VILNIAUS PEDAGOGINIS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA

Jurgita Balevičiūtė

Daugiamačių duomenų klasifikavimo rezultatų vizuali analizė

Magistro darbas

Darbo vadovas

dr. Olga Kurasova

Vilnius, 2008

TURINYS

ĮVADAS

Praktinio pobūdžio uždaviniuose analizuojami duomenys dažnai būna daugiamačiai, t.y. analizuojamus objektus apibūdina daugiau nei du parametrai. Daugiamačių duomenų analizė aktuali įvairiose srityse: moksle, versle, medicinoje, sociologijoje ir kt.

Apibrėžkime pagrindines sąvokas, naudojamas šiame darbe. Kalbant apie daugiamačius duomenis, svarbios yra dvi sąvokos, tai objektas ir parametras. Sąvoka objektas gali apimti įvairius dalykus: žmones, įrenginius, gamybos produktus, augalus ir kt. Objektai sudarantys konkrečią analizuojamų objektų aibę yra apibūdinami bendrais parametrais, dar dažnai vadinamais požymiais, savybėmis, ypatybėmis. Visų parametrų reikšmių junginys nusako vieną konkretų analizuojamos aibės objektą $X_i = (x_{i1}, x_{i2}, ..., x_{im})$, čia *m* – parametrų skaičius, dar vadinamas duomenų matavimų skaičiais. Kai objektą X_i apibūdina daugiau nei vienas parametras, duomenys (objektai) yra daugiamačiai. Dažnai daugiamačiai duomenys dar vadinami daugiamačiais taškais.

Taigi galima suformuoti analizuojamų duomenų aibę *D*, sudarytą iš vektorių $X_i \in R^m$, t.y.

$$D = \{X_1, X_2, ..., X_n\} = \{X_{ij}, i = 1,...,n,\ j = 1,...,m\}.$$

Matricos *D* i-oji eilutė yra vektorius $X_i \in R^n$, čia $X_i = (x_{i1}, x_{i2}, ..., x_{im})$, $i = 1,...,n$, n – analizuojamų objektų (vektorių) skaičius. Duomenys X_i dar vadinami įvesties duomenimis, klasifikavimo algoritmuose ir mokymo duomenimis.

Kuo didesnis skaičius matmenų duomenyse, tuo sunkiau iš lentelės išgauti informacijos apie atskirų objektų ryšius. Tokių duomenų analizė reikalauja intensyvių turimų. Tai gana sudėtingi uždaviniai. Spręsdamas juos žmogus gali įsiginti į duomenis ir daryti išvadas. Tam tikslui yra naudojami įvairūs duomenų analizės metodai: klasifikavimo, klasterizavimo, vizualizavimo ir kt. Labai svarbu duomenis pateikti žmogui suprantama forma, padedančia geriau juos suvokti: nustatyti struktūrą, tarpusavio ryšius, susidariusias grupes, prognozuojamus įverčius ir pan.

Siekiant gauti daugiau žinių iš analizuojamų duomenų, yra naudojami gavybos ir analizės metodai: klasifikavimo, klasterizavimo, vizualizavimo ir kt. Klasifikavimo uždavinių tikslas – turint aibę duomenų, kurių klasės įprastai yra žinomos, sukurti taisykles, pagal kurias duomenys, kurie nebuvo naudojami tų taisyklių kūrime, automatiškai bus priskirti vienai ar kitai žinomai klasei. Daugiamačių duomenų vizualizavimo, dar kitaip vadinamais dimensijos mažinimo metodais, didelės dimensijos matmenys transformuojami į mažesnės dimensijos erdvę taip, kad išliktų arba būtų atrastos „užslėptos“ analizuojamų duomenų savybės [6].

Darbo tikslas ir uždaviniai

Tikslas – ištirti daugiamačių duomenų klasifikavimo pateikimo vaizdinėmis priemonėmis galimybes.

Uždaviniai:

1) Išanalizuoti kelis automatinio klasifikavimo metodus.
2) Išnagrinėti kelias programines sistemas, kurios realizuoja klasifikavimo metodus vizualiomis iliustracijomis (pvz. klasifikavimo medis, taisyklės).
3) Palyginti sistemų pateiktus klasifikavimo rezultatus analizuojant kelias duomenų aibes.
4) Klasifikavimo rezultatus integruoti į daugiamačių duomenų projekcijų vaizdus, gautus daugiamačių skalių metodu.

1. DAUGIAMAČIŲ DUOMENŲ KLASIFIKAVIMAS

Klasifikavimas yra vienas iš žinomiausių ir populiariausių duomenų analizės metodų. Klasifikavimas naudojamas įvairiose srityse, pavyzdžiui medicinoje, paskolų davimo klausimais, aptinkant klaidas pramonėje ir klasifikuojant finansines rinkos kryptis. Elementarus klasifikavimo uždavinys – žmogaus amžiaus priskyrimas prie vienos iš klasių (jaunas, pagyvenęs ar senas). Tai dažniausiai vaizduojama kaip nuspėjimas kitos reikšmės kai klasifikavimo prognozės rezultatas yra diskretus dydis. Prieš naudojant kurį nors duomenų mokymo metodų, klasifikavimas būna atliktas paprasčiausiai panaudojant žinomus duomenis. Tai iliustruota pavyzdyje 1.1. [5]

Pavyzdys 1.1

Mokytoja pagal balus suskirsto mokinius į grupes A, B, C, D arba F, naudojant paprastas ribas (60, 70, 80, 90) suklasifikuoti mokinių gautus balus galima taip:

Sąlyga					Grupė
90	<=	balas			A
80	<=	balas	<	90	B
70	<=	balas	<	80	C
60	<=	balas	<	70	D
		balas	<	60	F

Bendras klasifikavimo uždavinio apibrėžimas:

Apibrėžimas 1.1. Duota aibė $D = \{X_1, X_2, ..., X_n\}$ duomenų (elementų, įrašų) ir aibė

klasių $C = \{C_1,...,C_m\}$. **Klasifikavimo uždavinys** yra apibrėžti atvaizdavimą (angl. mapping) $f: D \to C$, kur kiekvienas t_i yra susiejamas su viena klase. **Klasei** C_j yra priskirta dalis analizuojamų duomenų aibės *D* duomenų, t.y. $C_j = \{X_i \mid f(X_i) = C_j, 1 \le i \le n \; ir \; X_i \in D\}$.

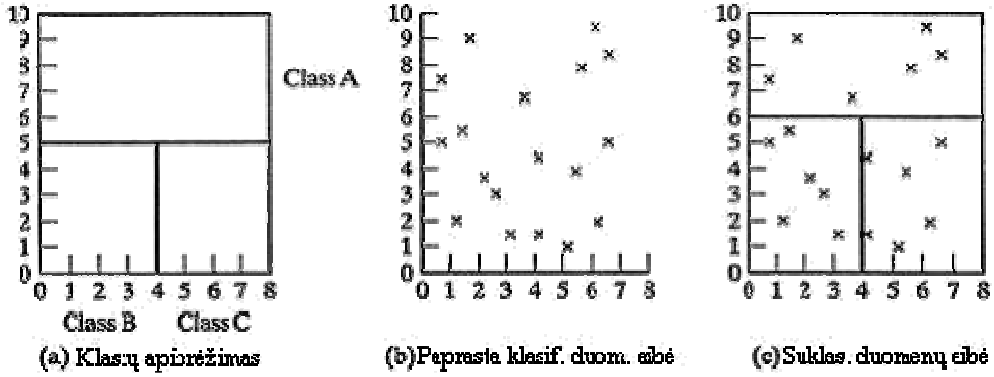
Mūsų apibrėžimas parodo klasifikaciją kaip atvaizdavimą iš duomenų aibės į klasių aibę. Klasės yra iš anksto apibrėžtos, nesutampančios, visos aibės dalis. Kiekvienas duomuo esantis duomenų aibėje yra susietas tik su viena klase. Klasifikavimo uždaviniai dažniausiai įgyvendina dvi tokias fazes:

1. Specifinio modelio mokymo duomenims įvertinti sukūrimas. Šis žingsnis turi įvesties mokymo duomenis ir modelio apibrėžimo vystymą kaip išvesties duomenis. Sukurtas modelis klasifikuoja mokymo duomenis taip tiksliai kaip įmanoma.

2. Vykdyti modelio vystymą pirmame žingsnyje klasifikuojant duomenis iš planuojamos aibės duomenų.

Iš esmės yra trys pagrindiniai metodai klasifikavimo uždaviniams spręsti:

- **Tiksliai apibrėžiant ribas**. Klasifikacija yra atliekama dalinant įvesties aibę duomens į sritis, kur kiekviena sritis yra susieta su viena klase.
- **Naudojant išsibarstymo (angl. distribution) tikimybę**. Jei atsitiktinė tikimybė kiekvienai klasei $P(C_j)$ yra žinoma, kai $P(C_j)P(X_i \mid C_j)$ yra naudojama apskaičiuoti tikimybę to X_i klasėje C_j.
- **Naudojant vėlesnes (angl. posterior) tikimybes**. Duotas duomenų vektorius X_i, mes turėtume nustatyti tikimybę to X_i esančio klasėje C_j. Tai išreiškiama per $P(C_j \mid X_i)$ ir yra vadinama *vėlesne tikimybė* (angl. posterior probability). Klasifikacija priartėja nustatant tolimesnę tikimybę kiekvienai klasei ir kai priskiriame X_i klasei su aukščiausia tikimybe.



1. 1 pav. Klasifikavimo iliustracija

Tariame, kad turime duomenų aibę sudarytą iš tokio pavidalo $t = \langle x, y \rangle$ duomenų, kur $0 \le x \le 8$ ir $0 \le y \le 10$. 1.1 pav. iliustruoja klasifikavimo metodų tipus. 1.1 pav.(a) parodo prieš

apibrėžiant klases, visa suskirstoma į nurodytus plotus, 1.1 pav.(b) pateikti įvesties duomenys, ir 1.1 pav.(c) parodo duomenų klasifikavimą atitinkamai klasei.

Klasifikavimo strategija gali tiksliai tikti mokymų duomenims, tačiau gali būti nepritaikoma kitiems apmokyme naudojamiems duomenims tirti. Klasifikavimo metu turi būti sukurtas toks klasifikatorius, kuris gerai klasifikuotų ne tik duomenis, kurie naudojami klasifikatoriaus apmokymui, bet ir naujiems duomenims nenaudotiems klasifikatoriaus sukūrime.

Vardas (angl. name)	Lytis (angl. gender)	Ūgis (angl. height)	Išvestis1	Išvestis2
Kristina	M	1.6 m	Žema	Vidutinė
Jim	V	2 m	Aukštas	Vidutinis
Maggie	M	1.9 m	Vidutinė	Aukšta
Martha	M	1.88 m	Vidutinė	Aukšta
Stephanie	M	1.7 m	Žema	Vidutinė
Bob	V	1.85 m	Vidutinis	Vidutinis
Kathy	M	1.6 m	Žema	Vidutinė
Dave	V	1.7 m	Žemas	Vidutinis
Worth	V	2.2 m	Aukštas	Aukštas
Steven	V	2.1 m	Aukštas	Aukštas
Debbie	M	1.8 m	Vidutinė	Vidutinė
Todd	V	1.95 m	Vidutinis	Vidutinis
Kim	M	1.9 m	Vidutinė	Aukšta
Amy	M	1.8 m	Vidutinė	Vidutinė
Wynette	M	1.75 m	Vidutinė	Vidutinė

Lentelė 1. Klasifikavimo duomenys naudojant ūgio duomenis

Lentele 1 galime iliustruoti daugelį klasifikavimo metodų. Mūsų iškeltas uždavinys yra suklasifikuoti žmones pagal ūgį į žemus, vidutinius ir aukštus. Lentelė 1 ūgis yra nurodytas metrais. Paskutiniai šios lentelės du stulpeliai parodo kaip gali būti atliktas klasifikavimas, antraštė Išvestis1 ir Išvestis2. Išvestis1 klasifikacija naudoja paprastą išdalinimą:

2 m <= Ūgis	Aukštas
1.7m < Ūgis < 2 m	Vidutinis
Ūgis <= 1.7 m	Žemas

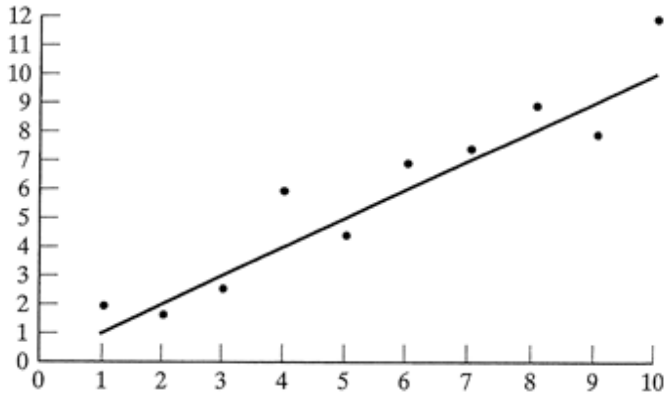
Išvestis2 rezultatai reikalauja daug daugiau išdalijimo naudojant abiejų parametrų ūgio ir lyties specifiką, t.y. Išvestis1 klasifikuojama pagal visus neatsižvelgiant į lytį, o Išvestis2 atsižvelgiama ir į lytį. Moteris gali būti vidutinio ūgio moterų grupėje, bet žema visų žmonių grupėje. [5]

1.1 Klasifikavimo metodai

1.1.1 Statistika grįsti algoritmai

Regresija (angl. regresion).

Tegul turime duomenis $X_1, X_2, ..., X_m$, kuriuos vadinsime įvesties duomenimis, *regresijos* rezultate gauname išvesties duomenis $Y_1, Y_2, .., Y_m$, kur $Y_i = f(X_1, X_2, ..., X_m)$. Įvesties duomenys naudojami iš bazės *D*, o išvesties duomenys atspindi klasę. Regresija gali būti naudojama išspręsti klasifikavimo uždavinius, taip pat gali būti pritaikoma tokiai sferai kaip prognozavimas (angl. forecasting) arba naudojama daugelyje įvairių metodų, įskaitant ir artimiausių kaimynų (angl. NN). Realybėje, regresija paima duomenų aibę ir įstato tuos duomenis į formulę.



1. 2 pav. Paprasta linijinė regresija

Žiūrint į 1.2 pav. matome, kad paprastas linijinės regresijos uždavinys gali būti išspręstas kai iš formulės gaunama tiesė. Tai gali būti prilyginta padalijus duomenis į dvi klases. Jei imtume banko pavyzdį, tai turėtų būti kaip patvirtinimas arba atmetimas duodant paskolą. Tiesė būtų dalyba tarp dviejų klasių.

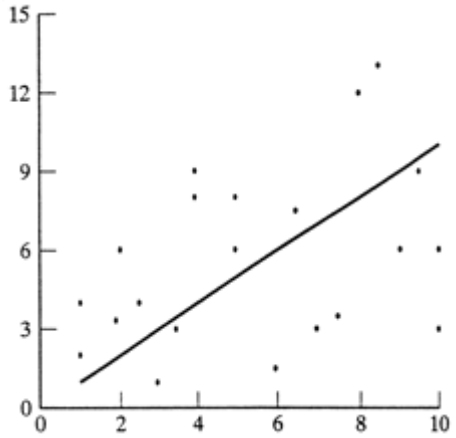
Linijinė regresija aprašoma formule:

$$y = c_0 + c_1 x_1 + ... + c_n x_n \qquad (1.1)$$

kur $x_1, ..., x_n$ yra duomenis apibūdinantys parametrai (kintamieji), $c_0, c_1, ..., c_n$ - *regresijos koeficientai*, o *y* išvesties duomuo. Įvesties parametrai $x_1, ..., x_n$ gali būti apskaičiuojami žinant *regresijos koeficientus* $c_0, c_1, ..., c_n$. Turint tik vieną parametrą x_i, tiesę $y = c_0 + c_1 x_i$ galima nubrėžti per du duotus taškus *xy* plokštumoje. Tiesę nusako regresijos koeficientai *m* ir *b*.

1.3 pav. iliustruoja bendrą linijinės regresijos naudojimą su viena įvesties reikšme. Tai yra pavyzdys duomenų, kuriuos mes norime modeliuoti (parodyta kaip išsibarstę taškai) naudojant tiesės modelį. Suprantame, kad tikri duomenų taškai netinka tiesės modeliui, kadangi jie yra gana toli. Taigi, šitas modelis yra skirtas įvertinti įvesties ir išvesties ryšius. Mes galime nutolę nuo tiesės sukurti tiesės modelį, numatyti išvesties duomenį duotą įvesties duomeniui,

bet duomenų apskaičiavimui gerai įvertinti turėti tą tikrą išvesties reikšmę. Jei mes bandome pritaikyti duomenis, kurie nėra tiesiniai, rezultatas bus prastas duomenų pavyzdys, kaip pavaizduota 1.3 pav.



1. 3 pav. Pavyzdys kaip netinkamai pritaikoma tiesinė regresija

Yra daugybė priežasčių, kodėl tiesinės regresijos modelis netinka įvertinti išvesties duomenų. Pirmiausia būna taip, kad duomenys netinka tiesiniam modeliui. Taip būna, kai duomenys faktiškai atstovauja linijiniam modeliui, tačiau linijinio modelio sugeneravimas yra menkas, nes triukšmai ir taškai atsiskyrėliai egzistuoja duomenyse. *Triukšmas* (angl. noise) yra klaidingi duomenys. *Taškai atsiskyrėliai* yra duomenų reikšmės, kurios yra išimtys įprastiniuose ir tikėtinuose duomenyse. Pavyzdys 1.2 iliustruoja taškus atsiskyrėlius. Šiuo atveju, matomos reikšmės gali būti aprašytos:

$$y = c_0 + c_1 x_1 + ... + c_n x_n + \varepsilon \qquad (1.2)$$

kur ε atsitiktinė paklaida artima 0. Mes galime apskaičiuoti tikslumą tiesinės regresijos tinkamumui esantiems duomenims naudojant vidutinės kvadratinės paklaidos funkciją.

Pavyzdys 1.2

Sakykime, kad turime 100 studentų, kurie yra išklausę abstrakčiosios algebros kursą. Kristina nuolatos pralenkia kitus studentus per įskaitas. Per egzaminą Kristina gavo 99 balus. Kitas aukščiausias įvertinimas buvo 75, o balų intervalas nuo 5 iki 99. Kiti tos grupės studentai neabejotinai pasipiktino Kristina, nes ji nepasirodė tam pačiam lygyje kaip ir jie. Ji peržengė juos. Jei mes bandytume pritaikyti modelį balams, šis vienas taškas atsiskyrėlis balas sukeltų problemų, nes kiekvienas modelis, kuris bandytų įtraukti tai, nebūtų tikslus likusiems duomenims.

Mes iliustravome kaip naudojama paprasta tiesinės regresijos formulė ir priskinti *k* taškai mūsų mokymo pavyzdyje. Tokiu būdu mes turime tokias formules su *k*:

$$y_i = c_0 + c_1 x_{1i} + \varepsilon_i,\ i = 1...k \qquad (1.3)$$

Su paprasta tiesine regresija, su duotomis reikšmėmis (x_{1i}, y_i), ε_i yra klaida ir tokiu

būdu kvadratinė palaidos metodas gali būti rodyti klaidą. Minimizuoti šią paklaidą yra naudojamas mažiausių kvadratų metodas, minimizuojantis mažiausią kvadratinę paklaidą. Šis pastebėjimas randa koeficientus c_0, c_1 tai kvadratinė paklaida yra sumažinta iki minimumo analizuojamų (angl. observable) duomenų aibei. Kvadratinių paklaidų suma (angl. squared errors) yra:

$$L = \sum_{i=1}^{k} \varepsilon_i^2 = \sum_{i=1}^{k} \left(y_i - c_0 - c_1 x_{1i} \right)^2 \qquad (1.4)$$

Imant dalinį išvedimą (su išreikštais koeficientais) ir prilyginus nuliui, mes galime gauti *mažiausius kvadratinių apskaičiavimus* koeficientams $\hat{c}_0$ ir $\hat{c}_1$.

Naudojant du skirtingus metodus vykdant klasifikavimą gali būti panaudota regresija:

- **Dalijimas (angl. division):** duomenys yra dalijami į klasių sritis.
- **Prognozavimas (angl. prediction):** formulės yra skirtos numatyti išvesties klasės reikšmę.

Pirmas atvejis rodo duomenis padalintus į n-mates erdves be bet kokio aiškaus klasės reikšmių pavaizdavimo. Regresijos būdu erdvė suskaidoma į sritis kiekvienai klasei. Naudojant regresiją yra sukuriama tiesės formulė numatyti klasės reikšmes.

Toliau pateiktas pavyzdys 1.3 iliustruoja dalijimo procesą, o Pavyzdys 1.4 iliustruoja prognozavimo procesą naudojant duomenis iš Lentelės 1. Kad būtų paprasčiau, mes imsime mokymo duomenis, kur įtraukti tik žemi ir vidutiniai žmonės ir klasifikavimas yra atliekamas naudojant Išvestis1 stulpelio reikšmes.

Pavyzdys 1.3

Žiūrint į duomenis Lentelėje 1 stulpelyje Išvestis1 suprantame, kad klasę lemia tik skaitinė moters arba vyro ūgio reikšmė, kuriai priskirtas žmogus. Šiame pavyzdyje mes pritaikysime tiesinės regresijos modelį nustatyti skirtumą tarp klasių vidutinis ir žemas. 1.4 pav.(a) rodo nagrinėjamus taškus. Mes turime tiesinės regresijos formulę $y = c_0 + \varepsilon$. Vadinasi ieškosime tos geriausiųjų ūgių skaitinės reikmės c_0 dalijimo į tuos, kurie yra žemi ir tuos kurie yra vidutiniai. Žiūrint į duomenis Lentelėje 1 matome, kad tik 12 iš 15 elementų gali būti panaudoti diferencijavimui tarp žemų ir vidutinių žmonių. Tokiu būdu gauname reikšmes y_i mūsų mokymo duomenyse: {1,6, 1,9, 1,88, 1,7, 1,85, 1,6, 1,7, 1,8, 1,95, 1,9, 1,8, 1,75}. Mes norime minimizuoti:

$$L = \sum_{i=1}^{12} \varepsilon_i^2 = \sum_{i=1}^{12} \left(y_i - c_0 \right)^2$$

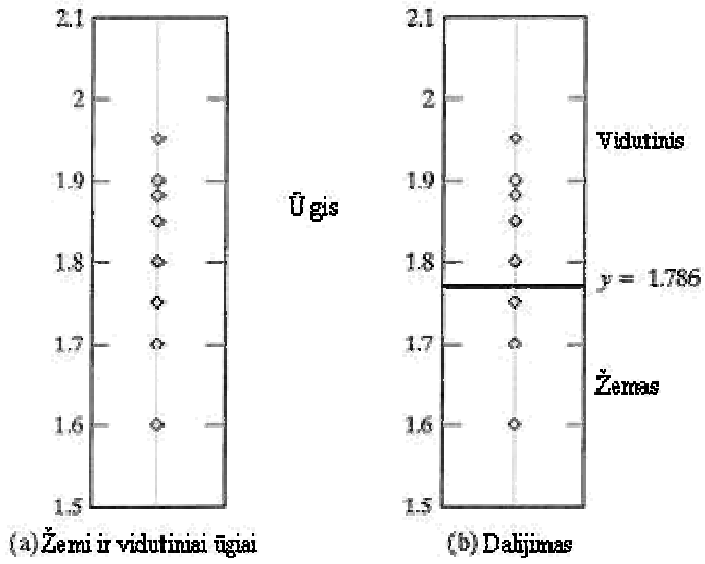
Paskaičiavę išvestinę pagal c_0 ir prisilyginę nuliui gauname,

$$-2 \sum_{i=1}^{12} y_i + \sum_{i=1}^{12} 2c_0 = 0$$

Išreiškus c_0 gauname, kad

$$c_0 = \frac{\sum_{i=1}^{12} y_i}{12} = 1.786$$

Taip mes turime dalijimą tarp žemų ir vidutinių žmonių kaip buvo apibrėžta *y=1.786*, kaip matome 1.4 pav.(b).



