

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Automātikas un datortehnikas institūts

2.praktiskais darbs

mācību priekšmetā

“Mākslīgā intelekta pamati(1),21/22-P”

**Indeksi. Datu glabāšanas struktūras**

Izmantotie faili praktiskajā darbā : <https://github.com/itsBoulder/MI_PD2>

Izstrādāja: Elvis Reinis Zeltītis

181REC034

Pārbaudīja:

2021./22. māc. Gads

**Saturs**

[**1.** **Datu kopas informācija** 3](#_Toc104319578)

[**2.** **Datu izpēte** 4](#_Toc104319579)

[**2.1.** **Datu pazīmju statistika** 4](#_Toc104319580)

[**2.2.** **Izkliedes diagrammas** 5](#_Toc104319581)

[**2.3.** **Histogramma** 7](#_Toc104319582)

[**2.4.** **Datu izpētes secinājumi** 8](#_Toc104319583)

[**3.** **Nepārraudzītā mašīnmācīšanās** 9](#_Toc104319584)

[**3.1.** **Hierarhiskā klasterizācija** 9](#_Toc104319585)

[**1. Eksperiments** 10](#_Toc104319586)

[**2. Eksperiments** 11](#_Toc104319587)

[**3. Eksperiments** 12](#_Toc104319588)

[**3.1.1.** **Hiperparametru skaidrojumi** 13](#_Toc104319589)

[**3.2.** **K-vidējo algoritms** 14](#_Toc104319590)

[**3.2.1.** **Hiperparametru skaidrojumi** 16](#_Toc104319591)

[**3.3.** **Nepārraudzītās mašīnmācīšanās secinājumi** 17](#_Toc104319592)

[**4.** **Pārraudzītā mašīnmācīšanās** 18](#_Toc104319593)

[**4.1.** **KNN** 19](#_Toc104319594)

[**4.1.1.** **KNN hiperparametru skaidrojumi** 20](#_Toc104319595)

[**4.2.** **Neironu tīkli** 21](#_Toc104319596)

[**4.2.1.** **Neironu tīklu hiperparametru skaidrojumi** 22](#_Toc104319597)

[**4.3.** **Lineārā regresija** 23](#_Toc104319598)

[**4.3.1.** **Lineārās regresijas hiperparametru skaidrojumi** 24](#_Toc104319599)

[**4.4.** **Pārraudzītās mašīnmācīšanās secinājumi** 25](#_Toc104319600)

[**5.** **Izmantotā literatūra** 26](#_Toc104319601)

[**6.** **Orange rīka pilnskats** 27](#_Toc104319602)

# **Datu kopas informācija**

Datu kopas nosaukums: “*Crab Age Prediction*”

Datu kopas autors: GURSEWAK SINGH SIDHU

Datu kopai nav pieejama informācija par licencēšanu, vai par datu ievākšanas veidu.

Datu kopas tīmekļa vietne : https://www.kaggle.com/datasets/sidhus/crab-age-prediction

Datu kopas mērķis ir noteikt krabja vecumu pēc pazīmēm kuras ir klasificētas pēc dzimuma, un to sakitliskie raksturojumi ir : (skatīt 1.1 attēlu)

* Garums ( Length )
* Diametrs ( Diameter )
* Augstums ( Height )
* Svars ( Weight )
* Svars bez čaulas ( Shucked Weight )
* Orgānu svars ( Viscera Weight )
* Čaulas svars ( Shell Weight )

Kopējais datu skaits datu kopā ir - 3893

**Graphical user interface

Description automatically generated with low confidence**

1.1 attēls. Datu kopa

# **Datu izpēte**

## **Datu pazīmju statistika**

Lai izprastu datu nozīmīgumu vēlamajam rezultātam, darba autors datus salīdzināja pēc vecuma. 1. Attēlā redzams ka datiem ir nozīmīgs trūkums, kas šajā gadījumā ir vecumu atšķirība, var novērot ka mediāna ir 10, un esošo datu apjoms kas tuvojas minimumam vai maksimumam nav balancēts, kas var ietekmēt gala rezultātu vecuma noteikšanai no datiem.

Trūkst datu konkrētās vecumu grupās pie īpašībām, kas norāda ka dati nav balancēti.

