KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Intelektikos pagrindai (P176B101)

Projekto darbo ataskaita

Atliko:

IFF-6-8 gr. Edvinas Repečka

IFF-6-8 gr. Lukas Šivickas

??????? gr. Ignas Kucinas

Priėmė:

Doc. Germanas Budnikas

KAUNAS 2019

TURINYS

[1. Santrauka 3](#_Toc9255972)

[2. Atlikėjų sąrašas 4](#_Toc9255973)

[3. Programinės sistemos sukūrimas/ pritaikymas duomenims surinkti 5](#_Toc9255974)

[4. Duomenų surinkimas (iš sukurto programinio produkto) 8](#_Toc9255975)

[5. Duomenų paruošimas ir valymas (pilnai suprogramuotas) 9](#_Toc9255976)

[6. Dimensijų sumažinimas (nepriklausoma realizacija) 10](#_Toc9255977)

[7. 1-ojo mašininio mokymosi metodo su mokytoju panaudojimas (pilnai suprogramuotas) 11](#_Toc9255978)

[8. 2-ojo mašininio mokymosi metodo su mokytoju panaudojimas (pilnai suprogramuotas) 12](#_Toc9255979)

[9. Mašininio mokymosi metodų su mokytoju rezultato parinkimas balsavimo principu (???) 13](#_Toc9255980)

[10. Balsavimo principu gautų rezultatų pritaikymas (???) 14](#_Toc9255981)

[11. Mašininio mokymosi be mokytojo metodo panaudojimas (pilnai suprogramuotas) 15](#_Toc9255982)

[12. Mokymosi be mokytojo metodo gautų rezultatų pritaikymas (programoje) / pakomentavimas ataskaitoje (tik rezultatų pakomentavimas) 16](#_Toc9255983)

[13. Literatūra 17](#_Toc9255984)

# Santrauka

Kadangi stiklų gali būti daug, mes sugalvojome sukurti klasifikavimo sistema, kuri paėmus duomenis iš sistemos, šitam projektui esame susikūrę savą, juos suformavus ir suteikus galimybę pašalinti nutolusius duomenis mūsų programa naudodama tris skirtingus būdus: neuroninį tinklą su „back propogation“, bajeso ir kNN (K-nearest neighbours). Programa taip pat naudodama kryžminę patikra suranda apytiklsų tų duomenų klasifikavimo metodo tiklsumą.

