Rīgas Tehniskā Universitāte

Datorzinātnes un informācijas tehnoloģijas fakultāte

2.Praktiskais darbs

Mācību priekšmetā

**Mākslīgā intelekta pamati**

Autors: Maksims Golovašs

Apliecības numurs: 211RDB273

2. grupa

Saite uz projektu un datu kopu: <https://github.com/maxon28oo/MIP_2023/tree/master/Pr.d.2>

2022/2023 māc. gads

**Saturs**

[**I daļa - Datu pirmapstrāde/izpēte** 3](#_Toc134747841)

[**Datu kopu analīze** 3](#_Toc134747842)

[**Secinājumi** 11](#_Toc134747843)

[**II daļa – Nepārraudzītā mašīnmācīšanās** 13](#_Toc134747844)

[**Hierarhiskā klasterizācija[3]** 13](#_Toc134747845)

[**K-vidējo algoritms[4]** 15](#_Toc134747846)

[**Secinājumi** 18](#_Toc134747847)

[**III daļa – Pārraudzītā mašīnmācīšanās** 19](#_Toc134747848)

[**kNN algoritms[5]** 19](#_Toc134747849)

[**Tree algoritms[6]** 20](#_Toc134747850)

[**Testi** 21](#_Toc134747851)

[**1.** **Tests** 21](#_Toc134747852)

[**2.** **Tests** 22](#_Toc134747853)

[**3.** **Tests** 23](#_Toc134747854)

[**Apmācīto modeļu testēšanas rezultāti un to veiktspējas salīdzinājums un interpretācija** 24](#_Toc134747855)

[**Secinājumi** 25](#_Toc134747856)

[**IZMANTOTIE AVOTI UN LITERATŪRA** 26](#_Toc134747857)

# **I daļa - Datu pirmapstrāde/izpēte**

## **Datu kopu analīze**

Datu kopu nosaukums: “**Pima Indians Diabetes Database**”[1]

Autors: **UCI MACHINE LEARNING**

Link: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>

Es nolēmu izvēlēties datu kopu ar informāciju par pacientiem, norādot viņu medicīnisko informāciju un diabēta klātbūtni.

Šo datu kopu sākotnēji sagatavoja Diabēta un gremošanas un nieru slimību valsts institūts. Datu kopas mērķis ir diagnostiski prognozēt, vai pacientam ir vai nav diabēts, pamatojoties uz konkrētiem datu kopā iekļautajiem diagnostiskajiem mērījumiem. Šo gadījumu atlasei no lielākas datubāzes tika piemēroti vairāki ierobežojumi. Jo īpaši visi pacienti ir vismaz 21 gadu vecas sievietes, kas ir Pima indiāņu izcelsmes

Autors raksta, ka šis datu kopums ir piemērots klasifikācijai (lai norādītu, vai pacientam ir diabēts).

Persona, kas ir saistījusi darbu ar šo aktu, ir veltījusi darbu publiskai lietošanai, atsakoties no visām savām autortiesībām uz darbu visā pasaulē saskaņā ar autortiesību likumu, tostarp visām blakustiesībām, ciktāl to atļauj likums.

Ir iespēja kopēt, pārveidot, izplatīt un izpildīt darbu, pat komerciālos nolūkos, un to visu bez atļaujas pieprasīšanas.

Datu kopā ir 768 ierakstu.

Zemāk ir paradītas visi datu kopas atribūti

A screenshot of a computer

Description automatically generated with medium confidence

1 att.

A screenshot of a graph

Description automatically generated with low confidence

2 att.

A picture containing text, screenshot, diagram, plot

Description automatically generated

3 att.

A screenshot of a graph

Description automatically generated with low confidence

4 att.

Visi iepriekš minētie attēli vēlreiz parāda visus manas datu kopas atribūtus (attēli ir ņemti no tīmekļa vietnes, kurā es ieguvu datu kopu [Keggle.com](https://www.kaggle.com), un viņiem ir informācija par atribūtiem).

