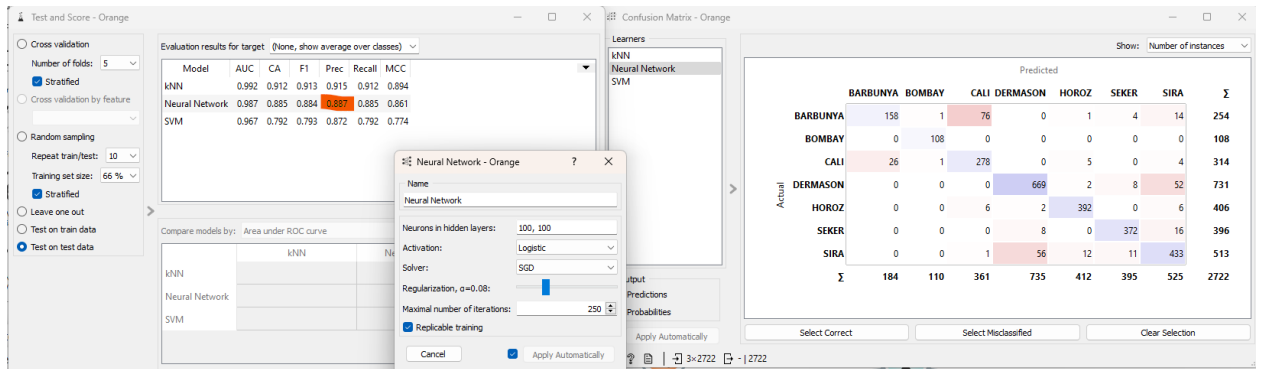
[33]



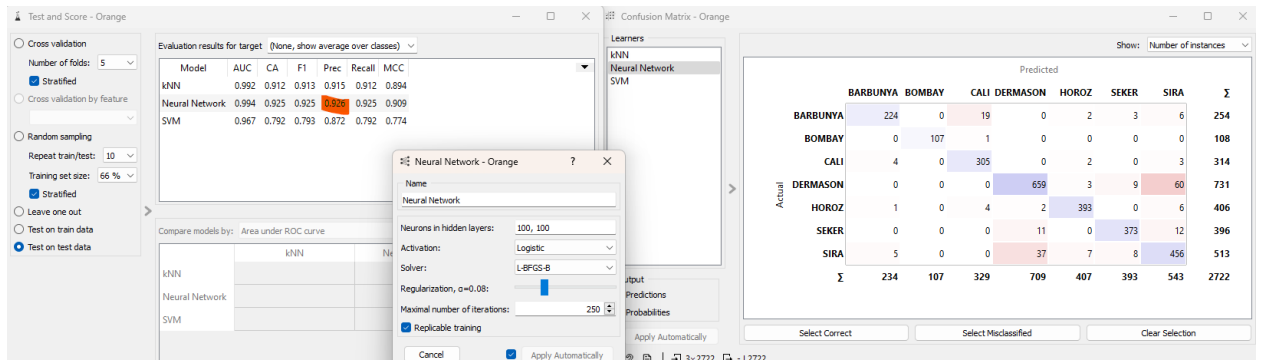
[34]



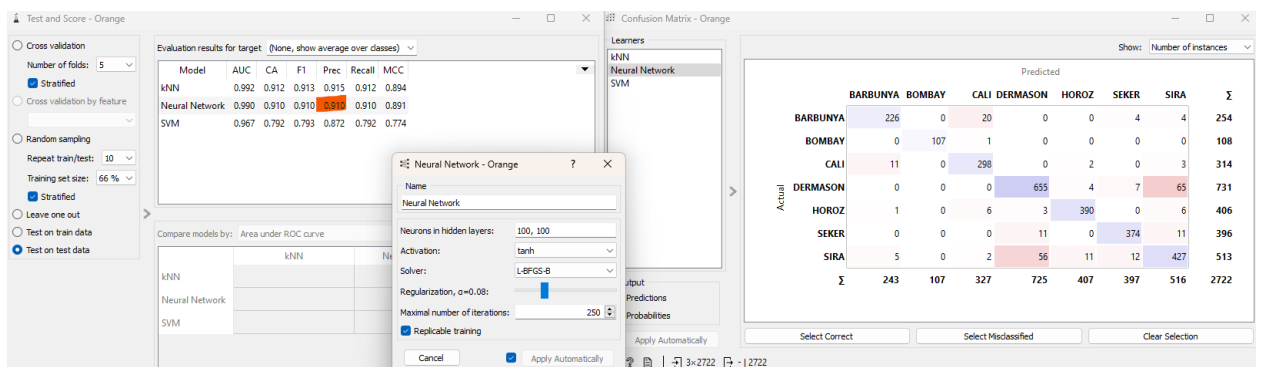
[35]