Chart

Description automatically generated

2.1.1 attēls. Datu pazīmju statistika

## **Izkliedes diagrammas**

Pēc izkliedes diagrammām, 2.Attēlā un 3.Attēlā, var novērot labu pazīmju sadalījumu, ir novērojama saistība starp pazīmēm un vecumu. Minētajos attēlos pazīmes tika attēlotas uz X ass, un vecums uz Y ass, pēc šo datu attēlošanas var secināt par labu datu korelāciju.

Datu krāsa tika vizualizēta pēc vecuma, un ir novērojams ka dati ir atdalāmi.

Pētot datus, 2. attēlā iespējams ir novērojami kļūdaini dati, jo ir ievērojama atšķirība pie vecuma grupas, un augstuma.

A screenshot of a computer

Description automatically generated with medium confidence

2.2.1 attēls.

A screenshot of a computer

Description automatically generated with medium confidence

2.2.2 attēls

## **Histogramma**

Aplūkojot Attēlus 2.3.1 un 2.3.2, ir novērojamas iepriekš minētās īpašības pie datu pazīmju statistikas.

Chart, histogram

Description automatically generated

2.3.1 attēls.

Chart, histogram

Description automatically generated

2.3.2 attēls.

## **Datu izpētes secinājumi**

Salīdzinot izmantotās datu izpētes metodes var secināt ka dati ir labi atdalāmi pēc pazīmēm, no kā var secināt par labu datu korelāciju.

# **Nepārraudzītā mašīnmācīšanās**

## **Hierarhiskā klasterizācija**

Ar hierarhisko klasterizāciju tika veikti 3. eksperimenti kuros tika mainīta distance starp klasteriem, minimāli tika mainīts dendogrammas dziļums, liela datu objektu skaita dēļ, lai informācija būtu pārskatāma, un klasteri tika atdalīti mainot “Height ratio” un “Top n” hiperparametru vērtības.

Eksperimentos ir manāms ka tiek atdalīti mazi klasteri, kas varētu būt kļūdainu datu dēļ. Kopumā skatoties klasterus var atdalīt sabalancēti.

### **1. Eksperiments**

Šajā eksperimentā maksimālais dziļums tika uzstādīts ar vērtību 10, un “Top N” vērtība uz 10. Linkage hiperparametra vērtība ir “Complete”.

Dendogrammā ir redzams liels, nesabalancēts datu kopas sadalījums klasteros.

**Chart, box and whisker chart

Description automatically generated**

3.1.1 attēls.

### **2. Eksperiments**

1. Eksperimentā klasteri tika sadalīti ar hiperparametru “Heightratio” kā vērtība ir 15.9%. Šajā eksperimentā veikts sadalījums trīs klasteros, un vizuāli izskatās sabalancēts. Hiperparametrs “Linkage” sājā eksperimentā ir “Average”, un “Max depth” uz 10.

(skatīt 3.1.2 attēlu)

**A picture containing graphical user interface

Description automatically generated**

3.1.2 attēls.

### **3. Eksperiments**

3. Eksperimentā tika paaugstināta “Max depth vērtība” un mazliet tika palielināta “Height Ratio” vērtība, klāsteru sadalījums ir ļoti līdzīgs otrā eksperimenta sadalījumam. (skatīt 3.1.3 attēlu)

**Graphical user interface, chart, application, table, Excel

Description automatically generated**

3.1.3 attēls.

## **Hiperparametru skaidrojumi**

* Linkage – Parametrs nosaka distanci starp klasteriem.
  + Single linkage – Aprēķina attālumu starp tuvākajiem klastera elementiem
  + Average linkage – Aprēķina vidējo attālumu starp klasteru elementiem
  + Weighted linkage – Izmanto WPGMA (“Weighted Pair Group Method with Arithmetic Mean”) metodi.
  + Complete linkage – Aprēķina attālumu starp tālākajiem klastera elementiem
  + Ward linkage – Aprēķina kļūdas kvadrātu summas pieaugumu.
* Annotations – Attēlo nosaukumus dendogrammā.
* Pruning – Iespēja mainīt dendogrammas attēloto dziļumu.
* Selection – Klasteru sadalīšana
  + Manual – Ar kreiso peles taustiņu var sadalīt klasterus dendogrammā.
  + Height ratio – Dendogrammā ar lineālu, kas ir attēlots ar raustītu līniju, klasterus var sadalīt pēc augstuma, to pārvietojot, vai ievadot procentuālu vērtību.
  + Top N – Augstāko mezglu vērtība

(skatīt 3.1.1.1 attēlu)

