



VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS KATEDRA

**Skaitmeninis intelektas ir sprendimų priėmimas**

**III užduotis: Tiesioginio sklaidimo DNT naudojant sistemą WEKA**

Tomas Giedraitis  
VU MIF Informatika  
3 kursas 3 grupė

Vilnius 2021

## 1. Darbo tikslas

Išmokyti dirbtinį neuroninį tinklą (DNT) teisingai klasifikuoti duomenis naudojant sistemą WEKA, tuo pačiu susipažįstant su šios programos veikimu, bei, apmokius DNT ir išsiminus jo parametrus, pakartoti naujų duomenų klasifikavimą su MS Excel tipo programa.

## 2. Darbo eiga

### 2.1. Duomenų paruošimas

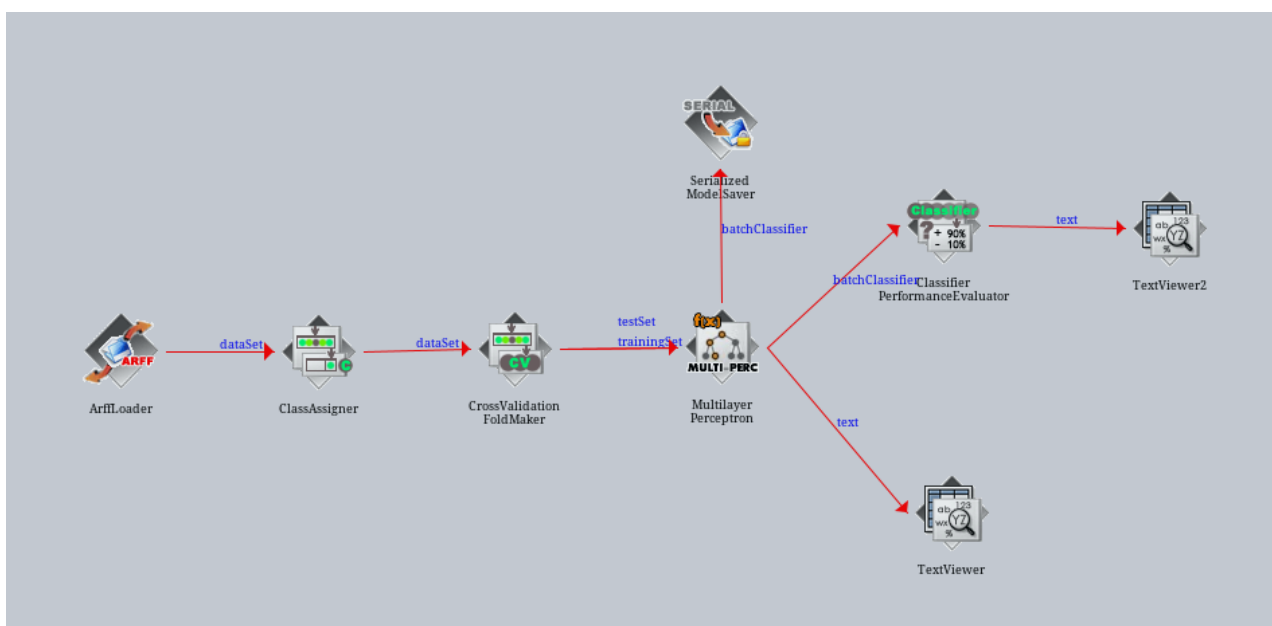
Šiame darbe naudojami irisų duomenys<sup>1</sup>, kurių klasės yra žinomos (Setosa, Versicolor ir Virginica). Parsisiųstas failas *iris.data* (.csv tipo) buvo padalintas į dvi dalis – mokymo ir validavimo failą (po 40 kiekvienos klasės duomenų) *iris\_train\_test.arff* ir naujų duomenų (testinių) failą *iris\_new.arff*, kuriame yra po 10 duomenų kiekvienai klasei. Failai su galūne *.arff* yra WEKA programos atpažįstami failai, kuriuos gauti reikėjo rankiniu būdu nežymiai pamodifikavus *iris.data* failo turinį (papildomai pridėdamos duomenų aibės pavadinimas, atributų pavadinimai, klasių reikšmės). Faile *iris\_new.arff* vietoj klasių reikšmių parašyti klaustukai – tai dirbtiniam neuroniniui tinklui nežinomos klasės, kurias jis turi nustatyti (mes jas žinome).

### 2.2. DNT darbo sekų konstravimas WEKA programoje

#### 2.2.1. DNT apmokymas

##### 2.2.1.1. Apmokymo seka

DNT mokymasis buvo realizuotas šia komponentių seka WEKA programoje:



1 pav. DNT mokymasis (WEKA)

<sup>1</sup> <https://archive.ics.uci.edu/ml/datasets/iris>

### 2.2.1.2. Sekos komponentės

Darbe atliktus veiksmus patogiu paašškinti nuosekliai pereinant per kiekvieną sekos komponentę, kartu ir paašškinant, ką jos daro:

**ArrfLoader** – priima duomenų failą *.arrf* tipo (naudojamas failas *iris\_train\_test.arrf*). Toliau perduoda įvesties duomenų aibę iš šio failo.

**ClassAssigner** – nurodoma, kuris atributas duomenų faile nusako klasę (buvo pasirinktas *class* atributas). Toliau perduoda duomenų aibę.