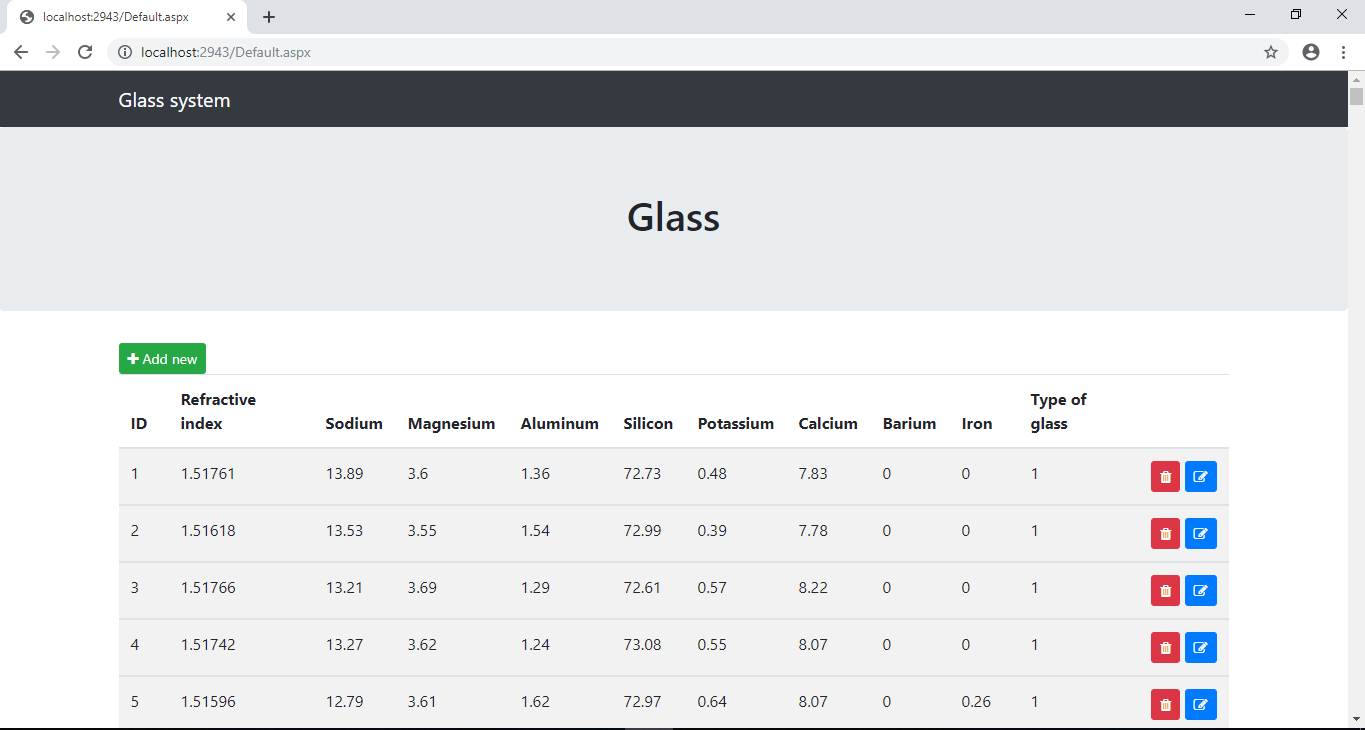
# Atlikėjų sąrašas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Vardas Pavardė | Užsiėmimo laikas | Atsakomybės projekte | Parengti skyriai ataskaitoje |
| 1 | Lukas Šivickas | Tr. 09:00 | Demensijų sumažinimas Duomenų paruošimas ir valymas Metodas be mokytojo | 3, 4, 10, 12 |
| 2 | Edvinas Repečka | Pr. 17:00 | Sukurtas neuroninis tinklas  Back propogation metodas(Metodas su mokytoju)  Duomenų valdymo sistema  Duomenų surinkimas iš sukurtos sistemos | 1, 2, 6, 7 |
| 3 | Ignas Kucinas | Pr. 17:00 |  | 12 |

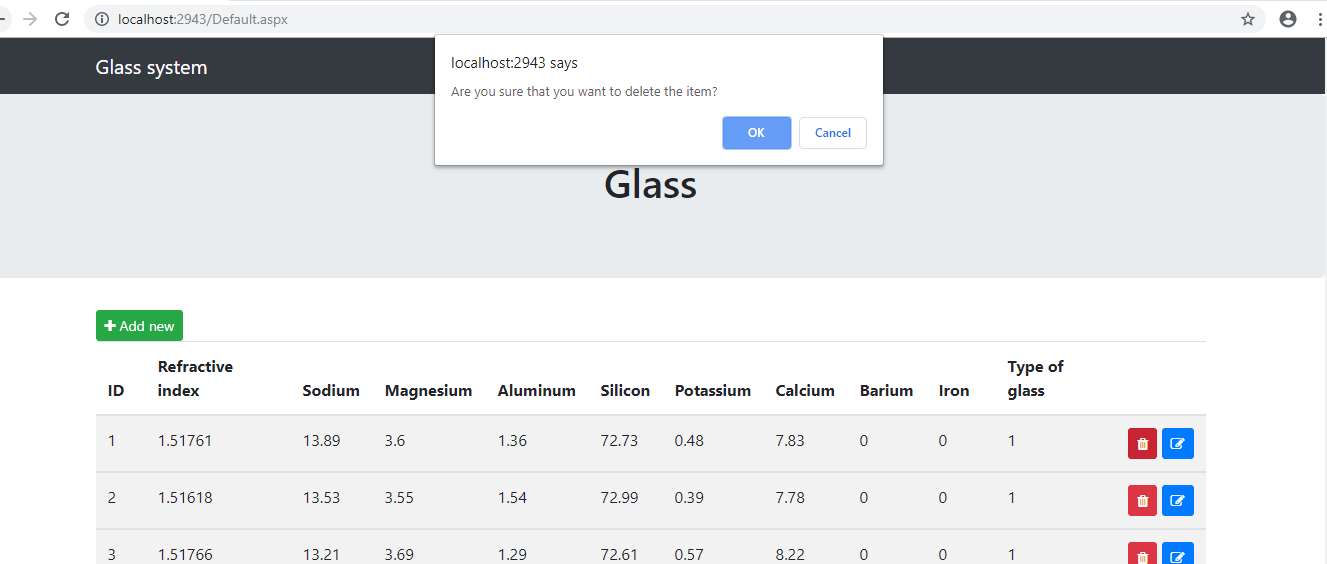
# Programinės sistemos sukūrimas/ pritaikymas duomenims surinkti

