

**Rīgas Tehniskā universitāte**

**Datorzinātnes un informācijas tehnoloģijas fakultāte**

**Viedo datortehnoloģiju institūts**

**2. Praktiskas darbs**

**priekšmetā**

**Mākslīga intelekta pamati:  
Datu klasifikācija izmantojot “Orange” riku**

|  |  |
| --- | --- |
| **Izstrādāja:** | Nikita Terentjevs |
|  |  |
|  | 2.grupa |
|  |  |
|  | 211RDB228 **2. kurss** |
|  |
| **Pārbaudīja:** |  |
| **Darba nodošanas datums:** | 11.05.23 |

Saturs

[**Darba izpilde** 3](#_Toc134743657)

[1. Datu pirmapstrāde /izpēte 3](#_Toc134743658)

[2. Nepārraudzītā mašīnmācīšanās 4](#_Toc134743659)

[1.1.Hierarhiskā klasterizācija: 4](#_Toc134743660)

[2.1.K-vidējo algoritms 5](#_Toc134743661)

[3.1.Nepārraudzītā mašīnmācīšanās secinajums 5](#_Toc134743662)

[3. Pārraudzītā mašīnmācīšanās 5](#_Toc134743663)

[1.1.Algoritmu izvēle 5](#_Toc134743664)

[2.1.Modeles izmantošana 6](#_Toc134743665)

[3.1.Pārraudzītā mašīnmācīšanās secinājums 8](#_Toc134743666)

[4. Kopejais secinajums 8](#_Toc134743667)

**Darba izpilde**

1. Datu pirmapstrāde /izpēte

Datu kopas apraksts

* Datu kopas nosaukums: “Forest type mapping Data Set” (Meža tipa kartēšanas datu kopa)
* Avots: UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/datasets/Forest+type+mapping>)
* Izveidotājs: Brian Johnson “Institute for Global Environmental Strategies”

Datu kopas apraksts un papildus informācija:

Šajā datu kopā ir iekļauti apmācības un testēšanas dati no attālinātās zondēšanas pētījuma, kas kartēja dažādus meža tipus, pamatojoties uz to spektrālajām īpašībām redzamajā un tuvā infrasarkanajā diapazonā, izmantojot ASTER satelīta attēlus. Iegūtais rezultāts (meža tipa karte) var tikt izmantots, lai identificētu un/vai kvantificētu ekosistēmas funkcijas (piemēram, ogļskābās gāzes uzglabāšanu, erozijas aizsardzību), ko sniedz mežs.

Datu kopas satura apraksts

1. Klasi: **Mēžu tips** (‘s’ – “Sugi” tipu mežs; ‘h’ – “Hinoki” koku mežs; ‘d’ – jauktu lapu mežs; ‘o’ – citi, ne-mežsaimniecības zemes gabali)
2. īpašība: **b1 - b9** – ASTER attēla joslas, kas satur spektrālo informāciju zaļajā, sarkanajā un tuvā infrasarkanajā diapazonā. (Tikai pozitīvi skaitļi)
3. īpašība: **pred\_minus\_obs\_S\_b1 - pred\_minus\_obs\_S\_b9** – prognozētā spektrālā vērtība (balstoties uz telpiskās interpolācijas metodi) mīnus faktiskā spektrālā vērtība klasei 's' (b1-b9) (Visi reālie skaitļi)
4. īpašība: **pred\_minus\_obs\_H\_b1 - pred\_minus\_obs\_H\_b9** – prognozētā spektrālā vērtība (balstoties uz telpiskās interpolācijas metodi) mīnus faktiskā spektrālā vērtība klasei 'h' (b1-b9) (Visi reālie skaitļi)

Datu objektu skaits datu kopā:

* Mācības kopā: **198**
* Testēšanas kopa: **325**
* Kopā: **523** objekti
* Objektu pa klasēm: ‘s’ – **199**, ‘h’ – **86**, ‘d’ – **159**, ‘o’ – **83**
* Kopa katram objektam ir **27** atribūti

Tā kā es neatradu iespēju pielietot dažādus datu failus, lai apmācīt un pēc tām testēt pārraudzītās mašīnmācīšanās modeli, es datus saliku viena tabulā.