**CrossValidationFoldMaker** – sudaro kryžminės patikros aibes – pasirenkama, į kiek dalių bus suskaidoma duomenų aibė kryžminei patikrai. Buvo pasirinkta suskaidyti į 10 dalių. Kadangi įvesties duomenų yra 120, tai jie buvo padalinti į 10 aibių po 12 elementų, iš kurių viena pasirenkama kaip validavimo aibė (*testSet*), o likusios – mokymo (*trainingSet*), ir visos jos perduodamos toliau sekančiai komponentei (perdavimas įvyksta 10 kartų, kiekvieną kartą vis kita aibė iš 10-ies pasirenkama kaip validavimo).

**MultilayerPerceptron** – daugiasluoksnio perceptrono komponentė. Joje galima keisti įvairius parametrus, arba pakrauti jau išsaugotą DNT modelį. Buvo keičiami šie parametrai:

- paslėptų neuronų sluoksnių skaičius
- paslėptų neuronų skaičius kiekviename sluoksnyje
- mokymosi greitis,  $\eta$
- mokymosi pagreitis,  $p$  (momentum)

Buvo siekiama parinkti tokius parametrus, kad klasifikavimo tikslumas būtų didžiausias, ir kartu neuroninis tinklas nebūtų per sudėtingas (paslėptų neuronų skaičius nebūtų itin didelis).

Mokymo epochų skaičius visuomet buvo fiksuotas ir lygus 500.

Ši komponentė toliau perduoda jau sukurtą naują klasifikatorių (daugiasluksnį perceptroną).

Taip pat, prie jos prijungta komponentė **TextViewer**, kuriai perduodama yra tekstinė informacija apie nustatytas svorių reikšmes. TextViewer sudaro galimybę peržiūrėti šią informaciją.

**ClassifierPerformanceCalculator** – įvertina klasifikavimo rezultatus. Prijungus komponentę **TextViewer2**, galima matyti rezultatus tekstinio failo pavidalu.

Pateikiamas teisingai ir neteisingai suklasifikuotų klasių skaičius, klasifikavimo tikslumas, klasifikavimo rezultatų matrica (confusion matrix), teisingų teigiamų rezultatų santykis, teisingų neigiamų rezultatų santykis, preciziškumas, įvairios paklaidos (vidutinė absoliuti (*mean absolute error*), šakninė vidutinė kvadratinė (*root mean squared error*) ir kt.) ir keli kiti rezultatai.

Derinant DNT modelį (t.y. keičiant jau minėtus parametrus) pagrinde buvo remiamasi bendru klasifikavimo tikslumu.

**SerializedModelSaver** – ši komponentė, prijungta prie **MultilayerPerceptron** komponentės, išsaugo esamą DNT modelį į failą. Tai bus naudinga vėliau testuojant DNT su naujais duomenimis.

#### 2.2.1.3. Mokymo rezultatai

Geriausias DNT apmokymo klasifikavimo tikslumas, lygus 0,975, buvo gautas su šiais parametrais:

- Vienas paslėptų neuronų sluoksnis, jame – 5 neuronai.
- mokymosi greitis  $\eta = 0,3$
- mokymosi pagreitis  $p = 0,2$

Kelių tirtų atvejų klasifikavimo rezultatai palyginimui (pačių tirtų atvejų buvo daugiau):

1 lentelė. Klasifikavimo rezultatai su skirtingais parametrais

Paslėptų neuronų skaičius (per kablelį atskiriami keli paslėpti sluoksniai)	Mokymosi greitis $\eta$	Mokymosi pagreitis $p$	Teisingai nustatytų klasių skaičius	Neteisingai nustatytų klasių skaičius	Klas. tikslumas
5, 4, 3	0,5	0,5	112	8	0,933
2, 3	0,3	0,2	114	6	0,950
5, 5	0,3	0,2	114	6	0,950
5	1,0	0,2	115	5	0,958
5	0,7	0,4	116	4	0,967
5	0,3	0,2	117	3	0,975

Toks pat klasifikavimo tikslumas buvo gautas ir naudojant kelias kitas neuronų skaičiaus, mokymo greičio ir pagreičio kombinacijas, tačiau tada buvo papildomai žiūrima į rezultatų paklaidų dydžius. Jų palyginimai čia nėra atvaizduoti. Tačiau pateikiamas paveikslukas, kaip atrodo rezultatų aprašas:

```

=== Evaluation result ===

Scheme: MultilayerPerceptron
Options: -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H 5 -R
Relation: iris_train_test

=== Summary ===

Correctly Classified Instances      117           97.5 %
Incorrectly Classified Instances     3            2.5 %
Kappa statistic                     0.9625
Mean absolute error                  0.0339
Root mean squared error              0.1259
Relative absolute error              7.6169 %
Root relative squared error          26.714 %
Total Number of Instances           120

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      1.000    0.000    1.000     1.000    1.000     1.000    1.000    1.000    Iris-setosa
      0.950    0.013    0.974     0.950    0.962     0.944    0.996    0.993    Iris-versicolor
      0.975    0.025    0.951     0.975    0.963     0.944    0.996    0.992    Iris-virginica
Weighted Avg.    0.975    0.013    0.975     0.975    0.975     0.963    0.997    0.995

=== Confusion Matrix ===

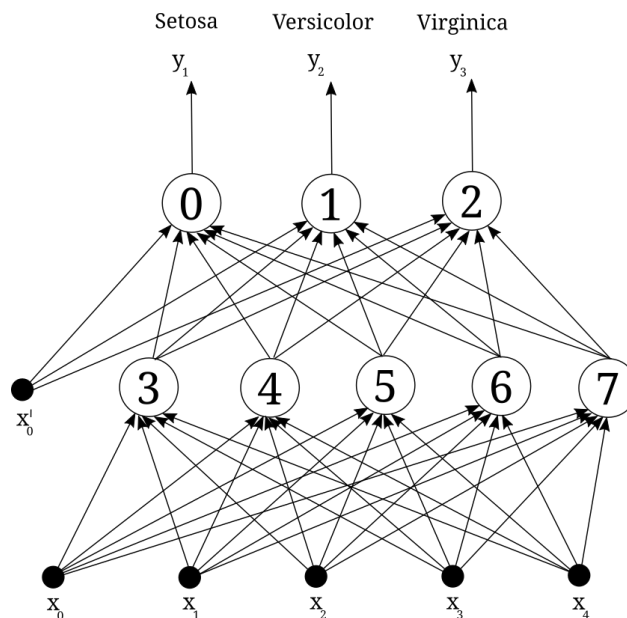
 a  b  c  <-- classified as
40  0  0 | a = Iris-setosa
 0 38  2 | b = Iris-versicolor
 0  1 39 | c = Iris-virginica

