**RĪGAS TEHNISKĀ UNIVERSITĀTE**

Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte

**Atskaite par otro praktisko darbu**

Studiju kurss “Mākslīgā intelekta pamati”

Komandas numurs: F\_7

Darba izpildītāji:

Andris Smirnijs 231DDB022,

Maksims Kaļiņins 231DDB029,

Natalja Veļičko 231DDB018,

Vladislavs Jegorovs 211DDB016,

Ēriks Kubanovs 201REB600

Mācībspēks: Alla Anohina-Naumeca

Saite uz projektu: [[saite]](https://github.com/Kladbicscze/Pd.2-MIP)

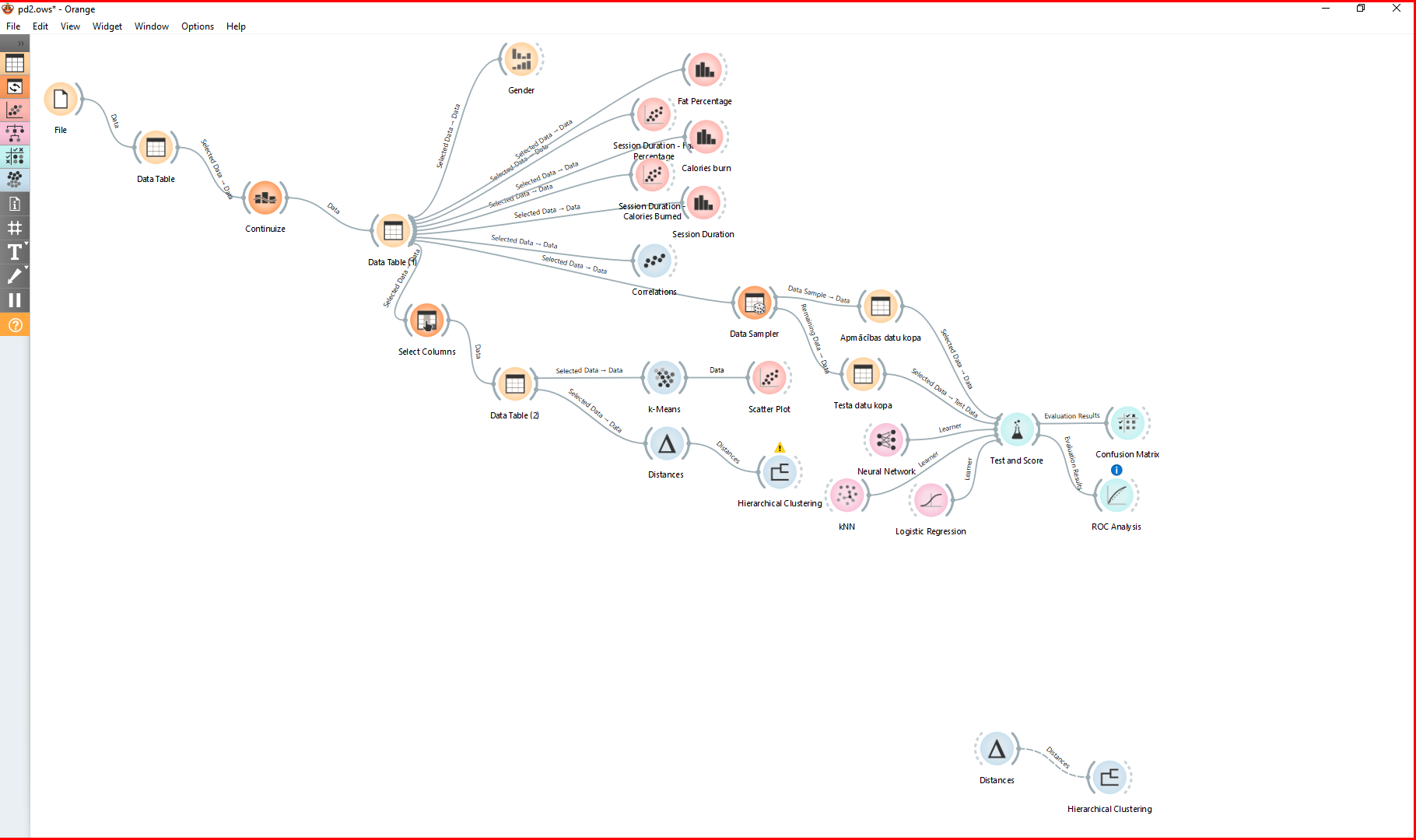
Saite uz datu kopu: [[saite]](https://www.kaggle.com/datasets/valakhorasani/gym-members-exercise-dataset)

2024./2025.studiju gads

# Mākslīgā intelekta rīku izmantošanas paziņojums

Mākslīgā intelekta rīki nebija lietoti šīs darbas izpildīšānas gaitā.

# Orange rīka darbplūsma



# I daļa

Informacija tika ņemta

## Datu kopas apraksts

### Datu kopas nosaukums: **Gym Members Exercise Dataset**

### Datu kopas avots: Datu kopa bija ģenerēta, izmantojot simulētus datus, kuru tiek pamatoti uz reālistisko vingrojumu modeļi sporta zālē, ņemot vērā: Publiski pieejamus fitnesa pētījumus; Nozares ziņojumus par vingrojumu un veselības tendencēm; Aptaujas, kas saistītas ar treniņu paradumiem, sirdsdarbības datiem un kaloriju patēriņu.

### Datu kopas izveidotājs un/vai īpašnieks: Seyed Vala Khorasani

Datu kopas problēmsfēras apraksts: Šis datu kopums sniedz detalizētu pārskatu par sporta zāles apmeklētāju vingrojumu režīmu, fiziskiem īpašībām un fiziskiem rādītājiem. Tas satur 973 ierakstus, tostarp galvenos snieguma rādītājus, piemēram, sirdsdarbības ātrumu, sadedzinātās kalorijas un treniņa ilgumu. Katrā ierakstā tiek iekļauti demogrāfiskie dati un pieredzes līmeņi, kas ļauj veikt visaptverošu fiziskās sagatavotības modeļu, sportista progresa un veselības trendu analīzi. [[saite]](https://www.kaggle.com/datasets/valakhorasani/gym-members-exercise-dataset)

Datu kopas licencēšanas nosacījumi:Apache 2.0[[saite]](https://www.apache.org/licenses/LICENSE-2.0)

Informācija par datu kopas savākšanas veidu vai procedūru:

Lai izveidotu datu kopu:

Galvenie mainīgie: Definēti, pamatojoties uz fitnesa pētījumos izplatītiem rādītājiem, tostarp sirdsdarbības ātrumu (BPM), sesijas ilgumu, sadedzinātās kalorijas, ķermeņa izmērus (vecumu, svaru, augumu) un pieredzes līmeni.

Dalībnieku profili: Ģenerēti, lai pārstāvētu dažādu sporta zāles apmeklētāju loku, ar treniņu intensitāti un pieredzes līmeni, kas nejaušināti sadalīts reālistiskos diapazonos.

Vingrinājumu modeļi: Simulēti, izmantojot datu vidējos rādītājus no fitnesa ziņojumiem, ar: Treniņu biežumu no 3 līdz 6 dienām nedēļā atkarībā no pieredzes līmeņa, Sesiju ilgumu no 0,5 līdz 2 stundām, Sadedzinātās kalorijas un ūdens uzņemšana tiek aprēķināta, pamatojoties uz treniņa veidu, BPM un demogrāfiskajiem faktoriem.

Datu strukturēšana: Dati tika strukturēti CSV formātā ar skaidri definētām kolonnām, lai nodrošinātu analīzes vienkāršību.

## Datu kopas satura apraksts

Datu objektu skaits datu kopā: 973

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

A screenshot of a computer

AI-generated content may be incorrect.