Secinājumi:

* Klases nav pilnīgi līdzsvaroti, bet man liekas tiem jābūt pietiekami sabalansētas jo 2 klases kas dominē ir līdzīgs objektu skaits, un parejam 2 klasēm ir ari līdzīgs objektu skaits sava starpa
* Veidojot “Scater plot” ar īpašībām **pred\_minus\_obs\_H\_b2** un **pred\_minus\_obs\_H\_b7**,kā arī **pred\_minus\_obs\_H\_b1** un **pred\_minus\_obs\_H\_b5** piedāvā vislabāk sadalīt tos “uz aci”
* Viennozīmīgi var atdalīt ka minimums 2 klases, dažreiz 3, atkarība no īpašībām
* Klases ‘s’ un ‘h’ vienmēr atrodas ļoti tuvu viens otram, bet ‘d’ un ‘o’ atrodas lielākajā attālumā salīdzinājuma ar ‘s’ un ‘h’ klasem

1. Nepārraudzītā mašīnmācīšanās
2. Hierarhiskā klasterizācija:

Vislabākais rezultāts man sanāca kad es mainīju parametru “Height ratio” kas pārvieto atdalošo līniju. Pārvietot to uz 40,1% - 40.5% dati sadalās 23 klasteros, kas var izskatīties daudz, bet lielāka daļa klašu atrodas klasteros ar tādiem pašam klasēm. Tīk daudz klasteru ir jo ir vairāki objekti kas ļoti augstajā diapazona. Tas ir izmatojot “vidējo” linkāžu.

Izmantojot “nosverto” linkāžu, klasteru skaits samazinās līdz 11 izmantojot parametru “Height ratio” no 44.5% līdz 49%

Vismazākās klasteru skaits man sanāca pie “Ward” linkāža kas minimizē variāciju vai kalkulē distances kvadrātu summu pirms saliekt klasterus kopā ([Wiki](https://en.wikipedia.org/wiki/Ward%27s_method)), un tas ir 4 klasteri – tik pat cik ir klases. Tā sanāc pie “Height ratio” vērtības no 29% līdz 52.2%

1. K-vidējo algoritms

A screenshot of a computer

Description automatically generated with medium confidenceK-vidējo algoritmam ir iespējams noteikt noteikto klasteru skaitu vai arī klasteru skaita diapazonu. Izmantojot diapazona funkciju programma man paradīja ka vislabāk tā var sadalīt 2 klasteros, nākamais labākais variants bija 4 klasteri. Es izvelējos 4 klasterus jo man ir 4 klases. Šeit ir visi saskaitītie “Silhouette Score”

* K = 2 –> 0.425
* K = 3 –> 0.224
* K = 4 –> 0.249
* K = 5 –> 0.219
* K = 6 –> 0.185
* K = 7 –> 0.189
* K = 8 –> 0.177

1. Nepārraudzītā mašīnmācīšanās secinajums

Salīdzinot “Scatter Plot” grafiku no k-vidējo algoritmā un reālo es varu pateikt ka tie izskatās līdzīgi, bet hierarhiskās klasterizācijas rezultāts ir, pēc manam domam, labāk, jo rezultāta hierarhiskā klasterizācija sadalīja 4 klasteros kuriem piederēja ļoti daudz vienādu klases objektu.

1. Pārraudzītā mašīnmācīšanās
2. Algoritmu izvēle

Es izmantoju Loģistisko regresiju un kNN algaritmu.

(Visa informācija par algoritmiem ir paņemta no [4.2 teksta konspekta](https://estudijas.rtu.lv/pluginfile.php/4123495/mod_resource/content/3/4_2_temas_konspekts_2019.pdf))

Loģistiskā regresija arī ir pārraudzītās mašīnmācīšanās algoritms. To var izmantot binārās klasifikācijas uzdevumā; tādējādi tas nosaka datu objekta piederību vienai no divām klasēm / kategorijām (piemēram, veiksme/neveiksme, labs/slikts, jā/nē, uzvara/zaudējums utt.). Šajā datu kopā man ir 4 klases un es gribēju uzzināt ka Loģistika regresija strādās ar tiem, jo tā vislabāk darbojas ar bināro klasifikāciju, bet šeit tādas nav.