```

2 pav. Mokymosi rezultatai (WEKA)

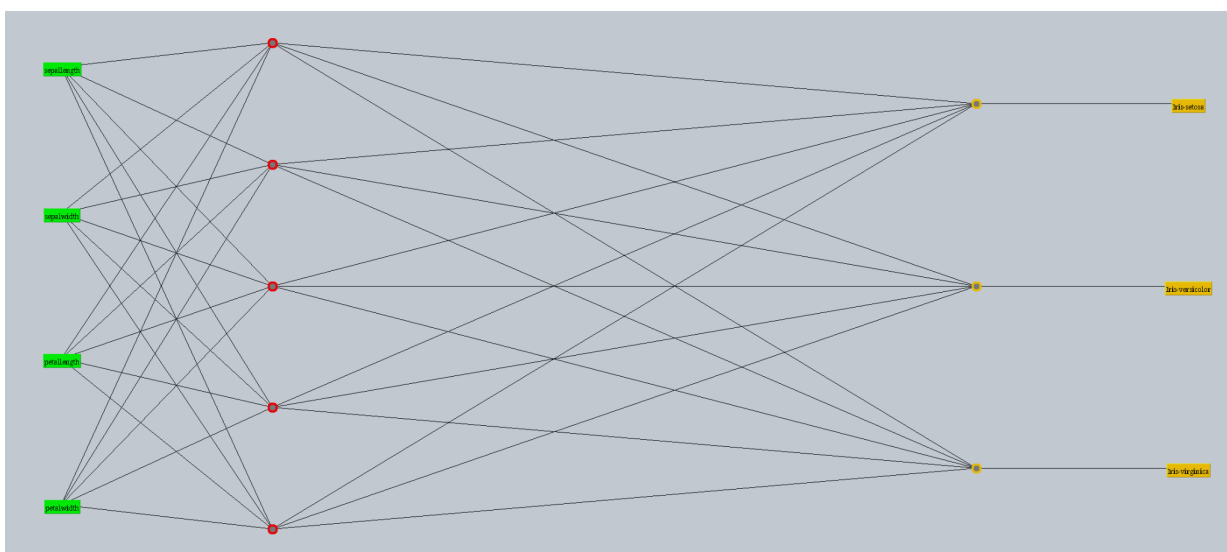
## 2.2.2. Neuroninio tinklo vaizdas

Neuroninio tinklo schema:



3 pav. Neuroninio tinklo schema

Neuroninio tinklo vaizdas WEKA programoje:

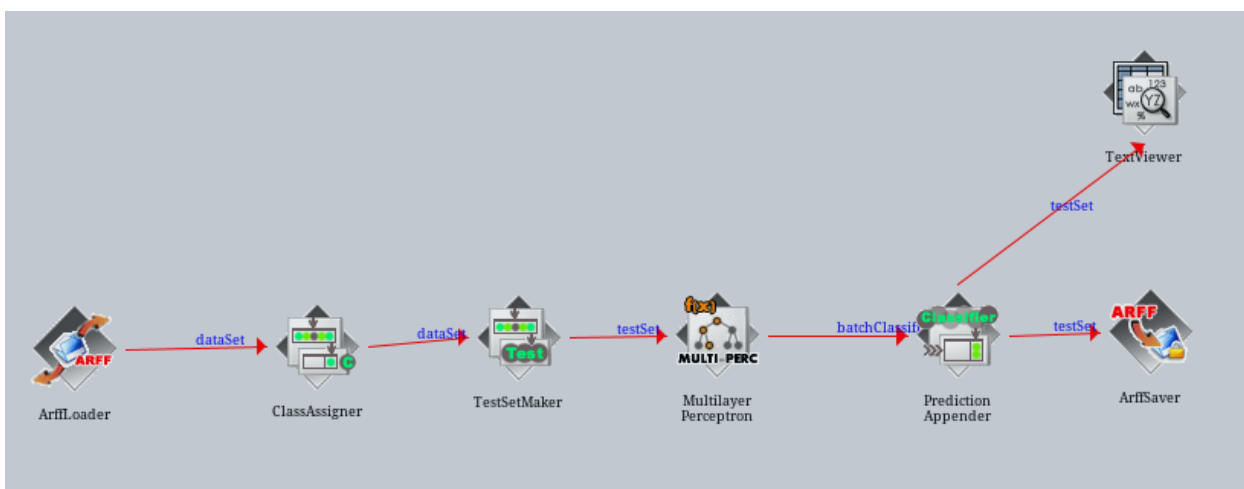


4 pav. DNT vaizdas (WEKA)

## 2.2.3. Naujų duomenų klasifikavimas

### 2.2.3.1. Testavimo seka

DNT testavimas su naujais duomenimis buvo realizuotas šia komponentių seka WEKA programoje:



5 pav. Testavimo seka (WEKA)

### 2.2.3.2. Sekos komponentės

Čia matome kelias dar neaptartas komponentes:

**TestSetMaker** – pažymi jai paduodamą duomenų aibę kaip testavimo duomenis, ir perduoda juos daugiasluoksniui perceptronui.