Mūsų sistema yra stiklo duomenų vadlymui. Joje vartotojas gali redaguoti, pridėti, šalinti ir peržiūrėti sistemoje esančius užregistruotus stiklus.

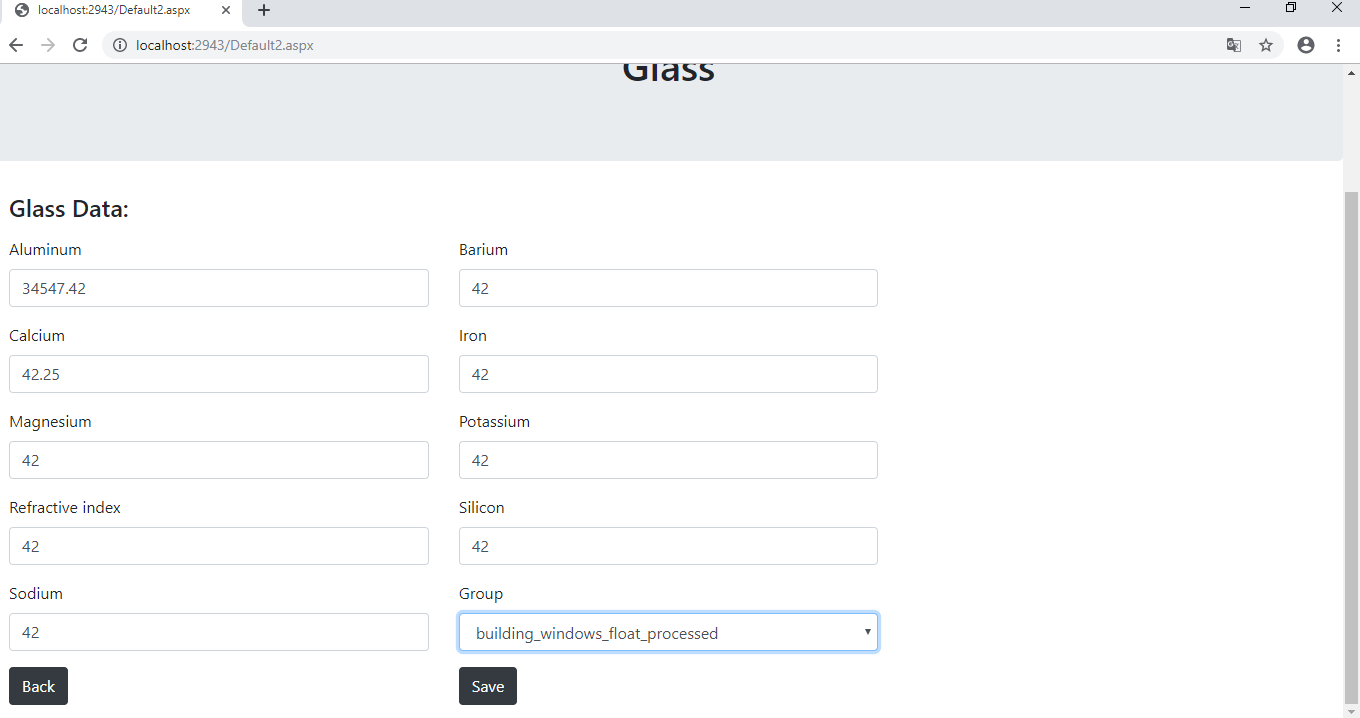
Sistemos vartotojo sąsajos ekrano kopijos.