Loģistika regresijā ir iespējā samainīt regulācijas tipu (Lasso un Ridge) ka arī “stiprums” kas var būt diapazonā no 0.001 līdz 1000

K-tuvāko kaimiņu metodes pamatā ir līdzības jēdziens (dažreiz saukts par līdzības metriku, attālumu vai tuvumu). Šajā gadījumā datu objektu līdzība tiek noteikta, pamatojoties uz pazīmju vērtībām (tā sauktā pazīmju līdzība) (Kubat, 2017). Izpratne par līdzības jēdzienu atšķiras dažādiem datu tipiem (Kubat, 2017). Es gribēju to pielietot jo līdz šī momenta es jau izveidoju vairākas “Scatter Plot” diagrammas un man bija interesanti ka algoritms sadalīs objektus pirms tam apmācoties “pēc parauga”.

kNN algoritmam var samainīt kaimiņu skaitu (visi naturālie skaitļi), kā arī metriku bet es atstāju “Euclidean” jo es to metriku izmantoju citas vietas šī projekta.

1. Modeles izmantošana

Izmantojot **mākslīgus neironu tīklus** es mainīju vairākas lietas

|  |  |  |  |
| --- | --- | --- | --- |
| Neironu skaits paslēptajā slāni | Paslēpto slānu skaits | Iterāciju skaits | F1 |
| 100 | 2 | 1000 | 0.509 |
| 100 | 4 | 1000 | 0.196 |
| 1000 | 1 | 1000 | 0.833 |
| 100 | 1 | 3000 | **0.840** |

Vislabāk man sanāca izmantojot 1 slēpto slānī ar lielu iterāciju skaitu. Izmantojot kļūdu matricu paradās tādā tabula:

A screenshot of a graph

Description automatically generated with medium confidence

Izmantojot **loģistisko regresiju** es mainīju tādus parametrus:

|  |  |  |
| --- | --- | --- |
| Regulācijas tips | Stiprums | F1 |
| Lasso | 0.001 | 0.129 |
| Lasso | 1000 | 0.872 |
| Lasso | 50 | **0.910** |
| Ridge | 0.001 | 0.196 |
| Ridge | 1000 | 0.891 |
| Ridge | 10 | 0.897 |

Izmantojot “Lasso” regulācijas tipu ar stiprumu 50 man sanāca šādi rezultāti

A screenshot of a graph

Description automatically generated with medium confidence

Izmantojot **kNN** **algoritmu** es nezināju cik man vērtību pārbaudīt jo man efektīvi ir tikai 1 parametrs un tas ir kaimiņu skaits. Rezultāti man sanāca sekojoši:

|  |  |
| --- | --- |
| kaimiņu skaits | F1 |
| 1 | 0.827 |
| 15 | 0.858 |
| 5 | **0.871** |

Uzstādot kaimiņu skaitu uz k = 5 man sanāca šāda kļūdu matrica:

A screenshot of a graph

Description automatically generated with medium confidence

1. Pārraudzītā mašīnmācīšanās secinājums

Projektā izstrādes laika es paeksperimentēju ar kNN un loģistisko algoritmu, kā arī ar mākslīgiem neironu tīkliem. Rezultāta vislabāk sevi paradīja loģistiskais algoritms ar F1 = 0.91 kas nozīmē ka tas pareizi klasificē datus ar precizitāti 91%. Vel bija interesanti uzzināt ka pievienojot slēptos slāņus neironu tīkliem nemainot iterāciju skaitu samazina efektivitāti.

1. Kopejais secinajums

Tas bija mana pirma reize kad es strādāju ar mākslīgo intelektu algoritmiem un es uzzināju daudz jaunu informāciju par datu kopnes apstrādi, vizualizaciju un ari ka tiek trenēti paredzēti un neparedzēti mašīnmācīšanās algoritmi. Apakša es pievienošu mana projekta loģikas ekrānšāvienu. Paldies par iespēju uzzināt vairāk par to ka strada klasifikacija.

A picture containing text, diagram, map

Description automatically generated