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribūts** | **Paskaidrojums** | **Vērtību tips** | **Diapazons** |
| Pregnancies | Šis atribūts norāda pacienta grūtniecību skaitu. | Skaitlis | 0 - 17 |
| Glucose | Plazmas glikozes koncentrācija 2 stundu laikā pēc glikozes tolerances testa perorāli. | Skaitlis | 0 - 199 |
| BloodPreassure | Asinsspiediens (mm Hg). | Skaitlis | 0 - 122 |
| SkinThickness | Tricepsa ādas krokas biezums (mm). | Skaitlis | 0 - 99 |
| Insulin | 2 stundu seruma insulīns (mu U/ml). | Skaitlis | 0 - 846 |
| BMI | Ķermeņa masas indekss (svars kg/(augums m)^2). | Skaitlis | 0 - 66.1 |
| Age | Norāda pacientu vecumu. | Skaitlis | 21 - 81 |
| Outcome | Klase, lai norādītu, ka pacientam ir diabēts (0 - nav, 1 - ir) | Skaitlis | 0 - 1 |

Šajā datumā vienīgā klasifikācijas iespēja ir diabēta esamība vai neesamība. Līdz ar to ir divas klases. 268 ieraksti ir klasificēti kā diabēta gadījumi (1), pārējie 500 ieraksti ir klasificēti kā bez diabēta (0).

A screenshot of a computer

Description automatically generated

5 att.

5. attēlā redzams datu kopas fails csv formātā, kas atvērts programmā Excel. Attēlā redzami visi kopas atribūti, kā arī ierakstu skaits.

A screenshot of a computer

Description automatically generated with medium confidence

6 att.

Kā redzat, šajā datu kopā nav trūkstošo vērtību, un visas vērtības ir skaitliskas, tāpēc nebija jāveic nekādi papildu pasākumi, tikai jāveic datu kopas apstrāde ar Orange Continuize, lai skaitliskās vērtības iekļautu diapazonā no 0 līdz 1.

A screenshot of a computer

Description automatically generated with low confidence

7 att.

Šādi izskatās dati pēc apstrādes.

Pēc tam es devos uz izkliedes diagrammu(Scatter Plot), lai redzētu, cik lielā mērā objekti ir atdalīti viens no otra. Šis grafiks attēlo vienu atribūtu X asij un otru Y asij. Turklāt, lai informācija būtu ērtāk uztverama, es iekrāsoju laukumus izvēlētās klasifikācijas klases krāsā. Lai nebūtu jātērē daudz laika, meklējot vairākus atribūtus ar labu objektu atdalīšanu, izmantoju funkciju "Find Informative Projections", kas sniedza sarakstu ar visām informatīvajām kombinācijām, no kurām izvēlējos un nofotografēju pirmās divas.

A screenshot of a computer

Description automatically generated with medium confidence

8 att. Vecums – Glikoze kombinācija

A screenshot of a computer screen

Description automatically generated with medium confidence

9 att. Ķermeņa masas indekss – Glikoze kombinācija

Pēc tam es devos uz histogrammu(Bar plot) un izvēlējos 2 grafikus (2 atribūti), kas labi atspoguļo datu kopu.

A red and blue lines

Description automatically generated with low confidence

10 att. Asinsspiediena histogramma

A picture containing text, screenshot, plot, line

Description automatically generated

11 att. Ķermeņa masas indeksa histogramma

Pēc tam dodieties uz sadaļu Izplatīšana (Distribution). Tas var sniegt detalizētu informāciju par pazīmju sadalījumu, tostarp par to, kuras vērtības ir visizplatītākās un cik bieži tās sastopamas. Rīkniņš parādīs histogrammu, vizualizējot pazīmes vērtību sadalījumu un izceļot bieži sastopamās vērtības.

A screenshot of a computer

Description automatically generated with medium confidence

12 att.

A screenshot of a graph

Description automatically generated with medium confidence

13 att.

Nākamais uzdevums ir aprēķināt statistikas datus. Orange vidē ir statistikas rīks(feauture statistics). Ar to es varu atrast uzdevuma mediānu un dispersiju.

A screenshot of a computer

Description automatically generated with medium confidence

14 att.

A screenshot of a computer

Description automatically generated with medium confidence

15 att.

## **Secinājumi**

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

Būtībā manā datu kopā ir tikai 2 klases, tomēr viena no tām dominē, proti, cilvēki bez diabēta. 500 pret 268.

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

Izkliedes diagrammā īsti neatspoguļo struktūru. Tā kā daži viena objekta punkti atrodas starp citiem objektiem, ir grūti noteikt struktūru. No manis izvēlētajām histogrammām (10. un 11. attēls) var redzēt datu grupēšanu (viens klases tips atrodas blakus citiem klases objektiem).