1. 4 pav. Klasifikavimas naudojant išdalijimą Pavyzdžiui 1.3

Pavyzdys 1.4

Dabar žiūrėsime į klasių spėjimą naudojant žemus ir vidutinius žmonių įvesties duomenis ir klasifikavimą Išvestis1. Duomenys yra tokie patys kaip Pavyzdyje 1.3 išskyrus tai, kad dabar žiūrime į klases nurodytas tik mokymo duomenyse. Nuo tada kai regresija įgyja skaitines reikšmes, mes imame, kad žemų klasės reikšmė yra 0, o vidutinių klasės reikšmė yra 1. 1.5 pav.(a) pavaizduoti duomenys šiam pavyzdžiui: {(1,6, 0), (1,9, 1), (1,88, 1), (1,7, 0), (1,85, 1), (1,6, 0), (1,7, 0), (1,8, 1), (1,95, 1), (1,9, 1), (1,8, 1), (1,75, 1)}. Šiuo atveju mes naudojame regresijos formulę su vienu kintamuoju:

$$y = c_0 + c_1 x_1 + \varepsilon$$

Taigi norima minimizuoti

$$L = \sum_{i=1}^{12} \varepsilon_i^2 = \sum_{i=1}^{12} (y_i - c_0 - c_1 x_{1i})^2$$

Paskaičiuojame išvestinę pagal c_0 ir prilyginus nuliui gauname

$$\frac{\partial L}{\partial c_0} = -2\sum_{i=1}^{12} y_i + \sum_{i=1}^{12} 2c_0 + \sum_{i=1}^{12} 2c_1 x_{1i} = 0$$

Toliau surandame c_0:

$$c_0 = \frac{\sum y_i - \sum c_1 x_{1i}}{12}$$

Įstatome c_0 reikšmę į reiškinį L ir apskaičiuojame išvestinę pagal c_1. Viską prilyginame nuliui

$$\frac{\partial L}{\partial c_1} = 2\sum \left(y_i - c_0 - c_1 x_{1i}\right)\left(-x_{1i}\right) = 0$$

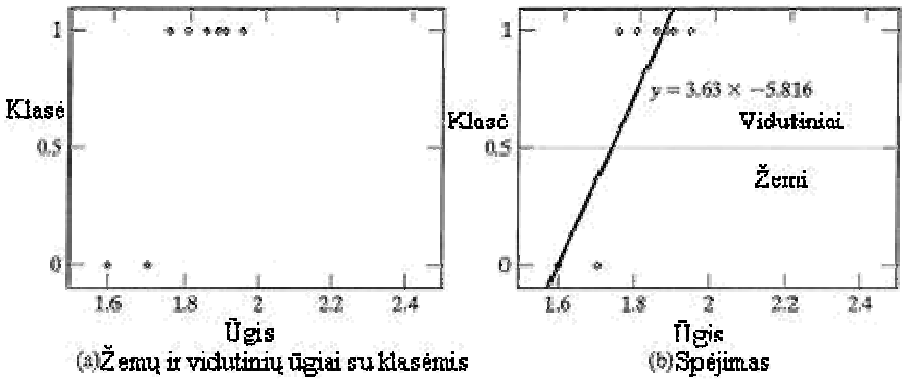
Išreiškę c_1 galiausiai gauname

$$c_1 = \frac{\sum \left(x_{1i} y_i\right) - \dfrac{\sum x_{1i} \sum y_i}{12}}{\sum \left(x_{1i}^2\right) - \dfrac{\left(\sum x_{1i}\right)^2}{12}}$$

Dabar galime apskaičiuoti c_0 ir c_1. Naudojant mokymo duomenis iš 12 taškų, mes turime $\sum x_{1i} = 21{,}43$, $\sum y_i = 8$, $\sum \left(x_{1i} y_i\right) = 14{,}83$ ir $\sum \left(x_{1i}^2\right) = 38{,}42$. Taigi, gauname $c_1 = 3{,}63$ ir $c_0 = -5{,}816$. Taigi klasės spėjimo reikšmė yra

$$y = -5{,}816 + 3{,}63 x_1$$

Ši riba yra pavaizduota 1.5 pav.(b).



1. 5 pav. Klasifikacija naudojant prognozavimo Pavyzdžiui 1.4

Pavyzdyje 1.4 tiesė, numato klasės reikšmes. Tai buvo padaryta dviem klasėm, bet tai taip pat galėtų būti padaryta ir visoms trims klasėms. Skirtingai nuo išdalijimo metodo, kur klasių sąjunga yra akivaizdžiai skirstoma į sritis, per atsitiktinį tašką su klasės tikimybe kam taškas priklauso yra mažiau akivaizdu. Čia mes numatome klasės reikšmę. 1.5 pav.(b) klasės reikšmė yra prognozuota remiantis tik ūgio reikšmėmis. Kadangi prognozavimo tiesė yra tolydi, priklausymas klasei nėra visada akivaizdus. Pavyzdžiui, jei numatytoji reikšmė yra 0,4, kokia tai klasė galėtų būti? Mes galime apibrėžti klasę, perskirtą tiesės. Taigi, žmogus pagal ūgį yra žemų klasėje, jei numatytoji reikšmė yra mažiau nei 0,5, ir vidutinių jei reikšmė yra didesnė už 0,5. Pavyzdyje 1.4 x_1 reikšmė, kur *y = 0,5* yra *1,74*. Taigi tai yra iš tikrųjų išdalijimas tarp žemų ir

vidutinio ūgio žmonių.

Jei analizuojamus duomenis apibūdinantys parametrai yra tiesinės regresijos funkcijoje yra paveikiami kažkokios funkcijos (kvadratas, šaknies traukimas, ir kt.), tuomet regresijos lygtis atrodo taip:

$$y = c_0 + f_1(x_1) + ... + f_n(x_n) \qquad (1.5)$$

kur f_i yra funkcija naudojama transformuoti parametrą. Šiuo atveju regresija vadinama *ne tiesine regresija*. Tiesinės regresijos modelis yra lengvai suprantamas, bet nėra pritaikoma daugeliui sudėtingų duomenų mokymo uždavinių. Jie neveiksmingi su neskaitiniais duomenimis. Jie taip pat daro prielaidą, kad ryšys tarp įvesties ir išvesties duomenų yra linijinis, ko faktiškai gali ir nebūti.

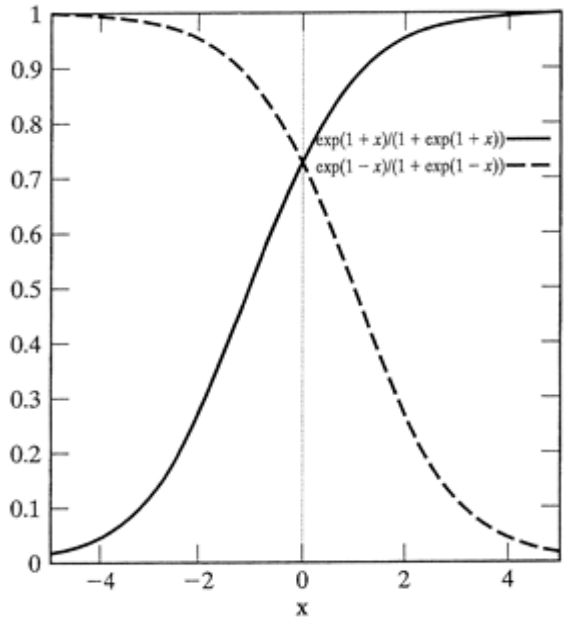
Tiesinė regresija ne visada yra tinkama, ne tik kad duomenys gali nesudaryti tiesios linijos, bet ir todėl, kad tiesės reikšmės gali būti didesnės nei 1 ir mažesnės nei 0. Tokiu būdu, jie tikrai negali būti panaudoti kaip atsitiktiniai numatytieji tos klasės. Kitas paprastas naudojimui regresijos modelis yra *logaritminė regresija*. Vietoj to, kad duomenis įstatinėtume į tiesę, logaritminė regresija naudoja logaritminę kreivę taip kaip parodyta 1.6 pav. Nubraižyti logaritminę kreivę yra naudojama tokia formulė

$$p = \frac{e^{(c_0 + c_1 x_1)}}{1 + e^{(c_0 + c_1 x_1)}} \qquad (1.6)$$

Logaritminės kreivės reikšmės yra tarp 0 ir 1, taigi tai gali būti interpretuota kaip tikėtinas klasių ryšys. Taip kaip su linijine regresija, tai gali būti naudojama ir kai norima klasifikuoti į dvi klases. Atlikti regresiją, logaritminė funkcija gali būti panaudoja tokiai funkcijai

$$\log_e\left(\frac{p}{1-p}\right) = c_0 + c_1 x_1 \qquad (1.7)$$

Čia *p* yra tikimybė būti klasėje, o *1 – p* yra tikimybė nebūti. Būdui pasirinktos reikšmės c_0 ir c_1 suteikia maksimalią tikimybę laikantis duotų reikšmių. [5]



1. 6 pav. Logaritminė kreivė

Bajeso (angl. Bayesian) klasifikatorius

Paprasta klasifikavimo schema vadinama Naive Bayes klasifikatoriumi remiasi Bayes taisyklėmis. Sakykime, kad visi parametrai yra nepriklausomi ir kad kiekvienas iš parametrų vienodai įtakoja klasifikavimo rezultatą.

Sakysime turime duomenis, susietus su klasėmis C_j, $j = 1,...,v$ (kur v yra klasių skaičius). Mūsų tikslas yra suklasifikuoti naujai įvestus duomenis, tai yra nuspręsti, kuriai apibrėžtai klasei jie priklauso, šiuo metu remiantis išvesties objektais. Spėjame, kai duomuo t_i turi *k* nepriklausomų parametrų reikšmių $\{x_{i1}, x_{i2},..., x_{ik}\}$. Mes turime suskaičiuoti vėlesnę (angl. posterior) tikimybę $P(C_j \mid X_i) = \frac{P(X_i \mid C_j)P(C_j)}{P(X_i)}$, tai yra tikimybė, kad X_i priklauso klasei C_j. Čia $P(C_j)$ yra priorinė klasės C_j tikimybė; $P(X_i \mid C_j)$ yra sąlyginė tikimybė, kuri gali būti užrašyta taip:

$$P(X_i | C_j) = \prod_{k=1}^{n} P(x_{ik} | C_j)$$

$P(X_i)$ yra tikimybė, kad X_i yra kiekvienoje klasėje, tai gali būti padaryta ieškant tikimybės, kad kiekvienas duomuo yra kiekvienoje klasėje ir įtraukiamos visos jų reikšmės. Vėlesnė tikimybė $P(C_j \mid X_i)$ yra apskaičiuojama kiekvienai klasei. Mes naują duomenį priskirsime klasei C_j, kuris įgyja aukščiausią vėlesnę tikimybę. [12]

Pavyzdys 1.5 iliustruoja Naive Bayes klasifikatoriaus panaudojimą.

Pavyzdys 1.5

Naudojant Išvestis1 klasifikavimo rezultatus iš Lentelės 1 yra keturi duomenys suklasifikuoti kaip žemas, aštuoni – vidutiniai ir trys – aukšti. Kad palengvinti klasifikavimą ūgio parametro reikšmes dalinsime į šešis intervalus:

(0, 1,6], (1,6, 1,7], (1,7, 1,8], (1,8, 1,9], (1,9, 2,0], (2,0, ∞)

Lentelė 2 parodo skaičiavimo rezultatus ir to po tikimybes susietas su parametrais. Su šitais mokymo duomenimis mes apskaičiuojame priorines išvestines:

P(žemas)=4/15=0,267, *P(vidutinis)=8/15=0,533*, *P(aukštas)=3/15=0,2*.

Parametras	Reikšmė	Skaičius			Tikimybės		
		Žemas	Vidutinis	Aukštas	Žemas	Vidutinis	Aukštas
Lytis	V	1	2	3	1/4	2/8	3/3
	M	3	6	0	3/4	6/8	0/3
Ūgis	(0, 1,6]	2	0	0	2/4	0	0
	(1,6, 1,7]	2	0	0	2/4	0	0
	(1,7, 1,8]	0	3	0	0	3/8	0
	(1,8, 1,9]	0	4	0	0	4/8	0
	(1,9, 2]	0	1	1	0	1/8	1/3
	(2, ∞)	0	0	2	0	0	2/3

Lentelė 2. Tikimybės susietos su parametrais

Mes naudojame tas reikšmes suklasifikuoti naujus duomenis. Pavyzdžiui, sakykime norime suklasifikuoti *t* = (Adam, V, 1,95m). Naudojant šias reikšmes, o kartu lyties ir ūgio išvestines mes gauname tokius paskaičiavimus:

P(t | žemas) = 1/4 x 0 = 0

P(t | vidutinis) = 2/8 x 1/8 = 0,031

P(t | aukštas) = 3/3 x 1/3 = 0,333

Sujungę tai, mes gauname

Tikimybė būti žemam = 0 x 0.267 = 0

Tikimybė būti vidutiniam = 0,031 x 0,533 = 0,0166

Tikimybė būti aukštam = 0,33 x 0,2 = 0,066

Sumuodami šias tikimybių reikšmes kol *t* bus arba žemas, arba vidutinis, arba aukštas apskaičiuojame *P*(t):

P(t) = 0 + 0,0166 + 0,066 = 0,0826

Galiausiai, gauname tikimybes kiekvienam dydžiui:

$$P(\text{žemas} \mid t) = \frac{0 \times 0{,}0267}{0{,}0826} = 0$$

$$P(\text{vidutinis} \mid t) = \frac{0{,}031 \times 0{,}533}{0{,}0826} = 0{,}2$$

$$P(\text{aukštas} \mid t) = \frac{0{,}333 \times 0{,}2}{0{,}0826} = 0{,}799$$

Taigi, remiantis šiomis tikimybėmis, mes priskiriame naują objektą t = (Adam, V, 1,95m) aukštųjų klasei, nes jis turi aukščiausią tikimybę.

Naive Bayes metodas turi keletą pranašumų. Pirma tai yra paprasta naudoti. Antra, skirtingai nei kitų metodų algoritmai, šis nėra iteracinis. Naive Bayes metodas gali lengvai susidoroja su trūkstamomis (angl. missing) reikšmes paprasčiausiai neįtraukiant to tikimybėje.

Nors Naive Bayes metodą yra lengva naudoti, jis ne visada duoda tinkamus rezultatus. Metodas netaikomas tolydiems duomenims. Norint išspręsti šią problemą, tolydžiai reikšmes galima sudalyti į intervalus, tačiau lieka problema, kokie tie intervalai turi būti. Nuo intervalų pasirinkimo gali priklausyti galutiniai klasifikavimo rezultatai.

1.1.2 Atstumo ieškojimu grįsti algoritmai

Kiekvienas elementas, kuris yra taikomas tai pačiai klasei gali būti daugiau panašus kitiems elementams esantiems klasėje, nei tiems elementams kitoje klasėje. Taigi, panašumo matai gali būti naudojami atpažinti skirtingų elementų „panašumus“ duomenų bazėje. Naudojantis paieška Internete, puslapiai (angl. Web pages) sudaro visą duomenų bazę ir jie yra padalinti į dvi klases: į tuos, kurie atsako į tavo užklausą ir tuos, kurie ne. Tie, kurie atsako į tavo užklausą turėtų būti daugiau panašūs į tuos kurie neatsako į užklausą. Panašumas šiuo atveju yra dažniausiai reikšmių žodžių sąrašas (angl. keyword lost) kuriame apibrėžta užklausa. Dažniausiai panašumas yra tas, kad juos sieja vienas ir tas pats užklausos žodis.

Panašumo matų idėja gali būti išskirta ir taikomas daugiau bendriems klasifikavimo uždaviniams. Sudėtingumas priklauso nuo to, kaip panašumo matai yra apibrėžti ir taikomi analizuojamiems duomenims. [5]

Paprastas metodas (angl. Simple Approach)

Jeigu mes turime tipišką (reprezentatyvią) klasę, galime kiekvieną duomenį priskirti kiekvienai labiausiai panašiai klasei atliekant klasifikavimą. Mes teigiame, kad kiekvienas duomuo X_i yra apibrėžtas kaip vektorius $\langle x_{i1}, x_{i2}, ..., x_{ik} \rangle$, kurį sudaro skaitinės reikšmės. Panašiai mes teigiame, kad kiekviena klasė C_j yra apibrėžta duomens $\langle C_{j1}, C_{j2}, ..., C_{jk} \rangle$ skaitinių reikšmių. Klasifikavimo uždaviniai suformuluoti 1.2 apibrėžimu.

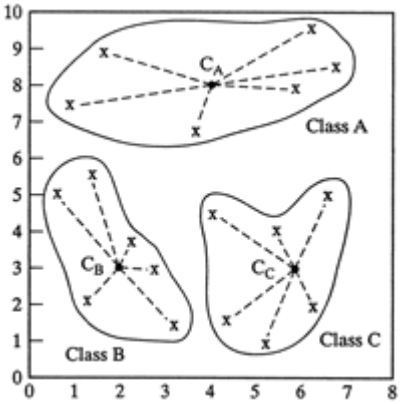
Apibrėžimas 1.2. Duota duomenų aibė $D = \{X_1, X_2, ..., X_n\}$, kur kiekviena duomuo $X_i = \langle x_{i1}, x_{i2}, ..., x_{ik} \rangle$ susideda iš skaitinių reikšmių ir aibė klasių $C = \{C_1, ..., C_m\}$ kur kiekviena klasė $C_j = \langle C_{j1}, C_{j2}, ..., C_{jk} \rangle$ turi skaitines reikšmes, klasifikavimo uždavinys yra priskirti

kiekvieną X_i klasei C_j taip kad $sim(X_i, C_j) \geq sim(X_i, C_l) \forall C, kur\ C_l \neq C_j$.

Norint apskaičiuoti panašumo matus, tipiškas (atstovaujantis) vektorius turi būti apibrėžtas kiekvienai klasei. Nurodant tris klases 1.1 pav.(a), mes galime apibrėžti atstovą kiekvienai klasei apskaičiavus kiekvienos srities centrą. Tokiu būdu klasė A yra sudaryta iš (4, 7,5), klasė B iš (2, 2,5) ir klasė C iš (6, 2,5). Paprasčiausias klasifikavimo metodas yra surasti tipiškiausią kiekvienos klasės elementą, kuris yra labiausiai panašus (arčiausias) į centrą tos klasės. Galima klasės tipiškiausio ieškoti ir kitu būdu. Pavyzdžiui, atpažinimo uždavinių metode, prieš apibrėžiant modelį galima pasinaudoti atvaizdavimu kiekvienos klasė. Kai panašumo matas yra apibrėžtas, kiekvienas elementas bus suklasifikuotas palyginus su kiekvienu prieš tai apibrėžtu metodu. Elementas bus toje klasėje, kur didžiausia panašumo reikšme. Algoritmas 1.1 iliustruoja nesudėtingą atstumu grįstą metodą tariant, kad c_i yra vaizduojama centre ar apie centrą. Kiekvienai klasei algoritme mes naudojame c_i kaip klasės centrą.

```
Algoritmas 1.1
Įvesties duomenys:
      c1, ..., cm // Centras kiekvienai klasei
      t // Įvesties duomuo klasifikavimui
Išvesties duomenys:
      c // Klasė, kuriai t yra priskirtas
Paprastas atstumo ieškojimu grįstas algoritmas
atst = ∞;
 for i := 1 to m do
      if atst(ci, t)< atst then
            c = i;
            atst = atst(ci, t);
```

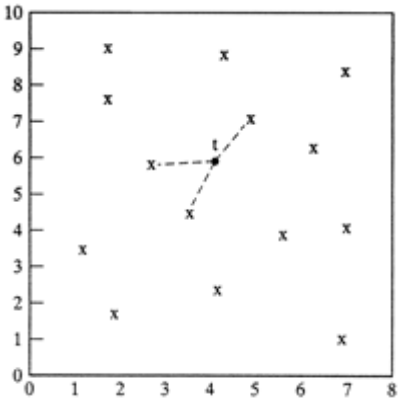
1.7 pav. iliustruoja šitą algoritmą atlikus klasifikaciją naudojant duomenis iš 1.1 pav. Trys dideli apskritimai yra tipiški (angl. representative) tų trijų klasių atstovai. Punktyrinės linijos rodo atstumą kiekvieno duomens atstumą iki artimiausio centro. [5]



1. 7 Klasifikacija naudojant paprastą atstumo algoritmą

K artimiausių kaimynų (angl. K Nearest Neighbors)

Viena labiausiai paplitusių klasifikavimo, pagrįsto atstumų skaičiavimu, schemų yra *K artimiausių kaimynų metodas* (kNN). KNN metodas sako, kad visa mokymo aibė sudaryta ne tik iš aibės duomenų esančių aibėje, bet ir norimo klasifikavimo kiekvienam elementui. Tiesą sakant, mokymo duomenys tampa modeliu. Kada klasifikacija yra sukuriama naujam elementui, tas atstumas kiekvienam elementui mokymo aibėje turi būti apibrėžtas. Tiktai *k* artimiausi įėjimai į mokymo aibę yra smulkiai apgalvoti. Naujas elementas yra patalpintas klasėje, kuri susideda iš *k* artimiausių elementų aibės. 1.8 pav. iliustruoja kNN metodo panaudojimą, kur parodyti taškai apmokymo aibėje ir *k*=*3*. Yra parodyti trys artimiausi elementai mokymo aibėje; *t* bus klasė, kurioje didžiausias kiekis narių. [9]



1. 8 pav. Klasifikacija naudojant kNN metodą

Algoritmas 1.2 nusako kNN algoritmo panaudojimą.

```
Algoritmas 1.2
Įvesties duomenys:
      T     //Mokymo duomenys
      K     //Kaimynų skaičius
      t     //Įvesties duomuo klasifikavimui
Išvesties duomenys:
      c     //Klasė kuriai t yra priskirtas
KNN algoritmas:
            //Algoritmas suklasifikuoti duomenį naudojant kNN
      N = 0;
            //Rasti aibę kaimynų N įvestam t
      for each d ∈ T do
            if |N| ≤ K
             then N = N ∪ {d};
             else
              if ∃u ∈ N taip kaip sim(t,u) ≤ sim(t,d)
```

then begin

$N = N - \{u\};$

$N = N \cup \{d\};$

end;

//Rasti klasifikavimui klasę

c = klasė kuriai dauguma $u \in N$ yra suklasifikuota;

Pavyzdys 1.6 iliustruoja šį metodą naudojant paprastus Lentelės 1 duomenis. kNN metodo rezultatas labai priklauso nuo *k* reikšmės pareiškimo. Pagrindinė taisyklė ta, kad $K \leq \sqrt{mokymo\ duomenų\ skaičius}$. Šiam pavyzdžiui imama reikšmė *3,46*. Komercinės sistemos algoritmai, kuriose realizuojamas kNN algoritmas dažniausiai naudoja nutylėtą (angl. default) reikšmę 10. [5]

Pavyzdys 1.6

Naudojant paprastus Lentelės 1 duomenis ir Išvestis1 klasifikacija kaip mokymo aibę išvesties reikšmėms, mes suklasifikuojame duomenį (Pat, M, 1,6). Atstumas yra tiesiog absoliutinė reikšmė skirtumo tarp reikšmių. Teigiame, kad duota *k = 5*. Mes turime, kad *k* artimiausių kaimynų įvestiems duomenims yra {(Kristina, M, 1,6), (Kathy, M, 1,6), (Stephanie, M, 1,7), (Dave, V, 1,7), (Wynette, M, 1,75)}. Iš šių penkių duomenų, keturi yra suklasifikuoti kaip žemi ir vienas kaip vidutinis. Vadinasi, kNN suklasifikuos Pat kaip žemą.

1.1.3 Sprendimų medžio sudarymu grįsti algoritmai

Klasifikavimo uždaviniams spręsti dažniausiai naudojamas sprendimo medžio sudarymo algoritmas. Klasifikavimo metu sukuriamas medžio modelis. Sukurtas medis pritaikomas kiekvienam duomeniui. Yra du pagrindiniai šio metodo žingsniai: sudaryti medį ir paskui panaudoti sukurtą medį naujų duomenų aibei. Daugelis tyrinėjimų yra nukreipti į tai, kaip sukurti efektyvius medžius per trumpiausią laiką.

Sprendimų medžio klasifikavimo esmės yra kaip padalinti duomenų aibės erdvę į stačiakampes sritis. Duomens suklasifikavimas priklauso nuo to, į kurią sritį jis paklius. Sprendimų medis naudojamas klasifikavime yra apibrėžtas Apibrėžime 1.3.

Apibrėžimas 1.3. Duota duomenų aibė $D = \{X_1,...,X_n\}$, kur $X_i = \langle x_{i1},...,x_{ih} \rangle$. Duomenis X_i apibūdina parametrai $\{x_1, x_2,..., x_h\}$. Taip pat duota klasių aibė $C = \{C_1,...,C_m\}$. Sprendimų medis ar klasifikavimo medis yra susietas su *D,* kuris turi tokias ypatybes:

- Kiekviena vidinė viršūnė yra pavadinta x_i parametru.

- Kiekvienas lankas yra pavadintas su predikatu, kuri gali būti taikoma pirminiam parametrui.
- Kiekvienas viršūnės lapelis yra pažymėtas (angl. labeled) klase C_j.

Sprendžiant klasifikavimo uždavinius, kuriuose naudojamas sprendimų medis, yra du žingsniai:

- Sprendimų medžio indukcija (angl. induction): sudaryti sprendimų medį naudojant mokymo duomenis.
- Kiekvieną $X_i \in D$ pagal sukurtą medį, priskirti vienai iš analizuojamų klasių.

Yra daug sprendimų medžio naudojimo klasifikavimui privalumų. Sukurtas taisykles lengva interpretuoti ir suprasti. Klasifikavimo medžiai tinka didelėms duomenų aibėms, nes medžio dydis nepriklauso nuo duomenų aibės dydžio. Kiekvienas analizuojamas duomuo turi prasiskverbti pro medį. Medis gali būti sukurtas duomenims su daug parametrų.