### Klašu skaits datu kopā: 3

*Klašu apraksts:*

**Experience Level – 1** – (Beginner) Tas ir cilvēks, kurš piedalās sportzalā mazāk, nekā sešus mēnešus. Viņš nevar pacelt lielu svaru, ir samērā mazs. Parasti, par Iesācējiem tiek parskatīti tie cilvēki, kuram nav pieredzes sportām. [[saite]](https://bonytobeastly.com/beginner-intermediate-advanced-lifter/)

**Experience Level – 2 –** (Intermediary) Tas ir cilvēks, kurš piedalās sportzalā lielāk, nekā sešus mēnešus, bet mazāk, nekā divi gadi. Parasti, par Starpniekiem tiek parskatīti tie cilvēki, kuri zino savu rutīnu un pie tās pieturas. [[saite]](https://bonytobeastly.com/beginner-intermediate-advanced-lifter/)

**Experience Level – 3 –** (Advanced) Tas ir cilvēks, kurš piedalās sportzalā lielāk, nekā divi gadi. Parasti, par Ekspertiem tiek parskatīti tie cilvēki, kuri atrodas sportzalē sacensību dēļ, kuri grīb sasniegt jaunus augstumus un rekordus. [[saite]](https://bonytobeastly.com/beginner-intermediate-advanced-lifter/)

*Datu objektu skaits, kas pieder katrai klasei:*

| **Pazīmes apzīmējums/nosaukums** | **Datu Objektu Skaits** |
| --- | --- |
| **Experience Level - 1** | 376 |
| **Experience Level - 2** | 406 |
| **Experience Level - 3** | 191 |

### Pazīmju apraksts:

**Age** – Vecums, gados.

**Gender** – Dzimums, (Male/Female).

**Weight (kg)** – Svars, kilogrammos.

**Height (m)** – Augstums, metros.

**Max\_BPM** – Maksimālī sasniegta sirdsdarbība minutē.

**Avg\_BPM** – Videja sirdsdarbība minutē.

**Resting\_BPM** - Atpūtas sirdsdarbība minutē.

**Session\_Duration (Hours)** – Sessijas Ilgums, stundos.

**Calories\_Burned** – Sadedzinātas Kalorijas vienā sessijā.

**Workout\_Type** – vingrinājumu komplekss (Yoga/HIIT/Cardio/Strength).

**Fat\_Percentage** - Tauku daļa ķermeņa svarā, procentos.

**Water\_Intake (Liters**) - Ūdens uzņemšana, litros.

**Workout\_Frequency (days/week)** - Treniņu biežums dienās/nedēļā.

**Experience\_Level** - Pieredzes līmenis (Beginner/Intermediary/Expert).

**BMI** – Ķermeņa Massas indekss, svars/augstums².

| **Klases iezīme** | **Pazīmes skaidrojums** | **Vērtību tips** | **Vērtību diapazons** |
| --- | --- | --- | --- |
| Age | Vecums | Numeric | [18 - 59] |
| Gender | Dzimums | Categorical | Male / Female |
| Weight (kg) | Svars | Numeric | [40 - 129.9] |
| Height (m) | Augstums | Numeric | [1.5 - 2] |
| Max\_BPM | Maksimālī sasniegta sirdsdarbība minutē. | Numeric | [160 - 199] |
| Avg\_BPM | Videja sirdsdarbība minutē. | Numeric | [120 - 169] |
| Resting\_BPM | Atpūtas sirdsdarbība minutē. | Numeric | [50 - 74] |
| Session\_Duration (Hours) | Sessijas Ilgstums, stundos. | Numeric | [0.5 - 2] |
| Calories\_Burned | Sadedzinātas Kalorijas vienā sessijā. | Numeric | [303 - 1783] |
| Workout\_Type | vingrinājumu komplekss. | Categorical | Yoga / HIIT / Cardio / Strength |
| Fat\_Percentage | Tauku daļa ķermeņa svarā. | Numeric | [10 - 35] |
| Water\_Intake (Liters) | Ūdens uzņemšana. | Numeric | [1.5 - 3.7] |
| Workout\_Frequency (days/week) | Treniņu biežums dienās/nedēļā. | Numeric | [2 - 5] |
| Experience\_Level | Pieredzes līmenis (Beginner/Intermediary/Expert). | Categorical | [1 - 3] |
| BMI | Ķermeņa Massas indekss, svars/augstums². | Numeric | [12.32 - 49.84] |

### Datu faila struktūra:

A screenshot of a computer

AI-generated content may be incorrect.

### A white background with black lines AI-generated content may be incorrect.

### Informācija par trūkstošajām vai izlecošajām vērtībām:

Mūsu datasetā nav trūkstošās vai izlecošās vērtības.

## Datu kopas vizuālais un statistiskais atspoguļojums

A screen shot of a computer

AI-generated content may be incorrect.

Session Duration – Fat Percentage Izkliedes diagrammas ekrānuzņēmums

A screen shot of a graph

AI-generated content may be incorrect.

Session Duration – Calories Burned Izkliedes diagrammas ekrānuzņēmums

A screenshot of a computer

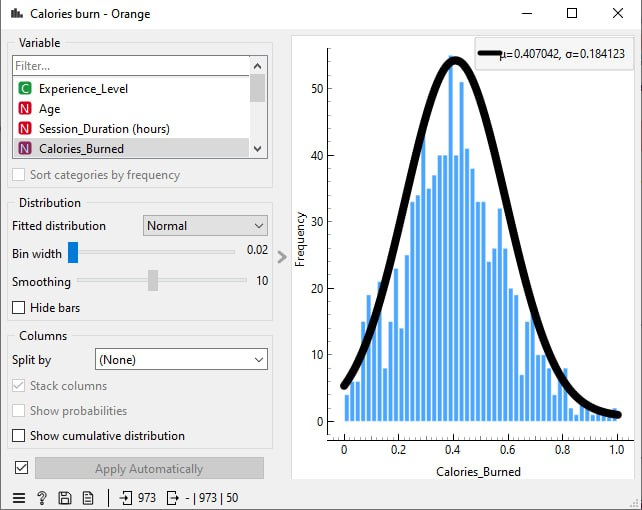
AI-generated content may be incorrect.

Calories Burned - Histogrammas ekrānuzņēmums

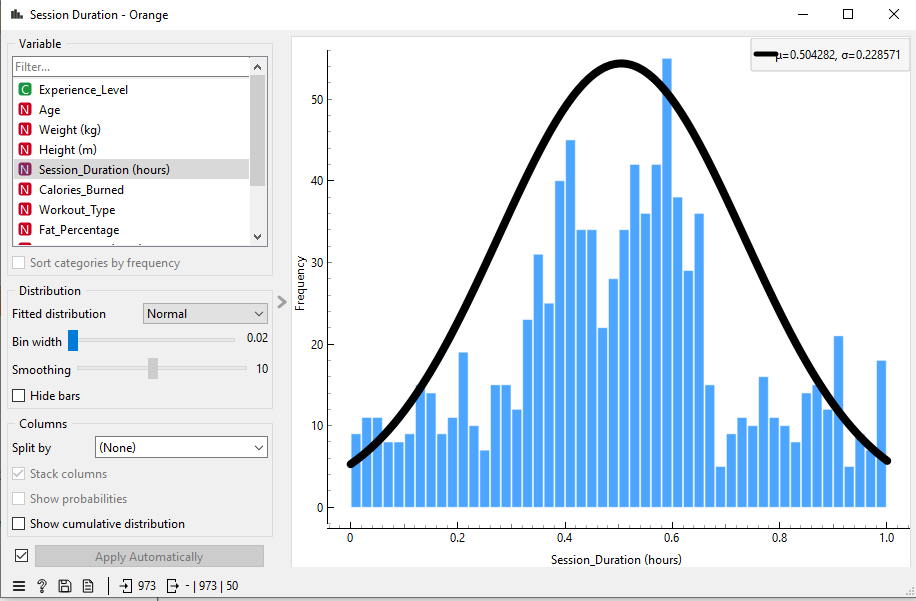
A screenshot of a computer

AI-generated content may be incorrect.