Īpaši KMI histogramma. No tās var redzēt, ka cilvēkiem bez KMI vidēji ir zemāks KMI nekā diabēta slimniekiem.

**Cik datu grupējums ir iespējams identificēt, pētot datu vizuālo atspoguļojumu?**

Aplūkojot divas izkliedes diagrammas. Pirmais grafiks (8. attēls) sniedz vislabāko grupējumu saskaņā ar citiem grafikiem. Pārējos grafikos punkti ir pārāk tuvu viens otram, tāpēc ir grūti precīzi noteikt klasi. Arī 8. attēla grafikā daži punkti pārklājas, bet šeit lielākā daļa objektu ir atdalīti un ir iespējams skaidri nodalīt grupas.

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

Grupēšana pēc atribūtiem vecums un glikoze diezgan labi atdala objektus vienu no otra. Tas nozīmē, ka tie ir normālā attālumā, lai gan joprojām ir situācijas, kad daži objekti "pārklājas" viens otram, bet šajā grupēšanā tas ir vismazāk ticams.

**Secinājumi, kas izriet no statistisko rādītāju** **analīzes.**

Tā kā esmu normalizējis ievades datus, visu atribūtu maksimālās un minimālās vērtības ir vienādas (min. - 0, maks. - 1).

Maksimālā dispersija (Dispersion) ir novērota insulīna parametram (1,44), bet mazākā - ĶMI parametram (0,25). Tas liecina, ka insulīna raksturlieluma vērtība variē visvairāk no visiem raksturlielumiem. Savukārt ĶMI atribūta vērtība svārstās vismazāk.

Kā redzams, lielākajai daļai pētījumā iesaistīto cilvēku nebija diabēta.

Asinsspiediena sadalījums ir ļoti līdzīgs normālajam sadalījumam.

Datu kopā bieži sastopama insulīna atribūta 0 vērtība. Es izlasīju vairākus komentārus un nonācu pie secinājuma, ka 0 nenozīmē, ka insulīna līmenis patiešām ir 0 (tas tā nevar būt), bet ka tas ir ļoti zems.

Arī bieža ādas tievuma vērtība arī ir 0. Tas var liecināt par to, ka cilvēkiem nav veikts ādas biezuma tests un no tā rodas 0 vērtība.

A screenshot of a computer

Description automatically generated with low confidence

16 att. Orange modelis

# **II daļa – Nepārraudzītā mašīnmācīšanās**

Šī mašīnmācīšanās veida uzdevums bija izvēlēties 2 algoritmus: hierarhiskā klasterizācija un K-vidējo algoritms.

## **Hierarhiskā klasterizācija[3]**

Hiperparametri:

Viens no šā algoritma hiperparametriem ir klasteru apvienošanas metodes (linkage) izvēle:

Single - aprēķina attālumu starp 2 blakus klasteriem

Average - aprēķina vidējo attālumu starp 2 klasteriem

Weighted - izmanto WPGMA metodi

Complete - aprēķina attālumu starp 2 vistālākajiem klasteriem

Ward - aprēķina summas kļūdas pieaugumu, minimizē kopējo iekšklasteru variāciju

Arī ir anotācijas – tas ir vienkārši komentāri zem asi.

Atzarošana – var ierobežot klasterizācijas dziļumu - tas ietekmē kopējo klasteru skaitu izvadē.

Selekcija:

Manual - iespēja atlasīt klasterus ar peli

Height ratio -veido līniju, zem kuras ir sadalījums klasēs. Ir iespējams mainīt augstumu attiecībā pret visu klasteru augstuma procentuālo daļu

Top N - augšējo N klasteru

A screenshot of a computer screen

Description automatically generated with low confidence

17 att. Hierarhiskā klasterizācija ar dziļumu 68, single

Kā redzams no galīgās dendagrammas, šī algoritma single metode manā datu kopā darbojas ļoti slikti. Tajā ir daudz objektu, un tie ir ļoti tuvu viens otram. Izmantojot sasaistes singlu, ir pārāk daudz mazu klasteru, un tie visi vienkārši neiekļaujas uz ekrāna, tāpēc man nācās ierobežot maksimālo dziļumu.