**Graphical user interface, text, application

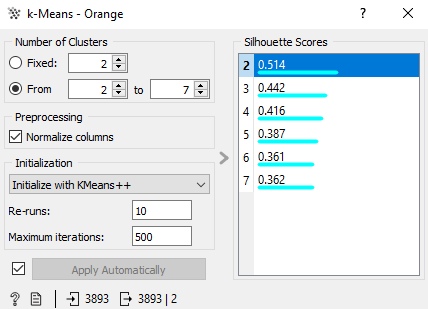
Description automatically generated**

3.1.1.1 attēls. Hiperparametri

## **K-vidējo algoritms**

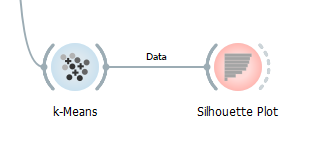
Veicot k-vidējo algoritma izpēti, rezultāti tiek atlasīti diapazonā no 2 līdz 7. Iegūtais rezultāts ir pozitīvs, pie 2 un 3 klasteriem rezultāts ir pieņemams, palielinot klasteru skaitu rezultāts krītas.

(skatīt 3.2.1 attēlu)



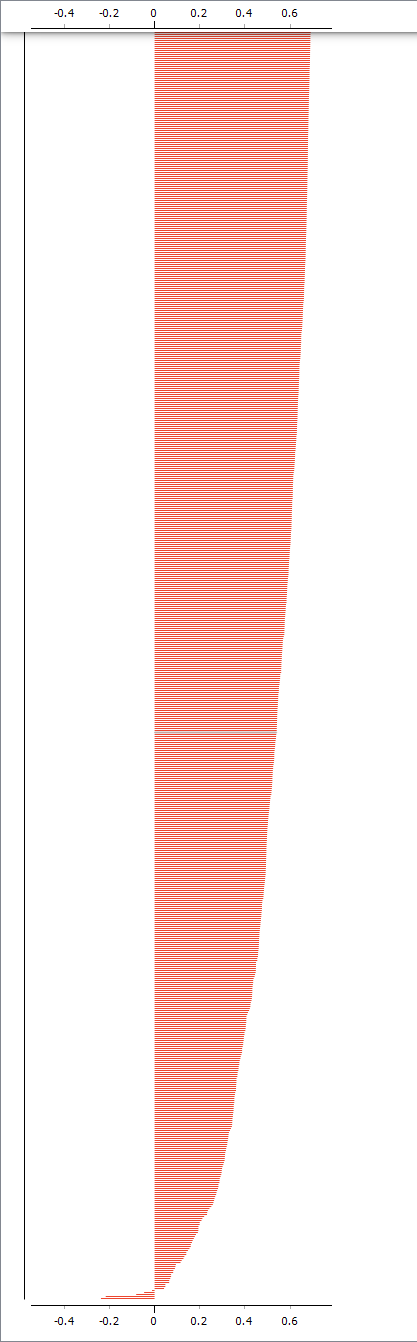
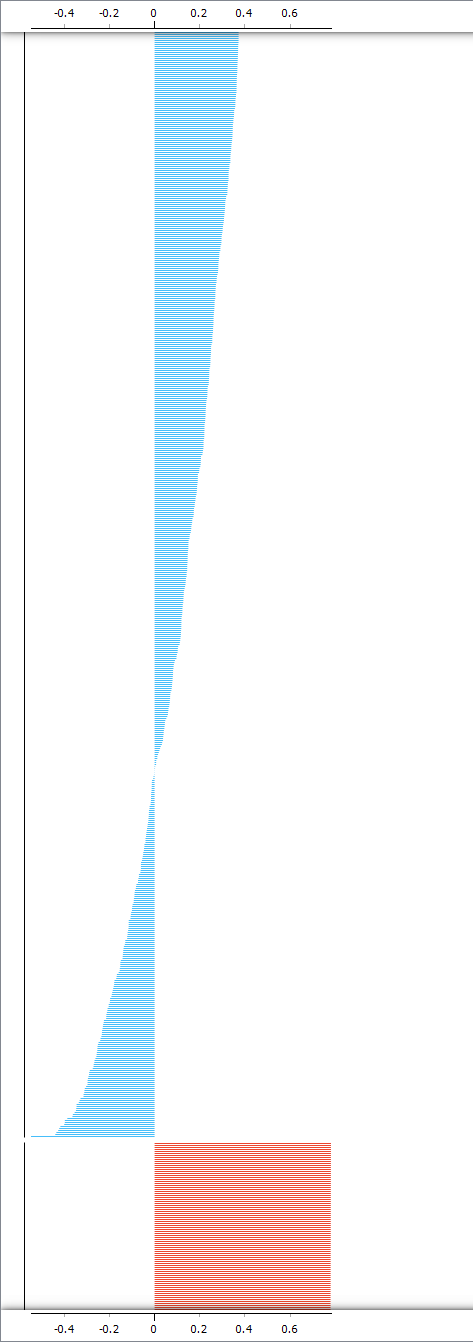
3.2.1 attēls. K-vidējo algoritms