Session Duration - Histogrammas ekrānuzņēmums



Calories Burned Pazīmju sadalījuma ekrānuzņēmums



Session Duration Pazīmju sadalījuma ekrānuzņēmums

A screenshot of a computer

AI-generated content may be incorrect.

Ekrānuzņēmums ar statistiskajiem rādītājiem

## Atbildes uz jautājumiem

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

*Datu kopā klases nav līdzsvarotas — Experience\_Level 1 atbilst pie 41.7% ierakstu darbām, bet Experience\_Level 3 atbilst tikai pie 19.6%.*

### Vai datu vizuālais atspoguļojums ļauj redzēt datu struktūru?

*Gandrīz. Atšķirt Experience\_Level 1 no Experience\_Level 2 būs izcīli grūti, bet Experience\_Level 3 gandrīz neparklājas ar Experience\_Level 1-2 lielākājā daļā grafiku.*

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

*Vizuāli aplūkojot datus, ir iespējams identificēt, kā Experience\_Level 1 un 2 ietilpst cītas pazīmes, neka Experience Level 3. Bez kolorēšēnas, vizuāli sadalīt Experience\_Level 2 no Experience\_Level 1 var būt iespējams tikai no Izkliedes diagrammas blīvuma.*

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

*Identificētie datu grupējumi atrodas iemērota attalūma no viens otru. 2 Experience\_Level parklājas ar 1, bet 1 pieņem divreiz daudzāku lauku. 3 Experience\_Level parasti neparklājas ar 1 vai 2.*

## Secinājumi, kas izriet no statistisko rādītāju analīzes

*Tiek noskatāms, kā pirmssvarigākais attribūts būs Experience\_Level, jo saistīti ar to attrībūti atšķīras radikāli.   
  
Kā mēs sacīm projektu, mūsu sākotnējā hipotēze bija ka Gender un Workout\_Type būs klases, kurās būs vislielākā atšķirība starp atribūtiem. Šī hipotēze bija nepareiza. Lielāko daļu parametru nebija atšķirības starp dzimumiem, un Workout\_Type klases nebija derīgas, jo spēka balstīti vingrinājumi (Strength) bija izkliedēti pa visu izkliedes diagrammu, bija grūti atšķirt citas klases.*

## Secinājumi par atlasītajām pazīmēm

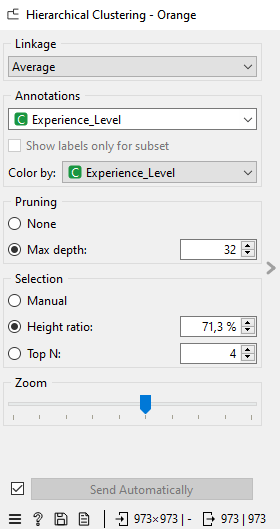
*Attrībūtos Gender, Weight, Height, Max\_BPM, Avg\_BPM, Resting\_BPM, Workout\_Type un BMI nav iespējams atšķirt klassus, un šī attribūti tiek izlaīsti.*

# II daļa

## Hierarhiskā klasterēšana

### Orange rīkā pieejamo hiperparametru apraksts:

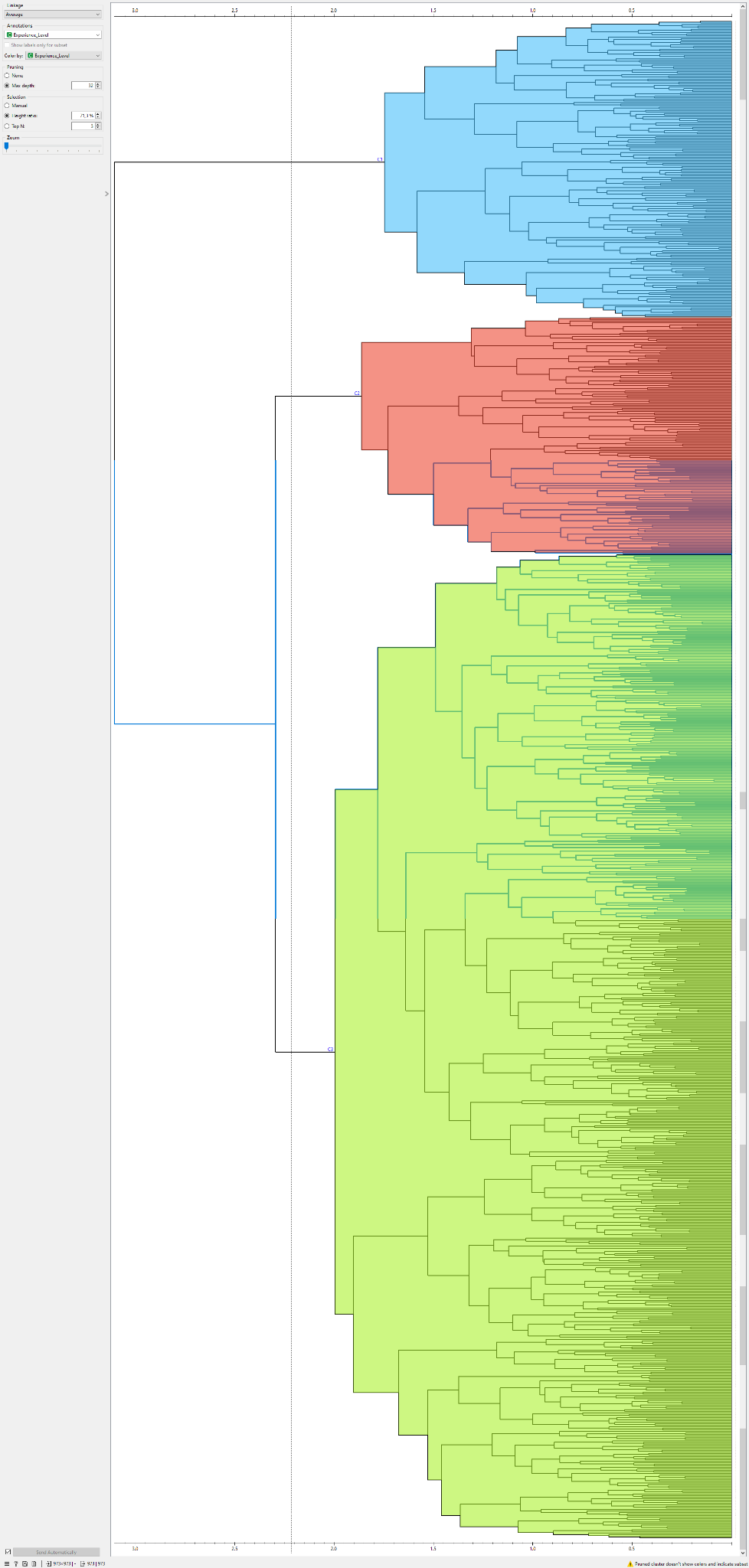
| **Hiperparametrs** | **Apraksts** |
| --- | --- |
| Linkage | Metode, kā aprēķina attālumu starp klasteriem. Eksperimentā izmantota "Average" metode, kas ņem vidējo attālumu starp visiem punktu pāriem no diviem klasteriem. Citas opcijas: "Single" (tuvākais punkts), "Complete" (tālākais punkts), "Weighted" (WPGMA metode), "Ward" (minimizē iekšējo klasteru dispersiju). |
| Annotations | Ļauj izvēlēties, ko rādīt dendrogrammas zaros. Eksperimentā izvēlēts "Experience\_Level", lai redzētu klašu sadalījumu (1, 2, 3 (Sācēji, Starpposms, Eksperti)). |
| Pruning (Max Depth) | Iespēja saīsināt dendrogrammu, izvēloties maksimālo dziļumu. Eksperimentā iestatīts uz 32, bet tas ietekmē tikai attēlojumu, nevis klasterēšanu. |
| Selection (Height Ratio) | Kā atlasīt klasterus: "Manual" (klikšķinot uz klasteriem, turot Ctrl/Cmd vairākiem klasteriem), "Height Ratio" (ar līniju dendrogrammā), vai "Top N" (noteikts klasteru skaits). Eksperimentā izmantota "Height Ratio". Eksperimentā mainīts no 71.3% uz 62.7%, un tad uz 60.0%, lai iegūtu 3, 4 un 5 klasterus. |
| Zoom | Ļauj pietuvināt vai attālināt dendrogrammu, lai labāk redzētu struktūru. Eksperimentā izmantots, lai pārbaudītu detaļas. |