Mēģināju atlasīt klasterizācijas augstumu tā, lai atdalītie klasteri netiktu veidoti no dažādiem objektiem. Kā redzams attēlā, man ir vairāki mazi klasteri, kas pilnībā neatbilst. Dažos klasteros var būt iekļauti vairāki objekti no citas klases. Taču, ja augstumu samazinās, klasteru skaits ievērojami palielinās. Tāpēc es atradu vairāk vai mazāk optimālu pozīciju

A picture containing text, screenshot, software, display

Description automatically generated

18 att. Hierarhiskā klasterizācija ar dziļumu 68, average

Vidējās metodes gadījumā klašu skaits ir ievērojami mazāks, un tās tiek klasterizēti daudz labāk. Kā redzams attēlā, augstumu izvēlējos tā, lai atdalītie klasteri tiktu sagrupēti vislabāk (lai vienā klasterī būtu vismazāk citas klases objektu). Mēģinājumu un kļūdu ceļā esmu atradis optimālo augstumu.

Aplūkojot visas dendagrammas, klasterizācija labi veic savu darbu, bet viena gadījumā ir ļoti grūti atrast augstumu, kas dotu atbilstošu klasteru skaitu ar pareizo klasteru izkārtojumu.

A picture containing text, screenshot, diagram, display

Description automatically generated

19 att. Hierarhiskā klasterizācija ar dziļumu 68, complete

Pēdējais, ko es paņēmu, bija savienojums pabeigts. Principā nav lielas atšķirības ar vidējo rādītāju, bet šeit es pamanīju labāku grupēšanu mazos klasteros. Daudzos klasteros ir daudz mazāk dažādas klases objektu.

## **K-vidējo algoritms[4]**

Hiperparametri:

Klasteru skaits:

Fixed - skaidri norādīt klasteru skaitu, kas būs algoritma rezultāts.

From \_ to \_ - diapazons, kurā algoritms izpilda klasterizāciju un noteiktu "Silhouette Score". Šis parametrs salīdzina vidējo attālumu starp elementiem, kas pieder vienā klasterī, ar vidējo attālumu starp elementiem citos klasteros un dot atizīmi.

Inicializācija:

Atkārtojumi - cik reizes algoritms sak darbību no nejauši izvelētā punkta

Maksimālo iterāciju skaits - iespējamo iterāciju skaits

Lai šajā algoritmā atrastu labāko sadalījumu skaitu vienam klasterim, es nolēmu izmantot funkciju no \_ līdz \_ ar diezgan lielu diapazonu. Es uzsāku ar diapazonu no 2 līdz 20 un pakāpeniski to samazināju, līdz momentam kad diapazons bija no 2 līdz 5.