**PredictionAppender** – paduodamas klasifikatorius, ir ši komponentė pateikia naujų duomenų klasių prognozių rezultatus .arrf formatu. Taip pat galima nustatyti, kad papildomai būtų išvestos kiekvienam įėjimo objektui priskirtų klasių tikimybės. Norint WEKA programoje peržiūrėti šią išvestį, panaudojama papildoma **TextViewer** komponentė, o išsaugoti išvesčiai į failą naudojama **ArrfSaver** komponentė.

#### 2.2.3.3. Naujų duomenų klasifikavimo rezultatai

Gautas minėtas klasifikavimo prognozių .arrf failas. Jo vaizdas **TextViewer** komponentėje:

```
@relation iris_new_set_1_of_1

@attribute sepallength numeric
@attribute sepalwidth numeric
@attribute petallength numeric
@attribute petalwidth numeric
@attribute class {Iris-setosa,Iris-versicolor,Iris-virginica}
@attribute 'class_predicted_by: MultilayerPerceptron' {Iris-setosa,Iris-versicolor,Iris-virginica}

@data
5,3.5,1.3,0.3,?,Iris-setosa
4.5,2.3,1.3,0.3,?,Iris-setosa
4.4,3.2,1.3,0.2,?,Iris-setosa
5,3.5,1.6,0.6,?,Iris-setosa
5.1,3.8,1.9,0.4,?,Iris-setosa
4.8,3.1,1.4,0.3,?,Iris-setosa
5.1,3.8,1.6,0.2,?,Iris-setosa
4.6,3.2,1.4,0.2,?,Iris-setosa
5,3.3,7,1.5,0.2,?,Iris-setosa
5,3.3,1.4,0.2,?,Iris-setosa
5.5,2.6,4.4,1.2,?,Iris-versicolor
6.1,3.4,6.1,1.4,?,Iris-versicolor
5.8,2.6,4.1,1.2,?,Iris-versicolor
5.2,3.3,3.1,?,Iris-versicolor
5.6,2.7,4.2,1.3,?,Iris-versicolor
5.7,3.4,2.1,1.2,?,Iris-versicolor
5.7,2.9,4.2,1.3,?,Iris-versicolor
6.2,2.9,4.3,1.3,?,Iris-versicolor
5.1,2.5,3.1,1.1,?,Iris-versicolor
5.7,2.8,4.1,1.3,?,Iris-versicolor
6.7,3.1,5.6,2.4,?,Iris-virginica
6.9,3.1,5.1,2.3,?,Iris-virginica
5.8,2.7,5.1,1.9,?,Iris-virginica
6.8,3.2,5.9,2.3,?,Iris-virginica
6.7,3.3,5.7,2.5,?,Iris-virginica
6.7,3.5,2.2,2.3,?,Iris-virginica
6.3,2.5,5.1,1.9,?,Iris-virginica
6.5,3.5,2.2,?,Iris-virginica
6.2,3.4,5.4,2.3,?,Iris-virginica
5.9,3.5,1.1,1.8,?,Iris-virginica
```

6 pav. Naujų duomenų klasifikavimo rezultatai (WEKA)

Gautos duomenų klasės parašytos prie kiekvieno duomens, po klaustuko. Klaustukas taip pat buvo įvesties faile – tarėme, kad naujų duomenų klasės mums nėra žinomos. Tačiau iš tiesų nauji duomenys turėjo nuosekliai išsidėsčiusias tris grupes po 10 įrašų – Iris-setosa, Iris-versicolor ir Iris-virginica klasių. Taigi matome, kad visi nauji duomenys suklasifikuoti teisingai. Klasifikavimo bendras tikslumas buvo geriausias įmanomas – 1,00, t.y. nė viena klasė nebuvo priskirta neteisingai.