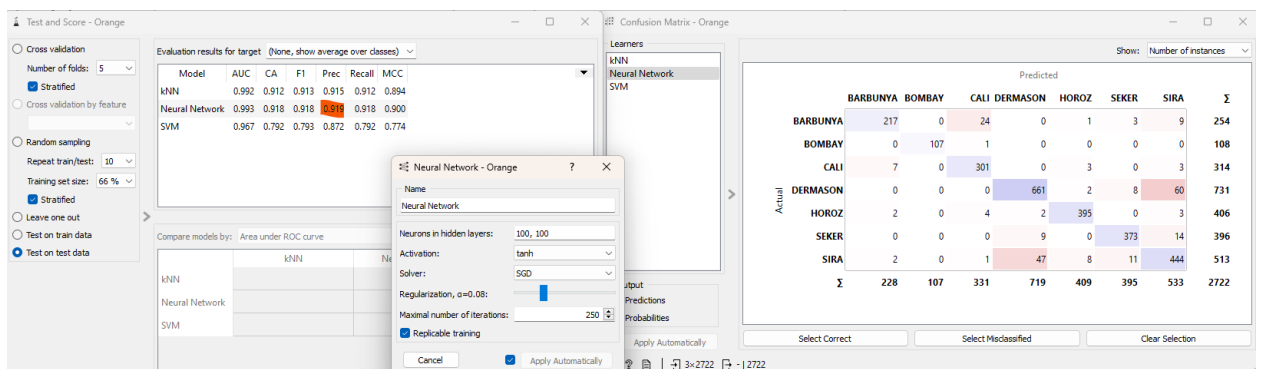
[36]



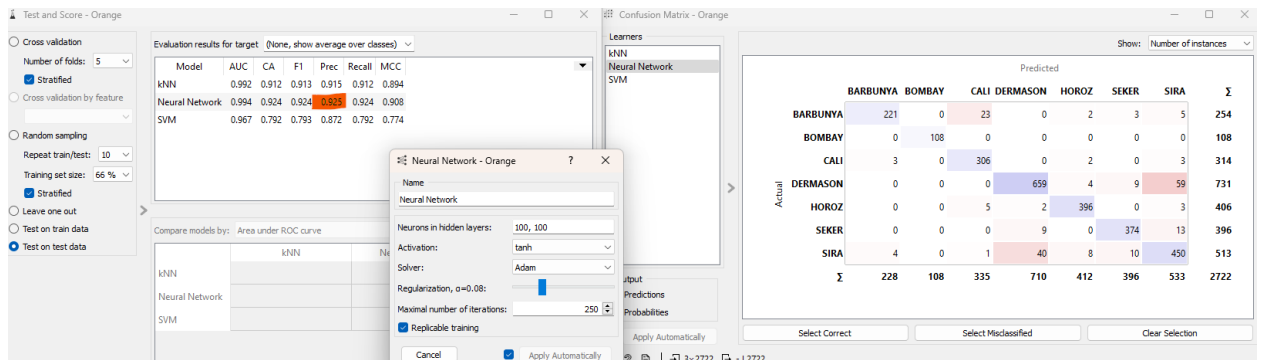
[37]



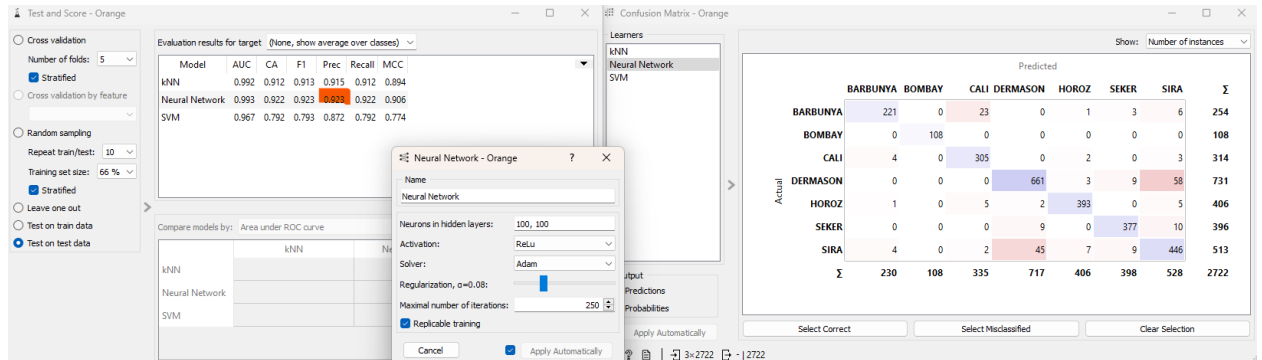
[38]