1 pav. Stiklo peržiūros forma.



2 pav. Stiklo šalinimo žinutė



3 pav. Stiklo redagavimo ir pridėjimo forma.

Kodo fragmentas, naudojamas duomenims surinkti:

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace WindowsFormsApp1.DataFormating  {  class GetDataFromWeb  {  public List<Glass> GetGlasses()  {  System.Net.WebClient wc = new System.Net.WebClient();  string webData = wc.DownloadString("http://localhost:2943/Default.aspx");  int textFrom = webData.IndexOf("<div class=\"table-responsive\">") + "<div class=\"table-responsive\">".Length;  int textTo = webData.LastIndexOf("</table>") - textFrom;  string data = webData.Substring(textFrom, textTo);  data = data.Replace("\r\n", "");  data = data.Replace(" ", "");  textFrom = data.IndexOf("<tbody") + "<tbody".Length;  string dataNew = data.Substring(textFrom);  int step = 0;  List<Glass> AllGlass = new List<Glass>();  foreach (var row in dataNew.Split(new string[] { "<tr>" }, StringSplitOptions.RemoveEmptyEntries))  {  step = 0;  string[] items = row.Split(new string[] { "<td>" }, StringSplitOptions.RemoveEmptyEntries);  if (items.Count() >= 7)  {  Glass newGlass = new Glass();  foreach (var item in items)  {  textFrom = item.IndexOf(">") + ">".Length;  textTo = item.IndexOf("</span></td>") - textFrom;  var info = item.Substring(textFrom, textTo);  switch (step)  {  case 0:  newGlass.id = int.Parse(info);  break;  case 1:  newGlass.refractive\_index = double.Parse(info);  break;  case 2:  newGlass.sodium = double.Parse(info);  break;  case 3:  newGlass.magnesium = double.Parse(info);  break;  case 4:  newGlass.aluminum = double.Parse(info);  break;  case 5:  newGlass.silicon = double.Parse(info);  break;  case 6:  newGlass.potassium = double.Parse(info);  break;  case 7:  newGlass.calcium = double.Parse(info);  break;  case 8:  newGlass.barium = double.Parse(info);  break;  case 9:  newGlass.iron = double.Parse(info);  break;  case 10:  newGlass.group\_type = int.Parse(info);  break;  default:  break;  }  step++;  }  AllGlass.Add(newGlass);  }  }  return AllGlass;  }  }  } |

# Duomenų surinkimas (iš sukurto programinio produkto)

Mes turime stiklą;

Jis turi 7 grupes:

* building\_windows\_float\_processed
* building\_windows\_non\_float\_processed
* vehicle\_windows\_float\_processed
* vehicle\_windows\_non\_float\_processed
* containers
* tableware
* headlamps

Stiklo sistemos paskirtis – aprašyti specifiinių grupių duomenis.

Stiklas turi 9 įvestis (be id ir grupės numerio):

* Aliuminis
* Baris
* Lūžio rodiklis
* Silikonas
* Natris
* Kalcis
* Magnis
* Geležis
* Kalis
* Tipas

Duomenų šaltinis: <http://localhost:2943/Default.aspx> - tai yra mūsų sukurtos sistemos tinklo puslapis.

Duomenų diapozonai:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Diapozonas | Aliuminis | Baris | Lūžio rodiklis | Silikonas | Natris | Kalcis | Magnis | Geležis | Kalis |
| Min | 0.29 | 0 | 1.5112 | 69.81 | 10.73 | 5.43 | 0 | 0 | 0 |
| Max | 3.5 | 3.15 | 1.5339 | 75.41 | 17.38 | 16.19 | 4.49 | 0.51 | 6.21 |

Pavyzdinis stiklos elementas:



4 pav. Pavyzdinio stiklo duomenys

# Duomenų paruošimas ir valymas (pilnai suprogramuotas)

Nutolusius duomenis pašaliname naudodami Tukey metodą. Naudojamas *k* yra 2. Kadangi pas mus yra 9 duomenų dimensijos, mes turėjome šiek tiek kitaip panaudoti šį metodą. Mes nusprendėme, kad jei tikrinamas duomenų vienetas turi 3 ar daugiau nukrypusius duomenis, jis visas skaitomas kaip nukrypęs. Po šio valymo iš visos mūsų duomenų imties (600) yra pašalinami 45 duomenys (lieka 555) . Nukrypusių duomenų pavyzdys:

|  |
| --- |
| Glass(id: 288, refractive\_index: 1.52211, so: 14.189, mg: 3.78, al: 0.91, si: 71.36, pt: 0.23, ca: 9.14, ba: 0, fe:0.37, group: 3)  Glass(id: 140, refractive\_index: 1.52664, so: 11.23, mg: 0, al: 0.77, si: 73.21, pt: 0, ca: 14.68, ba: 0, fe: 0, group: 2)  Glass(id: 177, refractive\_index: 1.52614, so: 13.7, mg: 0, al: 1.36, si: 71.24, pt: 0.189, ca: 13.44, ba: 0, fe: 0.1, group: 2)  Glass(id: 136, refractive\_index: 1.53125, so: 10.73, mg: 0, al: 2.1, si: 69.81, pt: 0.58, ca: 13.3, ba: 3.15, fe:0.28, group: 2)  Glass(id: 137, refractive\_index: 1.53393, so: 12.3, mg: 0, al: 1, si: 70.16, pt: 0.12, ca: 16.19, ba: 0, fe:0.24, group: 2)  Glass(id: 135, refractive\_index: 1.52475, so: 11.45, mg: 0, al: 1.88, si: 72.19, pt: 0.81, ca: 13.24, ba: 0, fe:0.34, group: 2) |

# Dimensijų sumažinimas (nepriklausoma realizacija)

Dimensijų sumažinimui buvo naudojama koreliacijos lentelė. Jei atributas labai koreliuoja su visais kitais atributais, tai reiškia, kad jis potencialiai gali būti pašalintas, nes jis nelabai įtakoja galutinį rezultatą ir skaičiavimus.



Kaip matome, neradome vieningo atributo, kuri galėtumėme pašalinti, kadangi, nei vienas atributas nekoreliuoja stipriai su visais kitais, tik su pavieniais. Taigi, mūsų atributų (dimensijų skaičius) nesumažėjo.

# 1-ojo mašininio mokymosi metodo su mokytoju panaudojimas (pilnai suprogramuotas)

Metodas: „Back propogation“

Metodo veikimas:

Metodo tiklsumas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eksperimentas | Duomenų kiekis | Mokymosi duomenų kiekis | Testavimo duomenų kiekis | Tikslumas |
| 1 | 555 | 479 | 120 | 62.498 % |
| 2 | 555 | 479 | 120 | 67.83 % |
| 3 | 555 | 479 | 120 | 70.002 % |
| Vidurkis | | | | 66.7766 % |

Mūsų metodas apmokydamas neuroninį tinklą sugeba klasifikuoti duomenis apytiklsiai ~66 % tikslumu.

Metodo kryžminė patikra:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteracija | Duomenų kiekis | Mokymosi duomenų kiekis | Testavimo duomenų kiekis | Tikslumas |
| 1 | 555 | 479 | 120 | 55.83 % |
| 2 | 555 | 479 | 120 | 65.83 % |
| 3 | 555 | 479 | 120 | 60 % |
| 4 | 555 | 479 | 120 | 65.83 % |
| 5 | 555 | 479 | 120 | 65 % |
| Vidurkis | | | | 62.498 % |

Gauname, kad įvykdžius kryžminę patikrą mūsų metodas sugeba klasifikuoti duomenis iki ~ 66 % tikslumo.

# 2-ojo mašininio mokymosi metodo su mokytoju panaudojimas (pilnai suprogramuotas)

Metodo pavadinimas; iki pusės puslapio metodo anotacija su nuorodą į šaltinį ir į iliustracinį pavyzdį.