Sprendimų medžių sudarymo algoritmai turi ir trūkumų. Pirmiausia jie sunkiai susitvarko su tolydžiais duomenimis. Šios parametrų sritys turi būti padalintos į kategorijas. Naudotas metodas parodo, kad erdvės dalis yra padalinta yra stačiakampes sritis (kaip matome 1.1 pav.(a)). Ne visi klasifikavimo uždaviniai yra tokio tipo. Apdoroti trūkstamus duomenis (angl. missing data) yra sunku, tuomet sprendimų medis yra sudaromas iš mokymo duomenų, gali įvykti per didelis tikimas (angl. overfitting[1]). Tai gali atitikti per medžio genėjimą (sumažinimą). Koreliacija tarp parametrų duomenų aibėje yra nepaisoma sprendimų medžio sudarymo procese. [5]

Algoritmas 1.3

Įvesties duomenys:

```
D     //Mokymo duomenys
```

Išvesties duomenys:

```
T     //Sprendimų medis
```

DTBuild algoritmas:

```
            //Paprastas algoritmas iliustruojantis sprendimų medžio
sudarymo metodą
            T = ∅;
            Apibrėžti geriausius skėlimo kriterijus;
            T = Sukuriamas pirminis taškas (mazgas) ir pavadinamas skėlimo
parametru;
            T = Pridedamas lankas pirminiam taškui kiekvienam skėlimo
predikatui ir pažymėjimui;
```

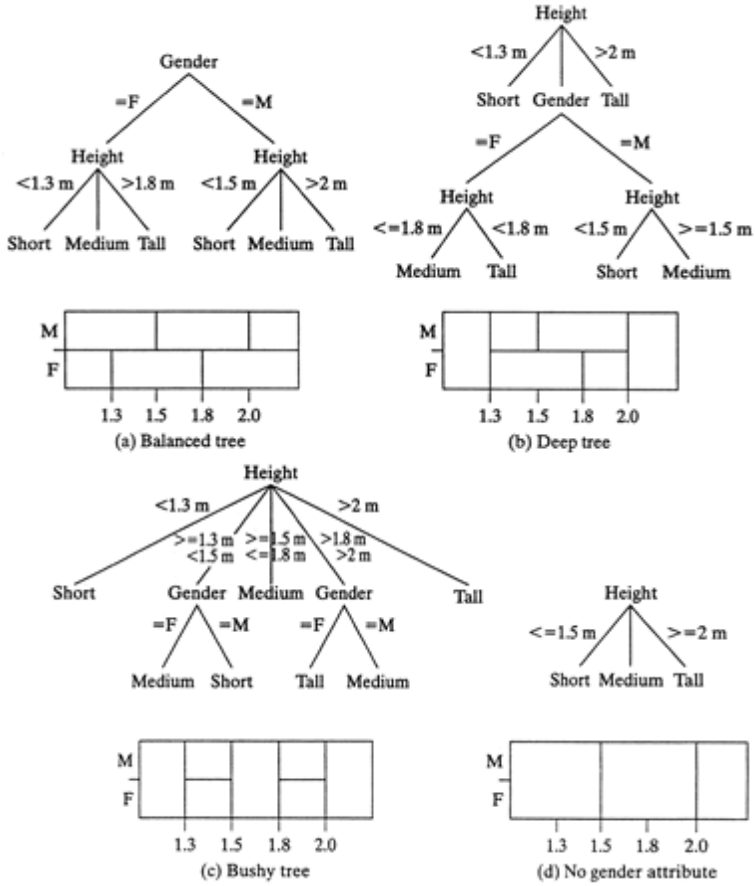
[1] Overfitting – tai atsitinka tada, aki sukurtas medis (ar taisyklės) puikiai tinka mokymo duomenims, tačiau visiškai netinka naujiems duomenims, kurie nebuvo mokymo duomenų aibėje, klasifikuoti, t.y. jie neteisingai priskiriami analizuojamoms klasėms.

```
            for each arc do
                 D = Sukuriame duomenų aibė D skėlimo predikatui;
                  if sustojimo taškas pasiekią šį savo kelią,
                       then T` = Sukuriamas lapelio taškas ir pažymimas
atitinkama klase;
                       else T` = DTBuild(D);
                 T = Pridėti T` tam lankui;
```

Mes iliustruosime medžio sudarymo fazes supaprastintu DTBuild Algoritmu 1.3. nuo viršūnės (angl. nodes) formuojamame medyje einantys parametrai yra vadinami *perskyrimo parametrais* (angl. splitting attributes), o nuo perskyrimo viršūnės einantys lanku parametrai vadinami *perskyrimo predikatai* (angl. splitting predicates). Sprendimų medyje parodytame 1.9 pav., perskyrimo parametrai yra {lytis, ūgis}. Perskyrimo predikatai lyčiai yra {= moteris, = vyras}, o ūgiui yra {< 1,3 m, > 1,8 m, < 1,5 m, > 2 m}. Perskyrimo predikatai ūgiui yra moteris arba vyras. Šitas algoritmas suformuoja nuo viršaus žemyn einantį medį. Naudojant mokymo duomenis pasirenkamas „geriausias“ perskyrimo parametras. Algoritmai skiriasi tuo, kaip jie apibrėžia „geriausią parametrą“ ir tas „geriausias predikatas“ naudojamas perskyrimui. Vieną kartą kai jis būna apibrėžtas, viršūnė ir jos lankas yra sukuriami ir įtraukiami į formuojamą medį. Algoritmas tęsiamas rekursiškai įtraukiant naujus pomedžius kiekvienam šakos lankui. Algoritmas baigiamas kai pasiekiamas kažkoks „sustojimo kriterijus“. Vėl kiekvienas algoritmas skirtingai apibrėžia kada sustoti. Vienas paprastas sustojimo metodas yra kada visi duomenys suskaidytoje mokymo duomenų aibėje priklauso tai pačiai klasei. Ta klasė tada naudojama pažymėti sukurtą viršūnę.

Dauguma faktorių atliekant sprendimų medžio sukūrimo algoritmą yra mokymo aibės dydis ir kaip gerai pasirinktas parametras. Keletas problemų pasitaikančių daugelyje sprendimų medžio sudarymo algoritmų:

- **Skėlimo parametrų pasirinkimas** (angl. choosing splitting attributes).
- **Skėlimo parametrų tvarka** (angl. ordering of splitting attributes).
- **Skėlimai** (angl. splits).
- **Medžio struktūra** (angl. tree structure).
- **Sustojimo kriterijus** (angl. stopping criteria).
- **Mokymo duomenys** (angl. training data).
- **Genėjimas** (sumažinimas) (angl. pruning).



1. 9 pav. Sprendimų medžių palyginimas

1.9 pav. parodo keturis skirtingus sprendimų medžius, kurie gali būti panaudoti suklasifikuoti žmones pagal jų ūgį. Šios iliustracijos pirmi trys medžiai atlieka tą pačią klasifikaciją, nors jie visi tai vykdo skirtingai. Apačioje kiekvieno medžio yra parodyti loginiai dalijimai panaudojus susietus medžius klasifikavimui. Puikus klasifikavimas 1.9 pav.(a), kur medis yra subalansuotas. Medis yra tokio pačio gilumo kiekvienam keliui nuo pirminio taško iki lapelio. 1.9 pav.(b) ir (c) nėra subalansuotos. Be to, medžio dydis esantis (b) yra didesnis už visus kitus, reiškia daug blogesnis klasifikavimas. Tai gal nebus kritinis atlikimo klausimas nebent jei duomenų aibė yra nepaprastai didelė, tuo atveju, subalansuotas mažesnis medis būtų nepageidautinas. Medis pavaizduotas 1.9 pav.(d) nevaizduoja tokios pačios klasifikavimo logikos kaip kiti.

Mokymo duomenys ir medžio indukcijos algoritmas apibrėžia medžio formą. Tokiu būdu geriausios formos medis, kuris įvykdo geriausiai mokymo duomenyse yra pageidautinas. Kai kurie algoritmai sukuria tik binarinius medžius. Binariniai medžiai yra lengvai sukuriami, bet jie linkę būti sunkesni. Atlikimo rezultatai, kada panaudojami šiems medžio tipams, klasifikavimui gali būti blogesni, nes dažniausiai reikia daugiau palyginimų. Bet tokiu atveju, kai šitie palyginimai yra lengvesni negu tie, kurie reikalauja daugiakrypčių išsišakojimų.

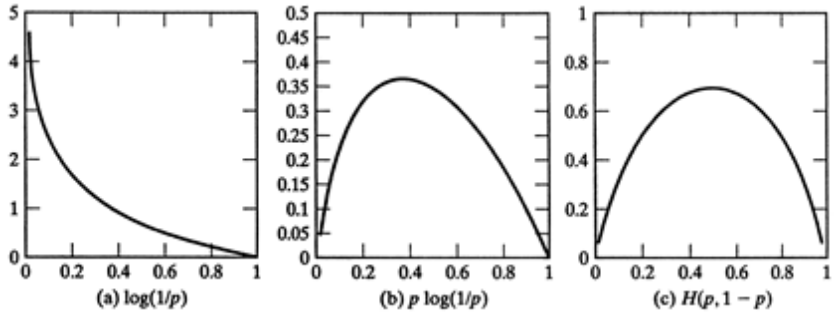
Sprendimų medžio sudarymo algoritmai gali iš pradžių sukurti medį ir tada apkarpyti jį efektyvesniam klasifikavimui. Su genėjimo metodu, dalis medžio gali būti ištrintas arba sujungtas sumažinti bendrą medžio dydį. Dalis medžio nesvarbius parametrus naudojamus

klasifikavimui galima ištrinti. Šis mažas pakeitimas su šakos viršūne gali nusiristi žemyn sukurti didelius pakeitimus žemiausioje medžio dalyje. Pavyzdžiui, su duomenimis 1.1 pav., jei medis, kur sukurtas žiūrint į parametrų reikšmių vardus, visos šakų viršūnės pavadinamos su tais parametrais turi būti ištrinta. Žemesnio lygio viršūnės gali būti pakeltos arba sujungtos kai kuriais atvejais. Per didelio tikimo atveju, žemesnis lygis submedžių gali būti galutinai ištrintas. Genėjimas gali būti atliktas kol medis yra kuriamas, tokiu būdu užkertamas kelias medžiui tapti per dideliu. Kitas genėjimo metodas, kai medis po to yra sukuriamas. [5]

ID3

ID3 metodas, pagal kurį sukuriamas sprendimų medis, yra pastanga minimizuoti tikėtiną palyginimų skaičių. Pagrindinis indukcijos algoritmo tikslas yra paklausti tokius klausimus, kurių atsakymai teikia daugiausiai informacijos. Tai yra panašu į intuityvų metodą paimtą iš suaugusiųjų, kada žaidžiamas „Dvidešimt klausimų“ žaidimas. Pirmas klausimas kuris suaugusioje gali būti užduotas galėtų būti „Ar valgomas ar ne?“, tuo tarpu vaikas galėtų paklausti „Ar čia mano tėtis?“. Pirmas klausimas padalina paieškos erdvę į dvi dideles paieškos sritis, kol kitas klausimas įvykdo mažą erdvės padalijimą. Pagrindinė strategija panaudota ID3 yra pasirinkti perskyrimo parametrus su aukščiausia informacijos nauda pirmai.

Sąvoka naudojama išmatuoti informaciją yra vadinama *entropija*. Sprendimų medžio klasifikavimo tikslas yra įtakoti duotų duomenų aibės dalinimus į poaibius, kur visi elementai kiekviename galutiniams poaibyje priklauso tai pačiai klasei. 1.10 pav.(a, b ir c) padės paaiškinti šią sąvoką. 1.11 pav.(a) parodo log(1/p) tikimybę *p* intervale nuo 0 iki 1. Tai intuityviai rodo kiekį tikimybių paremtų netikėtinumu. Kai *p = 1* nėra netikėtinumo (angl. surprise). Tai reiškia, kad jei įvykis turi tikimybę lygią 1 ir pasakyta, kad įvykis atsitiktinis, nebūsime nustebinti. Kai $p \to 0$, netikėtinumas didėja. 1.10 pav.(b) vaizduoja funkciją $p\log(1/p)$, kuri yra tikėtina informacija paremta tikimybe visų įvykių. Apibrėžti laukiamą informaciją susietą su dviem įvykiais, mes sudedam atskiras reikšmes kartu. Šita funkcija $p\log(1/p)+(1-p)\log(1/(1-p))$ yra pavaizduota 1.10 pav.(c). Pastebime kad maksimumas pasiekiamas kai dvi tikimybės yra lygios.



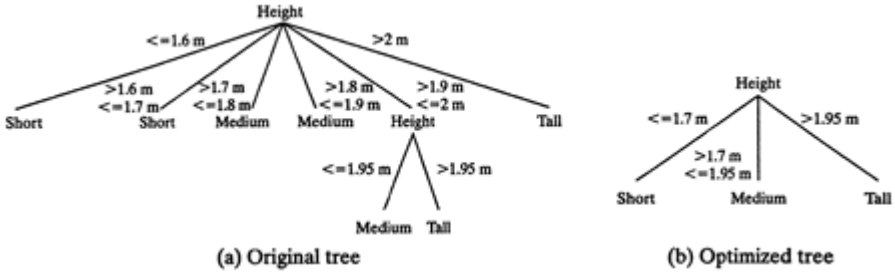
1. 10 pav. Entropija

Formalus entropijos apibrėžimas yra pateiktas Apibrėžime 1.4. Entropijos reikšmė yra tarp 0 ir 1 ir pasiekia maksimumą kai visos tikimybės yra vienodos.

Apibrėžimas 1.4. Duotoms tikimybėms $p_1, p_2, ..., p_s$ kur $\sum_{i=1}^{s} p_i = 1$, **entropija** yra apibrėžiama kaip

$$H(p_1, p_2, ..., p_s) = \sum_{i=1}^{s} (p_i \log(1/p_i)) \qquad (1.16)$$

Pavyzdys 1.7 ir 1.11 pav. iliustruoja šį procesą naudojant ūgių pavyzdį. Šiame pavyzdyje yra panaudoti šeši padalijimai iš galimų ūgio sričių. Šis dalijimas į sritis yra reikalingas, kada parametrų sritis yra tolydi ar (šiuo atveju) susideda iš daug reikšmių.



1. 11 pav. Klasifikavimo problema

Pavyzdys 1.7

Pradinė tikimybė mokymo duomenyse iš Lentelės 1 (su Išvestis1 klasifikavimu) yra tokia (4/15) yra žemas, (8/15) yra vidutinis ir (3/15) yra aukštas. Taigi, pradinė entropija yra

4/15 log(15/4) + 8/15 log(15/8) + 3/15 log(15/3) = 0,4384

Renkantis lytį kaip skėlimo parametrą, yra devyni duomenys kurie yra *M* ir šeši kurie yra *V*. Aibės entropija kur yra *M* yra

3/9 log(9/3) + 6/9 log(9/6) = 0,2764 (1.18)

Tuo tarpu aibei *V* yra

1/6 log(6/1) + 2/6 log(6/2) + 3/6 log(6/3) = 0,4392 (1.19)

ID3 algoritmas turi būti apibrėžtas, kad gauti informaciją naudojant šį skėlimą. Tai padarę mes apskaičiuojame svorį sumos šitų dviejų paskutinių entropijų gauname

((9/15) 0,2764) + ((6/15) 0,4392) = 0,34152 (1.20)

Entropijos įgyjama reikšmė naudojant lyties parametrą gauname

0,4384 – 0,34152 = 0,09688 (1.21)

Žiūrint į ūgio parametrus mes turime du duomenis kurie yra 1,6, du yra 1,7, vienas yra 1,75, du yra 1,8, vienas yra 1,85, vienas yra 1,88, du yra 1,9, vienas yra 2, vienas yra 2,1 ir vienas yra 2,2. Nustatyti skėlimo ūgio reikšmes nėra lengva. Nors lyginant apmokymo duomenų aibę turi tuos 11 reikšmių, mes žinome, kad jų bus daug daugiau. Taip kaip su tolygiais duomenimis, mes daliname į intervalus:

(0, 1,6], (1,6, 1,7], (1,7, 1,8], (1,8, 1,9], (1,9, 2,0], (2,0, ∞)

Yra 2 duomenys pirmame dalijime su entropija (2/2(0) + 0 + 0) = 0, 2 esantys (1,6, 1,7]

su entropija (2/2(0) + 0 + 0) = 0, 3 esantys (1,7, 1,8] su entropija (0 + 3/3(0) + 0) = 0, 4 esantys (1,8, 1,9] su entropija (0 + 4/4(0) + 0) = 0, 2 esantys (1,9, 2,0] su entropija (0 + 1/2(0,301) + 1/2(0,301) = 0,301 ir du esantys paskutinėje entropijoje (0 + 0 + 2/(0)) = 0. Visos šitos struktūros yra visiškai sutvarkytos ir nors entropija 0 tikėtina (1,9, 2,0] struktūroje. Entropijos nauda naudojant ūgio parametrus yra

$$0{,}4384 – 2/15(0{,}301) = 0{,}3983 \qquad (1.22)$$

Tokiu būdu, tai turi didesnę struktūrą ir mes pasirenkame tai dalijant iš ūgio kaip pirmas išdalijimo atributas. Šiame dalijime yra du vyriškiai, vienas vidutinis ir vienas aukštas. Tai atsitiko nes šis grupavimas buvo per didelis. Kitas padalijimas iš ūgio yra reikalingas ir sukurtas sprendimų medis pavaizduotas 1.11 pav.(a).

1.11 pav.(a) iliustruoja uždavinį kur medis turi daugiamačius dalinimus su identiškais rezultatais. Sudedant kur yra pasidalijimas srities (1,9, 2,0]. 1.11 pav.(b) parodo optimizuotą medžio sudarymo versiją. [5]

C4.5 and C5.0

Sprendimų medžio algoritmas C4.5 pagerina ID3 keletu būdų:

- **Trūkstami duomenys** (angl. missing data): kuriant sprendimo medį trūkstami duomenys yra tiesiog ignoruojami. Norint suklasifikuoti įrašus su trūkstamom parametrų reikšmėmis, tų parametrų reikšmės gali būti apskaičiuotos pagal žinomos kitų duomenų tų parametrų reikšmes.

- **Tolydūs duomenys** (angl. continuous data): pagrindinė idėja yra padalinti duomenis į sritis pagal parametrų reikšmes tiems duomenims, kurie yra mokymo šablone.

- **Genėjimas** (angl. pruning): yra dvi pirminės genėjimo strategijos taikomos C4.5 metode:

    - Su *pomedžio pakeitimu* (subtree replacement), medis yra pakeičiamas lapo viršūne, jei šie pakeitimo rezultatai klaidos koeficientas arti iki to originalaus medžio. Pomedžio pakeitimas veikia nuo pat medžio viršūnės iki šaknų.

    - Kita genėjimo strategija, vadinama *pomedžio pakėlimas*, pakeičia pomedį pačiu naudingiausiu. Čia pomedis yra pakeliamas nuo konkrečios vietos iki aukštesnės medžio viršūnės. Vėl mes turime apibrėžti šio pakeitimo paklaidos padidėjimą.

- **Taisyklės** (angl. rules): C4.5 leidžia klasifikavimą, pagal bet kurį sprendimo medį arba taisyklių sukūrimą. Be to, kai kurie metodai pasiūlyti supaprastinti sudėtingas taisykles. Vienas iš siūlymų yra pakeisti kairiąją pusę taisyklės paprastesne versija jei visi įrašai mokymo aibėje yra laikomi identiškais. Nors ši taisyklė gali būti naudojama parodyti, kad turėtų būti jei kitos taisyklės yra patvirtintos.

Perskyrimas (angl. splitting): ID3 metodas pateikia parametrus su daug išdalijimų ir kraštutiniu atveju gali atsitikti per dideli tikimai. Parametras kuris turi vienintelę reikšmę kiekvienam duomeniui mokymo aibėje būtų geriausias, nes ten būtų tik vienas duomuo (ir tokiu būdu viena klasė) kiekvienam dalijimui. [11]

C5.0 (vadinamas See 5 Windows operacinėje sistemoje) yra komercinė C4.5 versija dabar plačiai naudojama daugelyje duomenų klasifikavimo paketų tokių kaip Clementine ir RuleQuest. Sprendimų medžio induksija yra arčiausiai to C4.5, bet taisyklių sukūrimas yra skirtingas. Priešingai nei C4.5, tikslūs algoritmai, kurie naudoti C5.0, nebuvo atskleisti. C5.0 įtraukia ir priemones kurti taisykles. Rezultatai rodo, kad C5.0 patobulinta atminties sunaudojime apie 90 procentų, veikia tarp 5,7 ir 240 kartų greičiau nei C4.5 ir pateikia tikslesnes taisykles. [5]

1.1.4 Taisyklių sudarymu grįsti algoritmai

Vienas iš lengviausiai interpretuojamų klasifikatorių yra pagrįsti taisyklių sudarymu. Pavyzdžiui, mes turime tokius balus, pagal kuriuos sukuriame taisykles, pagal kurią duomenys gali būti suklasifikuojami į penkias klases (A, B, C, D, F):

Jei 90 <= balas, tada klasė = A
Jei 80 <= balas ir balas < 90, tada klasė = B
Jei 70 <= balas ir balas < 80, tada klasė = C
Jei 60 <= balas ir balas < 70, tada klasė = D
Jei balas < 60, tada klasė = F

Klasifikavimo taisyklė, *r=(a, c)*, susideda iš *a* dalis *if* arba pirmumas (anksčiau vykęs) ir *c* dalie *tada* arba rezultatas (pasekmė). Pirmumas susideda iš predikato, kuris gali būti įvertintas kaip tiesa arba melas susietas tiksliai su kiekvienu analizuojamu duomeniu (ir akivaizdžiai mokymo duomenyse). Šios taisyklės susieja tiesiai su atitinkamu sprendimų medžiu, kas gali būti sukurta. Sprendimų medis visada gali būti panaudota sukurti taisykles, bet jos nėra ekvivalenčios. Yra skirtumai tarp taisyklių ir medžių:

- Medis turi numatoma tvarką, kur atliekamas perskyrimas. Taisyklės neturi tvarkos.
- Medis yra sukuriamas remiantis visomis klasėmis. Kada kuriamos taisyklės, tik viena klasė gali būti nagrinėjama tuo momentu.

Yra algoritmai, kurie generuoja taisykles iš medžių taip pat puikiai kaip ir algoritmai, kurie generuoja taisykles iš pradžių nesukūrus sprendimo medžio. [5]

Taisyklių generavimas iš sprendimų medžio (angl. rule induction)

Procesas sukurti taisyklei iš sprendimo medžio, yra paprastas ir apibrėžtas 1.8 algoritme. Šis algoritmas sukurs taisyklę kiekvienam lapeliui sprendimų medyje. Visos taisyklės su vienodom pasekmėm turėtų būti sujungtos su O žiedu (angl. *ORing*) pirmenybė paprastų taisyklių.

Algoritmas 1.8

Įvesties duomenys:

T //Sprendimų medis

Išvesties duomenys:

R //Taisyklės

Gen algoritmas:

//Iliustruoja paprastą metodą, kaip iš sprendimų medžio kuriamos klasifikavimo taisyklės

R = 0;

for each keliui nuo šaknies iki lapelio esančiam T **do**

a = a^ (viršūnės žymėjimas sujungtas su atsitiktiniu išeinančiu lanku)

c = lapelio viršūnės žymėjimas

R = $R \cup r$ = (a, c)

Naudojant algoritmą, atitinkamai taisyklės yra sukuriamos iš sprendimų medžio 1.11 pav.(a):

{((Ūgis <= 1.6 m), Žemas)
(((Ūgis > 1.6 m) ^ (Ūgis <= 1.7 m)), Žemas)
(((Ūgis > 1.7 m) ^ (Ūgis <= 1.8 m)), Vidutinis)
(((Ūgis > 1.8 m) ^ (Ūgis <= 1.9 m)), Vidutinis)
(((Ūgis > 1.9 m) ^ (Ūgis <= 2 m) ^ (Ūgis <= 1.95 m)), Vidutinis)
(((Ūgis > 1.9 m) ^ (Ūgis <= 2 m) ^ (Ūgis > 1.95 m)), Aukštas)
((Ūgis > 2 m), Aukštas)}

Optimizuota šitų taisyklių versija būtų:

{((Ūgis <= 1.7 m), Žemas)
(((Ūgis > 1.7 m) ^ (Ūgis <= 1.95 m)), Vidutinis)
((Ūgis > 1.95 m), Aukštas)}

1.2 Klasifikavimo tikslumo įvertinimas

Lentelėje 1 pavaizduoti du skirtingi klasifikavimo rezultatai naudojant du skirtingus klasifikavimo metodus. Nustatyti, kuris yra geriausias labai subjektyvu ir priklauso nuo vartotojo, kuris tą uždavinį nagrinėja. Klasifikavimo algoritmas yra dažniausiai įvertinamas pagal klasifikavimo tikslumą. Nors pasitaiko, atvejų, kai neįmanoma formaliai įvertinti klasifikavimo tikslumo, įvertinant teisingai suklasifikuotų duomenų santykį.

Klasifikavimo tikslumas yra dažniausiai apskaičiuojamas, kurie yra teisingose klasėse. Tai atmeta faktą, kad gali taip pat būti įskaičiuoti ir duomenys, kurie buvo priskirti klaidingai tai

klasei.

Mes galime analizuoti klasifikacijos atlikimą tiek kiek yra susiję su informacijos išgavimu iš sistemos.

Tikrai teigiamas	Klaidingai neigiamas
Klaidingai teigiamas	Tikrai neigiamas

Klasės prognozavimas

1. 12 pav. Klasifikavimo atlikimas informacijos išgavimui

Suteikta apibrėžta klasė C_j ir duomenų aibė, X_i, kur duomenys gali būti arba ne priskirti tai klasei kol iš tikrųjų narystė priskiriama arba ne tai klasei. Tai mums vėl duoda keturis kvadratus kurie pavaizduoti 1.12 pav., kurie gali būti aprašyti būtent taip:

- *Tikrai teigiamas (TT)* (angl. True positive (TP): objektas X_i priskirtas klasei C_j ir iš tiesų jis jai priklauso;
- *Klaidingai teigiamas (KT)* (angl. False positive (FP): objektas X_i priskirtas klasei C_j, bet iš tiesų jis jai nepriklauso;
- *Tikrai neigiamas (TN)* (angl. True negative (TN): objektas X_i nepriskirtas klasei C_j ir iš tiesų jis jai nepriklauso;
- *Klaidingai neigiamas (KN)* (angl. False negative (FN): objektas X_i nepriskirtas klasei C_j bet iš tiesų jis jai priklauso.

Dviejų klasių atveju klasifikuojami objektai gali priklausyti klasei (tokius objektus vadinsime *teigiamais*) ir nepriklausyti klasei (neigiami objektai). Taip pat kiekvienas objektas klasifikatoriaus yra priskiriamas vienai iš klasių. Rezultatas taip pat gali būti arba teigiamas arba neigiamas (objektas klasifikatoriaus priskirtas nurodytai klasei arba ne). Teigiamas objektas ir klasifikatoriaus priskirtas nurodytai klasei, jis vadinamas *tikrai teigiamu* (TT). Jeigu neigiamas objektas klasifikatoriaus nepriskirtas nurodytai klasei, jis vadinamas *tikrai neigiamu* (TN). Teigiamas objektas, klasifikatoriaus priskirtas klaidingai klasei, vadinamas *klaidingai neigiamu* (KN), o neigiamas objektas, klasifikatoriaus priskirtas tiriamai klasei, vadinamas *klaidingai teigiamu* (KT).