Tiek pievienots “Silhouette plot” pie “k-Means” Orange rīkā (skatīt 3.2.2 attēlu)



3.2.2 attēls

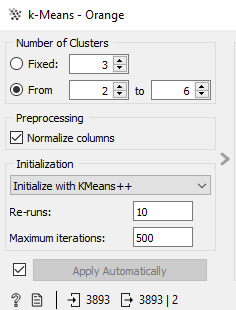
Apskatot “Silhouette plot”, var novērot ka dati ir labi atdalāmi. (skatīt attēlus 3.2.3, 3.2.4)



3.2.3, 3.2.4 attēli

### **Hiperparametru skaidrojumi**

* Klasteru skaits
  + Fixed - algoritms sagrupē datus noteiktam klasteru skaitam.
  + From - logrīks parāda klasterizācijas rezultātus atlasītajam klasteru diapazonam, izmantojot Silhouette punktu skaitu
* Preprocessing
  + Normalize columns – Kolonnas tiek normalizē1tas, vidējais centrēts līdz 0 un standarta novirze mērogota līdz 1
* Initialization
  + k-Means++ - pirmais centrs tiek izvēlēts nejauši, nākamie tiek izvēlēti no atlikušajiem punktiem ar varbūtību, kas proporcionāla kvadrātā attālumam no tuvākā centra
  + Random initialization - klasteri vispirms tiek piešķirti nejauši un pēc tam tiek atjaunināti ar turpmākām iterācijām
  + Re-runs - cik reizes algoritms tiek palaists no nejaušām sākuma pozīcijām, tiks izmantots rezultāts ar zemāko kvadrātu summu klasterī
  + Maximum iterations - maksimālais iterāciju skaits katrā algoritma izpildē



3.2.1.1 attēls.

# **Nepārraudzītās mašīnmācīšanās secinājumi**

Salīdzinot izmantotos algoritmus var secināt par labu datu atdalāmību klasteros.

# **Pārraudzītā mašīnmācīšanās**

Pārraudzītai mašīnmācīšanai tika izmantota Lineārā regresija, KNN un Neironu tīkli

lineārā regresija ir lineāra pieeja, lai modelētu attiecības starp skalāro reakciju un vienu vai vairākiem skaidrojošiem mainīgajiem ( *Wikipedia – Linear regression* )

Statistikā k-tuvāko kaimiņu algoritms (k-NN) ir neparametriska uzraudzīta mācību metode. To izmanto klasifikācijai un regresijai. Abos gadījumos ievade sastāv no k tuvākajiem apmācības piemēriem datu kopā. Izvade ir atkarīga no tā, vai k-NN tiek izmantots klasifikācijai vai regresijai. ( *Wikipedia – k-nearest neighbors algorithm* )

Algoritmu realizācijai tika izmantoti 40% datu apjoma un eksperimentējot ar 3. izvēlētajiem mašīnmācīšanās algoritmiem labākais rezultāts tika iegūts ar Lineāro regresiju.

# **KNN**

Veicot algoritmu, labākais rezultāts tika iegūts pie parametriem,

Number of neigbors - 90

Metric – Euclidian

Weight – Distance

Mainot parametrus rezultāts nemainās ievērojami, turklāt mainot “Number of neigbors” ievērojami mainās skaitlis aiz komata, ar ko izdevās pietuvināt rezultātu. (skatīt 4.1.1 attēlu)

Table

Description automatically generated

4.1.1 attēls

# **KNN hiperparametru skaidrojumi**

* Neighbours :
  + Number of neighbours – tuvāko attālumu skaits
  + Metric :
    - Euclidean – attālums starp diviem punktiem
    - Manhattan – visu atribūtu absolūto atšķirību summa
    - Maximal – lielākā no absolūtajām atšķirībām starp atribūtiem
    - Mahalanobis – attālums starp punktu un sadalījumu
  + Weight :
    - Uniform – visi punkti katrā apkaimē tiek svērti vienādi
    - Distance – tuvākiem vaicājuma punkta kaimiņiem ir lielāka ietekme nekā kaimiņiem, kas atrodas tālāk