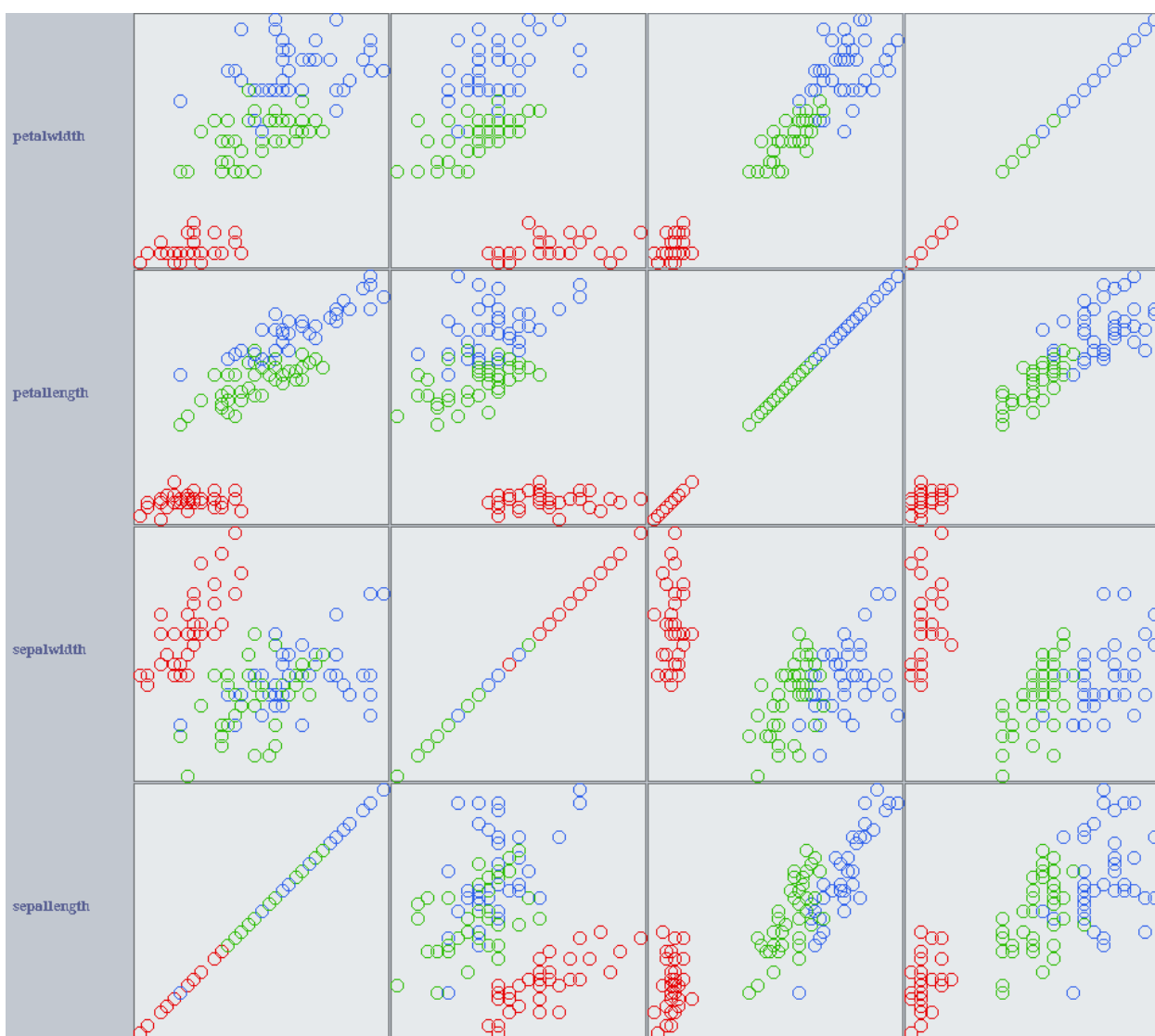
## 2.2.4. Duomenų požymių porų vaizdai

DNT mokymo schemoje prie **ClassAssigner** komponentės pridėjus komponentę **ScatterPlotMatrix**, galima pamatyti duomenų požymių (stulpelių) porų vaizdus Dekarto koordinatinių sistemoje.

Sutartinis spalvinis žymėjimas:

**Raudona** – Iris-setosa  
**Žalia** – Iris-versicolor  
**Mėlyna** – Iris-virginica

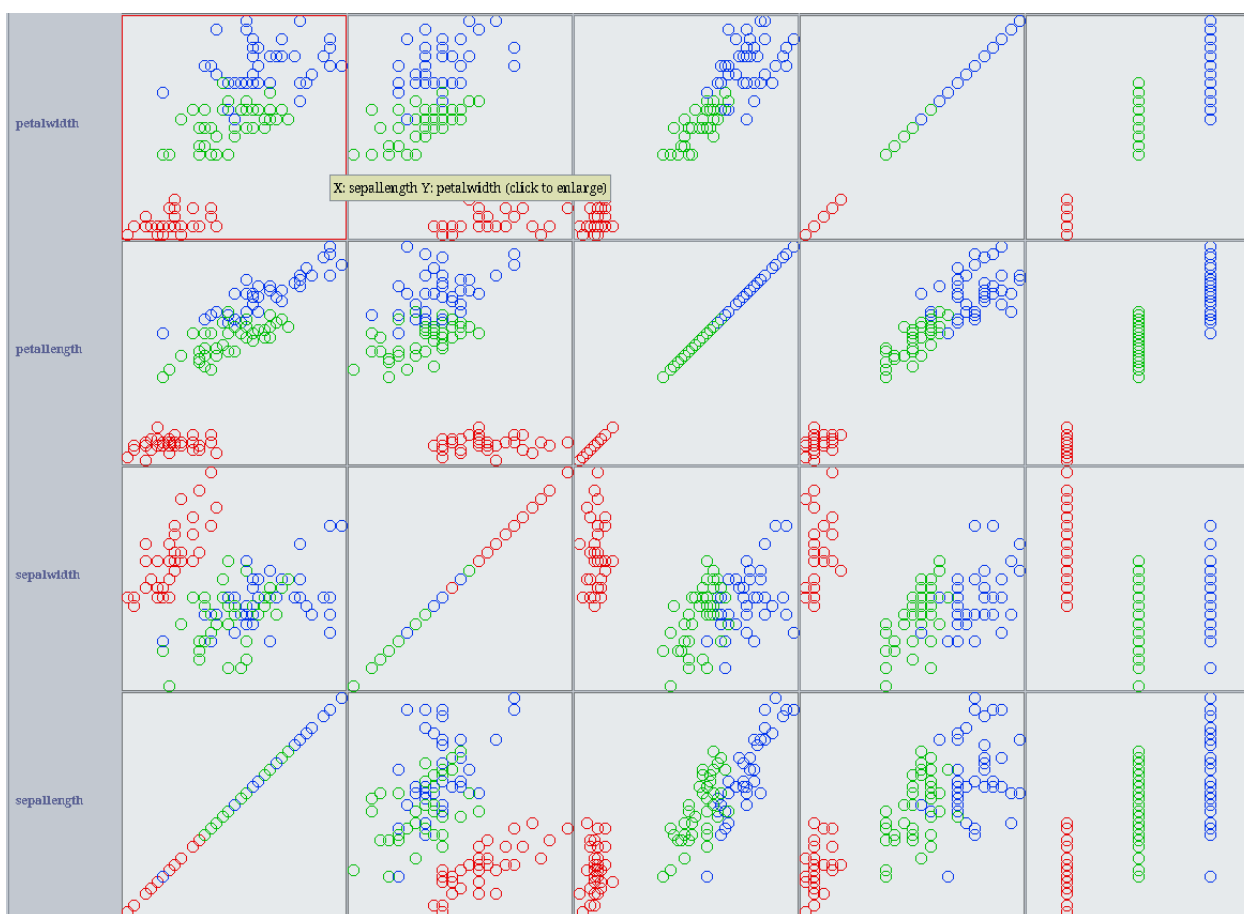
Mokymo duomenų požymių porų vaizdai:



7 pav. Mokymo duomenų požymių porų vaizdai (WEKA)



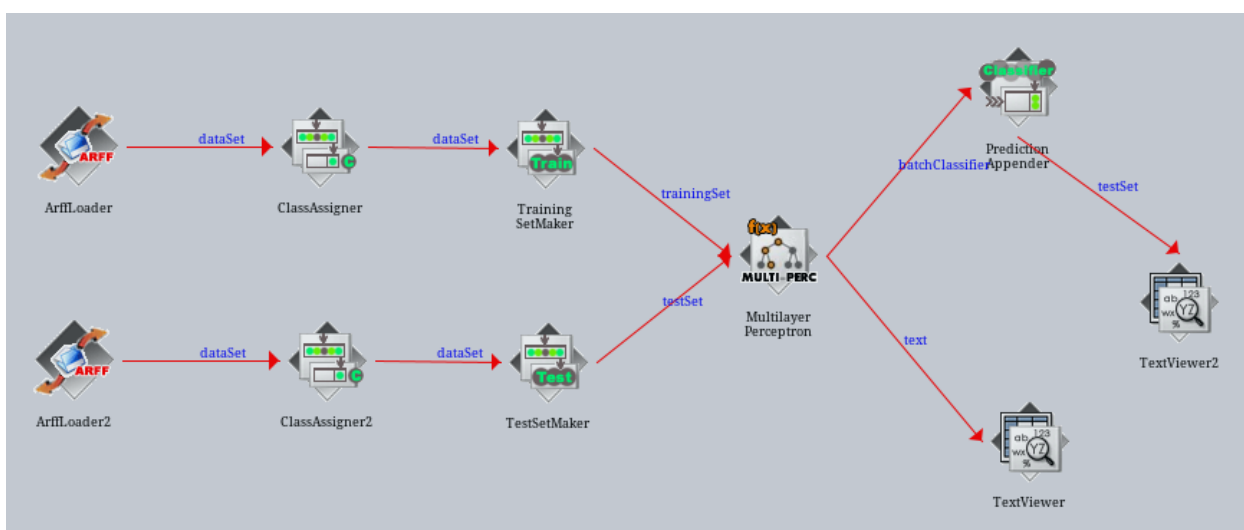
Naujų duomenų požymių porų vaizdai:



8 pav. Naujų duomenų požymių porų vaizdai (WEKA)

## 2.2.5. Klasifikavimas ir testavimas

Sukurta nauja komponentių seka WEKA programoje, kurioje vyksta ir DNT (su tais pačiais nustatytais parametrais) apmokymas, ir testavimas su naujais duomenimis:



9 pav. Klasifikavimas ir testavimas vienoje schemoje (WEKA)

Schemoje šalia **TestSetMaker** matome ir **TrainingSetMaker** komponentę, kuri savo ruožtu tiesiog pažymi jai paduotą duomenų aibę kaip mokymo aibę.

Šiame žingsnyje gautos neuronų svorių reikšmės pateikiamos lentelėse, kad būtų aišku, kurio sluoksnio kuris svorių rinkinys yra:

2 lentelė. Neuronų svorių reikšmės (į paslėptą sluoksnį)

	w_0	w_1	w_2	w_3	w_4
3	-1,444107	0,2364103705	2,088346904	-2,623024709	-2,771155606
4	-0,4539474624	0,5304868773	-0,654151762	0,8899942229	0,7938384577
5	-5,165369585	-1,256461703	-2,351790193	6,962986773	6,516774088
6	-6,256330346	-1,50121545	-2,837604676	8,057893309	7,801477824
7	2,705516769	0,8573184137	-2,231413815	3,106899437	3,133679697