A screenshot of a computer

Description automatically generated with medium confidence

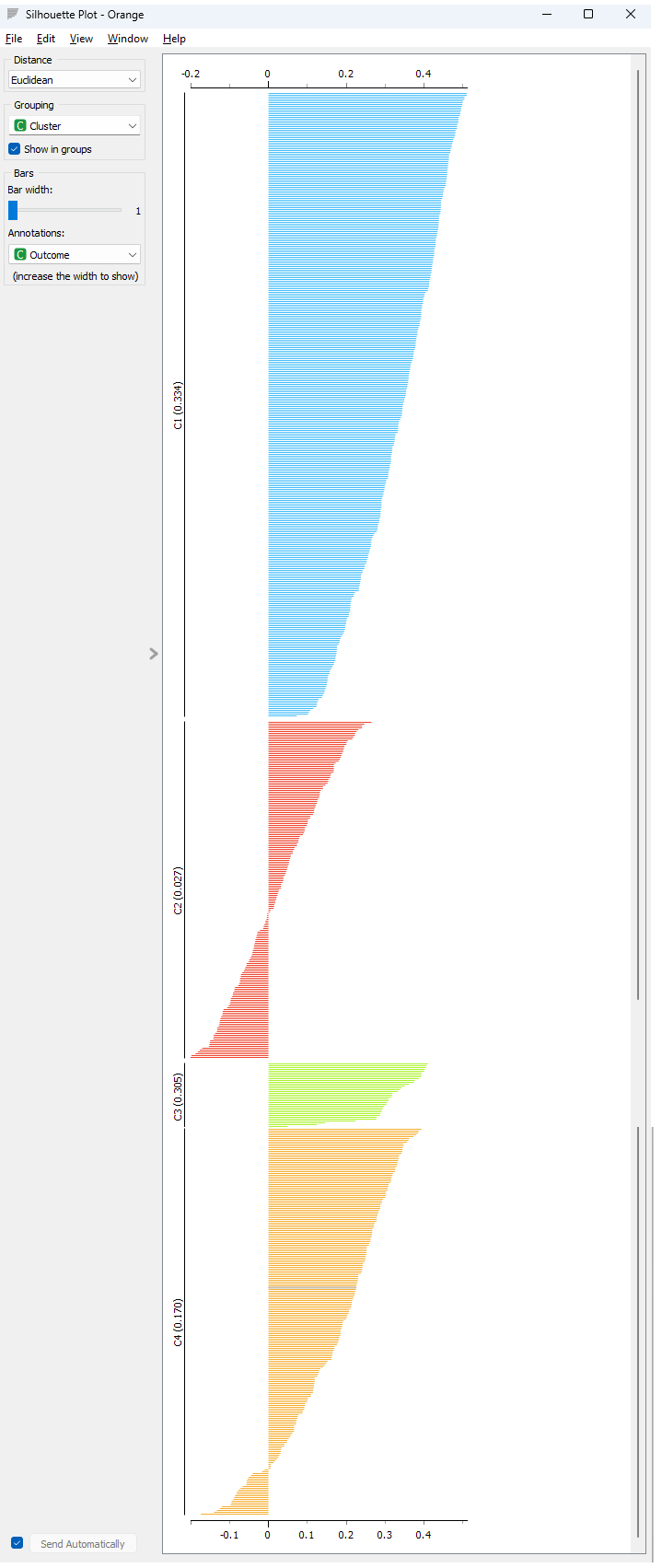
20 att. Sakuma diapazons

A screenshot of a computer

Description automatically generated with medium confidence

21 att. Beigas diapazons

Kad es sāku meklēt pareizo klasteru skaitu, rezultāts 4 klasteriem bija augsts, sākot no lielākā diapazona. Man šķita, ka šis skaitlis paliks labākais, kas izrādījās taisnība.



22 att.

Rezultāts ir šāds attēls. Kā es saprotu, negatīvās vērtības klasteros norāda, ka klasterizācija nav veikta pareizi un ka vienā klasterī ir dažādu klasteru objekti. Kā redzams, nav daudz negatīvu vērtību, tikai 2 klasteri C2 un C4, arī C3, bet to var uzskatīt par algoritma kļūdu. Taču grafikā redzams liels klasteris C1, kurā nav nevienas kļūdas, kas liecina par labu algoritma rezultātu.

## **Secinājumi**

Veicot nepārraudzītās mašīnmācīšanās, var redzēt, ka abi algoritmi neveic ļoti labu darbu. Lai gan manā gadījumā k-vidējais ar 3 klasteriem darbojās daudz labāk nekā hierarhiskā klasterizācija. Pēdējam algoritmam bija vairāk klasterizācijas kļūdu, turklāt ārkārtīgi mazos augstumos, kas padara klasterizāciju bezjēdzīgu, kas liecina, ka šis algoritms darbojās sliktāk nekā k-vidējais. Bija dažas kļūdas, bet kopumā redzams, ka sniegums ir labāks.

Tomēr kopējais sniegums liecina, ka šāda veida mašīnmācīšanos nevajadzētu izmantot, ja ir nepieciešams precīzi noteikt jauna objekta klasi.

# **III daļa – Pārraudzītā mašīnmācīšanās**

|  |  |  |
| --- | --- | --- |
|  | 0(nav diab.) | 1(ir diab.) |
| Sakumā | 500 | 268 |
| Mācīšanai(80%) | 400 | 214 |
| Testam(20%) | 100 | 54 |

Pārraudzītai mašīnmācīšanai nolēmu izvēlēties 2 algoritmus (kNN un Gradient Boosting), un 3 obligātais algoritms bija neironu tīkli.

A screenshot of a computer

Description automatically generated with medium confidence

## **kNN algoritms[5]**

k-tuvākie kaimiņi (kNN) ir mašīnmācīšanās algoritms, ko izmanto, lai klasificētu datus, pamatojoties uz to tuvumu jau zināmiem datiem. Vienkārši sakot, kNN algoritms atrod k tuvāko kaimiņu jauniem datiem un klasificē tos atbilstoši klasei, kas ir visbiežāk sastopama starp šiem k tuvākajiem kaimiņiem.