### Eksperimentu apraksts

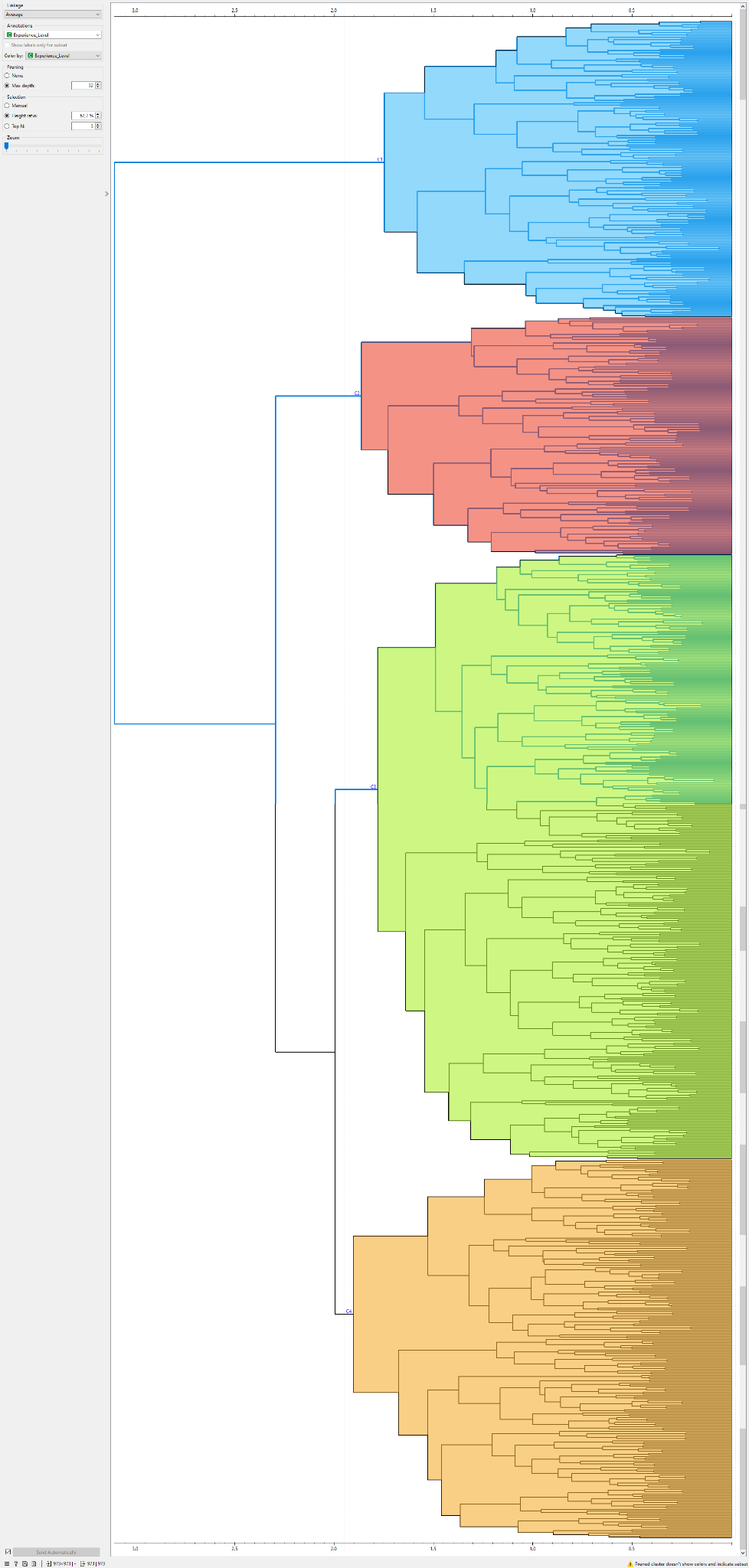
**1. eksperiments**: Horizontālā atdalošā līnija tika pārbīdīta uz 71.3% augstumu, kas sadalīja objektus 3 klasteros. Rezultāti:

* Pirmajā klasterī (C1) ir tikai klase 3
* Otrajā klasterī (C2) galvenokārt ir klase 1, bet ir 1 elements no klases 2
* Trešajā klasterī (C3) ir aptuveni vienādas daļas no klases 1 un klases 2.
* Dominē klasteris C3, jo tajā ir visvairāk objektu, bet C1 un C2 ir gandrīz vienāda izmēra, ar C1 nedaudz lielāku.
* Tas nozīmē, ka visi objekti netika pilnīgi atdalīti tīri savās klasēs, jo C2 un C3 satur sajauktas klases.



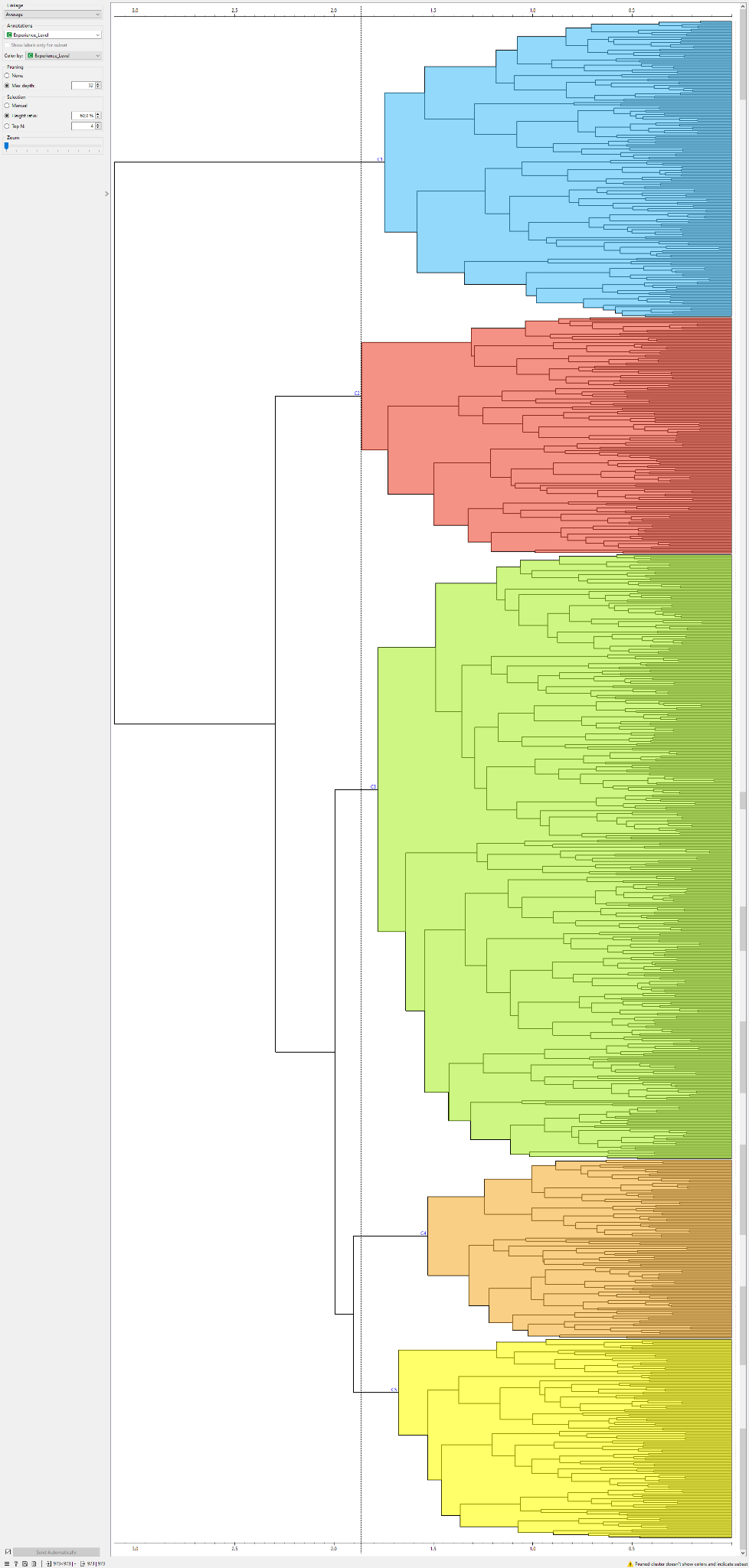
**Otrais eksperiments**: Otrajā eksperimentā atdalošā līnija tika pārbīdīta uz 62.7% augstumu un tas sadalīja objektus 4 klasteros. Rezultāti:

* Pirmajā klasterī (C1) ir tikai klase 3,
* Otrajā klasterī (C2) galvenokārt ir klase 1, bet ir 1 elements no klases 2,
* Trešajā klasterī (C3) ir aptuveni vienādas daļas no klases 1 un klases 2,
* Ceturtajā klasterī (C4) dominē klase 2, bet ir daudz elementu no klases 1.
* Dominē klasteris C3, jo tajā ir visvairāk objektu, seko C4, bet C1 un C2 ir mazāki, ar C1 nedaudz lielāku par C2.
* Tas nozīmē, ka visi objekti netika pilnīgi atdalīti tīri savās klasēs, jo C2, C3 un C4 satur sajauktas klases.



**Trešais eksperiments**: Trešajā eksperimentā atdalošā līnija tika pārbīdīta uz 60.0% augstumu un tas sadalīja objektus 5 klasteros. Rezultāti:

* Pirmajā klasterī (C1) ir tikai klase 3,
* Otrajā klasterī (C2) galvenokārt ir klase 1, bet ir 1 elements no klases 2,
* Trešajā klasterī (C3) ir aptuveni vienādas daļas no klases 1 un klases 2,
* Ceturtajā klasterī (C4) dominē klase 2, bet ir daudz elementu no klases 1,
* Piektajā klasterī (C5) dominē klase 2, bet ir daudz elementu no klases 1.
* Dominē klasteris C3, jo tajā ir visvairāk objektu, seko C1 un C2, ar C1 nedaudz lielāku par C2, bet C4 un C5 ir mazāki un aptuveni vienāda izmēra.
* Tas nozīmē, ka visi objekti netika pilnīgi atdalīti tīri savās 3 klasēs, jo C2, C3, C4 un C5 satur sajauktas klases.



### Secinājumi no eksperimentiem:

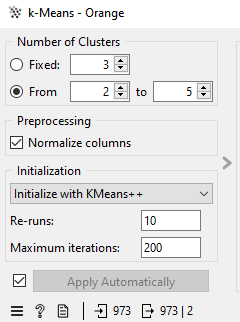
Mainot horizontālās atdalošās līnijas novietojumu no 71.3% uz 60.0%, klasteru skaits palielinājās no 3 līdz 5. Visos eksperimentos C1 satur tikai klasi 3, bet pārējie klasteri (C2, C3, C4, C5) satur sajauktas klases – klase 1 un klase 2 ir sajauktas, un klase 2 bieži dominē C4 un C5. C3 vienmēr ir lielākais klasteris, bet tas satur sajauktas klases, kas apgrūtina atdalīšanu. Ja Height Ratio turpinātu samazināt zem 60.0%, klasteru skaits palielinātos vēl vairāk, bet klašu atdalīšana, visticamāk, neuzlabotos, jo klase 1 un klase 2 turpinātu sajaukties. Savukārt, palielinot Height Ratio, piemēram, līdz 90%, būtu tikai 2 klasteri – viens klasteris (C1) saturētu klasi 3, bet otrs klasteris saturētu sajauktas klases 1 un 2. Tas nozīmē, ka visi objekti netika pilnīgi atdalīti tīri savās 3 klasēs. Veicot datu analīzi un mainot līnijas novietojumu, var secināt, ka nav iespējams panākt tīru klašu atdalīšanu – klase 1, klase 2 un klase 3 ir slikti atdalāmas, jo datu pazīmes (piemēram, sesijas ilgums, tauku procents) nav pietiekami atšķirīgas.

## K-vidējo algoritms

### Orange rīkā pieejamo hiperparametru apraksts:

<pievienojiet tabulai rindas pēc nepieciešamības>

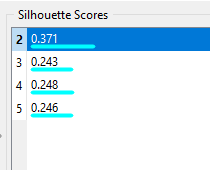
| **Hiperparametrs** | **Apraksts** |
| --- | --- |
| Number of Clusters | Norāda, cik klasteru algoritms izveidos. Var izvēlēties fiksētu skaitu (piemēram, 3) vai diapazonu (no 2 līdz 5), lai redzētu Silueta rezultātus. |
| Preprocessing | Ja izvēlēts "Normalize columns", dati tiek pielāgoti (vidējā vērtība 0, standarta novirze 1). |
| Initialization | Sākotnējā klasteru izveide: "KMeans++" izvēlas pirmo centru nejauši, bet nākamos punktus tā, lai tie būtu tālāk no jau izvēlētajiem centriem (balstoties uz attāluma kvadrātu), kas palīdz labāk sadalīt klasterus. "Random initialization" sāk ar pilnīgi nejaušu klasteru piešķiršanu, ko vēlāk uzlabo ar iterācijām. |
| Re-runs | Cik reizes algoritms sākas no jauna ar citu sākuma pozīciju (piemēram, 10 reizes), lai atrastu labāko rezultātu. |
| Maximum Iterations | Maksimālais iterāciju skaits vienā algoritma palaišanā (piemēram, 200), lai pārliecinātos, ka algoritms beidz darboties. |

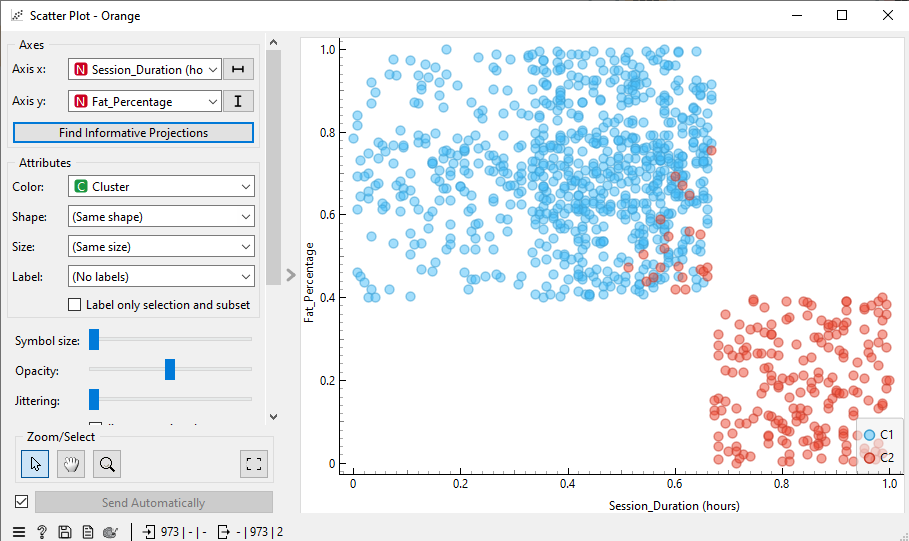


### Eksperimentu apraksts

K-vidējo algoritmam aprēķinājām Silueta koeficientu, lai pārbaudītu dažādas k k k vērtības un analizētu, kā dati sadalās klasteros. Iestatījumi tika mainīti, kā redzams 4.2 attēlā:

* Klasteru skaits – no 2 līdz 5,
* Normalizācija – ieslēgta, lai uzlabotu datu apstrādi (pirmapstrāde jau veikta iepriekš, bet, izslēdzot normalizāciju, rezultāti pie k bija sliktāki),
* Sākotnējā centroīdu ģenerēšana – "KMeans++" ar maksimālo iterāciju skaitu 200.