Tikros klasės	Priskyrimas prie klasių		
	Žemas (-a)	Vidutinis (-ė)	Aukštas (-a)
Žemas (-a)	0	4	0
Vidutinis (-ė)	0	5	3
Aukštas (-a)	0	1	2

Lentelė 3. Sumaišymo matrica

Sumaišymo matrica iliustruoja klasifikavimo tikslumą. Duota *m* klasių, sumaišymo matrica yra *m x m* matrica kur c_{ij} parodo skaičių duomenų iš *D,* kuri yra priskirta klasei C_j, bet teisinga klasė yra C_i. Akivaizdu, kad geriausias sprendimas bus, kai įstrižainė yra sudaryta iš nulių. Lentelė 3 parodo sumaišymų matricą ūgio pavyzdžiui iš lentelės 1, kur Išvestis1 yra tariamai teisingi, o Išvestis2 rezultatas yra tikrasis.

Klasifikavimo kokybė yra apskaičiuojama naudojant tokias formules:

$$\textit{specifiškumas} = \frac{TN\ skaičius}{TN\ skaičius + KT\ skaičius} \tag{1.27}$$

$$\textit{jautrumas} = \frac{TT\ skaičius}{TT\ skaičius + KN\ skaičius} \tag{1.28}$$

$$\textit{bendras klasifikavimo tikslumas} = \frac{TT\ skaičius + TN\ skaičius}{visų\ objektų\ skaičius} \tag{1.29}$$

Pateiktų tikslumo matų prasmę iliustruosime medicininių duomenų klasifikavimo pavyzdžiu. Tarkime, kad grupei žmonių atliekamas tyrimas tam tikrai ligai nustatyti. Keli žmonės iš šios grupės tikrai serga šia liga ir atliktas tyrimas rodo, kad šie žmonės serga. Tokius žmones vadinsime tikrai teigiamais (TT). Kita grupelė tiriamųjų serga šia liga, tačiau tyrimas rodo, kad jie yra sveiki. Juos vadinsime klaidingai neigiamais (KN). Keli žmonės iš šios grupės neserga šia liga ir atliktas tyrimas rodo, kad šie žmonės neserga – juos vadinsime tikrai neigiamais (TN). Galiausiai, yra žmonių kurie neserga nurodyta liga, tačiau tyrimas parodė, kad jie serga. Tokius tiriamuosius vadinsime klaidingai teigiamais (KT). Visų tikrai teigiamų, klaidingai neigiamų, tikrai neigiamų ir klaidingai teigiamų tiriamųjų skaičius sudaro 100% visų tiriamųjų aibės.

Jautrumo matas parodo santykį sergančiųjų, kurių liga patvirtinta tam tikru diagnostikos metodu arba tyrimu (tikrai teigiami tiriamieji), su visų sergančiųjų skaičiumi; šis matas rodo tikimybę, kad sergančio žmogaus tyrimo duomenys patvirtins ligą. Kuo didesnis jautrumo matas, tuo mažesnė tikimybė, kad tikrai sergančiam žmogui liga bus diagnozuota.

Specifiškumo matas parodo santykį tarp tikrai neigiamų (neserga šia liga ir atliktas tyrimas rodo, kad neserga) ir visų nesergančių žmonių. Panašiai kaip ir jautrumo matas rodo tikimybę, kad sveiko žmogaus tyrimo duomenys patvirtina, jog jis yra sveikas. Kuo didesnė specifiškumo mato reikšmė, tuo mažiau sveikų žmonių yra priskirta sergantiems. [11]

2. KLASIFIKAVIMO REZULTATŲ VIZUALIZAVIMO TYRIMAS

2.1 Analizuojamos sistemos

2.1.1 *Orange* sistema

Orange sistemoje yra realizuota daug sistemos mokymo (angl. machine learning) algoritmų, taip pat papildomų įrankių duomenų įvedimui ir apdorojimui.

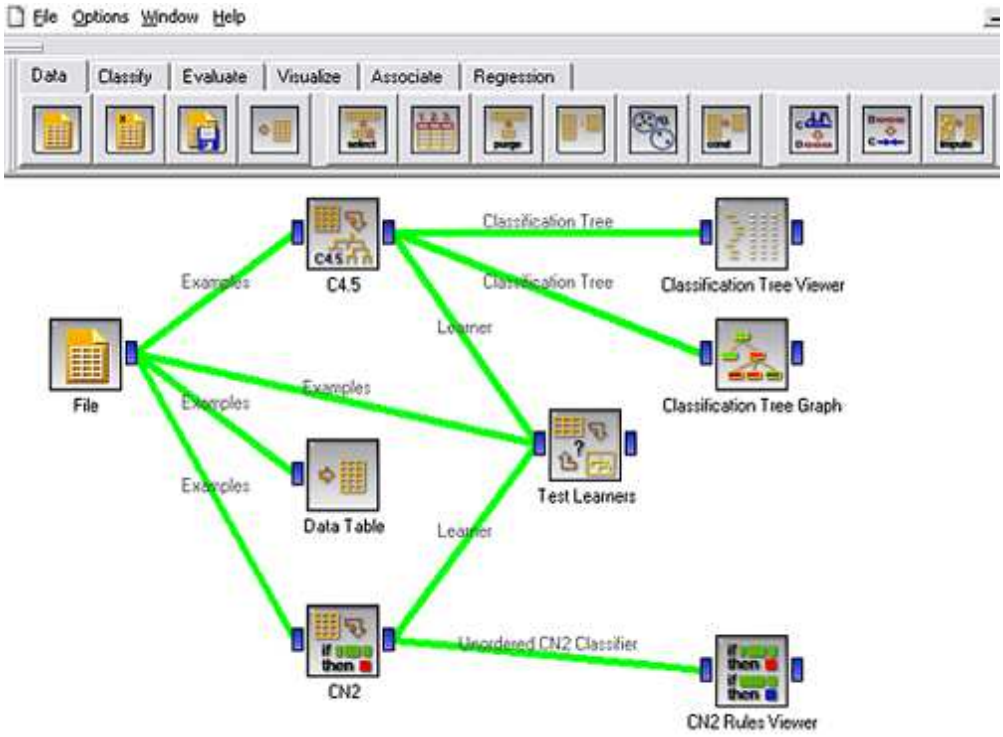
Orange yra sudedamoji dalis iš *Python* pagrindinių modulių ir vykdo įvairias funkcijas, kurios yra lengviau padaromas *Python* nei *C++* kalba. Tai apima daugelį uždavinių, tokių kaip pasirinkimų medžio atspausdinimas, parametrų aibė ir kita.

Orange taip pat turi aibę grafinių objektų, kurie naudoja metodus iš pagrindinės bibliotekos ir *Orange* modulių. Vizualinio programavimo dėka, grafiniai objektai gali būti surinkti į aplikaciją pasinaudojus vizualinio programavimo įrankiu pavadintu *Orange Canvas*.

Orange, kuri skirta sistemos mokymui ir duomenų analizei, yra visapusiška, nes tinka tiek patyrusiam vartotojui, tiek pradedančiajam sistemos mokymesi, kuris nori plėtoti ir testuoti savo paties algoritmus.

Orange yra komponentėmis paremta struktūra. Šioje sistemoje galima naudoti jau sukurtas komponentes arba susikurti savo, kurios paskui naudojamos *Orange* klasifikavimo medžio indukcijos algoritmuose. *Orange* naudoja *Python* kaip jungiamąją programavimo kalbą. Kai kurios *Orange* savybės:

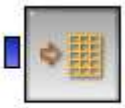
- Duomenų įvedimas/išvedimas: Orange gali nuskaityti iš ir įrašyti tekstą su tarpais failus ir kt. formatus.
- Iki duomenų apdorojimas: galimybė pažymėti aibę, diskretizacija.
- Prognozavimo modeliai: klasifikavimo medis, Naive Bayesian klasifikatorius, kNN (k artimiausių kaimynų), loginė regresija, taisyklėmis grįsti algoritmai (pavyzdžiui CN2).
- Duomenų aprašymo metodai, įvairios vizualizacijos priemonės, saviorganozuojantys tinklai, hierarchinis klasterizavimas, daugiamačių skalių metodas ir kita. [4]



2. 1 pav. Orange sistemos pagrindinis langas

Orange Canvas sistemos pagrindinis langas pavaizduotas 2.1 pav. Kortelėje *Data* įrankis

nuskaito duomenis iš failo. Nustatyti ar duomenys buvo gerai nuskaityti, prie įrankio



File prijungiame toje pačioje kortelėje esantį įrankį . Atsidarę *Data Table* nuskaitytus duomenis matome lentelės pavidalu kur antraštėje surašyti parametrų pavadinimai. Toliau iš kortelės *Classify* pasirenkame klasifikavimui reikalingus metodus. Pavyzdžiui įkeliamas

klasifikatorius . Kad duomenys būtų klasifikuojami šiuo klasifikatoriumi, reikia įrankį *File* sujungti tempiant pelyte su *CN2*. Paskutinis etapas yra duomenų atvaizdavimas. Šiuo atveju tam yra visa kortelė *Vizualine*. Vadinasi sujungę įrankį *CN2*, kuris jau yra sujungtas su

File, su mes išvysime vaizdą pavaizduotą 2.6 paveikslėlyje. Norint įvertinti ar

klasifikatorius gerai apsimokė, galime įkelti kortelėje *Evaluate* esantį įrankį ir jį sujungti su įrankiais *File* ir pasirinktu klasifikatoriumi, mūsų atveju *CN2*. Paskui atidarius mums

bus pateikta lentelė kaip 2.3 paveiksliukas.

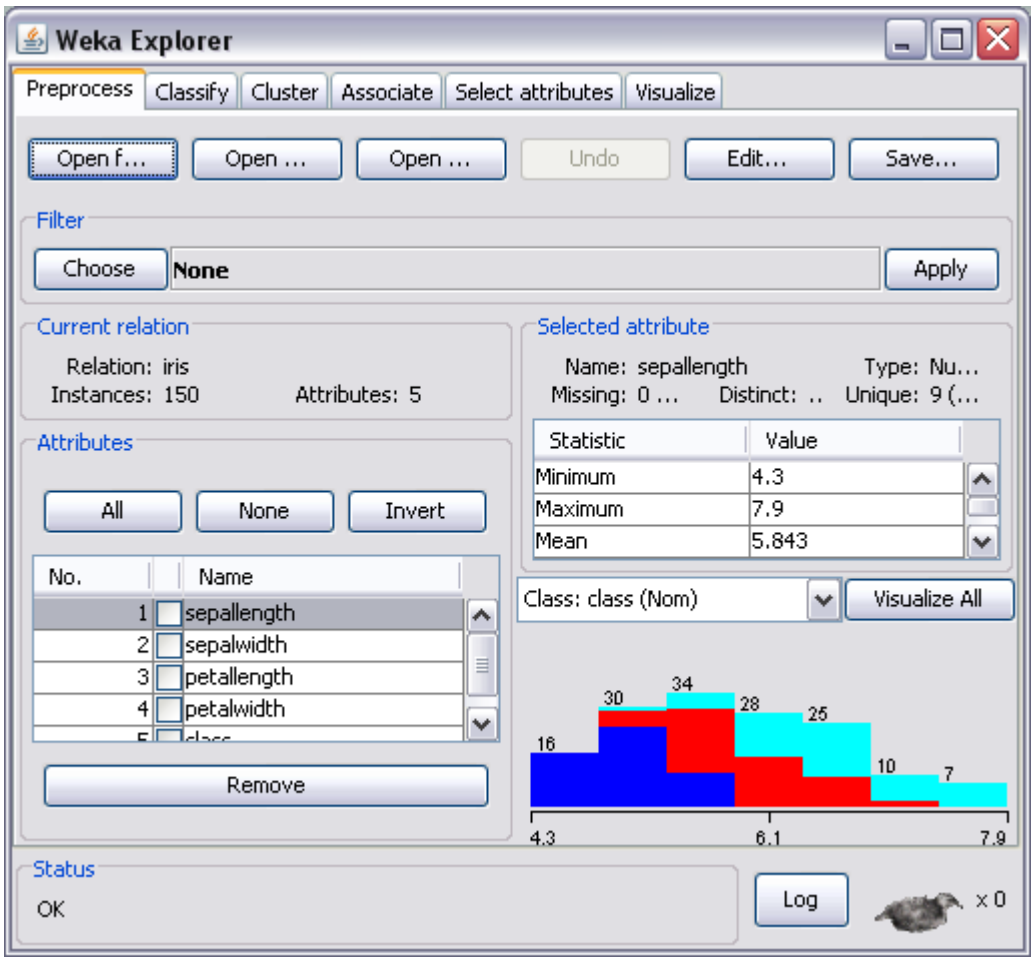
2.1.2 *Weka* sistema

Weka sistemos darbastalis turi vizualizavimo įrankius ir algoritmus, skirtus duomenų analizei ir prognozavimo modelį, kartu su patogia grafine vartotojo sąsaja. Nuo *Weka 3* versijos visa programa paremta Java kalba, kuri pradėta platinti 1997 metais ir dabar naudojama daugelyje mokslo sferų. Šios sistemos stipriosios savybės yra:

- atvirojo kodo programa (bendroji viešoji licenzija (angl. General Public License)),
- labai patogi, nes realizuota Java programavimo kalba ir todėl veikia beveik su visomis operacinėmis sistemomis,
- yra paprasta naudoti pradedančiajam vartotojui.

Weka sistema palaiko keletą standartinių duomenų mokymo uždavinių, tokių kaip, duomenų iki apdirbimas, klasterizavimas, klasifikavimas, regresija, vizualizavimas ir savybių (parametrų, požymių) atrinkimas. Visos *Weka* sistemos grįstos prielaida, kad duomenys yra galimi kaip vienas failas, kur kiekvienas duomuo yra aprašytas fiksuotu parametrų skaičiumi. *Weka* suteikia galimybę jungtis prie SQL duomenų bazės naudojant Java Database Connectivity ir gali rezultatus pateikti naudojant duomenų bazės užklausas.

Weka sistemos pagrindinė vartotojui skirta sąsaja yra *Explorer*, bet iš tikrųjų su tomis pačiomis savybėmis galima prisijungti ir per komponentėmis paremtą versiją *Knowledge Flow* ir komandų juostą. Taip pat yra *Experimenter* vartotojo sąsaja, kur leidžia sistemingai palyginti *Weka* sistemos duomenų aibę per mašininius apmokymo algoritmus.



2. 2 pav. Weka sistemos Explorer pagrindinis langas

Explorer vartotojo sąsaja turi keletą kortelių (žr. 2.2 pav.), kurios suteikia mums galimybę prisijungti prie pagrindinių darbastalio komponenčių. *Preprocess* kortelė turi priemones duomenų nuskaitymui iš duomenų bazės, CSV failo ir kt., taip pat duomenų filtravimui, naudojamas *filtering* algoritmas. Šie filtrai gali būti naudojami pakeisti duomenis ir suteikia galimybę ištrinti analizuojamus duomenų objektus ar parametrus remiantis specialiais kriterijais. *Classify* kortelėje leidžiama vartotojui patvirtinti klasifikavimo ar regresijos algoritmus atvaizduojančius duomenis, apskaičiuoti tikslumą gauto modelio ir pateikti vizualiai duomenų apskaičiavimus, ROC kreives ir kt, arba pačius modelius (pavyzdžiui klasifikavimo medžius). *Associate* kortelėje leidžiama prisijungti prie bendrų taisyklių apmokymų kur leidžia identifikuoti visus svarbius tarpusavio ryšius tarp parametrų duomenyse. *Cluster* kortelėje suteikiama galimybė prisijungti prie klasterizavimo metodų *Weka* sistemoje. Kita kortelė yra *Selecte attributes* leidžia algoritmams identifikuoti pačius svarbiausius (labiausiai lemiančius) parametrus duomenų aibėje. Pati paskutinė kortelė *Visualize* parodo taškinių grafikų Dekarto koordinačių sistemoje matricą, kur kiekvienas grafikas gali būti pažymėtas ir padidintas, analizuojamas naudojant įvairius pasirinkimo operatorius. [10]

2.2 Analizuojami duomenys

Analizavimui naudojami *Fišerio irisų* duomenys, kurie kartais vadinami tiesiog irisais arba irisų duomenimis. Duomenys yra vieni iš klasikinių tekstinių duomenų, naudojamų daugiamačių duomenų analizei. Yra išmatuota 150-ies gėlių – irisų:

- taurėlapių pločiai (angl. *sepal width*)
- taurėlapių ilgiai (angl. *sepal length*)
- vainiklapių pločiai (angl. *petal width*)
- vainiklapių ilgiai (angl. *petal length*)

Matuotos trijų veislių gėlės: *Iris Setosa* (I klasė), *Iris Versicolor* (II klasė) ir *Iris Virginica* (III klasė). Sudaryti 4-mačiai ($n = 4$) vektoriai $X_1, X_2, ..., X_{150}$ ($D_i = (x_{i1}, x_{t2}, x_{i3}, x_{i4})$, $i = 1,...,150$). Įvairiais metodais nustatyta, kad I klasės irisai nuo kitų skiriasi nuo kitų dviejų (II ir III) klasių, o pastarųjų – labiau giminingi. Žinome, kas po 50 duomenų yra kiekvienoje klasėje. Vadinasi *Iris Setosa* yra 50, *Iris Versicolor* – 50, *Iris Virginica* – 50. Priede Nr. 1 pateikiami irisų duomenys. [7]

Kiti analizuojami duomenys buvo vynų (angl. wine), kuriuos kaip ir irisus galima rasti duomenų bazėje „UCI Repository of Machine Learning Databases“ (http://archive.ics.uci.edu/ml/). Duomenys buvo surinkti tiriant 178 vynų rūšių iš vieno Italijos regiono cheminę analizę. Buvo tiriama 13 vyno komponentų (tokių kaip skonis, kvapas, cheminė sudėtis ir kt.) rastų trijose skirtingose vynų rūšyse. Kiekvienas iš komponentų (toliau vadinsime parametrais) buvo pažymėtas *A1*, *A2*, ..., *A11*, *A13*. Vadinasi yra 178 13-mačiai vektoriai $X_1, X_2, ..., X_{178}$. Žinoma, kad yra 3 vynų rūšys, kur pirmajai priklauso 59, antrajai 71 ir trečiajai 48 duomenys. Priede Nr. 2 pateikiami vynų duomenys. [1]

2.3 Irisų duomenų klasifikavimo rezultatai

2.3.1 *Orange Canvas* sistema

Pirmiausia prieš atliekant klasifikavimą, duomenys yra nuskaitomi (file). Norint įsitinkinti ar gerai duomenys tai buvo atlikta, galima „išsitraukti“ *Data Table* įrankį, jį sujungti su *File* įrankiu ir pažiūrėti ar gerai suformuota duomenų aibė, kuri bus pateikta klasifikavimo analizei. Toliau seka klasifikatoriaus apmokymas su CN2 ir C4.5 algoritmais. Norint nustatyti sukurto klasifikatoriaus tikslumą, galima tai padaryti „ištraukus“ įrankį *Test Learners* ir aktyvavus jį (žr. 2.3 pav.).

Evaluation Results

	Classifier	CA	Sens	Spec
1	C4.5	0.9533	0.9800	1.0000
2	CN2 rules	0.9333	1.0000	0.9400

2. 3 pav. Klasifikavimo tikslumas irisų duomenims

2.3 pav. *klasifikatorius* (angl. Classifier) – nurodomas koks klasifikatorius naudojamas klasifikuojant duomenis. Šiuo atveju naudojami C4.5 ir CN2 taisyklių (angl. rules) klasifikatoriai.

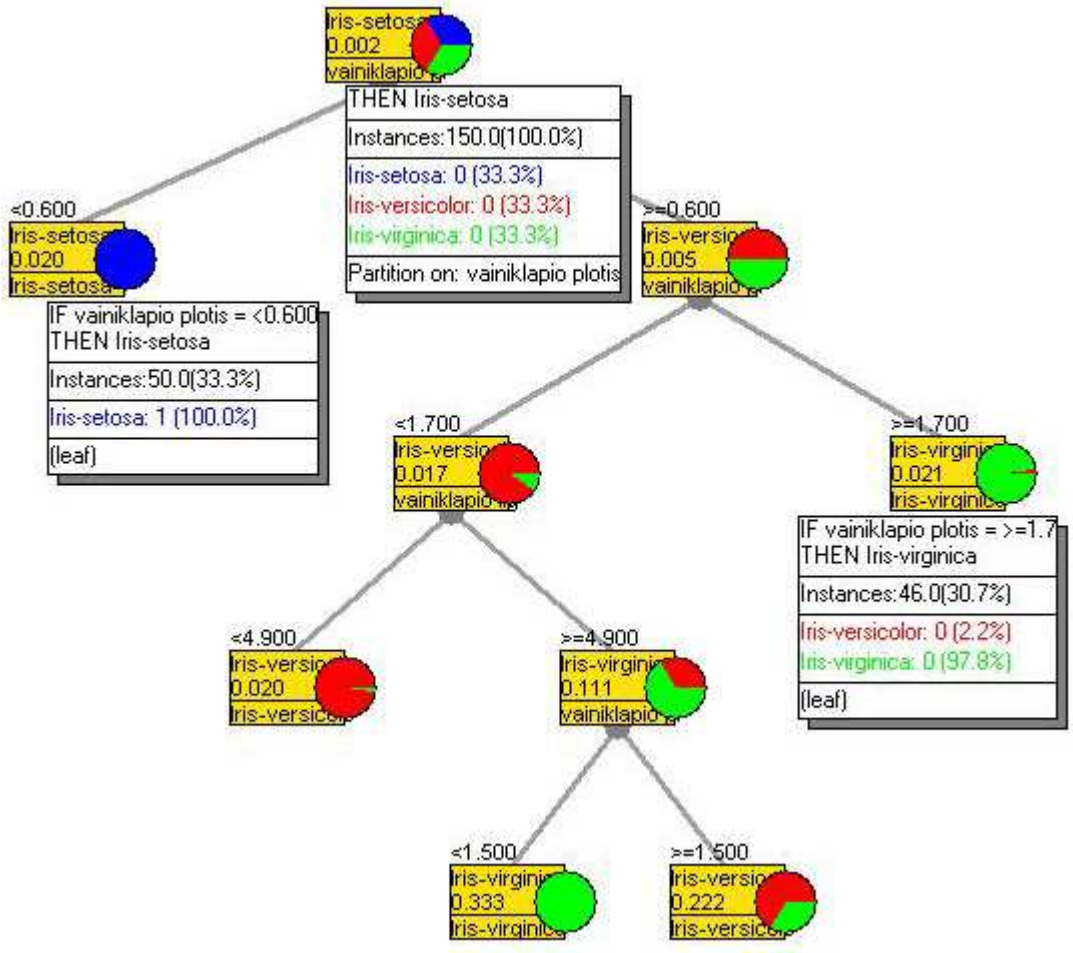
CA (angl. clasification accuaracy) yra save mokančiose sistemose ir žinių gavyboje klasifikavimo kokybės įvertinimui dažniausiai naudojamas matas vad. klasifikavimo *tikslumu.*

Jautrumas (angl. Sens) parodo santykį tarp teigiamo objekto ir klasifikatoriaus priskirto nurodytai klasei, tai pat vadinamo *tikrai teigiamu (TT)* skaičius dalintas iš tų pačių teigiamų objektų ir klasifikatoriaus priskirto nurodytai klasei skaičius objektų (TT) sudėto su teigiamais objektais priskirtais klasifikatoriaus klaidingai klasei, dar vadinamas *klaidingai neigiamais (KN)* skaičiaus. Jautrumas apskaičiuojamas formule (1.27).

Specifiškumas (angl. Spec) yra santykis tarp jeigu neigiamas objektas klasifikatoriaus nepriskirtas nurodytai klasei, taip pat vadinamas *tikrai neigiamu (TN)* dalytas iš neigiamų objektų klasifikatoriaus nepriskirtų klasei (TN) sudėto su neigiamais objektais, klasifikatoriaus priskirtais tiriamajai klasei dar vadinamai *klaidingai teigiamais (KT)*. Specifiškumas paskaičiuojamas pasinaudojus formule (1.27).

Klasifikavimo medžio grafinis atvaizdavimas.

Naudojamas klasifikatorius: C4.5



2. 4 pav. Klasifikavimo medžio grafinis atvaizdavimas irisams

Sukurtame medyje 2.4 pav. suklasifikuoti duomenys vaizduojami apskritimo dalimi (išpjova) priskirtų vienai iš klasių, santykinė dalis. Šie apskritimai vadinami medžio lapais (angl. leaf).

Kiekviena klasė yra vaizduojama skirtingomis spalvomis. Mėlynos spalvos dalis atitinka *Iris-setosa*, *Iris-versicolor* klasė vaizduojama raudona spalva, o žalios spalvos irisai yra trečios klasės, kuri vadinasi *Iris-virginica*.

Iš pradžių turime 150 irisų ir pirmas „pyragas“ yra suskirstytas į tris lygias dalis. Kiekviena dalis nudažyta mėlyna, raudona ar žalia spalvomis. Kadangi visos „pyrago“ dalys yra lygios, tai kiekviena dalis yra lygiai po 33.3%. Pirmas klasifikavimo etapas yra atliekamas imant parametrą *vainiklapio plotis*. Viena medžio šaka atsiranda kai *vainiklapio plotis* <0.600 (iš kairės), kita šaka kai *vainiklapio plotis* >= 0.600 (atitinkamai iš dešinės).