Lai klasificētu datus, izmantojot kNN algoritmu, ir jānosaka, kā katram jaunam novērojumam tiks noteikti tuvākie kaimiņi. To parasti dara, mērot attālumu, piemēram, izmantojot Eiklīda attālumu.

Kad katram jaunam novērojumam ir atrasti k tuvākie kaimiņi, kNN algoritms izmanto balsošanas metodi, lai noteiktu klasi, kurai piederēs katrs jaunais novērojums. Tas nozīmē, ka algoritms saskaita katrai klasei piederošo kaimiņu skaitu, un klase ar lielāko balsu skaitu tiek piešķirta kā jaunā novērojuma klase.

Es nolēmu izmantot šo algoritmu, jo tas šķita pietiekami vienkāršs, un mēs šo algoritmu esam aplūkojuši arī lekcijās.

Hiperparametri kNN ietver:

Parametrs k ir tuvāko kaimiņu skaits, kas tiek izmantots, lai klasificētu vai regresētu jaunos datus. Liela k

Attāluma metrika - metode attāluma mērīšanai starp novērojumiem. Visizplatītākā metode ir Eiklīda attālums, bet var izmantot arī citas metrikas, piemēram, Manhetenas.

Svari - nosaka, kā svērt attālumu starp kaimiņiem. Svērtie svari ņem vērā attālumu līdz katram kaimiņam.

## **Tree algoritms[6]**

Koku algoritmu (jeb lēmumu koku) izmanto klasifikācijas un regresijas problēmu risināšanai. Tas darbojas, sadalot datus mazākās apakšgrupās, pamatojoties uz pazīmju vērtībām. Sadalīšana balstās uz jautājumiem, kurus var uzdot par katru atribūtu.

Koks sākas ar saknes mezglu, kurā ir visi dati. Pēc tam katrs mezgls sadala datus divās vai vairākās grupās, izmantojot izvēlēto atribūtu un robežvērtību.

Hiperparametri kNN ietver:

Iekļaut bināro koku - ģenerē bināro koku (bērnu skaits -2)

Objektu min. skaits “lapās”– algoritms neveidos nodalījumus, kuru rezultātā lapās būs pārāk maz mācību piemēru.

Nesadaliet apakškopas, kas ir mazākas par - aizliedz algoritmam sadalīt mezglus, kuru gadījumu skaits ir mazāks par norādīto

Ierobežot maksimālo koka dziļumu - ierobežo koka gereāciju ierobežotā dziļumā

Es izvēlējos šo algoritmu, jo tas ir arī ļoti viegli saprotams, un man nebija problēmu izlasīt dokumentāciju oficiālajā Orange vietnē un saprast, kā algoritms darbojas.

## **Testi**

### **Tests**

A screenshot of a computer

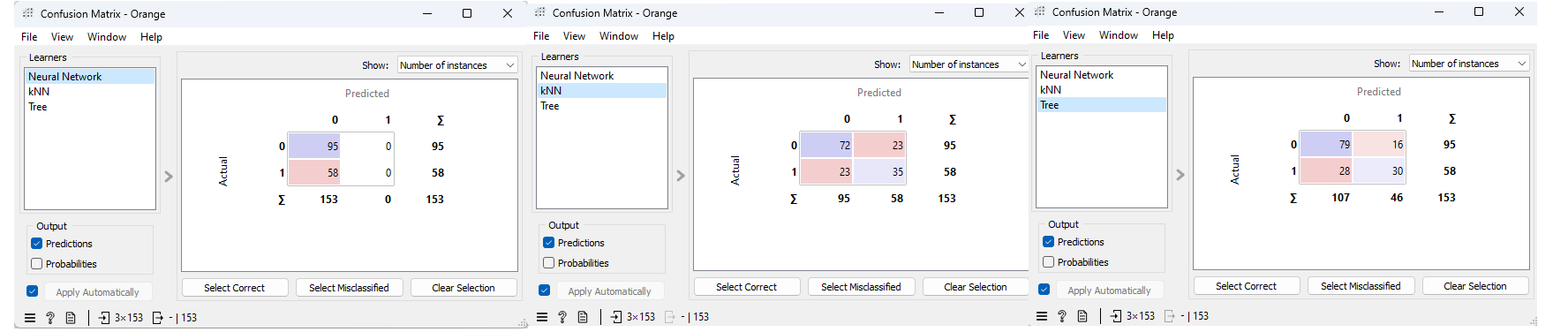
Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