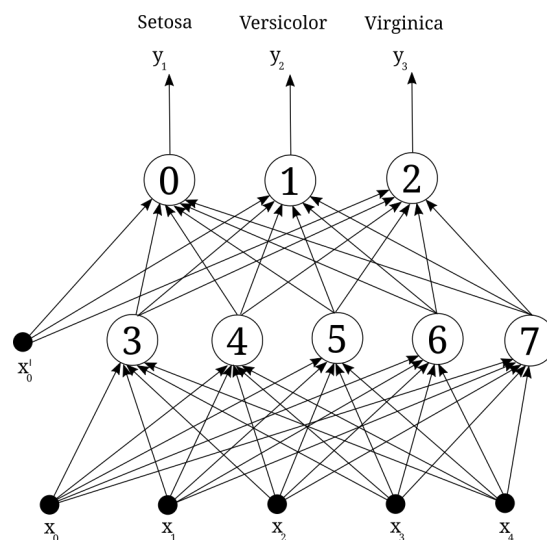
3 lentelė. Neuronų svorių reikšmės (iš paslėpto sluoksnio)

	w'_0	w_3	w_4	w_5	w_6	w_7
0	0,3409621996	4,860791324	-1,975106426	-1,558605013	-1,70816831	-4,736806698
1	-0,3434147466	-4,835390471	0,270534453	-5,750800556	-7,093165007	5,509634477
2	-4,956559933	-5,563880921	0,4153193082	5,495738789	6,5872398	-0,01303788128

#### Paaiškinimas:

2-oje lentelėje stulpeliuose yra svorių reikšmės iš  $i$ -ojo įėjimo,  $i \in \{0, 1, 2, 3, 4\}$ , o eilutėse – svorių reikšmės įėjimuose į paslėptus sluoksnius, jų numeriai:  $\{3, 4, 5, 6, 7\}$ .

3-oje lentelėje stulpeliuose yra svorių reikšmės iš paslėptų sluoksnių, o eilutėse – svorių reikšmės įėjimuose į išėjimo mazgus (išėjimų numeriai: 0, 1, 2).



10 pav. DNT schema

### 2.3. Pakartojamas duomenų klasifikavimas su MS Excel tipo programa

Dirbtinis neuroninis tinklas buvo konstruojamas naudojant daugiasluoksnį perceptroną su vienu paslėptu sluoksniu, kuriame yra 5 neuronai. Duomenų atributai (keturi) buvo pateikiami kaip įėjimai kartu su slenksčiu  $x_0$ , kuris yra lygus 1. Visi įėjimai sujungti su kiekvienu paslėptu neuronu, ir kiekvienai iš šių jungčių priskiriamas svoris. Svoriai sudauginami su įėjimų reikšmėmis, ir gaunami įėjimai į paslėptą sluoksnį. Gauti paslėptų neuronų išėjimai (paveikus įėjimus sigmoidine aktyvacijos funkcija) pasiunčiami į išėjimo sluoksnį – sudauginami su likusiais svoriais, taip pat pridodamas slenkstis  $x'_0$ , kurio reikšmė taip pat lygi 1. Išėjimo mazgai yra trys, nusakantys, kokia tikimybė, kad duotas duomuo priklauso vienai iš trijų klasių.

Su naujais duomenimis WEKA programoje gautas klasifikavimo rezultatas:

4 lentelė. Klasifikavimo rezultatai (WEKA)

Klasių priskyrimo tikimybės			Priskirta klasė:
Iris-setosa	Iris-versicolor	Iris-virginica	
0,992604	0,007364	0,000032	Iris-setosa
0,968445	0,031473	0,000082	Iris-setosa
0,992188	0,007778	0,000034	Iris-setosa
0,988668	0,011291	0,000041	Iris-setosa
0,991913	0,008053	0,000034	Iris-setosa
0,989134	0,010825	0,000041	Iris-setosa
0,993278	0,006691	0,00003	Iris-setosa
0,991871	0,008094	0,000035	Iris-setosa
0,993135	0,006834	0,000031	Iris-setosa
0,992158	0,007808	0,000034	Iris-setosa
0,006849	0,984404	0,008747	Iris-versicolor
0,006169	0,98376	0,01007	Iris-versicolor
0,008831	0,985466	0,005703	Iris-versicolor
0,017441	0,978669	0,003889	Iris-versicolor
0,007402	0,984817	0,007782	Iris-versicolor
0,013732	0,98213	0,004138	Iris-versicolor
0,009476	0,984656	0,005869	Iris-versicolor
0,008282	0,985813	0,005905	Iris-versicolor
0,027463	0,969689	0,002849	Iris-versicolor
0,008956	0,984972	0,006072	Iris-versicolor
0,000116	0,000614	0,99927	Iris-virginica
0,000149	0,001125	0,998726	Iris-virginica

0,000208	0,002069	0,997723	Iris-virginica
0,000117	0,00062	0,999263	Iris-virginica
0,000116	0,000592	0,999292	Iris-virginica
0,000134	0,000817	0,999049	Iris-virginica
0,0002	0,00279	0,997011	Iris-virginica
0,000223	0,003296	0,996481	Iris-virginica
0,000156	0,000778	0,999066	Iris-virginica
0,000476	0,026456	0,973069	Iris-virginica

Naujų duomenų klasifikavimas, remiantis jau aprašytu DNT veikimu, buvo pakartotas ir su MS Excel programa, skaičiavimams naudojant patogias Excel formules. Gautas rezultatas:

5 lentelė. Klasifikavimo rezultatai (MS Excel)