Turime, kad kai *vainiklapio plotis* yra <0.600 lygiai 50 irisų yra priskiriama *Iris-setosa* klasei, atitinkamai apskritimas tampa mėlynas.

Parametras *Instances* rodo kiek duomenų yra padengti šia taisykle. Pavyzdžiui, jei pirmu klasifikavimo etapu teigėme kad *vainiklapio plotis* yra >=0.600, tuomet lygiai 50 irisų priskyrėme *Iris-setosa*. Vadinasi mūsų *Instances* yra 50, o kiti 100 duomenų buvo priskirti kitai šakai, kur *Instances* gauname 100. Skaidant medį toliau pagal parametrą *vainiklapio plotis* >=1.700 turime, kad į šią šaką papuola 46 duomenys, vadinasi kiti 54 yra kitoje šakoje. Prisimename, kad iš viso turėjome 100 irisų, būtent sudėję abejas šakas tiek ir gauname.

Procentai prie *Instances* parodo, kurią dalį tos šakos duomenų sudaro nuo bendro skaičiaus, tai yra nuo 150 irisų.

Klasifikavimo medžio struktūrinė peržiūra (2.5 pav).

Naudojamas klasifikatorius: C4.5

Classification Tree	Class	P(Class)	P(Target)	#Inst	Rel. distr.	Abs. distr.
⊟ <root>	Iris-setosa	33	33	150	33:33:33	0:0:0
vainiklapio plotis <0.600	Iris-setosa	100	100	50	100:0:0	1:0:0
⊟ vainiklapio plotis >=0.600	Iris-versicolor	50	0	100	0:50:50	0:1:1
⊟ vainiklapio plotis <1.700	Iris-versicolor	91	0	54	0:91:9	0:1:0
vainiklapio ilgis <4.900	Iris-versicolor	98	0	48	0:98:2	0:1:0
⊟ vainiklapio ilgis >=4.900	Iris-virginica	67	0	6	0:33:67	0:0:1
vainiklapio plotis <1.500	Iris-virginica	100	0	3	0:0:100	0:0:1
vainiklapio plotis >=1.500	Iris-versicolor	67	0	3	0:67:33	0:1:0
vainiklapio plotis >=1.700	Iris-virginica	98	0	46	0:2:98	0:0:1

2. 5 pav. Klasifikatoriaus C4.5 grafinis atvaizdavimas irisams

Iš viso turime 150 duomenų (trys irisų klasės po 50).

Sukuriame taisykles, pagal kurias irisai bus priskirti klasėms. Kaip matome iš 2.5 pav. taisyklės kuriamos pagal parametrus *vainiklapio plotis* ir *vainiklapio ilgis.*

Pirmiausia klasifikatorius sukūrė taisyklę, kad jei *vainiklapio plotis* yra <0.600, gauname, kad mūsų naudojamas klasifikatorius C4.5 iš 150 turimų duomenų, 50 irisų priskyrė klasei *Iris-Setosa*. Vadinasi mūsų klasifikatoriaus atpažinimas lygus 100%.

P(Class) nurodo procentus objektų tos klasės, kuriai taisyklė yra sukurta. Mūsų pirmoji sukurta taisyklė yra, kad *vainiklapio plotis* < 0.600. Pagal šią pirmą taisyklę gauname, kad klasifikatorius 50 irisų priskyrė klasei *Iris-Setosa*. Antroji taisyklė yra tokia – *vainiklapio plotis* >= 0.600. Klasifikatorius 50 irisų priskyrė *Iris-Versicolor* klasei, kitus 50 kitai klasei. Tačiau toks klasifikavimas nėra tikslus ir sukuriamos papildomai dar 6 taisyklės.

#Inst parametras parodo, kiek duomenų priskyrė būtent nurodytai klasei pagal nurodytą taisyklę.

Rel. Distr. duomenų išsibarstymas pagal klases.

Jei (if) tada (then) taisyklių peržiūra.

Naudojamas klasifikatoriaus: CN2

Length	Quality	Coverage	Class	Distribution	Rule
1	0.981	50.0	Iris-setosa	<50.0,0.0,0.0>	IF vainiklapio ilgis<=1.900 THEN iris=Iris-setosa
3	0.978	44.0	Iris-versicolor	<0.0,44.0,0.0>	IF vainiklapio ilgis<=4.700 AND vainiklapio ilgis>1.900 AND vainiklapio plotis<=1.600 THEN iris=Iris-versicolor
3	0.800	3.0	Iris-versicolor	<0.0,3.0,0.0>	IF vainiklapio ilgis>1.900 AND vainiklapio plotis<=1.600 AND vainiklapio ilgis<=4.900 THEN iris=Iris-versicolor
2	0.978	43.0	Iris-virginica	<0.0,0.0,43.0>	IF vainiklapio plotis>1.700 AND vainiklapio ilgis>4.800 THEN iris=Iris-virginica
2	0.750	2.0	Iris-virginica	<0.0,0.0,2.0>	IF vainiklapio ilgis>4.700 AND vainiklapio ilgis>5.100 THEN iris=Iris-virginica
3	0.750	2.0	Iris-virginica	<0.0,0.0,2.0>	IF taurelapio plotis<=2.900 AND vainiklapio plotis>1.400 AND vainiklapio plotis>1.600 THEN iris=Iris-virginica

2. 6 pav. Jei... tada... taisyklių grafinis atvaizdavimas irisams

Iš viso turime 150 duomenų (trys irisų klasės, kiekvienoje klasėje po 50 irisų).

Sukuriamos taisyklės (jei..., tada...), pagal kurias kiekvienas irisas yra priskiriamas vienai iš trijų klasių. Kaip matyti iš 2.6 pav. taisyklės kuriamos pagal parametrus *vainiklapio plotis* ir *vainiklapio ilgis*. Iš viso sukurtos šešios taisyklės.

Geru klasifikatoriumi laikomas tas, kuriame taisyklių (jei... tada...) nėra daug. Turint mokymo duomenis galima sukurti tiek taisyklių, kad jomis bus parengti visi mokymo duomenys, t.y. mokymo duomenys bus klasifikuojami 100% tikslumu. Tačiau toks klasifikatorius bus visiškai netinkamas naujiems, mokymo aibėje esantiems duomenims. [3]

Čia *Length* yra vienos taisyklės sąlygų skaičius. Kuo mažiau yra sąlygų, tuo klasifikavimas yra efektyvesnis.

Kokybė (angl. quality) mums parodo sukurtos taisyklės efektyvumą. Kuo reikšmė yra

arčiau 1, tuo pagal šią taisyklę tiksliau duomenys bus priskirti klasėms.

Persidengimas (angl. coverage) parodo, kiek irisų yra priskirta būtent tai klasei, t.y. kiek irisų bus padengti šia taisykle pagal nurodytą taisyklę.

Klasė (angl. class) parodo, kokiai klasei yra priskiriami klasifikuojami irisai.

Išsibarstymas (angl. distribution) yra grafiškas persidengimo (angl. coverage) atvaizdavimas. Žinome, kad turime iš viso 3 klases: *Iris-Setosa*, *Iris-Versicolor* ir *Iris-Virginica*. Kiekviena klasė vaizduojama skirtingomis spalvomis, *Iris-setosa* yra mėlyna, *Iris-versicolor* – raudona, o *Iris-virginica* vaizduojama žalia.

Taip pat pateikiamos sukurtos *taisyklės* (angl. rule) yra tikslios naudojamos taisyklės. Kiek yra iš viso taisyklių mums pasako duomuo *length* (liet. sąlygų skaičius). Pavyzdžiui pirmoji taisyklė sako, kad jei *vainiklapio ilgis* <= 1.900, tuomet turime, kad 50 irisų yra priskiriama pirmai klasei, tai yra *Iris-Setosa*. Taip ir yra.

2.3.2 *Weka* sistema

Iš viso turime 150 irisų, kurie priskirti vienai iš trijų klasių. Pirmoji klasė yra *Iris-setosa*, antroji – *Iris-virginica* ir trečioji klasė – *Iris-versicolor*.

Duomenims klasifikuoti *Weka* sistema, naudotasi *NBTree* klasifikatoriumi.

Gavome suformuotą klasifikavimo modelį kurio dydis yra 7 šakos ir 4 stilizuoti lapeliai arba taip vadinamos šakų viršūnės. Klasifikavimui naudoti tik du irisų parametrai:

petallength – vainiklapio ilgis

petalwidth – vainiklapio plotis

kaip ir *Orange Canvas* sistemoje naudojami klasifikavimui medžio ir CN2 taisyklių generavimo algoritmai.

```
petallength <= 2.45: NB 1
petallength > 2.45
|   petalwidth <= 1.75
|   |   petallength <= 4.95: NB 4
|   |   petallength > 4.95: NB 5
|   petalwidth > 1.75: NB 6

Leaf number: 1 Naive Bayes Classifier
          Class Iris-setosa: Prior probability = 0.96
          Class Iris-versicolor: Prior probability = 0.02
          Class Iris-virginica: Prior probability = 0.02

Leaf number: 4 Naive Bayes Classifier
          Class Iris-setosa: Prior probability = 0.02
          Class Iris-versicolor: Prior probability = 0.94
          Class Iris-virginica: Prior probability = 0.04

Leaf number: 5 Naive Bayes Classifier
          Class Iris-setosa: Prior probability = 0.11
          Class Iris-versicolor: Prior probability = 0.33
          Class Iris-virginica: Prior probability = 0.56

Leaf number: 6 Naive Bayes Classifier
          Class Iris-setosa: Prior probability = 0.02
          Class Iris-versicolor: Prior probability = 0.04
          Class Iris-virginica: Prior probability = 0.94
```

2. 7 pav. Klasifikatoriaus NBTree duomenų atvaizdavimas Weka sistemoje irisams

Duomenys klasifikuojami naudojant Naive Bayes algoritmą, o atvaizduojami kaip matome iš 2.7 pav. stilizuotu medžiu, sudarytu iš 6 šakų. Pirmiausia yra sukuriamos taisyklės, pagal kurias duomenys yra klasifikuojami.

Pirmoji taisyklė mums sako, kad *vainiklapio plotis* yra <= 2.45. Iš pačios pirmosios taisyklės gaunamas lapelis numeris vienas (angl. leaf number: 1). Toliau pateikta suvestinė parodo, kad pirmai klasei Iris-setosa tikimybė pakliūti į pirmą klasę yra 0.96. Kitų dviejų klasių *Iris-versicolor* ir *Iris-virginica* tikimybės lygios 0.02.

Tolimesnis klasifikavimas vykdomas imant parametrą *vainiklapio plotis* > 2.45, tačiau ši sukurtoji taisyklė dar skaidoma į *vainiklapio ilgis* <= 1.75 ir *vainiklapio ilgis* > 1.75. Tokiu būdu skaidant suformuojamas medis.

Teisingai suklasifikuoti objektai	141	94%
Neteisingai suklasifikuoti objektai	9	6%
Viso klasifikuojamų objektų	150	

Lentelė 4. Suklasifikuotų duomenų statistika

Lentelėje 4 pateikiama klasifikavimo statistika. Teisingai suklasifikuota duomenų yra 141, tai yra 94% nuo visų duomenų. Iš viso mes turime 150 irisų. Atitinkamai turime, kad neteisingai suklasifikuota buvo 9 irisai ir tai yra 6% nuo visų duomenų.

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	0	1	1	1	Iris-setosa
0.92	0.05	0.902	0.92	0.911	Iris-versicolor
0.9	0.04	0.918	0.9	0.909	Iris-virginica

Lentelė 5. Detalizuotas tikslumas pagal klases

Lentelėje 5 matome tokius klasifikavimo tikslumo parametrus:

TP Rate (True Positive) – tikrai teigiamas. Kuo TP Rate skaičius arčiau vieneto, tuo klasifikavimo tikslumas yra didesnis.

FP Rate (False Positive) – klaidingai teigiamas. Jei FT Rate yra arčiau nulio, galime teigti, kad klasifikavimas yra tikslesnis.

Precision – tikimybinis klasifikavimo tikslumas.

Class – nurodoma klasė. Iš viso yra 3 klasė: *Iris-setosa*, *Iris-versicolor* ir *Iris-virginica*.

a	b	c	<-- priskirti klasei
50	0	0	a = Iris-setosa
0	46	4	b = Iris-versicolor
0	5	45	c = Iris-virginica

Lentelė 6. Sumaišymo matrica

Sumaišymo matrica (lentelė 6) mums parodo kaip buvo suklasifikuoti duomenys. Iš šios matricos matome, kad 50 duomenų buvo priskirta Iris-setosa klasei. Visi šios klasės irisai buvo priskirti teisingai, nes kiti rodikliai yra nuliai. Iris-versicolor klasei buvo priskirta 46 duomenys, o kiti 4 kitai klasei – Iris-virginica. Galime sakyti, kad 4 duomenys buvo priskirti neteisingai. Žinome, kad kiekvienoje klasėje yra lygiai po 50 duomenų. Paskutinei klasei priskiriami 45 irisai, kiti likę 5 neteisingai suklasifikuojami ir priskiriami klasei Iris-virginica.

2.4 Vynų duomenų klasifikavimo rezultatai

2.4.1 *Orange Canvas* sistema

Analizuotos trys vynų rūšys (klasės). Kiekvieną vyną charakterizuoja 13 parametrų *A1*, *A2*, ..., *A13*.

Pirmiausia prieš atliekant klasifikavimą, vynų duomenys yra nuskaitomi (file). Norint įsitinkinti ar gerai duomenys tai buvo atlikta, galima „išsitraukti“ *Data Table* įrankį, jį sujungti su *File* įrankiu ir pažiūrėti ar gerai suformuota duomenų aibė, kuri bus pateikta klasifikavimo analizei. Toliau seka klasifikatoriaus apmokymas su CN2 ir C4.5 algoritmais. Norint nustatyti sukurto klasifikatoriaus tikslumą, galima tai padaryti „ištraukus“ įrankį *Test Learners* ir aktyvavus jį (žr. 2.8 pav.).

Evaluation Results

	Classifier	CA	Sens	Spec
1	C4.5	0.8940	0.9153	0.9412
2	CN2 rules	0.9211	0.8983	0.9832

2. 8 pav. Klasifikavimo tikslumas vynų duomenims

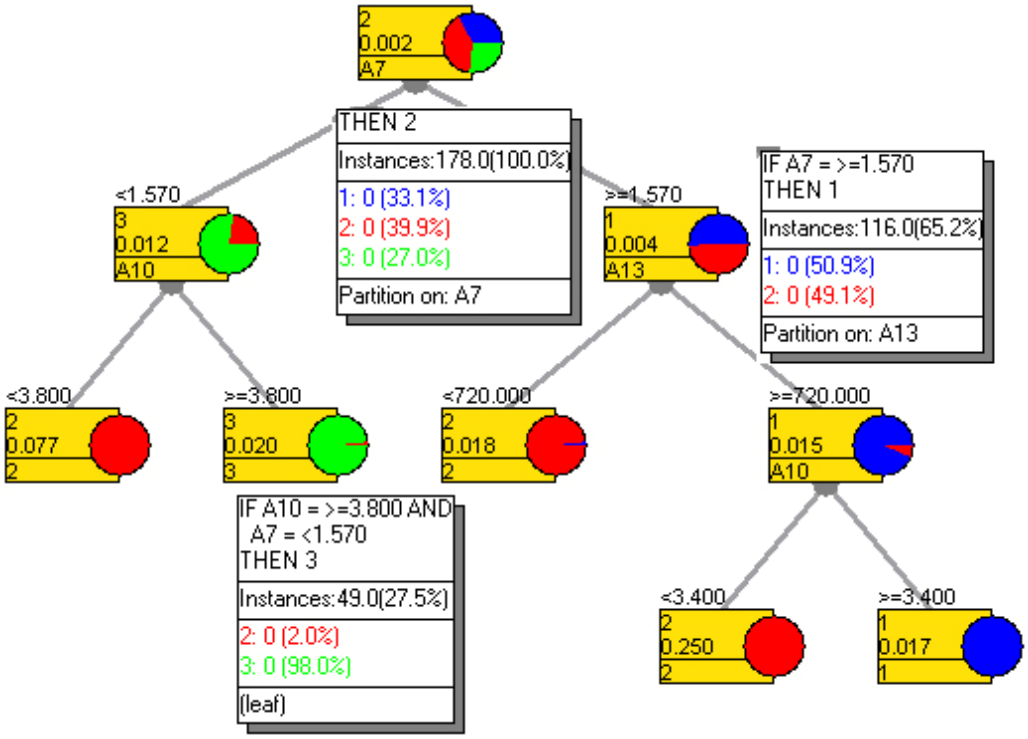
2.8 pav. *klasifikatorius* (angl. Classifier) – nurodomas koks klasifikatorius naudojamas klasifikuojant duomenis. Šiuo atveju naudojami C4.5 ir CN2 taisyklių (angl. rules) klasifikatoriai. *CA* (clasification accuaracy) yra save mokančiose sistemose ir žinių gavyboje klasifikavimo kokybės įvertinimui dažniausiai naudojamas matas vad. klasifikavimo *tikslumu.*

Jautrumas (angl. Sens) parodo santykį tarp teigiamo objekto ir klasifikatoriaus priskirto nurodytai klasei, tai pat vadinamo *tikrai teigiamu (TT)* skaičius dalintas iš tų pačių teigiamų objektų ir klasifikatoriaus priskirto nurodytai klasei skaičius objektų (TT) sudėto su teigiamais objektais priskirtais klasifikatoriaus klaidingai klasei, dar vadinamas *klaidingai neigiamais (KN)* skaičiaus. Jautrumas apskaičiuojamas pasinaudojus formule (1.28).

Specifiškumas (angl. Spec) yra santykis tarp jeigu neigiamas objektas klasifikatoriaus nepriskirtas nurodytai klasei, taip pat vadinamas *tikrai neigiamu (TN)* dalytas iš neigiamų objektų klasifikatoriaus nepriskirtų klasei (TN) sudėto su neigiamais objektais, klasifikatoriaus priskirtais tiriamajai klasei dar vadinamai *klaidingai teigiamais (KT)*. Specifiškumas paskaičiuojamas pasinaudojus formule (1.27).

Klasifikavimo medžio grafinis atvaizdavimas.

Naudojamas klasifikatoriaus: C4.5



2. 9 pav. Klasifikavimo medžio grafinis atvaizdavimas vynams

Sukurtame medyje 2.9 pav. suklasifikuoti duomenys vaizduojami apskritimo dalimis. Šie apskritimai vadinami medžio lapais (angl. leaf). Iš viso turime 5 apskritimus, kuriuos galime vadinti medžio lapais ir 3 pagrindinius apskritimus (angl. root). Visi apskritimai yra sujungti medžio šakomis.

Kiekviena klasė yra vaizduojama skirtingomis spalvomis. Pirmos klasės vynai yra vaizduojami mėlyna spalva, antros klasės – raudona, trečios klasės – žalia.

Iš pradžių turime 178 vynų duomenų ir pirmasis „pyragas“ yra suskirstomas į tris dalis. Kiekviena dalis yra nudažyta mėlyna, raudona ar žalia spalvomis. Pirma klasė užims 33.1% apkritimo, antroji klasė – 39.9%, o trečioji – 27.0%. Pirmas klasifikavimo etapas yra pradedamas imant parametrą *A7*. Viena medžio šaka atsiranda kai *A7* < 1.570 (iš kairės), o kita šaka sukuriama kai parametras *A7* => 1.570 (atitinkamai iš dešinės).

Turime, kad kai *A7* < 1.570 iš bendros skaičiaus vynų, t.y. 178 atskiriami lygiai 62 vynai, kurie priskiriami 3 klasei. Pagal klasifikavimo taisyklę turime, kad tik 77,6% vynų yra suklasifikuojami teisingai. Mums reikalingas tolimesnis medžio kūrimas. Toliau kuriame taisykles su parametrais *A7* ir *A10*. Pirmu atveju jei *A10* < 3.800 ir *A7* < 1.570, tuomet 13 vynų yra priskiriama 2 klasei. Šiuo atveju mūsų klasifikatoriaus atpažino 100%. Šita šaka buvo kuriama kai *A10* => 3.800 ir *A7* < 1.570. Turime, kad lygiai 49 vynai yra priskiriami 3 klasei, tačiau vėl mūsų klasifikatoriaus nebuvo tikslus ir atpažinimas siekė 98%.

Kitose šakose klasifikavimas atliekamas analogiškai.

Klasifikavimo medžio struktūrinė peržiūra.

Naudojamas klasifikatorius: C4.5

Classification Tree	Class	P(Class)	P(Target)	#Inst	Rel. distr.	Abs. distr.
⊟ <root>	2	40	33	178	33:40:27	0:0:0
⊟ A7 <1.570	3	77	0	62	0:23:77	0:0:1
A10 <3.800	2	100	0	13	0:100:0	0:1:0
A10 >=3.800	3	98	0	49	0:2:98	0:0:1
⊟ A7 >=1.570	1	51	51	116	51:49:0	1:0:0
A13 <720.000	2	98	2	54	2:98:0	0:1:0
⊟ A13 >=720.000	1	94	94	62	94:6:0	1:0:0
A10 <3.400	2	100	0	4	0:100:0	0:1:0
A10 >=3.400	1	100	100	58	100:0:0	1:0:0

2. 10 pav. Klasifikatoriaus C4.5 grafinis atvaizdavimas vynams

Iš viso turime 178 duomenų (trys vynų klasės, kur pirmai priklauso 59 duomenys, antrai 71 ir trečiajai 48 duomenys).

Sukuriame taisykle, pagal kurias vynai bus priskirti klasėms. Kaip matoma 2.10 pav. taisyklės kuriamos pagal parametrus *A7*, *A10* ir *A13*.

Pirmiausia klasifikatoriaus sukūrė taisyklę, kad jei *A7* < 1.570, gauname, kad mūsų naudojamas klasifikatorius C4.5 iš visų 178 duomenų, 67 vynus priskyrė trečiai klasei. Tačiau mūsų klasifikatoriaus atpažinimas tik 77%. Reikia tolimesnio taisyklių kūrimo. Sukuriama papildoma dar viena taisyklė su parametru *A10*. Šiuo atveju turime, kad jei *A10* < 3.800 antrai klasei priskiriama 13 vynų, atpažinimas 100%. Kai *A10* => 3.800 lygiai 49 vynus priskyrė 3 klasei, atpažinimas yra 98%. Analogiškai kuriamos kitos taisyklės ir visi 178 vynai yra suklasifikuojami tiksliau arba mažiau tiksliai (priklausomai nuo klasifikatoriaus ir parametrų).

P(Class) nurodo procentus objektų tos klasės, kuriai taisyklė buvo sukurta.

#Inst (Instances) parametras parodo, kiek duomenų priskyrė būtent nurodytai klasei pagal nurodytą taisyklę.

Rel. Distr. vaizduojamas duomenų išsibarstymas procentais pagal klases.

Jei (if) tada (then) taisyklių peržiūra.

Naudojamas klasifikatoriaus: CN2

Length	Quality	Coverage	Class	Distribution	Rule
2	0.979	45.0	1	<45.0,0.0,0.0>	IF A13.000>750.000 AND A1.000>13.360 THEN Wine=1.000
3	0.933	13.0	1	<13.0,0.0,0.0>	IF A13.000>725.000 AND A7.000>2.140 AND A10.000>3.350 THEN Wine=1.000
2	0.667	1.0	1	<1.0,0.0,0.0>	IF A4.000<=18.100 AND A2.000>3.430 THEN Wine=1.000
1	0.982	55.0	2	<0.0,55.0,0.0>	IF A10.000<=3.400 THEN Wine=2.000
2	0.909	9.0	2	<0.0,9.0,0.0>	IF A5.000<=87.000 AND A12.000>1.860 THEN Wine=2.000
1	0.800	3.0	2	<0.0,3.0,0.0>	IF A1.000<=11.760 THEN Wine=2.000
3	0.833	4.0	2	<0.0,4.0,0.0>	IF A2.000<=1.360 AND A6.000>2.000 AND A12.000<=2.770 THEN Wine=2.000
2	0.976	39.0	3	<0.0,0.0,39.0>	IF A11.000<=0.780 AND A10.000>3.940 THEN Wine=3.000
2	0.909	9.0	3	<0.0,0.0,9.0>	IF A7.000<=0.920 AND A3.000>1.360 THEN Wine=3.000

2. 11 pav. Jei... tada... taisyklių grafinis atvaizdavimas vynams

Iš viso turime 178 duomenų (trys klasės, kur pirmai priklauso 59 vynai, antrai 71 ir paskutinei trečiai klasei 48 duomenys). Taisyklės kuriamos pagal parametrus *A13*, *A1*, *A7*, *A10*, *A4*, *A2*, *A5*, *A12*, *A6*, *A11*, *A3*.

Čia *Length* yra vienos taisyklės sąlygų skaičius. Kuo mažiau yra sukuriama taisyklių, tuos klasifikavimas yra efektyvesnis.

Kokybė (angl. quality) informuoja apie sukurtos taisyklės efektyvumą. Kuo reikšmė arčiau 1, tuo pagal šią taisyklę tiksliau duomenys bus priskirti klasėms.

Persidengimas (anlg. coverage) parodo, kiek vynų yra priskirta būtent tai klasei, t.y. kiek vynų bus padengti šia taisykle pagal nurodytą taisyklę.

Klasė (angl. class) parodo, kokiai klasei yra priskiriami klasifikuojami vynai. Žinome, kad turime 3 vynų klases.

Išsibarstymas (angl. distribution) yra grafiškas persidengimo atvaizdavimas. Žinome, kad turime 3 klases. Kiekviena klasė vaizduojama skirtingomis spalvomis, pirma klasė yra mėlyna, antra klasė raudona ir trečioji klasė žalia.