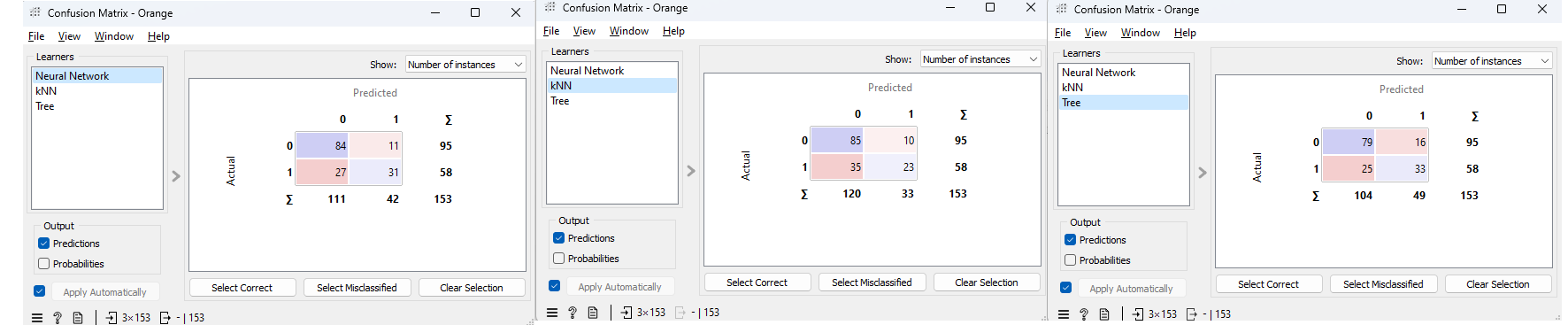
### **Tests**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### **Tests**

A screenshot of a computer

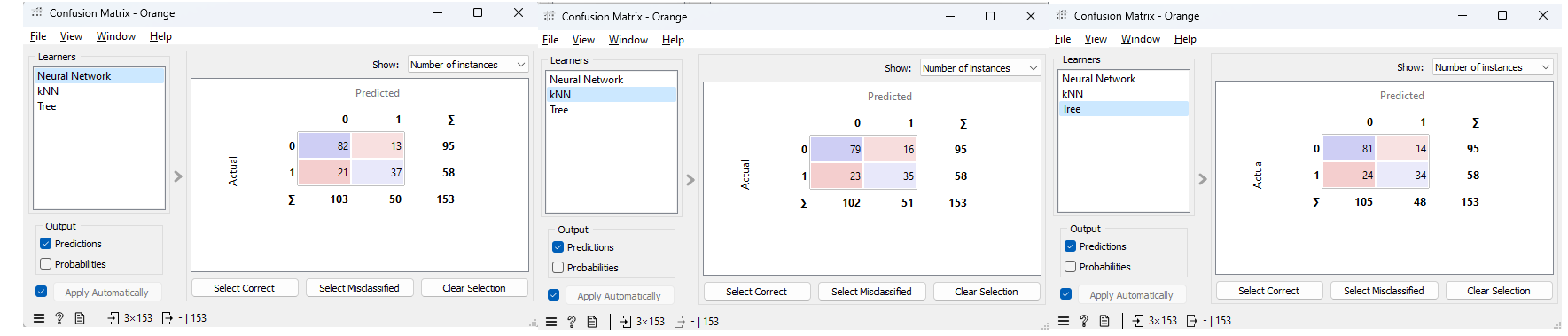
Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



### **Apmācīto modeļu testēšanas rezultāti un to veiktspējas salīdzinājums un interpretācija**

Esmu sācis veikt 3 testus, lai palielinātu 3 algoritmu precizitāti.

Pirmajā novērtējuma tabulā redzams, ka Tree ir labākais algoritms ar augstākajām CA un Prec metriku vērtībām – 71.2% un 70.6% 1. testā, kaut gan AUC vērtība tam ir zemāka nekā kNN un neironu tiklu algoritmiem.

Sākotnēji kNN es iestatīju Eiklīda metriku un uzskatāmo kaimiņu skaitu 2 - tas uzrādīja vissliktāko rezultātu no visiem 3 testiem. Neironu tīkliem es sāku ar 2 slēptajiem slāņiem, es nolēmu neradīt mazāk, jo tas neuzlaboja algoritma kopējo novērtējumu.