Klasių priskyrimo tikimybės			Priskirta klasė:
Iris-setosa	Iris-versicolor	Iris-virginica	
0,9933944285	0,006125189337	0,00002890173217	Iris-setosa
0,9653033378	0,0316343495	0,00008518559788	Iris-setosa
0,9931347152	0,006409348902	0,00003016603958	Iris-setosa
0,992071136	0,007172003911	0,00003161805411	Iris-setosa
0,993253568	0,006213472673	0,00002905178679	Iris-setosa
0,9912059832	0,007921132669	0,00003411663696	Iris-setosa
0,993734401	0,00588653434	0,00002813819841	Iris-setosa
0,9929169229	0,006546198172	0,00003046151159	Iris-setosa
0,9935995487	0,005947953602	0,00002829749875	Iris-setosa
0,9930238394	0,006390922433	0,00002968617251	Iris-setosa
0,004070247274	0,9804404269	0,02066445534	Iris-versicolor
0,004441935028	0,9811966579	0,01616202869	Iris-versicolor
0,005927727406	0,9909062521	0,007782532044	Iris-versicolor
0,009859121935	0,9901671983	0,005544195848	Iris-versicolor
0,004969200724	0,9860599682	0,01311962834	Iris-versicolor
0,01112113703	0,986813034	0,004578281177	Iris-versicolor
0,007193230153	0,9882555981	0,007423900679	Iris-versicolor
0,006197198548	0,9897118404	0,007209574042	Iris-versicolor
0,01742054619	0,98409925	0,003683392807	Iris-versicolor
0,006539418411	0,9890400946	0,007996079459	Iris-versicolor
0,0000971289503	0,0005866965378	0,9993917194	Iris-virginica
0,0001138780821	0,0007917963527	0,9991461248	Iris-virginica
0,000129828902	0,0007548318285	0,9991565831	Iris-virginica
0,00009741936136	0,00058686825	0,9993880688	Iris-virginica

0,00009928278059	0,0005790895104	0,9993911568	Iris-virginica
0,0001053412698	0,0006518255334	0,9993081917	Iris-virginica
0,0001147547967	0,0008258104013	0,9991350478	Iris-virginica
0,0001386904172	0,001127784524	0,9987470102	Iris-virginica
0,0001306539113	0,0006366626283	0,9992416686	Iris-virginica
0,0002181407692	0,002753823122	0,9967370434	Iris-virginica

Absoliutaus skirtumo tarp WEKA ir MS Excel gautų tikimybių lentelė:

6 lentelė. Absoliutus skirtumas tarp WEKA ir MS Excel gautų rezultatų

Iris-setosa	Iris-versicolor	Iris-virginica
0,0007904285346	0,001238810663	0,000003098267829
0,003141662248	0,0001613494957	0,000003185597878
0,0009467151939	0,001368651098	0,00000383396042
0,003403135977	0,004118996089	0,000009381945895
0,001340567972	0,001839527327	0,000004948213214
0,002071983234	0,002903867331	0,000006883363038
0,0004564009837	0,0008044656596	0,000001861801587
0,001045922926	0,001547801828	0,00000453848841
0,0004645487194	0,0008860463981	0,000002702501249
0,00086583939	0,001417077567	0,000004313827494
0,002778752726	0,003963573121	0,01191745534
0,001727064972	0,002563342084	0,00609202869
0,002903272594	0,005440252149	0,002079532044
0,007581878065	0,0114981983	0,001655195848
0,002432799276	0,001242968168	0,005337628343
0,002610862968	0,004683034019	0,0004402811774
0,002282769847	0,003599598056	0,001554900679
0,002084801452	0,00389884037	0,001304574042
0,01004245381	0,01441024999	0,0008343928068
0,002416581589	0,0040680946	0,001924079459
0,0000188710497	0,00002730346225	0,0001217194154
0,00003512191793	0,0003332036473	0,0004201248223
0,00007817109796	0,001314168171	0,001433583121
0,00001958063864	0,00003313175005	0,000125068809
0,00001671721941	0,00001291048957	0,00009915684362
0,0000286587302	0,0001651744666	0,0002591916824
0,00008524520326	0,001964189599	0,002124047822
0,00008430958279	0,002168215476	0,002266010204

0,00002534608872	0,0001413373717	0,0001756685848
0,0002578592308	0,02370217688	0,02366804337

Matome, kad skirtumai nežymūs, prasidedantys nuo 3 ir tolimesnių skaičių po kablelio. Šis nedidelis nesutapimas susidarė dėl realių skaičių su skaitmenimis po kablelio saugojimo kompiuterio atmintyje specifikos, ir kadangi buvo atliekama grupė veiksmų su pradiniais skaičiais, tai ši paklaida skaičiavimų eigoje vis didėjo.

### **3. Išvados**

WEKA programoje buvo sukurtas dirbtinis neuroninis tinklas ir išmokytas teisingai klasifikuoti irisų duomenis, derinant ir parenkant geriausius mokymo parametrus. Tuo pačiu buvo susipažinta su šios programos veikimu, bei dar labiau įsisavintas DNT veikimas naujus duomenis klasifikavus ne tik su WEKA, bet ir su MS Excel programa, kurioje detalai matomas kiekvienas skaičiavimas.

Matoma, kad DNT kūrimas ir naudojimas laisvai galimas ir naudojant nemažą sluoksnį abstrakcijos, kaip kad WEKA programoje, tačiau svarbu ir turėti gilesnį suvokimą apie kiekvienos komponentės funkcijas, kiekvieno žingsnio atlikimą, ir, reikalui esant, sumažinti abstrakciją. Tokiu būdu su DNT galima dirbti greitai ir efektyviai.

## **A Priedai**

### **A.1 MS Excel programa *DNT\_skaiciavimai***