Vislabākais Silueta koeficients (0.371) tika iegūts pie k=2, bet atdalījums nebūs tik labs, jo rezultāts nav tuvu 1. Mēs pievienojām izkliedes diagrammu, lai redzētu klasterus. Izkliedes diagramma rāda divus klasterus (C1 un C2) ar pazīmēm "Session Duration (hours)" un "Fat Percentage". Dati daļēji atdalās, bet ir pārklāšanās.  
  


### Secinājumi no eksperimentiem:

Silueta koeficients rāda, ka labākais rezultāts ir pie k=2 ar 0.371, taču atdalījums nav labs, jo koeficients ir tālu no 1, kas norāda uz vāju klasteru kvalitāti. Mūsu datu kopā ir 3 klases (Experience\_Level 1, 2, 3), tāpēc k=2 neatbilst gaidītajam klašu skaitam. Izkliedes diagramma ar k=2 parāda, ka klasteri ir daļēji atdalīti, bet ir pārklāšanās, īpaši sesijas ilguma (0.5–0.7 stundas) apgabalā. Normalizācijas ieslēgšana palīdzēja pielāgot datus, bet tas neuzlaboja klasteru atdalīšanu pietiekami, jo ar k=3 Silueta koeficients palika zemāks (0.243). Tas liecina, ka klases ir grūti atdalāmas.

## Noslēguma secinājumi

Pamatojoties uz divu algoritmu – K-vidējā un Hierarhiskās klasterēšanas – darbības salīdzinājumu, var secināt, ka datu kopā esošās klases (klase 1, klase 2 un klase 3) ir slikti atdalāmas. K-vidējais algoritms, kura Silueta koeficients pie labākā klasteru skaita (k = 2) sasniedza tikai 0.371, rādīja vāju klasteru atdalīšanu, ar nelielu pārklāšanos izkliedes diagrammā. Palielinot klasteru skaitu līdz 3, kas atbilst faktiskajām klasēm, Silueta koeficients samazinājās līdz 0.243, kas apstiprina, ka klases nav skaidri nodalāmas ar izvēlētajām pazīmēm (piemēram, sesijas ilgums, tauku procents). Savukārt Hierarhiskā klasterēšana, mainot Height Ratio no 71.3% līdz 60.0%, palielināja klasteru skaitu no 3 līdz 5, taču klase 3 tika atdalīta tikai vienā klasterī (C1), bet klase 1 un klase 2 palika sajauktas visos pārējos klasteros (C2, C3, C4, C5), pat pie dažādiem Height Ratio iestatījumiem, tostarp pie 90%, kad tika iegūti tikai 2 klasteri ar klasi 3 atsevišķi, bet klase 1 un klase 2 joprojām bija sajauktas. Abu algoritmu rezultāti liecina, ka datu kopas klases ir slikti atdalāmas, visticamāk, tāpēc, ka izmantotās pazīmes nav pietiekami atšķirīgas, lai nodrošinātu skaidru klasifikāciju.

# III daļa

## Izvēlēto algoritmu apraksts

### **Pirmā algoritma nosaukums: kNN**

### **Pirmā algoritma apraksts:**

Vienkāršs, bet efektīvs uzraudzītās mācīšanās algoritms, ko izmanto klasifikācijā. Lai klasificētu jaunu datu punktu, algoritms aprēķina attālumu starp šo punktu un visiem punktiem apmācības datos. Mēs izmantosim Eiklīda attālumu. Izvēlas k punktus, kuri ir vistuvāk jaunajam punktam, tiek apskatītas to klases, un visbiežāk sastopamā klase tiek piešķirta jaunajam punktam.

### **Otrā algoritma nosaukums: Loģistiskā regressija**

### **Otrā algoritma apraksts:**

Galvenokārt izmanto binārās klasifikācijas uzdevumiem. Lai gan nosaukumā ir vārds "regresija", tas ir klasifikācijas algoritms, nevis regresijas.

Datus apstrādā, veidojot lineāru kombināciju:

Rezultātu z transformē ar sigmoīda funkciju:

​

Ja šī varbūtība ir virs 0.5, tad punkts tiek klasificēts kā "1", pretējā gadījumā kā "0".

## Hiperparametru apraksts

| **Hiperparameters** | **Apraksts un vērtības** |
| --- | --- |
| Mākslīgo neironu tīkli | |
| 1. Neironi paslēptos slāņos | Definēts kā i-tais elements, kas apzīmē neironu skaitu i-tajā slēptajā slānī |
| 2. Activation | Slēptā slāņa aktivizēšanas funkcija:  **Identity**: bezoperācijas aktivizēšana, noderīga, lai īstenotu lineāro sašaurinājumu  **Logistic**: loģistikas sigmoīda funkcija  **tanh**: hiperboliskā iedeguma funkcija  **ReLu**: rektificētas lineārās vienības funkcija |
| 3. Solver | Risinātājs svara optimizēšanai:  **L-BFGS-B**: optimizētājs kvaziņūtona metožu saimē  **SGD**: stohastiskā gradienta nolaišanās  **Adam**: uz stohastisku gradientu balstīts optimizētājs |
| 4. Alfa | Apmācības ātrums: L2 soda (regulēšanas termiņa) parametrs (**0 – 1000**) |
| 5. Iterāciju maksimālais skaits | Pašsaprotami |
| kNN | |
| 1. Kaimiņu skaits | Cik tuvāko punktu būs izmantoti (**1-100**) |
| 1. Metrika | Metode, ar kuru būs skaitīts attālums starp punktiem:  **Euclidean:** "taisna līnija", attālums starp diviem punktiem  **Manhattan:** visu atribūtu absolūto atšķirību summa  **Maximal:** lielākā absolūtā atšķirība starp atribūtiem  **Mahalanobis:** attālums starp punktu un sadalījumu |
| 1. Svari | Vai kaimiņu svars tiek modificēts:  **Uniform**: visi punkti katrā apkaimē tiek svērti vienādi  **Distance**: tuvākiem kaimiņiem ir lielāka ietekme nekā tālākiem |
| Loģistiskā regressija | |
| 1. Regularizācijas tips | Pārvērš problēmas atbildi uz vienkāršāku:  **L1**: noved pie retiem modeļiem, pievienojot sodu, pamatojoties uz koeficientu absolūto vērtību  **L2**: veicina mazāku, vienmērīgāk sadalītu svaru, pievienojot sodu, pamatojoties uz koeficientu kvadrātu |
| 1. Vērtības spēks | w koeficients  (**0-1000**) |

## Informācija par testa un apmācības datu kopām

***A screenshot of a computer

AI-generated content may be incorrect.***

*Datu kopas sadalīšana testa un apmācības datu kopās (Att. 3.1)*

### **Datu objektu skaits apmācības datu kopā: 682**

### **Datu objektu % proporcija apmācības datu kopā: 70%**

| **Klases iezīme** | **Datu objektu skaits apmācības datu kopā** | **Datu objektu % proporcija apmācības datu kopā** |
| --- | --- | --- |
| 1 | 266 | 39 % |
| 2 | 274 | 40 % |
| 3 | 142 | 21 % |

### **Datu objektu skaits testa datu kopā: 291**

### **Datu objektu % proporcija testa datu kopā: 30%**

| **Klases iezīme** | **Datu objektu skaits testa datu kopā** | **Datu objektu % proporcija testa datu kopā** |
| --- | --- | --- |
| 1 | 110 | 38 % |
| 2 | 132 | 45 % |
| 3 | 49 | 17 % |