Graphical user interface, text, application

Description automatically generated

4.1.1.1 attēls

# **Neironu tīkli**

Eksperimentējot ar neironu tīklu, brīvi mainot parametrus, labāko rezultātu izdevās iegūt pie parametriem “Activation” uz “Identity”, “Solver” uz “Adam” un ar neironu sadalījumu 2, 4, 4. (skatīt attēlu 4.2.1)

**Table

Description automatically generated with medium confidence**

4.2.1 attēls

# **Neironu tīklu hiperparametru skaidrojumi**

* Neurons in hidden layers – definēts kā i-tais elements, apzīmē neironu skaitu i-tajā slēptajā slānī.
* Activation :
  + Identity – bezoperācijas aktivizēšana, noderīga, lai īstenotu lineāro sašaurinājumu
  + Logistic – loģistikas sigmoīda funkcija
  + Tanh – hiperboliskā funkcija
  + ReLu – rektificētās lineārās vienības funkcija
* Solver :
  + L-BFGS-B – optimizētājs kvaziņūtona metožu saimē
  + SGD – stohastiskā gradienta nolaišanās
  + Adam – stohastisko gradientu optimizētājs
* Regularization – L2 soda parametrs
* Maximal number of iterations – Maksimālais atkārtojumu skaits

**Graphical user interface, application

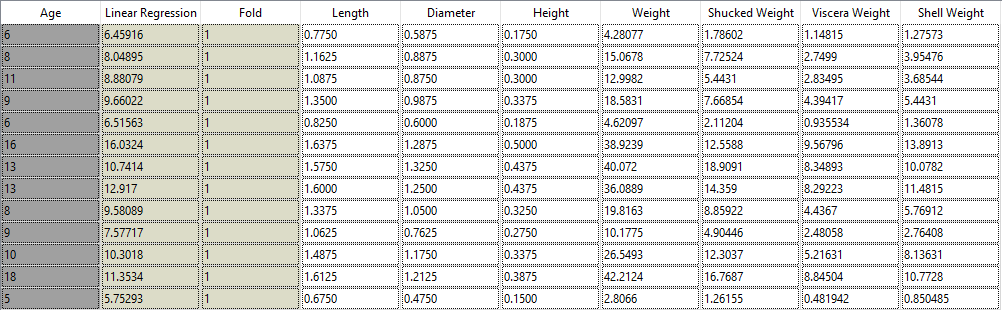
Description automatically generated**

4.2.1. attēls

# **Lineārā regresija**

Izmantojot lineāro regresiju, pie parametriem atstājot “No regulation” izdevās iegūt labākus rezultātus kā ar “Ridge regression” vai pārējām piejamajām iespējām.

Iegūtie resultāti bija apmierinoši. (skatīt 4.3.1 attēlu)

****

4.3.1 attēls

# **Lineārās regresijas hiperparametru skaidrojumi**

* Parameters :
  + Fit intercept – Noņemot atzīmi no opcijas, pārtveršana tiek piespiesta uz nulli
* Regularization :
  + Ridge regression – palielinot sodu, visi parametri tiek samazināti, bet joprojām nav nulle
  + Lasso regression – arvien vairāk parametru tiks nospiesti uz nulli.
  + Elastic net regression – elastīgais tīkls sodam pievieno kvadrātisko daļu

Graphical user interface, text, application, email

Description automatically generated

4.3.1.1 attēls

# **Pārraudzītās mašīnmācīšanās secinājumi**

Izpētot pielietotos algoritmus var secināt ka Lineārā regresija sniedza labākos rezultātus, lai gan Neironu tīklu rezultāti bija tuvi. (Skatīt 4.4.1 attēlu)

**Table

Description automatically generated**

4.4.1 attēls

# **Izmantotā literatūra**

Orange Data Mining Hierarchical Clustering :

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

Orange Data Mining k-Means :

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

Orange Data Mining KNN :

<https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/knn.html>

Orange Data Mining Neural Network :

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

Wikipedia Linear Regression :

<https://en.wikipedia.org/wiki/Linear_regression>

# **Orange rīka pilnskats**

**Diagram

Description automatically generated**