Taip pat pateikiamos sukurtos *taisyklės* (angl. rule). Kiek iš viso yra taisyklių mums nurodo duomuo *length (*liet. sąlygų skaičius). Pavyzdžiui pirmoji taisyklė sako, kad jei *A13* yra

daugiau už 750.000 ir *A1* yra daugiau už 13.360, gauname, kad 45 vynai yra priskiriami pirmajai klasei.

2.4.2 *Weka* sistema

Iš viso turime 178 vynų duomenis, kurie sudaro tris klases. Kiekvienoje klasėje yra po skirtingą skaičių duomenų, vadinasi turime, kad pirmoje klasėje yra 59 vynai, antroje 71, o paskutiniojoje yra lygiai 48 duomenys.

Duomenims klasifikuoti Weka sistema naudotasi *NBTree* klasifikatoriumi.

Gauname suformuluotą klasifikavimo medį, kurį sudaro 9 šakos su 5 viršūnėmis, kurie vadinami lapeliais (angl. leaves).

Klasifikavimui naudoti parametrai žymimi nuo A1 iki A13. Kiekvienas iš tų parametrų pagal kuriuos buvo tirti vynai (pavyzdžiui skonis, cheminė sudėtis, spalva ir kt.) apibūdiną vyną, kuris suskirstomas į tris klases.

```
A10 <= 3.46: NB 1
A10 > 3.46
|   A7 <= 1.58
|   |   A2 <= 1.89: NB 4
|   |   A2 > 1.89: NB 5
|   A7 > 1.58
|   |   A2 <= 1.3: NB 7
|   |   A2 > 1.3: NB 8

Leaf number: 1 Naive Bayes Classifier
            Class 1: Prior probability = 0.02
            Class 2: Prior probability = 0.97
            Class 3: Prior probability = 0.02

Leaf number: 4 Naive Bayes Classifier
            Class 1: Prior probability = 0.09
            Class 2: Prior probability = 0.36
            Class 3: Prior probability = 0.55

Leaf number: 5 Naive Bayes Classifier
            Class 1: Prior probability = 0.02
            Class 2: Prior probability = 0.02
            Class 3: Prior probability = 0.96

Leaf number: 7 Naive Bayes Classifier
            Class 1: Prior probability = 0.1
            Class 2: Prior probability = 0.8
            Class 3: Prior probability = 0.1

Leaf number: 8 Naive Bayes Classifier
            Class 1: Prior probability = 0.88
            Class 2: Prior probability = 0.1
            Class 3: Prior probability = 0.01
```

2. 12 pav. Klasifikatoriaus NBTree duomenų atvaizdavimas Weka sistemoje vynams

Duomenys suklasifikuojami naudojant Naive Bayes algoritmą, o atvaizduojami stilizuotu medžiu, sudarytu iš 9 šakų (žr. 2.12 pav.). Pirmiausia yra sukuriamos taisyklės, pagal kurias duomenys yra klasifikuojami.

Pirmoji taisyklė mums sako, kad parametras *A10* yra <= 3.46. Iš pačios pirmosios taisyklės gaunamas lapelis pirmas (angl. Leaf Number 1). Toliau pateikta suvestinė parodo, kad tikimybinis pakliuvimas į antrą klasę yra 0.97, o į pirmą ir trečią klases tik 0.02.

Tolimesnis klasifikavimas vykdomas imant parametrą *A10* kuris yra > 3.46, kuris dar toliau skaldomas į dvi šakas, kur *A7* <= 1.58 ir *A7* > 1.58. Vėliau parametras *A7* skeliamas ir taip toliau, kai galiausiai gaunamos klasifikavimo tikimybės. Taip suklasifikavus gaunamos medis, kuris vadinamas klasifikavimo medžiu.

Teisingai suklasifikuoti objektai	172	96.6292 %
Neteisingai suklasifikuoti objektai	6	3.3708 %
Viso klasifikuojamų objektų	178	

Lentelė 7. Suklasifikuotų duomenų statistika

Lentelėje 7 teisingai suklasifikuota lygiai 172 duomenys, o tai sudaro 96.63% visų klasifikuojamų duomenų. Iš viso turime 178 vynų duomenis, vadinasi 6 vynai liko neteisingai suklasifikuoti. Suklasifikuotų neteisingai duomenų yra 3.37% skaičiuojant nuo bendros sumos.

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.983	0.017	0.967	0.983	0.975	1
0.944	0.019	0.971	0.944	0.957	2
0.979	0.015	0.959	0.979	0.969	3

Lentelė 8. Detalizuotas tikslumas pagal klasę

Lentelėje 8 matome tokius klasifikavimo tikslumo parametrus:

TP Rate (True Positive) – tikrai teigiamas. Kuo TP Rate skaičius arčiau vieneto, tuo klasifikavimo tikslumas yra didesnis.

FP Rate (False Positive) – klaidingai teigiamas. Jei FT Rate yra arčiau nulio, galime teigti, kad klasifikavimas yra tikslesnis.

Precision – tikimybinis klasifikavimo tikslumas.

Class – nurodoma klasė. Žinome, kad iš viso yra trys klasės, kurios numeruojamos 1, 2 ir 3.

a	b	c	<-- priskirti klasei
58	1	0	a = 1
2	67	2	b = 2
0	1	47	c = 3

Lentelė 9. Sumaišymo matrica

Sumaišymo matrica (žr. Lentelė 9) mums parodo kaip buvo suklasifikuoti duomenys. Iš šios matricos matome, kad 58 duomenys buvo priskirti pirmai klasei, o vienas duomuo antrai klasei. Trečiai klasei buvo nieko nepriskirtą ką ir rodo esantis nulis. Antroje klasėje 67 duomenys suklasifikuoti teisingai, o likę 4 po du priskirti pirmai ir antrai klasėms. Paskutinėje, trečiojoje klasėje, yra 47 vynų duomenys suklasifikuoti teisingai ir 1 vynas, kuris priskirtas antrai klasei.

2.5 Rezultatų apibendrinimas ir išvados

Šiame skyriuje buvo lyginti sistemų *Orange Canvas* (toliau *Orange*) ir *Weka 3.4* (toliau *Weka*) sistemų vaizdžiai pateikti suklasifikuoti duomenys. Buvo klasifikuoti irisų ir vynų duomenų aibės ir lyginti gautas klasifikavimo rezultatų grafines iliustracijas. Ir sistemai *Orange*, ir *Weka* buvo pateikti tie patys irisų ir vynų duomenys. *Orange* sistemoje naudoti trys klasifikavimo rezultatų atvaizdavimo būdai, o *Weka* tik vieną. Orange sistemoje naudojome klasifikavimo medžio metodą ir taisyklių generavimo metodą, o *Weka* sistemoje pasirinkę vieną atvaizdavimo būdą, naudojome ir vieną NBTree klasifikatorių. *Orange* sistemoje gavome po tris spalvotas suklasifikuotas duomenų iliustracijas ir vieną *Weka* sistemoje.

Orange sistemoje suklasifikuoti daugiamačiai duomenys atvaizduojami spalvotomis, vaizdžiomis iliustracijomis. Spalvomis išskiriama kiekviena klasė. Spalvotas iliustracijas yra lengviau suprasti. Taip pat *Orange* sistema suteikia galimybę to paties klasifikatoriaus suklasifikuotus duomenis pažiūrėti keliais skirtingais atvaizdavimo būdais. Pavyzdžiui, klasifikatorius C4.5 mums pateikia klasifikavimo medžio grafinį atvaizdavimą ir struktūrinę peržiūrą. Klasifikavimo medžio grafinio atvaizdavimo (2.4 ir 2.9 pav.) šakų skaičių galima keisti. Tokią galimybę mums suteikia *Orange* sistema. Tai pat lengvai brėžinį galima sumažinti ar padidinti, išplėsti ar suspausti. Kiekvienoje iliustracijoje iš karto pateikiami ir minimalūs duomenys apie klasifikavimo tikslumą, tai naudinga norint iš karto pamatyti ar klasifikatorius efektyviai apmokomas.

Weka sistemoje suklasifikuoti duomenys vizualiai atvaizduojami „skurdžiai“. Sukuriamas medis, tačiau tolimesnis jo tyrimas priklauso nuo vartotojo įgūdžių klasifikavime. Weka pateikia rezultatus tik tekstiniu pavidalu, vadinasi vaizdinius reikia kurti galvoje. Kiekvienas žingsnis smulkiai aprašomas, tačiau iš karto pažiūrėjus į medį informacijos galima gauti mažai.

Lyginant *Orange* ir *Weka* sistemos daugiamačių klasifikavimo rezultatų pavaizdavimą vizualiomis priemonėmis, *Orange* gerokai lenkia *Weka*. Ne tik, kad schemos lengvai suprantamos, bet ir lengviau interpretuojamos. Be to, *Orange* tuo pačiu klasifikatoriumi suklasifikuotus duomenis gali pavaizduoti keliais būdais.

Tačiau pati *Orange* sistema turi ir trūkumų. *Orange* sistemą neįgudusiam vartotojui sunku suprasti. *Weka* sistemoje reikia tik atidaryti daugiamačius duomenis, paskui pasirinkti klasifikatorių pagal kurį bus klasifikuojami duomenys ir įvykdyti klasifikavimą. O *Orange* sistema veikia visai kitaip. Šioje sistemoje reikia dėlioti įrankius. Sudėjus keletą paskui jas sujungti. Neturint bent jau minimalių žinių apie programą, pradedančiajam yra sunku susigaudyti ką su kuo jungti. Tam, kad duomenys būtų atvaizduoti vizualiai, dar reikia sujungti daugiamačius turimus duomenimis su vizualizacijos priemonėmis. Tiesa, pati *Orange* yra gerokai išplėsta su daugiau galimybių.

3. DAUGIAMAČIŲ SKALIŲ METODAS

3.1 Daugiamačių skalių metodas

Daugiamačių skalių (DS) (*multidimensional scaling*, MDS) metodas – tai grupė metodų, plačiai naudojamų daugiamačių duomenų analizei įvairiose šakose, ypač ekonomikoje, socialiniuose moksluose ir kt. Naudojantis DS, ieškoma daugiamačių duomenų projekcijų mažesnio skaičiaus matmenų erdvėje (dažniausiai į R^2), siekiant išlaikyti analizuojamos aibės objektų panašumus [2]. Gautuose vaizduose panašūs objektai išdėstomi arčiau vieni kitų, o mažiau panašūs – toliau vieni nuo kitų.

Šiuo metodu analizuojami duomenys yra kvadratinė simetrinė matrica, kurios elementai yra analizuojamų duomenų aibės elementų ryšiai. Tai gali būti arba panašumų arba skirtingumų matrica. Ryšiais tarp aibės elementų gali būti Euklido atstumai. Tačiau bendruoju atveju, tai nebūtinai turi būti atstumai griežtai matematine prasme.

Vienas DS tikslų yra rasti optimalią daugiamačių duomenų konfigūraciją mažo skaičiaus matmenų erdvėje. Yra daugybė skirtingų DS variantų su skirtingomis paklaidų funkcijomis ir jas optimizuojančiais algoritmais [2].

Tarkime, kiekvieną *n*-matį vektorių $X_i \in R^n$, $i \in \{1,...,m\}$, atitinka mažesnio skaičiaus matmenų vektorius $Y_i \in R^d$, $d < n$. Atstumą tarp vektorių X_i ir X_j pažymėkime $d(X_i, X_j)$, o atstumą tarp vektorių Y_i ir Y_j – $d(Y_i, Y_j)$, $i, j = 1,...,m$. Naudojantis DS algoritmu, bandoma atstumus $d(Y_i, Y_j)$ priartinti prie atstumų $d(X_i, X_j)$. Jei naudojama kvadratinė paklaidos funkcija, tai minimizuojama tikslo funkcija E_{MDS} gali būti užrašyta taip:

$$E_{DS} = \sum_{i<j} w_{ij} \left(d(X_i, X_j) - d(Y_i, Y_j) \right)^2 . \qquad (2.5)$$

Paklaidos funkcija E_{DS} dar vadinama *Stress* funkcija. Dažnai naudojami tokie svoriai w_{ij}:

$$w_{ij} = \frac{1}{\sum_{k<l} \left(d(X_k, X_l) \right)^2},$$

$$w_{ij} = \frac{1}{d(X_i, X_j)} \sum_{k<l} d^{-1}(X_k, X_l),$$

$$w_{ij} = \frac{1}{m d(X_k, X_l)}.$$

3.2 Daugiamačių skalių metodas irisų duomenims

Turime 150 4-mačių irisų vektorių. Kiekvienas vektorius susideda iš 4 parametrų, tokių kaip taurėlapio plotis, taurėlapio ilgis, vainiklapio plotis ir vainiklapio ilgis. Kiekvieną vektorių nusako šie keturi parametrai. Žinome, kad 50 duomenų priklauso pirmai klasei, vadinamai *Iris-setosa*, kiti 50 duomenų – antrai klasei, *Iris-versicolor*, o likę 50 duomenų priskiriami trečiajai klasei – *Iris-virginica*.

Mūsų pirmoji užduotis yra 4-mačius vektorius, pasitelkus daugiamačių skalių metodą, transformuoti į 2-mačius vektorius. Matmenų erdvės keitimą iš R^4 į R^2 atliekame *Orange Canvas* sistema. Atlikę transformavimą gauname duomenis, kurie pateikiame Priede Nr. 3.

Kitas žingsnis yra pasinaudojus jau seniau nagrinėtu CN2 klasifikatoriumi ir jo sukurtomis šešiomis taisyklėmis atrinkti duomenis kiekvienai tai taisyklei. Vadinasi turime šešias taisykles:

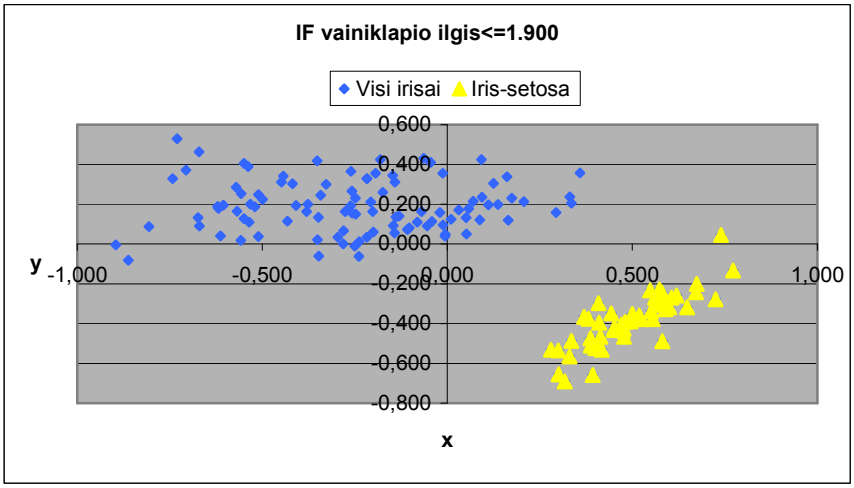
1. **Jeigu** vainiklapio ilgis $\leq$ 1.900 **tada** iris = *Iris-setosa*
2. **Jeigu** vainiklapio ilgis $\leq$ 4.700 **ir** vainiklapio ilgis $>$ 1.900 **ir** vainiklapio plotis $\leq$ 1.600 **tada** iris = *Iris-versicolor*
3. **Jeigu** vainiklapio ilgis $>$ 1.900 **ir** vainiklapio plotis $\leq$ 1.600 **ir** vainiklapio ilgis $\leq$ 4.900 **tada** iris = *Iris-versicolor*
4. **Jeigu** vainiklapio plotis $>$ 1.700 **ir** vainiklapio ilgis $>$ 4.800 **tada** iris = *Iris-virginica*
5. **Jeigu** vainiklapio ilgis $>$ 4.700 **ir** vainiklapio ilgis $>$ 5.100 **tada** iris = *Iris-virginica*
6. **Jeigu** taurėlapio plotis $\leq$ 2.900 **ir** vainiklapio plotis $>$ 1.400 **ir** vainiklapio plotis $>$ 1.600 **tada** iris = *Iris-virginica*

Iš jau klasifikatoriumi CN2 suklasifikuotų duomenų žinome, kiek unikalių duomenų turėsime. Lentelėje pavaizduota kiekvienos sukurtos taisyklės numeris ir tą taisyklę atitinkančių duomenų skaičius kiekvienai klasei.

Nr.	Taisyklė	Duomenų skaičius	Klasė
1	Jeigu vainiklapio ilgis $\leq$ 1.900	50	Iris-setosa
2	Jeigu vainiklapio ilgis $\leq$ 4.700 ir vainiklapio ilgis $>$ 1.900 ir vainiklapio plotis $\leq$ 1.600	44	Iris-versicolor
3	Jeigu vainiklapio ilgis $>$ 1.900 ir vainiklapio plotis $\leq$ 1.600 ir vainiklapio ilgis $\leq$ 4.900	3	Iris-versicolor
4	Jeigu vainiklapio plotis $>$ 1.700 ir vainiklapio ilgis $>$ 4.800 tada	43	Iris-virginica
5	Jeigu vainiklapio ilgis $>$ 4.700 ir vainiklapio ilgis $>$ 5.100	2	Iris-virginica
6	Jeigu taurėlapio plotis $\leq$ 2.900 ir vainiklapio plotis $>$ 1.400 ir vainiklapio plotis $>$ 1.600 tada	2	Iris-virginica

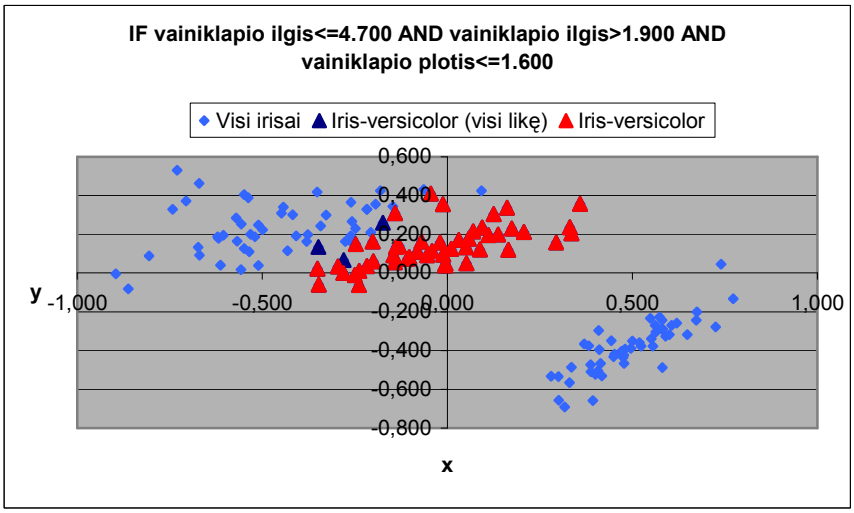
Lentelė 10. Suklasifikuotų duomenų skaičius kiekvienai sukurtai taisyklei

Galiausiai visas šias taisykles turime vaizdžiai pavaizduoti. Kiekvienai taisyklei nubraižome atskirą grafiką Dekarto koordinačių sistemoje atvaizduodami kiekvieną klasę. Tokiu būdu gauname šešis atskirus brėžinius. Dabar kiekvieną grafiką panagrinėsime plačiau.



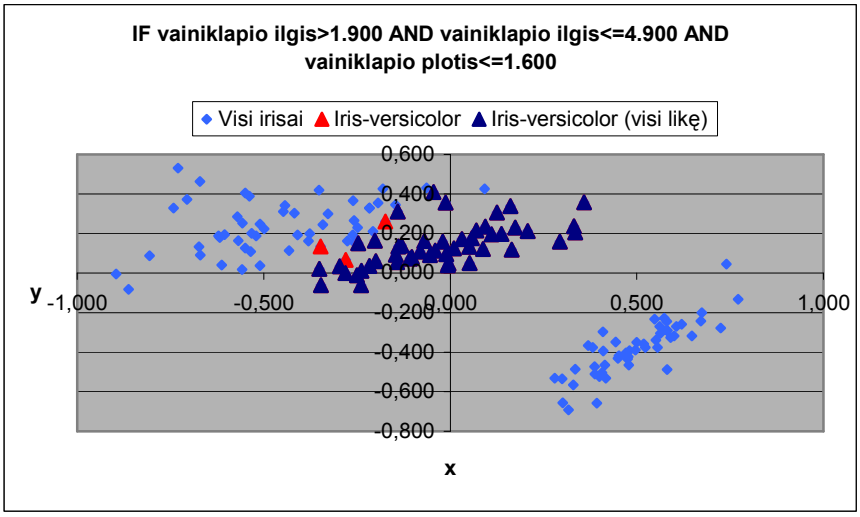
3. 1 pav. Grafikas Nr 1. taisyklei iš Lentelės 10

3.1 pav. pagal pirmąją sukurtą taisyklę gavome, kad 50 irisų (t.y. visi tos klasės atstovai) priklauso pirmajai klasei. Pirmosios klasės irisai vadinami *Iris-setosa* ir jie nudažyti ryškiai geltona spalva. Visi kiti likę irisai yra žymimi šviesiai mėlyna spalva.



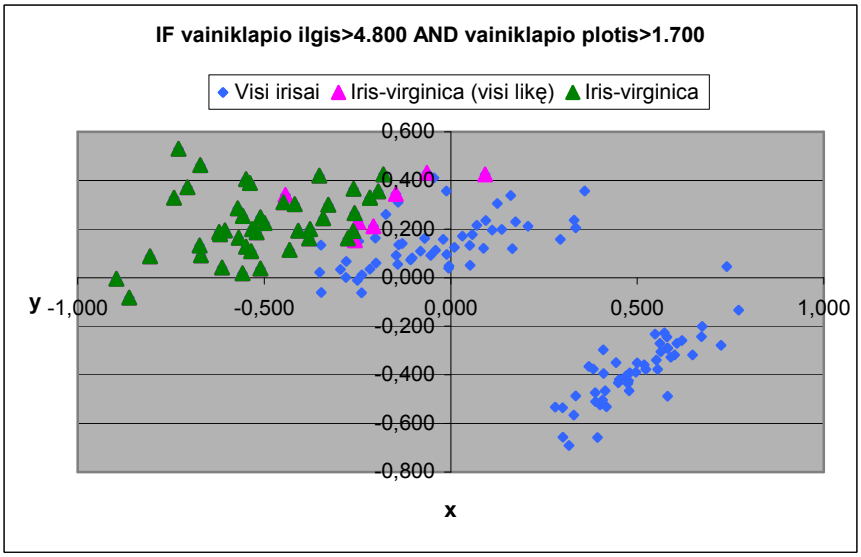
3. 2 pav. Grafikas Nr. 2 taisyklei iš Lentelės 10

Pagal antrąją taisyklę, kuri pavaizduota 3.2 pav., gauname, kad 44 irisai, kurie priklauso *Iris-setosa* klasei yra nudažomi raudona spalva. Visi likę tos pačios klasės atstovai yra tamsiai mėlynos spalvos, o visi kitę irisai kurie nėra antrosios klasės atstovai nuspalvinti šviesiai mėlyna spalva.



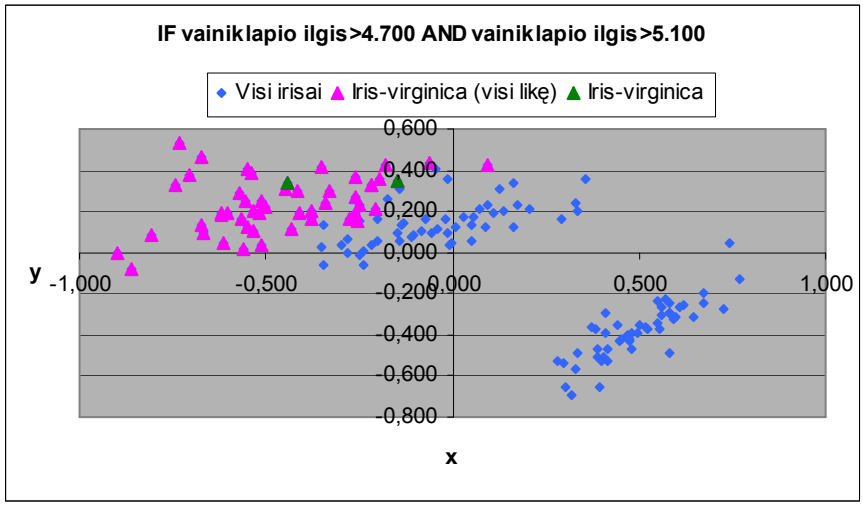
3. 3 pav. Grafikas Nr. 3 taisyklei iš Lentelės 10

Pagal trečiuoju numeriu pažymėtą taisyklę, kuri pavaizduota 3.3 pav., gauname, kad tik trys irisai priklauso *Iris-versicolor* klasei. Tos taisyklės atstovai yra žymima raudona spalva. Tamsiai mėlyna nudažyti likę *Iris-setosa* klasės irisai, o šviesiai mėlyna visi kiti likę duomenys.