## Eksperimenti ar mākslīgo neironu tīklu

| **Eksperiments** | **Hiperparametru vērtības** |
| --- | --- |
| 1.eksperiments | Neironi: 100; 100  Alfa: 0,1  Iterācijas: 1000 |
| 2.eksperiments | Neironi: 300; 300  Alfa: 0,1  Iterācijas: 1000 |
| 3.eksperiments | Neironi: 100; 100  Alfa: 1  Iterācijas: 1000 |
| 4.eksperiments | Neironi: 100; 100  Alfa: 0,05  Iterācijas: 10.000 |
| 5.eksperiments | Neironi: 500; 500  Alfa: 0,05  Iterācijas: 2000 |

*A screenshot of a computer

AI-generated content may be incorrect.*

*1.eksperimenta hiperparametru vērtības(Att. 3.2)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*1.esperimenta veiktspējas metrikas(Att. 3.3)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*2.eksperimenta hiperparametru vērtības(Att. 3.4)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*2.esperimenta veiktspējas metrikas(Att. 3.5)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*3.eksperimenta hiperparametru vērtības(Att. 3.6)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*3.esperimenta veiktspējas metrikas(Att. 3.7)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*4.eksperimenta hiperparametru vērtības(Att. 3.8)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*4.esperimenta veiktspējas metrikas(Att. 3.9)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*5.eksperimenta hiperparametru vērtības(Att. 3.10)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*5.esperimenta veiktspējas metrikas(Att. 3.11)*

*Secinājumi no eksperimentiem:*

No visiem šajā darbā izmantotajiem algoritmiem neironu tīkls ir visnepiemērotākais mūsu uzdevumam, ņemot vērā tā prasīgo skaitļošanas jaudu. Šis uzdevums viņam ir pārāk vienkāršs. Neironu tīkls labi strādā tikai tad, ja tam ir pietiekami daudz neironu (Att. 3.4-5). Pārāk augsts apmācības ātrums var paaugstināt precizitāti vienā vietā bet samazināt citā (Att. 3.6-7). Pārmērīgs resursu pārpalikums uzlabo prognožu precizitāti(Att. 3.10-11), taču ne tik daudz, lai tas būtu tā vērts.

### Testēšanai izvēlētais modelis:

Otro eksperimentu var uzskatīt par optimālu – tas ir pietiekami precīzs un ne pārāk lēns.

## Eksperimenti ar kNN

| **Eksperiments** | **Hiperparametru vērtības** |
| --- | --- |
| 1.eksperiments | Kaimiņu skaits: 5  Attāluma metrika: Eiklīda |
| 2.eksperiments | Kaimiņu skaits: 1  Attāluma metrika: Eiklīda |
| 3.eksperiments | Kaimiņu skaits: 90  Attāluma metrika: Manhettena |
| 4.eksperiments | Kaimiņu skaits: 100  Attāluma metrika: Eiklīda |

*A screenshot of a computer

AI-generated content may be incorrect.*

*1.eksperimenta hiperparametru vērtības(Att. 3.12)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*1.esperimenta veiktspējas metrikas(Att. 3.13)*

*A screenshot of a computer

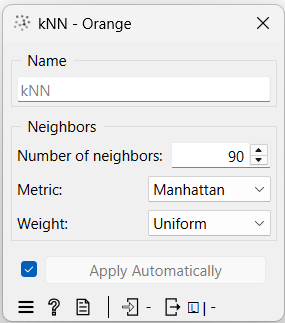
AI-generated content may be incorrect.*

*2.eksperimenta hiperparametru vērtības(Att. 3.14)*

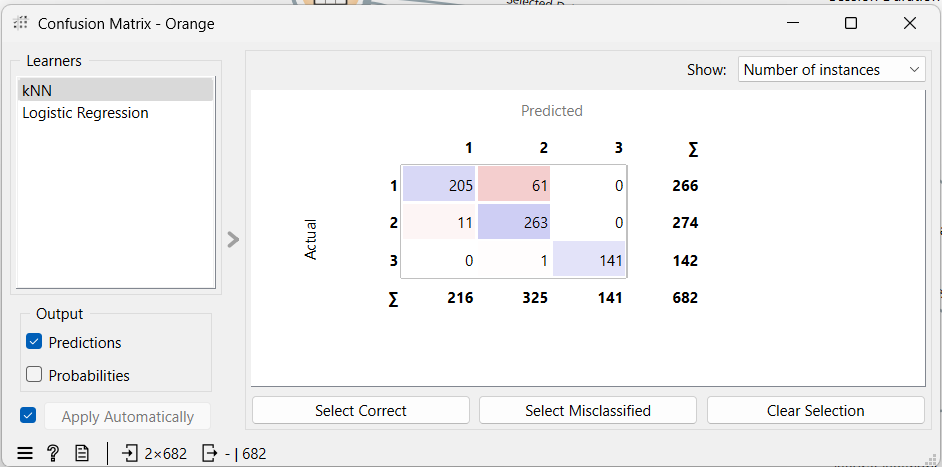
*A screenshot of a computer

AI-generated content may be incorrect.*

*2.esperimenta veiktspējas metrikas(Att. 3.15)*

**

*3.eksperimenta hiperparametru vērtības(Att. 3.16)*

**

*3.esperimenta veiktspējas metrikas(Att. 3.17)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*4.eksperimenta hiperparametru vērtības(Att. 3.18)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*4.esperimenta veiktspējas metrikas(Att. 3.19)*

*Secinājumi no eksperimentiem:*

Ļoti vienkāršs algoritms, bet ar savu vāju vietu - tam ir ļoti grūti atšķirt klases, kad ir nozīmīga pārklāšanās starp tām, kā redzams izkliedes diagrammās no 1. daļas. Pat labākajā rezultātā(Att. 3.16-17) ir 73 kļūdas pret 468 pareizām atbildēm starp 1. un 2. klasi - 15.6% atbilžu ir kļūdainas. Tā kā vairāk par 100 kaimiņiem Orange rīkā uzstādīt nevar(Att. 3.18-19), absurda atbildes ar pietiekami lielu datu bāzi saņemt nav iespējams.

### Testēšanai izvēlētais modelis:

Ar izmēģinājumu un kļūdu palīdzību tika noteikts, ka 3.eksperimenta hiperparametri darbojas vislabāk.

## Eksperimenti ar loģistisko regressiju

| **Eksperiments** | **Hiperparametru vērtības** |
| --- | --- |
| 1.eksperiments | Regularizācijas veids: L2  Spēks: 1 |
| 2.eksperiments | Regularizācijas veids: L2  Spēks: 0,01 |
| 3.eksperiments | Regularizācijas veids: L2  Spēks: 1000 |
| 4.eksperiments | Regularizācijas veids: L1  Spēks: 1 |
| 5.eksperiments | Regularizācijas veids: None  Spēks: N/A |

*A screenshot of a computer

AI-generated content may be incorrect.*

*1.eksperimenta hiperparametru vērtības(Att. 3.20)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*1.esperimenta veiktspējas metrikas(Att. 3.21)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*2.eksperimenta hiperparametru vērtības(Att. 3.22)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*2.esperimenta veiktspējas metrikas(Att. 3.23)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*3.eksperimenta hiperparametru vērtības(Att. 3.24)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*3.esperimenta veiktspējas metrikas(Att. 3.25)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*4.eksperimenta hiperparametru vērtības(Att. 3.26)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*4.esperimenta veiktspējas metrikas(Att. 3.27)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*5.eksperimenta hiperparametru vērtības(Att. 3.28)*