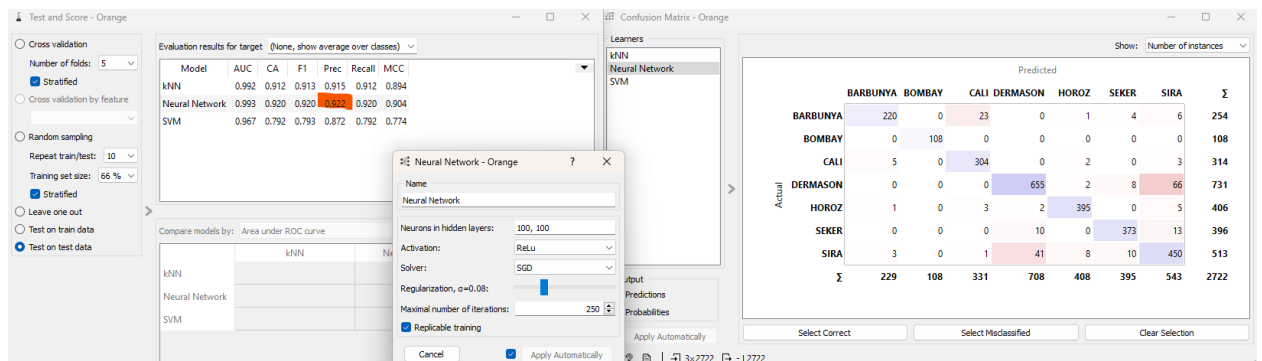
[39]



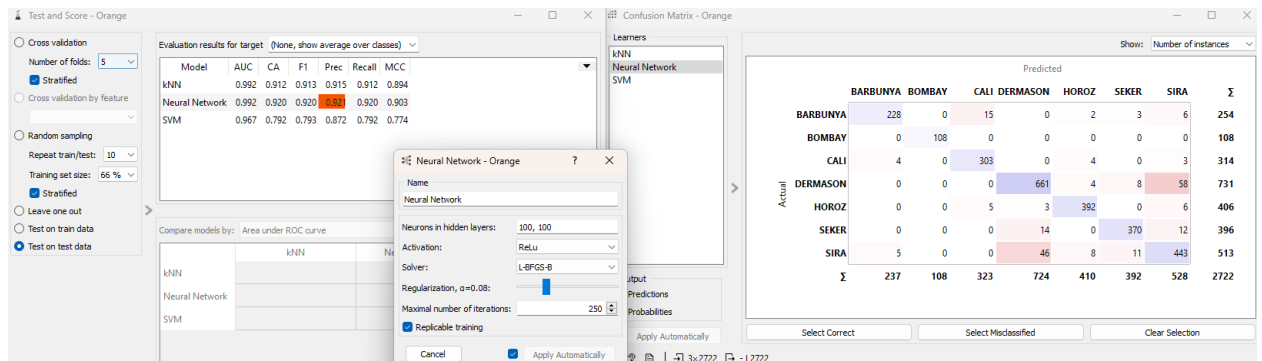
[40]



[41]



[42]



[43]

Secinot par neirona tikliem no 12 dažādu eksperimentu varu secināt ka vislielākā precizitāte ir 0.926 ja neirona tikls ir uzstādīts [37] bildē uz Activation Logical un Solver L-BFGS-B, lielam monotonam datu kopām var derēt tādi uzstādījumi. Bet vismazākā vērtībā bija ja bija uzstādīti parametri: Logical + SGD bildē[36].

Secinājumi:

Pēc visam veiktam darbībām ar mašīnmācīšanās algoritmiem, var secināt ka lielumam datu kopām, jāpielieto specializēto algoritmu, kurš varēs izmantot lielo datu skaitu, manā gadījumā tas bija 13600 datu ieraksti. Piemēram SVM algoritms nevarēja sasniegt pat 0.9 punktu precizitāti, un doda maksimum 0.872 bildē [30]. Neirona Tīkla algoritms varēja dot precizitāti jau 0.926 [37] bildē un sanāca līderis šajā jomā. Bet visvienkāršākais algoritms kNN, kurš sasniedza 0.924 bildē[22], arī bija ne tik tālu ja runa iet par precizitāti. Bet ja runāt par ātrumu, Neirona tīkls – ir vislabākais algoritms, jo ir ātrs un precīzs.