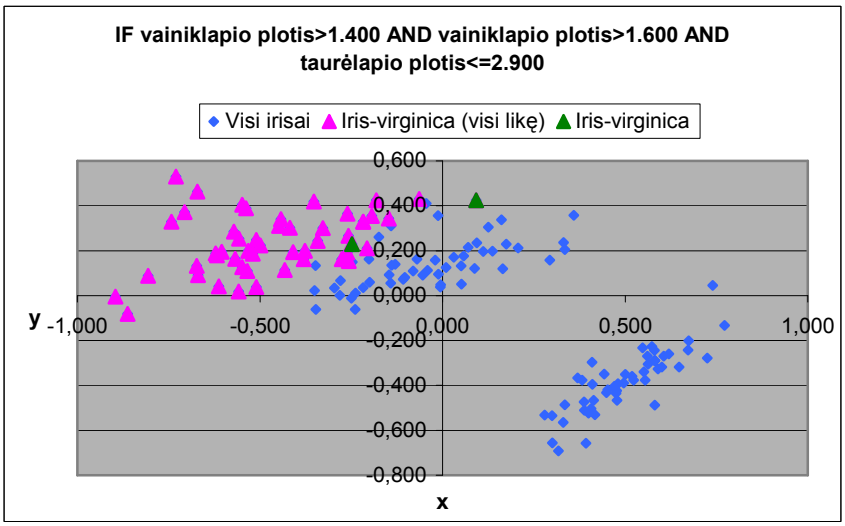
3. 4 pav. Grafikas Nr. 4 taisyklei iš Lentelės 10

3.4 pav. pavaizduota ketvirtajai taisyklei sukurtas Dekarto koordinatėse brėžinys. Iš Lentelės 10 žinome, kad lygiai 43 irisai priklauso šiai sukurtai taisyklei. Šios taisyklės atstovai yra *Iris-virginica* klasės atstovai. 3.4 pav. tai taisyklei priklausantys irisai nuspalvinti ryškiai žalia spalva, likę šios klasės atstovai yra rožiniai. Visi likę irisai yra šviesiai mėlynos spalvos.



3. 5 pav. Grafikas Nr. 5 taisyklei iš Lentelės 10

Penktajai taisyklei sukurtame grafike (3.5 pav.), gauname, kad du irisai priklauso šiai klasei, kurie nudažyti žalia spalva. Ryškiai rožine spalva nuspalvinti likę *Iris-virginica* klasės atstovai. Šviesiai mėlyni irisai yra iš pirmosios ir antrosios klasių.



3. 6 pav. Grafikas Nr. 6 taisyklei iš Lentelės 10

Paskutiniajai šeštajai taisyklei, kuri pavaizduota 3.6 pav. lygiai 2 irisai, kaip ir penktojoje taisyklėje priklauso *Iris-virginica* klasei. Šie irisai yra žymimi ryškiai žalia spalva, o likę tos klasės atstovai nuspalvinti ryškiai rožinė. Irisai, kurie nepriklauso trečiajai klasei yra šviesiai mėlynos spalvos.

Iš sudarytų grafikų matome, kad geriausiai atsiskiria pirmoji klasė. Kitų likusių dvejų klasių atstovų duomenys yra labai artimi. Vadinasi galime sakyti, kad *Iris-versicolor* ir *Iris-virginica* klasių atpažinimas yra sunkesnis. Pagal brėžinius galime sakyti, kad *Iris-setosa*, t.y. pirmosios klasės atstovai iš karto visi buvo ir „atskirti“. Nubraižyti pirmosios klasės irisus mums prireikė tik vieno brėžinio, o likusiom dviem klasėm net penkios iliustracijos.

IŠVADOS

Šiame magistro diplominiame darbe nagrinėjamos daugiamačių duomenų klasifikavimo rezultatų vizualios analizės problemos.

Darbe nagrinėtos dvi sistemos (*Orange Canvas* ir *Weka*), kuriose yra realizuoti keli klasifikavimo metodai. *Orange* sistemoje analizuoti du klasifikavimo algoritmai: klasifikavimo medis ir taisyklių generatorius. Nagrinėti iš viso trys klasifikavimo rezultatų grafiniai atvaizdavimai. *Weka* sistemoje nagrinėtas vienas klasifikatorius: Naive Bayes algoritmas, kurio rezultatas atvaizduotas stilizuotu medžiu.

Atlikus lyginamąją analizę galima daryti tokias išvadas:

1. *Orange* sistemoje realizuoti C4.5 metodo ir CN2 taisyklių generavimo algoritmo rezultatai pateikti grafiškai, o taip pat *Weka* sistemoje realizuotas NBTree algoritmas yra tinkami įrankiai klasifikavimo rezultatų interpretavimui.
2. *Orange* sistemoje realizuotas klasifikavimo rezultatų grafinis atvaizdavimas yra pranašesnis už *Weka* sistemoje realizuotą atvaizdavimą.
3. Sukurti klasifikatoriai yra gana tikslūs, t.y. gerai klasifikuotų naujus duomenis. Irisų ir vynų duomenų bendras klasifikavimo tikslumas yra ne mažiau nei 90%.
4. Klasifikavimo medžio grafinis atvaizdavimas, realizuotas *Orange* sistemoje yra informatyviausias lyginant su kitais analizuotais klasifikavimo rezultatų grafiniais atvaizdavimais.
5. Klasifikavimo rezultatų integravimas į daugiamačių duomenų projekcijų vaizdus, gautus daugiamačių skalių metodu, leidžia vizualiai stebėti klasių tarpusavio išdėstymą ir matyti, kurie taškai yra automatiškai priskirti nurodytai klasei pagal klasifikatoriaus sukurtą taisyklę.

LITERATŪRA

1. Asuncion, A.; Newman, D. J. UCI *Machine Learning Repository. Irvine*, CA: University of California, School of Information and Computer Science, 2007. http://www.ics.uci.edu/~mlearn/MLRepository.html

2. Borg, I; Groenen, P. *Modern Multidimensional Scaling. 2nd ed*. Springer, New York, 2005.

3. Clark, P.; Nibblet, T. *The CN2 Induction Algorithm, Machine Learning, Vol. 3, No. 4*. 1989. 261-283 p.

4. Demsar, J.; Zupan, B.; Leban, G. *Orange: From Experimental Machine Learning to Interactive Data Mining, White Paper*. Faculty of Computer and Information Science, University of Ljubljana, 2004. Available http://www.ailab.si/orange

5. Dunham, M. H. *Data Mining Introductory and Advanced Topics. Pearson Education*. Inc. Prentice Hall, 2003.

6. Dzemyda, G.; Kurasova, O.; ir Medvedev, V. *Dimension Reduction and Data Visualization Using Neural Networks*. In I. Maglogiannis, K. Karpouzis, M. Wallace and J. Soldatos, editors, Emerging Artificial Intelligence Applications in Computer Engineering. Real Word AI Systems with Applications in eHealth .Frontiers in Artificial Intelligence and Applications. 2007, 160: 25-49, IOS Press.

7. Fisher, R. A. *The Use of Multiple Measurements in Taxonomic Problems, Annals of Eugenics*. 1936, 7: 179-188.

8. Flach, P.; Lavrač, N. *Rule Induction*, In: Berthold, M., Hand, D. J. (eds.), Intelligent Data Analysis: an Introduction. Springer-Verlag, 2003. 230-267 p.

9. Ham, J.; Kamber, M. *Data Mining, Concepts and Techniques*. Elsevier, 2006.

10. Ian, H.; Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann*. San Francisco, 2005.
http://www.cs.waikato.ac.nz/ml/weka/

11. Lorose, D. T. *Discovery Knowledge in Data: An Introduction to Data Mining*. Wiley-Intersience, 2004.

12. Ramoni, M and Sebastiani, P. *Bayesian methods*. Springer-Verlag New York, 2003. 131-168 p. ISBN:3-540-43060-1.

13. Two pages about Orange data mining framework http://www.ailab.si/orange/wp/orange-leaflet.pdf; prisijungimo laikas 2008-04-23.

14. Weka (machine learning) – Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/weka_%28machine_learning%29; prisijungimo laikas 2008-04-23.

SANTRAUKA

Šiame magistro diplominiame darbe nagrinėjamos daugiamačių duomenų klasifikavimo rezultatų vizualios analizės problemos. Klasifikavimo uždavinių tikslas yra ne tik sukurti tikslų klasifikatorių, bet ir gautus rezultatus pateikti tokia atvaizdavimo forma, kuri padėtų tyrinėtojui lengvai interpretuoti gautus rezultatus, patvirtinti ar paneigti pradžioje iškeltas hipotezes ar formuoti naujas.

Darbe buvo nagrinėtos dvi sistemos (*Orange Canvas* ir *Weka*), kuriose buvo realizuoti keli klasifikavimo metodai. *Orange* sistemoje analizuoti du klasifikavimo algoritmai: klasifikavimo medis ir taisyklių generatorius, klasifikavimo medžio rezultatai buvo pateikti dviem formom: grafinis klasifikavimo medžio atvaizdavimas ir klasifikavimo medžio struktūrinė peržiūra. Vadinasi *Orange* sistemoje iš viso nagrinėti trys klasifikavimo rezultatų grafiniai atvaizdavimai. *Weka* sistemoje nagrinėtas vienas klasifikatorius: Naive Bayes algoritmas, kurio rezultatas atvaizduotas stilizuotu medžiu. Gauti dviejų sistemų rezultatai buvo lyginami norint sužinoti, kuris efektyvesnis. Nustatyta, kad geriausiai daugiamačių duomenų klasifikavimo rezultatus atvaizduoja klasifikavimo medžio grafinis atvaizdavimas.

Klasifikavimo rezultatus integravus į daugiamačių duomenų projekcijų vaizdus, gautus daugiamačių skalių metodu, nubraižyti grafikai stebėti irisų duomenų išsibarstimą pagal tiriamas klases. Grafikai nubraižyti pasinaudojus taisyklių generatoriaus sukurtomis taisyklėmis. Viso gauti šeši grafikai, kurie atspinti klasių išsibarstimą *xy* plokštumoje.

SUMMARY

Visual Analysis of the Multidimensional Data Classification Results

Visual analysis of the multidimensional data classification results were analyzed in this master thesis. Classification problem is not only to create the right classifier, but also to present the obtained results by such a visual form, that help us to interpret the obtained results, to confirm, reject or form new hypothesis.

Two systems (*Orange Canvas* and *Weka*) were analyzed in this work, where some classification approaches were realized. Two classification algorithms (classification tree and rule induction method) were analyzed. The classification tree results were performed in two ways: graphic classification tree and structure classification tree. Thus, three graphic classification results were analyzed in *Orange* system. One classifier was approached in *Weka* system. It was Naïve Bayes algorithm and results were pictured in stylized tree. The obtained results of two systems were compared to find which of them is more effective. Therefore, the best multidimensional data classification results are shown up by the graphic classification tree picture.

Classification results integrated into mapping of multidimensional data to plane, obtained by multidimensional scaling, graphs are draw to watch iris data scatter by classes. The graphs were drawn using rule induction. Six graphs demonstrate three classes in xy plane.

PRIEDAI

Priedas Nr. 1

150 irisų suskirstytų į 3 klases: Iris Setosa, Iris Versicolor, Iris Virginica.

A – taurėlapio ilgis

B – taurėlapio plotis

C – vainiklapio ilgis

D – vainiklapio plotis

class – irisų klasė (iš viso galimos trys klasės)

A	B	C	D	class															
5.1	3.5	1.4	0.2	Iris-setosa	4.9	3.1	1.5	0.1	Iris-setosa	6.6	3.0	4.4	1.4	Iris-versicolor	5.7	2.5	5.0	2.0	Iris-virginica
4.9	3.0	1.4	0.2	Iris-setosa	4.4	3.0	1.3	0.2	Iris-setosa	6.8	2.8	4.8	1.4	Iris-versicolor	5.8	2.8	5.1	2.4	Iris-virginica
4.7	3.2	1.3	0.2	Iris-setosa	5.1	3.4	1.5	0.2	Iris-setosa	6.7	3.0	5.0	1.7	Iris-versicolor	6.4	3.2	5.3	2.3	Iris-virginica
4.6	3.1	1.5	0.2	Iris-setosa	5.0	3.5	1.3	0.3	Iris-setosa	6.0	2.9	4.5	1.5	Iris-versicolor	6.5	3.0	5.5	1.8	Iris-virginica
5.0	3.6	1.4	0.2	Iris-setosa	4.5	2.3	1.3	0.3	Iris-setosa	5.7	2.6	3.5	1.0	Iris-versicolor	7.7	3.8	6.7	2.2	Iris-virginica
5.4	3.9	1.7	0.4	Iris-setosa	4.4	3.2	1.3	0.2	Iris-setosa	5.5	2.4	3.8	1.1	Iris-versicolor	7.7	2.6	6.9	2.3	Iris-virginica
4.6	3.4	1.4	0.3	Iris-setosa	5.0	3.5	1.6	0.6	Iris-setosa	5.5	2.4	3.7	1.0	Iris-versicolor	6.0	2.2	5.0	1.5	Iris-virginica
5.0	3.4	1.5	0.2	Iris-setosa	5.1	3.8	1.9	0.4	Iris-setosa	5.8	2.7	3.9	1.2	Iris-versicolor	6.9	3.2	5.7	2.3	Iris-virginica
4.4	2.9	1.4	0.2	Iris-setosa	4.8	3.0	1.4	0.3	Iris-setosa	6.0	2.7	5.1	1.6	Iris-versicolor	5.6	2.8	4.9	2.0	Iris-virginica
4.9	3.1	1.5	0.1	Iris-setosa	5.1	3.8	1.6	0.2	Iris-setosa	5.4	3.0	4.5	1.5	Iris-versicolor	7.7	2.8	6.7	2.0	Iris-virginica
5.4	3.7	1.5	0.2	Iris-setosa	4.6	3.2	1.4	0.2	Iris-setosa	6.0	3.4	4.5	1.6	Iris-versicolor	6.3	2.7	4.9	1.8	Iris-virginica
4.8	3.4	1.6	0.2	Iris-setosa	5.3	3.7	1.5	0.2	Iris-setosa	6.7	3.1	4.7	1.5	Iris-versicolor	6.7	3.3	5.7	2.1	Iris-virginica
4.8	3.0	1.4	0.1	Iris-setosa	5.0	3.3	1.4	0.2	Iris-setosa	6.3	2.3	4.4	1.3	Iris-versicolor	7.2	3.2	6.0	1.8	Iris-virginica
4.3	3.0	1.1	0.1	Iris-setosa	7.0	3.2	4.7	1.4	Iris-versicolor	5.6	3.0	4.1	1.3	Iris-versicolor	6.2	2.8	4.8	1.8	Iris-virginica
5.8	4.0	1.2	0.2	Iris-setosa	6.4	3.2	4.5	1.5	Iris-versicolor	5.5	2.5	4.0	1.3	Iris-versicolor	6.1	3.0	4.9	1.8	Iris-virginica
5.7	4.4	1.5	0.4	Iris-setosa	6.9	3.1	4.9	1.5	Iris-versicolor	5.5	2.6	4.4	1.2	Iris-versicolor	6.4	2.8	5.6	2.1	Iris-virginica
5.4	3.9	1.3	0.4	Iris-setosa	5.5	2.3	4.0	1.3	Iris-versicolor	6.1	3.0	4.6	1.4	Iris-versicolor	7.2	3.0	5.8	1.6	Iris-virginica
5.1	3.5	1.4	0.3	Iris-setosa	6.5	2.8	4.6	1.5	Iris-versicolor	5.8	2.6	4.0	1.2	Iris-versicolor	7.4	2.8	6.1	1.9	Iris-virginica
5.7	3.8	1.7	0.3	Iris-setosa	5.7	2.8	4.5	1.3	Iris-versicolor	5.0	2.3	3.3	1.0	Iris-versicolor	7.9	3.8	6.4	2.0	Iris-virginica
5.1	3.8	1.5	0.3	Iris-setosa	6.3	3.3	4.7	1.6	Iris-versicolor	5.6	2.7	4.2	1.3	Iris-versicolor	6.4	2.8	5.6	2.2	Iris-virginica
5.4	3.4	1.7	0.2	Iris-setosa	4.9	2.4	3.3	1.0	Iris-versicolor	5.7	3.0	4.2	1.2	Iris-versicolor	6.3	2.8	5.1	1.5	Iris-virginica
5.1	3.7	1.5	0.4	Iris-setosa	6.6	2.9	4.6	1.3	Iris-versicolor	5.7	2.9	4.2	1.3	Iris-versicolor	6.1	2.6	5.6	1.4	Iris-virginica
4.6	3.6	1.0	0.2	Iris-setosa	5.2	2.7	3.9	1.4	Iris-versicolor	6.2	2.9	4.3	1.3	Iris-versicolor	7.7	3.0	6.1	2.3	Iris-virginica
5.1	3.3	1.7	0.5	Iris-setosa	5.0	2.0	3.5	1.0	Iris-versicolor	5.1	2.5	3.0	1.1	Iris-versicolor	6.3	3.4	5.6	2.4	Iris-virginica
4.8	3.4	1.9	0.2	Iris-setosa	5.9	3.0	4.2	1.5	Iris-versicolor	5.7	2.8	4.1	1.3	Iris-versicolor	6.4	3.1	5.5	1.8	Iris-virginica
5.0	3.0	1.6	0.2	Iris-setosa	6.0	2.2	4.0	1.0	Iris-versicolor	6.3	3.3	6.0	2.5	Iris-virginica	6.0	3.0	4.8	1.8	Iris-virginica
5.0	3.4	1.6	0.4	Iris-setosa	6.1	2.9	4.7	1.4	Iris-versicolor	5.8	2.7	5.1	1.9	Iris-virginica	6.9	3.1	5.4	2.1	Iris-virginica
5.2	3.5	1.5	0.2	Iris-setosa	5.6	2.9	3.6	1.3	Iris-versicolor	7.1	3.0	5.9	2.1	Iris-virginica	6.7	3.1	5.6	2.4	Iris-virginica
5.2	3.4	1.4	0.2	Iris-setosa	6.7	3.1	4.4	1.4	Iris-versicolor	6.3	2.9	5.6	1.8	Iris-virginica	6.9	3.1	5.1	2.3	Iris-virginica
4.7	3.2	1.6	0.2	Iris-setosa	5.6	3.0	4.5	1.5	Iris-versicolor	6.5	3.0	5.8	2.2	Iris-virginica	5.8	2.7	5.1	1.9	Iris-virginica
4.8	3.1	1.6	0.2	Iris-setosa	5.8	2.7	4.1	1.0	Iris-versicolor	7.6	3.0	6.6	2.1	Iris-virginica	6.8	3.2	5.9	2.3	Iris-virginica
5.4	3.4	1.5	0.4	Iris-setosa	6.2	2.2	4.5	1.5	Iris-versicolor	4.9	2.5	4.5	1.7	Iris-virginica	6.7	3.3	5.7	2.5	Iris-virginica
5.2	4.1	1.5	0.1	Iris-setosa	5.6	2.5	3.9	1.1	Iris-versicolor	7.3	2.9	6.3	1.8	Iris-virginica	6.7	3.0	5.2	2.3	Iris-virginica
5.5	4.2	1.4	0.2	Iris-setosa	5.9	3.2	4.8	1.8	Iris-versicolor	6.7	2.5	5.8	1.8	Iris-virginica	6.3	2.5	5.0	1.9	Iris-virginica
4.9	3.1	1.5	0.1	Iris-setosa	6.1	2.8	4.0	1.3	Iris-versicolor	7.2	3.6	6.1	2.5	Iris-virginica	6.5	3.0	5.2	2.0	Iris-virginica
5.0	3.2	1.2	0.2	Iris-setosa	6.3	2.5	4.9	1.5	Iris-versicolor	6.5	3.2	5.1	2.0	Iris-virginica	6.2	3.4	5.4	2.3	Iris-virginica
5.5	3.5	1.3	0.2	Iris-setosa	6.1	2.8	4.7	1.2	Iris-versicolor	6.4	2.7	5.3	1.9	Iris-virginica	5.9	3.0	5.1	1.8	Iris-virginica
					6.4	2.9	4.3	1.3	Iris-versicolor	6.8	3.0	5.5	2.1	Iris-virginica					

Priedas Nr. 2

178 vynų rūšys suskirstytos į 3 klase.

Wine – vynas

A1 - Alcohol – alkoholis

A2 - Malic acid – obuolių rūgštis

A3 - Ash – pelenai (uosis)

A4 - Alcalinity of ash

A5 - Magnesium – magnis

A6 - Total phenols

A7 - Flavanoids

A8 - Nonflavanoid phenols

A9 - Proanthocyanins

A10 - Color intensity – spalvos intensyvumas

A11 - Hue – atspalvis

A12 - OD280/OD315 of diluted wines

A13 – Proline

Wine d class	A1 c	A2 c	A3 c	A4 c	A5 c	A6 c	A7 c	A8 c	A9 c	A10 c	A11 c	A12 c	A13 c
1	14.23	1.71	2.43	15.6	127	2.8	3.06	.28	2.29	5.64	1.04	3.92	1065
1	13.2	1.78	2.14	11.2	100	2.65	2.76	.26	1.28	4.38	1.05	3.4	1050
1	13.16	2.36	2.67	18.6	101	2.8	3.24	.3	2.81	5.68	1.03	3.17	1185
1	14.37	1.95	2.5	16.8	113	3.85	3.49	.24	2.18	7.8	.86	3.45	1480
1	13.24	2.59	2.87	21	118	2.8	2.69	.39	1.82	4.32	1.04	2.93	735
1	14.2	1.76	2.45	15.2	112	3.27	3.39	.34	1.97	6.75	1.05	2.85	1450
1	14.39	1.87	2.45	14.6	96	2.5	2.52	.3	1.98	5.25	1.02	3.58	1290
1	14.06	2.15	2.61	17.6	121	2.6	2.51	.31	1.25	5.05	1.06	3.58	1295
1	14.83	1.64	2.17	14	97	2.8	2.98	.29	1.98	5.2	1.08	2.85	1045
1	13.86	1.35	2.27	16	98	2.98	3.15	.22	1.85	7.22	1.01	3.55	1045
1	14.1	2.16	2.3	18	105	2.95	3.32	.22	2.38	5.75	1.25	3.17	1510
1	14.12	1.48	2.32	16.8	95	2.2	2.43	.26	1.57	5	1.17	2.82	1280
1	13.75	1.73	2.41	16	89	2.6	2.76	.29	1.81	5.6	1.15	2.9	1320
1	14.75	1.73	2.39	11.4	91	3.1	3.69	.43	2.81	5.4	1.25	2.73	1150
1	14.38	1.87	2.38	12	102	3.3	3.64	.29	2.96	7.5	1.2	3	1547
1	13.63	1.81	2.7	17.2	112	2.85	2.91	.3	1.46	7.3	1.28	2.88	1310
1	14.3	1.92	2.72	20	120	2.8	3.14	.33	1.97	6.2	1.07	2.65	1280
1	13.83	1.57	2.62	20	115	2.95	3.4	.4	1.72	6.6	1.13	2.57	1130
1	14.19	1.59	2.48	16.5	108	3.3	3.93	.32	1.86	8.7	1.23	2.82	1680
1	13.64	3.1	2.56	15.2	116	2.7	3.03	.17	1.66	5.1	.96	3.36	845
1	14.06	1.63	2.28	16	126	3	3.17	.24	2.1	5.65	1.09	3.71	780
1	12.93	3.8	2.65	18.6	102	2.41	2.41	.25	1.98	4.5	1.03	3.52	770
1	13.71	1.86	2.36	16.6	101	2.61	2.88	.27	1.69	3.8	1.11	4	1035
1	12.85	1.6	2.52	17.8	95	2.48	2.37	.26	1.46	3.93	1.09	3.63	1015
1	13.5	1.81	2.61	20	96	2.53	2.61	.28	1.66	3.52	1.12	3.82	845
1	13.05	2.05	3.22	25	124	2.63	2.68	.47	1.92	3.58	1.13	3.2	830
1	13.39	1.77	2.62	16.1	93	2.85	2.94	.34	1.45	4.8	.92	3.22	1195
1	13.3	1.72	2.14	17	94	2.4	2.19	.27	1.35	3.95	1.02	2.77	1285
1	13.87	1.9	2.8	19.4	107	2.95	2.97	.37	1.76	4.5	1.25	3.4	915
1	14.02	1.68	2.21	16	96	2.65	2.33	.26	1.98	4.7	1.04	3.59	1035
1	13.73	1.5	2.7	22.5	101	3	3.25	.29	2.38	5.7	1.19	2.71	1285
1	13.58	1.66	2.36	19.1	106	2.86	3.19	.22	1.95	6.9	1.09	2.88	1515
1	13.68	1.83	2.36	17.2	104	2.42	2.69	.42	1.97	3.84	1.23	2.87	990
1	13.76	1.53	2.7	19.5	132	2.95	2.74	.5	1.35	5.4	1.25	3	1235
1	13.51	1.8	2.65	19	110	2.35	2.53	.29	1.54	4.2	1.1	2.87	1095
1	13.48	1.81	2.41	20.5	100	2.7	2.98	.26	1.86	5.1	1.04	3.47	920
1	13.28	1.64	2.84	15.5	110	2.6	2.68	.34	1.36	4.6	1.09	2.78	880
1	13.05	1.65	2.55	18	98	2.45	2.43	.29	1.44	4.25	1.12	2.51	1105
1	13.07	1.5	2.1	15.5	98	2.4	2.64	.28	1.37	3.7	1.18	2.69	1020
1	14.22	3.99	2.51	13.2	128	3	3.04	.2	2.08	5.1	.89	3.53	760
1	13.56	1.71	2.31	16.2	117	3.15	3.29	.34	2.34	6.13	.95	3.38	795
1	13.41	3.84	2.12	18.8	90	2.45	2.68	.27	1.48	4.28	.91	3	1035
1	13.88	1.89	2.59	15	101	3.25	3.56	.17	1.7	5.43	.88	3.56	1095
1	13.24	3.98	2.29	17.5	103	2.64	2.63	.32	1.66	4.36	.82	3	680
1	13.05	1.77	2.1	17	107	3	3	.28	2.03	5.04	.88	3.35	885
1	14.21	4.04	2.44	18.9	111	2.85	2.65	.3	1.25	5.24	.87	3.33	1080
1	14.38	3.59	2.28	16	102	3.25	3.17	.27	2.19	4.9	1.04	3.44	1065