*A screenshot of a computer

AI-generated content may be incorrect.*

*5.esperimenta veiktspējas metrikas(Att. 3.29)*

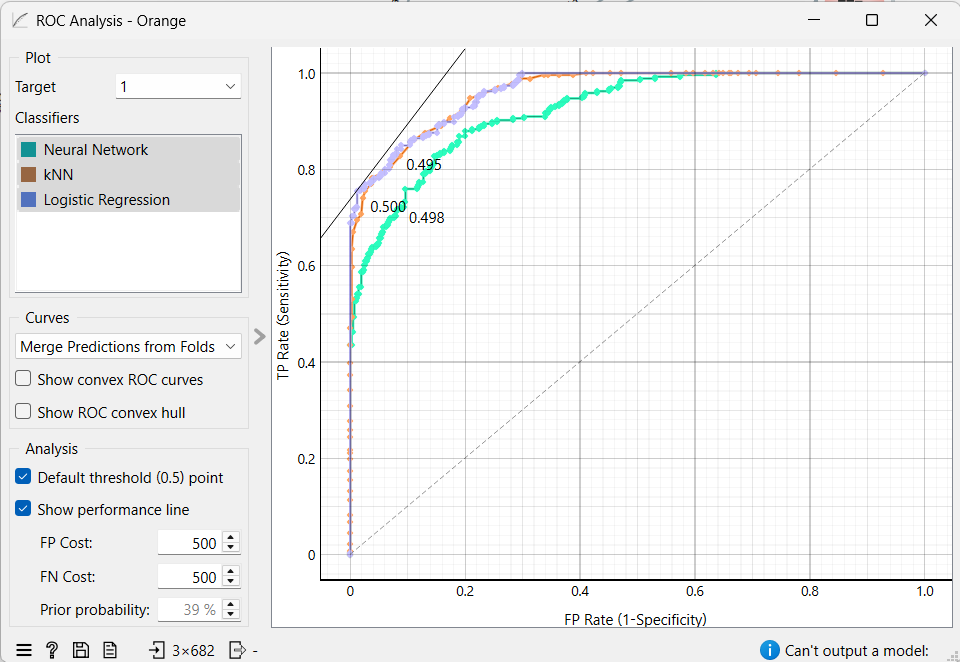
*Secinājumi no eksperimentiem:*

Visstabilākais algoritms. Vairumā gadījumu, izņemot C<0,1 (Att. 3.22-23), tas rada aptuveni vienu un to pašu rezultātu ar vidēji 80 kļūdām.

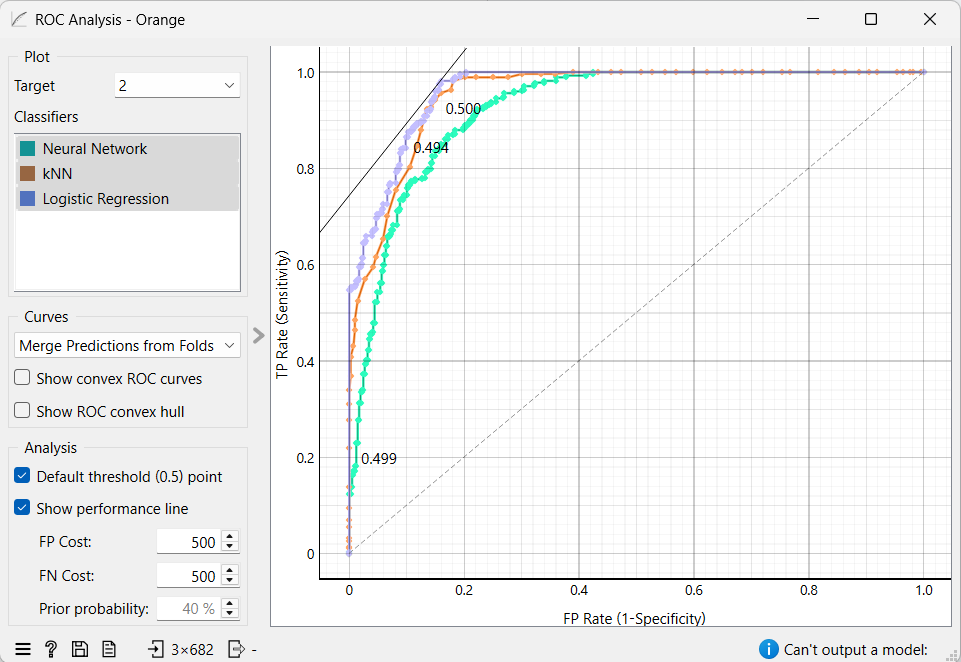
### Testēšanai izvēlētais modelis:

Lai gan kļūdu matricā tas nav redzams(Att. 3.28), 5.eksperiments parāda labāko rezultātu zemāk esošajos grafikos.

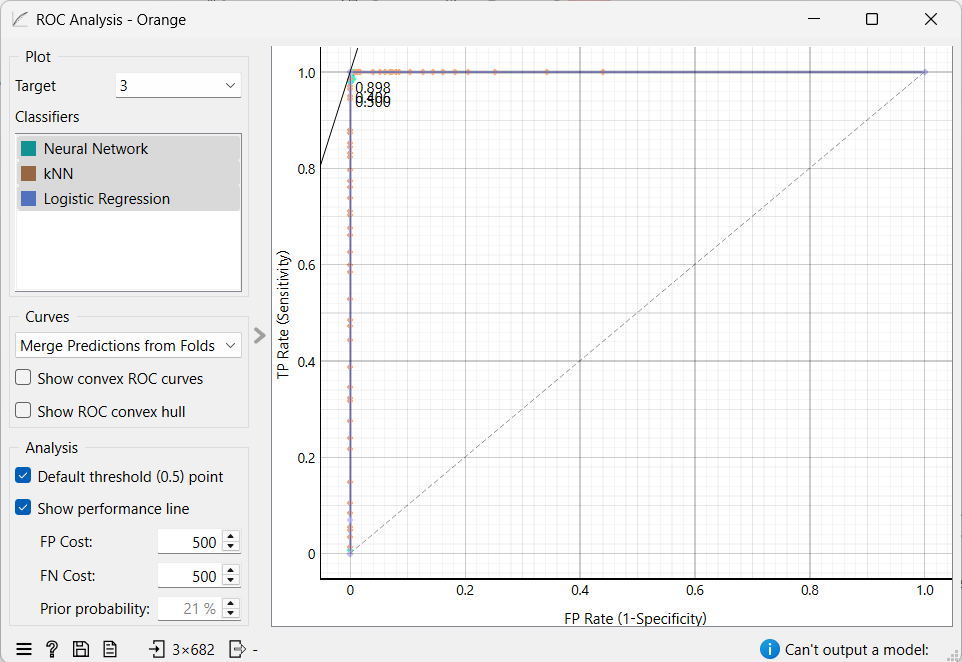
## Apmācīto modeļu testēšanas rezultāti



*Algoritmu salīdzinājums pēc panākumiem 1.klases izšķiršanā (Att. 3.30)*



*Algoritmu salīdzinājums pēc panākumiem 2.klases izšķiršanā (Att. 3.31)*



*Algoritmu salīdzinājums pēc panākumiem 3.klases izšķiršanā (Att. 3.32)*

*Secinājumi pēc testēšanas:*

Galvenā problēma šajā datubāzē ir atšķirt 1. un 2. klasi, kur 3. ir triviāli pamanāma. Pat labākie algoritmi, ko esam mēģinājuši, pieļauj daudz kļūdu. Kā izriet no 2. daļas, daudzus objektus nevar precīzi klasificēt, iespējams, mūsu mērķa klases rakstura dēļ. Mums ir 1. klase - cilvēki, kuri apmeklē sporta zāli mazāk nekā 6 mēnešus, un 2. klase - no 6 līdz 24. Viņus ir daudz grūtāk atšķirt vienu no otra nekā vidusmēra cilvēku no sportista.

# Izmantotie informācijas avoti

<https://orangedatamining.com/>

<https://orangedatamining.com/widget-catalog/unsupervised/hierarchicalclustering/>

<https://orangedatamining.com/widget-catalog/unsupervised/kmeans/>

# Pielikums