Tālāk kNN gadījumā es nolēmu palielināt attiecīgo kaimiņu skaitu, kā arī mēģināju mainīt metrisko parametru uz Manhetenas. Principā tas nedaudz uzlaboja kopējo rezultātu(kopējā precizitāte 69.9 % ->70.4%, klasifikācijas precizitāte 69.9% -> 70.6 %), taču ne tik ļoti, kā es būtu vēlējies. Turpmākie eksperimenti parādīja, ka manā modelī Manhattan nav labs, tāpēc es pārgāju atpakaļ uz eiklīdisko.

Neironu tīkliem es palielināju neironu skaitu katrā slēptajā slānī. Es arī nedaudz palielināju L2(aplha) parametru, kas uzlaboja šā modeļa precizitāti. (kopējā precizitāte 38,6 % -> 75 %, klasifikācijas precizitāte 62,1 % -> 75,2 %).

Attiecībā uz koku es nedaudz palielināju piemēru skaitu katrā mezglā, kā rezultātā precizitāte palielinājās tikai par 2 %.

Izmaiņas 3. testā visus 3 algoritmus noveda pie aptuveni vienādiem rezultātiem, kas liek domāt, ka mans mērķis maksimizēt visus algoritmus bija veiksmīgs.

kNN algoritma palielināju kaimiņu skaitu, kas palielināja precizitāti apmēram uz 4 procentiem.

Kokā palielināju dziļuma lielumu(turpmākas izmaiņas neietekmēja rezultātu), kā arī vēl vairāk palielināju objektu skaitu mezglos. Pieaugums bija 2 procenti, domāju, ka es izspiedu visu, ko es varētu no šī algoritma.

Neironu tīklos pievienoja vēl vienu slāni ar neironiem, kā arī sadalīja kā 30, 20, 100 un vēl palielināja L2 parametru, kas deva 2 procentu pieaugumu

# **Secinājumi**

Strādājot ar Orange rīku par mākslīgā intelekta pamatiem, tika pētīti un izmantoti dažādi mašīnmācīšanās algoritmi, piemēram, K-vidējie, hierarhiskā klasterizācija, kNN, lēmumu koks un neironu tīkli. Tika pētīta pamata metrika, lai novērtētu algoritmu darbības kvalitāti. Tika veikta arī dažādu datasetu analīze. Darbs bija interesants un izglītojošs, un tas ļāva padziļināt zināšanas par mašīnmācīšanos un mākslīgo intelektu. Orange rīka izmantošana bija ļoti ērta un palīdzēja paātrināt datu analīzes procesu. Darba rezultātus var izmantot turpmākajos mašīnmācīšanās pētījumos un pielietojumos.

Viens no grūtākajiem uzdevumiem bija atrast piemērotu datu kopu darbam, jo to ir daudz, taču ne visi atbilst konkrētam klasifikācijas uzdevumam. Arī daudzi neatbilda ierakstu un atribūtu kritērijiem.

Es uzskatu, ka pārraudzītā mašīnmācīšanās ir efektīvāka un kvalitatīvāka rezultātu precizitātē, tomēr labai precizitātei ir jāizvēlas pareizie parametri un ir nepieciešams pietiekams datu kopu izmērs, lai mācītos.

# **IZMANTOTIE AVOTI UN LITERATŪRA**

1. Kaggle, vietne meklēšanai datasetus. Saite: <https://www.kaggle.com>
2. Orange mājaslapā lai meklētu informāciju par logrīkiem. Saite: <https://orangedatamining.com/widget-catalog/>
3. Orange dokumentācija par hierarhisko klasterizāciju. Saite: <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/hierarchicalclustering.html>
4. Orange dokumentācija par K-vidējo algoritmu. Saite: <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/hierarchicalclustering.html>
5. Orange dokumentācija par kNN algoritmu. Saite: <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/knn.html>
6. Orange dokumentācija par Tree algoritmu. Saite: <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/tree.html>
7. Orange dokumentācija par neironu tīkliem. Saite: <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/neuralnetwork.html>
8. Ortus Mākslīgā intelekta pamatu priekšmeta lapa kur ir dažādas prezentācijas ar informāciju par kopējas darba tēmu. Saite: <https://estudijas.rtu.lv/course/view.php?id=252548>