Kryžminės patikros eksperimentai apibendrinti lentelėje. Lentelėje paryškinti didžiausią metodo tikslumo įvertį.

# Mašininio mokymosi metodų su mokytoju rezultato parinkimas balsavimo principu (???)

|  |  |
| --- | --- |
| balsavimo tvarka\_1 | Tikslumas |
| ... |  |
| balsavimo tvarka\_K | Tikslumas |

Aprašyti panaudotą (-as) balsavimo tvarką (-as)

# Balsavimo principu gautų rezultatų pritaikymas (???)

Aprašyti pritaikymo būdą (pvz. programoje įvedame grybų atributų rinkinį ir programa pateikia rekomendaciją - valgyti grybą arba ne).

# Mašininio mokymosi be mokytojo metodo panaudojimas (pilnai suprogramuotas)

Metodo pavadinimas; iki pusės puslapio metodo anotacija su nuorodą į šaltinį ir į iliustracinį pavyzdį.

Kryžminės patikros eksperimentai apibendrinti lentelėje. Lentelėje paryškinti didžiausią metodo tikslumo įvertį.

Pateikti apmokymo rezultatus bei juos pakomentuoti.

Mašininiam mokymuisi be mokytojo buvo panaudotas K-Nearest-Neighbors (kNN). kNN yra įgyvendinamas pasiemus *training* duomenis ir nurodant kiek kaimynų paimti klasifikavimui. Tuomet, kai norime suklasifikuoti nežinomą elementą, paskaičiuojame atstumus nuo jo iki visų kitų duomenų taškų. Tuomet pasiemame *k* artimiausius taškus ir pažiūrime kokios klasės egzempliorių yra daugiausia. (iliustracinis video: <https://www.youtube.com/watch?v=UqYde-LULfs>)

Kryžminė patikra:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteracijos # | Duomenų kiekis | Mokymo d. k. | Testavimo d. k. | Tikslumas |
| 1 | 555 | 443 | 112 | 13.39% |
| 2 | 555 | 443 | 112 | 20.54% |
| 3 | 555 | 443 | 112 | 18.75% |
| 4 | 555 | 443 | 112 | 11.61% |
| 5 | 555 | 443 | 112 | 15.32% |
| Vidutinis | 555 | 443 | 112 | 15.92% |

Metodo rezultatų apibendrinimas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eksperimento # | Duomenų kiekis | Mokymo d. k. | Testavimo d. k. | Vidutinis tikslumas |
| 1 | 555 | 443 | 112 | 15.92% |
| 2 | 555 | 443 | 112 | 17.35% |
| 3 | 555 | 443 | 112 | 16.46% |
| Vidutinis | 555 | 443 | 112 | 16.58% |

Taigi su kNN sugebėjome pasiekti apie 16.58% tikslumą. Tai įtakoja mūsų dimensijų skaičius, kadangi kNN geriau dirba, kai yra mažiau dimensijų. Taip pat tai gali įtakoti per mažas duomenų kiekis.

# Mokymosi be mokytojo metodo gautų rezultatų pritaikymas (programoje) / pakomentavimas ataskaitoje (tik rezultatų pakomentavimas)

Aprašyti pritaikymo būdą (pvz. programoje įvedame grybų atributų rinkinį ir programa pateikia rekomendaciją - valgyti grybą arba ne).

# Literatūra

* <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>
* <https://www.youtube.com/watch?v=tIeHLnjs5U8&t=4s>
* <https://www.youtube.com/watch?v=Ilg3gGewQ5U&t=623s>
* <https://www.sciencedirect.com/topics/earth-and-planetary-sciences/unsupervised-classification>
* <https://medium.com/@equipintelligence/java-algorithms-the-k-nearest-neighbor-classifier-4faca7ad26b2>
* <https://www.researchgate.net/figure/Pseudocode-for-KNN-classification_fig7_260397165>
* <https://medium.com/datadriveninvestor/classification-algorithms-in-machine-learning-85c0ab65ff4>
* <https://www.youtube.com/watch?v=UqYde-LULfs>