1	13.9	1.68	2.12	16	101	3.1	3.39	.21	2.14	6.1	.91	3.33	985
1	14.1	2.02	2.4	18.8	103	2.75	2.92	.32	2.38	6.2	1.07	2.75	1060
1	13.94	1.73	2.27	17.4	108	2.88	3.54	.32	2.08	8.90	1.12	3.1	1260
1	13.05	1.73	2.04	12.4	92	2.72	3.27	.17	2.91	7.2	1.12	2.91	1150
1	13.83	1.65	2.6	17.2	94	2.45	2.99	.22	2.29	5.6	1.24	3.37	1265
1	13.82	1.75	2.42	14	111	3.88	3.74	.32	1.87	7.05	1.01	3.26	1190
1	13.77	1.9	2.68	17.1	115	3	2.79	.39	1.68	6.3	1.13	2.93	1375
1	13.74	1.67	2.25	16.4	118	2.6	2.9	.21	1.62	5.85	.92	3.2	1060
1	13.56	1.73	2.46	20.5	116	2.96	2.78	.2	2.45	6.25	.98	3.03	1120
1	14.22	1.7	2.3	16.3	118	3.2	3	.26	2.03	6.38	.94	3.31	970
1	13.29	1.97	2.68	16.8	102	3	3.23	.31	1.66	6	1.07	2.84	1270
1	13.72	1.43	2.5	16.7	108	3.4	3.67	.19	2.04	6.8	.89	2.87	1285
2	12.37	.94	1.36	10.6	88	1.98	.57	.28	.42	1.95	1.05	1.82	520
2	12.33	1.1	2.28	16	101	2.05	1.09	.63	.41	3.27	1.25	1.67	680
2	12.64	1.36	2.02	16.8	100	2.02	1.41	.53	.62	5.75	.98	1.59	450
2	13.67	1.25	1.92	18	94	2.1	1.79	.32	.73	3.8	1.23	2.46	630
2	12.37	1.13	2.16	19	87	3.5	3.1	.19	1.87	4.45	1.22	2.87	420
2	12.17	1.45	2.53	19	104	1.89	1.75	.45	1.03	2.95	1.45	2.23	355
2	12.37	1.21	2.56	18.1	98	2.42	2.65	.37	2.08	4.6	1.19	2.3	678
2	13.11	1.01	1.7	15	78	2.98	3.18	.26	2.28	5.3	1.12	3.18	502
2	12.37	1.17	1.92	19.6	78	2.11	2	.27	1.04	4.68	1.12	3.48	510
2	13.34	.94	2.36	17	110	2.53	1.3	.55	.42	3.17	1.02	1.93	750
2	12.21	1.19	1.75	16.8	151	1.85	1.28	.14	2.5	2.85	1.28	3.07	718
2	12.29	1.61	2.21	20.4	103	1.1	1.02	.37	1.46	3.05	.906	1.82	870
2	13.86	1.51	2.67	25	86	2.95	2.86	.21	1.87	3.38	1.36	3.16	410
2	13.49	1.66	2.24	24	87	1.88	1.84	.27	1.03	3.74	.98	2.78	472
2	12.99	1.67	2.6	30	139	3.3	2.89	.21	1.96	3.35	1.31	3.5	985
2	11.96	1.09	2.3	21	101	3.38	2.14	.13	1.65	3.21	.99	3.13	886
2	11.66	1.88	1.92	16	97	1.61	1.57	.34	1.15	3.8	1.23	2.14	428
2	13.03	.9	1.71	16	86	1.95	2.03	.24	1.46	4.6	1.19	2.48	392
2	11.84	2.89	2.23	18	112	1.72	1.32	.43	.95	2.65	.96	2.52	500
2	12.33	.99	1.95	14.8	136	1.9	1.85	.35	2.76	3.4	1.06	2.31	750
2	12.7	3.87	2.4	23	101	2.83	2.55	.43	1.95	2.57	1.19	3.13	463
2	12	.92	2	19	86	2.42	2.26	.3	1.43	2.5	1.38	3.12	278
2	12.72	1.81	2.2	18.8	86	2.2	2.53	.26	1.77	3.9	1.16	3.14	714
2	12.08	1.13	2.51	24	78	2	1.58	.4	1.4	2.2	1.31	2.72	630
2	13.05	3.86	2.32	22.5	85	1.65	1.59	.61	1.62	4.8	.84	2.01	515
2	11.84	.89	2.58	18	94	2.2	2.21	.22	2.35	3.05	.79	3.08	520
2	12.67	.98	2.24	18	99	2.2	1.94	.3	1.46	2.62	1.23	3.16	450
2	12.16	1.61	2.31	22.8	90	1.78	1.69	.43	1.56	2.45	1.33	2.26	495
2	11.65	1.67	2.62	26	88	1.92	1.61	.4	1.34	2.6	1.36	3.21	562
2	11.64	2.06	2.46	21.6	84	1.95	1.69	.48	1.35	2.8	1	2.75	680
2	12.08	1.33	2.3	23.6	70	2.2	1.59	.42	1.38	1.74	1.07	3.21	625
2	12.08	1.83	2.32	18.5	81	1.6	1.5	.52	1.64	2.4	1.08	2.27	480
2	12	1.51	2.42	22	86	1.45	1.25	.5	1.63	3.6	1.05	2.65	450
2	12.69	1.53	2.26	20.7	80	1.38	1.46	.58	1.62	3.05	.96	2.06	495
2	12.29	2.83	2.22	18	88	2.45	2.25	.25	1.99	2.15	1.15	3.3	290
2	11.62	1.99	2.28	18	98	3.02	2.26	.17	1.35	3.25	1.16	2.96	345
2	12.47	1.52	2.2	19	162	2.5	2.27	.32	3.28	2.6	1.16	2.63	937
2	11.81	2.12	2.74	21.5	134	1.6	.99	.14	1.56	2.5	.95	2.26	625
2	12.29	1.41	1.98	16	85	2.55	2.5	.29	1.77	2.9	1.23	2.74	428
2	12.37	1.07	2.1	18.5	88	3.52	3.75	.24	1.95	4.5	1.04	2.77	660
2	12.29	3.17	2.21	18	88	2.85	2.99	.45	2.81	2.3	1.42	2.83	406
2	12.08	2.08	1.7	17.5	97	2.23	2.17	.26	1.4	3.3	1.27	2.96	710
2	12.6	1.34	1.9	18.5	88	1.45	1.36	.29	1.35	2.45	1.04	2.77	562
2	12.34	2.45	2.46	21	98	2.56	2.11	.34	1.31	2.8	.8	3.38	438
2	11.82	1.72	1.88	19.5	86	2.5	1.64	.37	1.42	2.06	.94	2.44	415
2	12.51	1.73	1.98	20.5	85	2.2	1.92	.32	1.48	2.94	1.04	3.57	672
2	12.42	2.55	2.27	22	90	1.68	1.84	.66	1.42	2.7	.86	3.3	315
2	12.25	1.73	2.12	19	80	1.65	2.03	.37	1.63	3.4	1	3.17	510
2	12.72	1.75	2.28	22.5	84	1.38	1.76	.48	1.63	3.3	.88	2.42	488
2	12.22	1.29	1.94	19	92	2.36	2.04	.39	2.08	2.7	.86	3.02	312
2	11.61	1.35	2.7	20	94	2.74	2.92	.29	2.49	2.65	.96	3.26	680
2	11.46	3.74	1.82	19.5	107	3.18	2.58	.24	3.58	2.9	.75	2.81	562
2	12.52	2.43	2.17	21	88	2.55	2.27	.26	1.22	2	.9	2.78	325
2	11.76	2.68	2.92	20	103	1.75	2.03	.6	1.05	3.8	1.23	2.5	607
2	11.41	.74	2.5	21	88	2.48	2.01	.42	1.44	3.08	1.1	2.31	434
2	12.08	1.39	2.5	22.5	84	2.56	2.29	.43	1.04	2.9	.93	3.19	385
2	11.03	1.51	2.2	21.5	85	2.46	2.17	.52	2.01	1.9	1.71	2.87	407
2	11.82	1.47	1.99	20.8	86	1.98	1.6	.3	1.53	1.95	.95	3.33	495
2	12.42	1.61	2.19	22.5	108	2	2.09	.34	1.61	2.06	1.06	2.96	345
2	12.77	3.43	1.98	16	80	1.63	1.25	.43	.83	3.4	.7	2.12	372
2	12	3.43	2	19	87	2	1.64	.37	1.87	1.28	.93	3.05	564
2	11.45	2.4	2.42	20	96	2.9	2.79	.32	1.83	3.25	.8	3.39	625
2	11.56	2.05	3.23	28.5	119	3.18	5.08	.47	1.87	6	.93	3.69	465
2	12.42	4.43	2.73	26.5	102	2.2	2.13	.43	1.71	2.08	.92	3.12	365
2	13.05	5.8	2.13	21.5	86	2.62	2.65	.3	2.01	2.6	.73	3.1	380
2	11.87	4.31	2.39	21	82	2.86	3.03	.21	2.91	2.8	.75	3.64	380
2	12.07	2.16	2.17	21	85	2.6	2.65	.37	1.35	2.76	.86	3.28	378
2	12.43	1.53	2.29	21.5	86	2.74	3.15	.39	1.77	3.94	.69	2.84	352
2	11.79	2.13	2.78	28.5	92	2.13	2.24	.58	1.76	3	.97	2.44	466
2	12.37	1.63	2.3	24.5	88	2.22	2.45	.4	1.9	2.12	.89	2.78	342
2	12.04	4.3	2.38	22	80	2.1	1.75	.42	1.35	2.6	.79	2.57	580
3	12.86	1.35	2.32	18	122	1.51	1.25	.21	.94	4.1	.76	1.29	630
3	12.88	2.99	2.4	20	104	1.3	1.22	.24	.83	5.4	.74	1.42	530
3	12.81	2.31	2.4	24	98	1.15	1.09	.27	.83	5.7	.66	1.36	560
3	12.7	3.55	2.36	21.5	106	1.7	1.2	.17	.84	5	.78	1.29	600
3	12.51	1.24	2.25	17.5	85	2	.58	.6	1.25	5.45	.75	1.51	650
3	12.6	2.46	2.2	18.5	94	1.62	.66	.63	.94	7.1	.73	1.58	695
3	12.25	4.72	2.54	21	89	1.38	.47	.53	.8	3.85	.75	1.27	720

3	12.53	5.51	2.64	25	96	1.79	.6	.63	1.1	5	.82	1.69	515
3	13.49	3.59	2.19	19.5	88	1.62	.48	.58	.88	5.7	.81	1.82	580
3	12.84	2.96	2.61	24	101	2.32	.6	.53	.81	4.92	.89	2.15	590
3	12.93	2.81	2.7	21	96	1.54	.5	.53	.75	4.6	.77	2.31	600
3	13.36	2.56	2.35	20	89	1.4	.5	.37	.64	5.6	.7	2.47	780
3	13.52	3.17	2.72	23.5	97	1.55	.52	.5	.55	4.35	.89	2.06	520
3	13.62	4.95	2.35	20	92	2	.8	.47	1.02	4.4	.91	2.05	550
3	12.25	3.88	2.2	18.5	112	1.38	.78	.29	1.14	8.21	.65	2	855
3	13.16	3.57	2.15	21	102	1.5	.55	.43	1.3	4	.6	1.68	830
3	13.88	5.04	2.23	20	80	.98	.34	.4	.68	4.9	.58	1.33	415
3	12.87	4.61	2.48	21.5	86	1.7	.65	.47	.86	7.65	.54	1.86	625
3	13.32	3.24	2.38	21.5	92	1.93	.76	.45	1.25	8.42	.55	1.62	650
3	13.08	3.9	2.36	21.5	113	1.41	1.39	.34	1.14	9.40	.57	1.33	550
3	13.5	3.12	2.62	24	123	1.4	1.57	.22	1.25	8.60	.59	1.3	500
3	12.79	2.67	2.48	22	112	1.48	1.36	.24	1.26	10.8	.48	1.47	480
3	13.11	1.9	2.75	25.5	116	2.2	1.28	.26	1.56	7.1	.61	1.33	425
3	13.23	3.3	2.28	18.5	98	1.8	.83	.61	1.87	10.52	.56	1.51	675
3	12.58	1.29	2.1	20	103	1.48	.58	.53	1.4	7.6	.58	1.55	640
3	13.17	5.19	2.32	22	93	1.74	.63	.61	1.55	7.9	.6	1.48	725
3	13.84	4.12	2.38	19.5	89	1.8	.83	.48	1.56	9.01	.57	1.64	480
3	12.45	3.03	2.64	27	97	1.9	.58	.63	1.14	7.5	.67	1.73	880
3	14.34	1.68	2.7	25	98	2.8	1.31	.53	2.7	13	.57	1.96	660
3	13.48	1.67	2.64	22.5	89	2.6	1.1	.52	2.29	11.75	.57	1.78	620
3	12.36	3.83	2.38	21	88	2.3	.92	.5	1.04	7.65	.56	1.58	520
3	13.69	3.26	2.54	20	107	1.83	.56	.5	.8	5.88	.96	1.82	680
3	12.85	3.27	2.58	22	106	1.65	.6	.6	.96	5.58	.87	2.11	570
3	12.96	3.45	2.35	18.5	106	1.39	.7	.4	.94	5.28	.68	1.75	675
3	13.78	2.76	2.3	22	90	1.35	.68	.41	1.03	9.58	.7	1.68	615
3	13.73	4.36	2.26	22.5	88	1.28	.47	.52	1.15	6.62	.78	1.75	520
3	13.45	3.7	2.6	23	111	1.7	.92	.43	1.46	10.68	.85	1.56	695
3	12.82	3.37	2.3	19.5	88	1.48	.66	.4	.97	10.26	.72	1.75	685
3	13.58	2.58	2.69	24.5	105	1.55	.84	.39	1.54	8.66	.74	1.8	750
3	13.4	4.6	2.86	25	112	1.98	.96	.27	1.11	8.5	.67	1.92	630
3	12.2	3.03	2.32	19	96	1.25	.49	.4	.73	5.5	.66	1.83	510
3	12.77	2.39	2.28	19.5	86	1.39	.51	.48	.64	9.899999	.57	1.63	470
3	14.16	2.51	2.48	20	91	1.68	.7	.44	1.24	9.7	.62	1.71	660
3	13.71	5.65	2.45	20.5	95	1.68	.61	.52	1.06	7.7	.64	1.74	740
3	13.4	3.91	2.48	23	102	1.8	.75	.43	1.41	7.3	.7	1.56	750
3	13.27	4.28	2.26	20	120	1.59	.69	.43	1.35	10.2	.59	1.56	835
3	13.17	2.59	2.37	20	120	1.65	.68	.53	1.46	9.3	.6	1.62	840
3	14.13	4.1	2.74	24.5	96	2.05	.76	.56	1.35	9.2	.61	1.6	560

Priedas Nr. 3

Naudojamos sąvokos:

Iris – klasifikavimui naudojami *irisų* duomenys

Class – klasė, kuri gali būti iris-setora, iris-versicolor ir iris-virginica.

X	Y	taurėlapio ilgis	taurėlapio plotis	vainiklapio ilgis	vainiklapio plotis	iris class
0,475	-0,435	5,1	3,5	1,4	0,2	Iris-setosa
0,580	-0,244	4,9	3,0	1,4	0,2	Iris-setosa
0,590	-0,327	4,7	3,2	1,3	0,2	Iris-setosa
0,606	-0,270	4,6	3,1	1,5	0,2	Iris-setosa
0,478	-0,466	5,0	3,6	1,4	0,2	Iris-setosa
0,300	-0,535	5,4	3,9	1,7	0,4	Iris-setosa
0,555	-0,377	4,6	3,4	1,4	0,3	Iris-setosa
0,496	-0,389	5,0	3,4	1,5	0,2	Iris-setosa
0,674	-0,201	4,4	2,9	1,4	0,2	Iris-setosa
0,581	-0,290	4,9	3,1	1,5	0,1	Iris-setosa
0,388	-0,510	5,4	3,7	1,5	0,2	Iris-setosa
0,523	-0,377	4,8	3,4	1,6	0,2	Iris-setosa
0,620	-0,259	4,8	3,0	1,4	0,1	Iris-setosa
0,725	-0,278	4,3	3,0	1,1	0,1	Iris-setosa
0,301	-0,656	5,8	4,0	1,2	0,2	Iris-setosa
0,772	-0,133	5,7	4,4	1,5	0,4	Iris-setosa
0,330	-0,565	5,4	3,9	1,3	0,4	Iris-setosa
0,452	-0,419	5,1	3,5	1,4	0,3	Iris-setosa
0,280	-0,532	5,7	3,8	1,7	0,3	Iris-setosa
0,400	-0,523	5,1	3,8	1,5	0,3	Iris-setosa
0,410	-0,395	5,4	3,4	1,7	0,2	Iris-setosa
0,387	-0,473	5,1	3,7	1,5	0,4	Iris-setosa
0,581	-0,488	4,6	3,6	1,0	0,2	Iris-setosa
0,409	-0,297	5,1	3,3	1,7	0,5	Iris-setosa
0,500	-0,351	4,8	3,4	1,9	0,2	Iris-setosa
0,548	-0,233	5,0	3,0	1,6	0,2	Iris-setosa
0,443	-0,349	5,0	3,4	1,6	0,4	Iris-setosa
0,449	-0,432	5,2	3,5	1,5	0,2	Iris-setosa
0,471	-0,404	5,2	3,4	1,4	0,2	Iris-setosa
0,563	-0,305	4,7	3,2	1,6	0,2	Iris-setosa
0,561	-0,270	4,8	3,1	1,6	0,2	Iris-setosa
0,382	-0,376	5,4	3,4	1,5	0,4	Iris-setosa
0,393	-0,658	5,2	4,1	1,5	0,1	Iris-setosa
0,317	-0,691	5,5	4,2	1,4	0,2	Iris-setosa
0,581	-0,290	4,9	3,1	1,5	0,1	Iris-setosa
0,551	-0,339	5,0	3,2	1,2	0,2	Iris-setosa
0,414	-0,466	5,5	3,5	1,3	0,2	Iris-setosa
0,581	-0,290	4,9	3,1	1,5	0,1	Iris-setosa
0,672	-0,243	4,4	3,0	1,3	0,2	Iris-setosa
0,480	-0,392	5,1	3,4	1,5	0,2	Iris-setosa
0,477	-0,424	5,0	3,5	1,3	0,3	Iris-setosa
0,740	0,045	4,5	2,3	1,3	0,3	Iris-setosa
0,648	-0,318	4,4	3,2	1,3	0,2	Iris-setosa
0,370	-0,366	5,0	3,5	1,6	0,6	Iris-setosa
0,335	-0,486	5,1	3,8	1,9	0,4	Iris-setosa
0,573	-0,227	4,8	3,0	1,4	0,3	Iris-setosa
0,417	-0,531	5,1	3,8	1,6	0,2	Iris-setosa
0,600	-0,318	4,6	3,2	1,4	0,2	Iris-setosa
0,407	-0,504	5,3	3,7	1,5	0,2	Iris-setosa
0,519	-0,359	5,0	3,3	1,4	0,2	Iris-setosa
-0,347	-0,061	7,0	3,2	4,7	1,4	Iris-versicolor
-0,238	0,011	6,4	3,2	4,5	1,5	Iris-versicolor
-0,351	0,022	6,9	3,1	4,9	1,5	Iris-versicolor
0,125	0,305	5,5	2,3	4,0	1,3	Iris-versicolor
-0,202	0,163	6,5	2,8	4,6	1,5	Iris-versicolor
-0,020	0,158	5,7	2,8	4,5	1,3	Iris-versicolor
-0,281	0,001	6,3	3,3	4,7	1,6	Iris-versicolor
0,335	0,205	4,9	2,4	3,3	1,0	Iris-versicolor
-0,200	0,060	6,6	2,9	4,6	1,3	Iris-versicolor
0,111	0,196	5,2	2,7	3,9	1,4	Iris-versicolor
0,359	0,357	5,0	2,0	3,5	1,0	Iris-versicolor
-0,108	0,073	5,9	3,0	4,2	1,5	Iris-versicolor
0,161	0,338	6,0	2,2	4,0	1,0	Iris-versicolor

-0,139	0,135	6,1	2,9	4,7	1,4	Iris-versicolor
0,052	0,051	5,6	2,9	3,6	1,3	Iris-versicolor
-0,250	-0,011	6,7	3,1	4,4	1,4	Iris-versicolor
-0,081	0,109	5,6	3,0	4,5	1,5	Iris-versicolor
0,088	0,121	5,8	2,7	4,1	1,0	Iris-versicolor
-0,045	0,410	6,2	2,2	4,5	1,5	Iris-versicolor
0,137	0,198	5,6	2,5	3,9	1,1	Iris-versicolor
-0,280	0,067	5,9	3,2	4,8	1,8	Iris-versicolor
-0,041	0,113	6,1	2,8	4,0	1,3	Iris-versicolor
-0,141	0,310	6,3	2,5	4,9	1,5	Iris-versicolor
-0,070	0,162	6,1	2,8	4,7	1,2	Iris-versicolor
-0,142	0,055	6,4	2,9	4,3	1,3	Iris-versicolor
-0,217	0,035	6,6	3,0	4,4	1,4	Iris-versicolor
-0,247	0,150	6,8	2,8	4,8	1,4	Iris-versicolor
-0,348	0,134	6,7	3,0	5,0	1,7	Iris-versicolor
-0,130	0,140	6,0	2,9	4,5	1,5	Iris-versicolor
0,165	0,119	5,7	2,6	3,5	1,0	Iris-versicolor
0,174	0,230	5,5	2,4	3,8	1,1	Iris-versicolor
0,207	0,212	5,5	2,4	3,7	1,0	Iris-versicolor
0,051	0,133	5,8	2,7	3,9	1,2	Iris-versicolor
-0,174	0,260	6,0	2,7	5,1	1,6	Iris-versicolor
-0,054	0,091	5,4	3,0	4,5	1,5	Iris-versicolor
-0,239	-0,062	6,0	3,4	4,5	1,6	Iris-versicolor
-0,296	0,034	6,7	3,1	4,7	1,5	Iris-versicolor
-0,012	0,356	6,3	2,3	4,4	1,3	Iris-versicolor
-0,005	0,048	5,6	3,0	4,1	1,3	Iris-versicolor
0,094	0,235	5,5	2,5	4,0	1,3	Iris-versicolor
0,070	0,215	5,5	2,6	4,4	1,2	Iris-versicolor
-0,146	0,092	6,1	3,0	4,6	1,4	Iris-versicolor
0,058	0,177	5,8	2,6	4,0	1,2	Iris-versicolor
0,331	0,236	5,0	2,3	3,3	1,0	Iris-versicolor
0,031	0,171	5,6	2,7	4,2	1,3	Iris-versicolor
-0,006	0,039	5,7	3,0	4,2	1,2	Iris-versicolor
-0,011	0,095	5,7	2,9	4,2	1,3	Iris-versicolor
-0,103	0,081	6,2	2,9	4,3	1,3	Iris-versicolor
0,294	0,158	5,1	2,5	3,0	1,1	Iris-versicolor
0,010	0,125	5,7	2,8	4,1	1,3	Iris-versicolor
-0,670	0,091	6,3	3,3	6,0	2,5	Iris-virginica
-0,217	0,329	5,8	2,7	5,1	1,9	Iris-virginica
-0,571	0,285	7,1	3,0	5,9	2,1	Iris-virginica
-0,342	0,244	6,3	2,9	5,6	1,8	Iris-virginica
-0,510	0,248	6,5	3,0	5,8	2,2	Iris-virginica
-0,706	0,372	7,6	3,0	6,6	2,1	Iris-virginica
0,092	0,425	4,9	2,5	4,5	1,7	Iris-virginica
-0,538	0,389	7,3	2,9	6,3	1,8	Iris-virginica
-0,352	0,419	6,7	2,5	5,8	1,8	Iris-virginica
-0,806	0,088	7,2	3,6	6,1	2,5	Iris-virginica
-0,432	0,114	6,5	3,2	5,1	2,0	Iris-virginica
-0,328	0,300	6,4	2,7	5,3	1,9	Iris-virginica
-0,499	0,224	6,8	3,0	5,5	2,1	Iris-virginica
-0,181	0,425	5,7	2,5	5,0	2,0	Iris-virginica
-0,510	0,038	5,8	2,8	5,1	2,4	Iris-virginica
-0,535	0,109	6,4	3,2	5,3	2,3	Iris-virginica
-0,377	0,199	6,5	3,0	5,5	1,8	Iris-virginica
-0,896	-0,004	7,7	3,8	6,7	2,2	Iris-virginica
-0,730	0,530	7,7	2,6	6,9	2,3	Iris-virginica
-0,064	0,431	6,0	2,2	5,0	1,5	Iris-virginica
-0,618	0,179	6,9	3,2	5,7	2,3	Iris-virginica
-0,194	0,355	5,6	2,8	4,9	2,0	Iris-virginica
-0,671	0,463	7,7	2,8	6,7	2,0	Iris-virginica
-0,258	0,266	6,3	2,7	4,9	1,8	Iris-virginica
-0,549	0,127	6,7	3,3	5,7	2,1	Iris-virginica
-0,557	0,253	7,2	3,2	6,0	1,8	Iris-virginica
-0,248	0,230	6,2	2,8	4,8	1,8	Iris-virginica
-0,276	0,162	6,1	3,0	4,9	1,8	Iris-virginica
-0,418	0,302	6,4	2,8	5,6	2,1	Iris-virginica
-0,443	0,341	7,2	3,0	5,8	1,6	Iris-virginica
-0,549	0,405	7,4	2,8	6,1	1,9	Iris-virginica
-0,862	-0,082	7,9	3,8	6,4	2,0	Iris-virginica
-0,448	0,310	6,4	2,8	5,6	2,2	Iris-virginica
-0,207	0,211	6,3	2,8	5,1	1,5	Iris-virginica
-0,147	0,344	6,1	2,6	5,6	1,4	Iris-virginica
-0,742	0,329	7,7	3,0	6,1	2,3	Iris-virginica
-0,613	0,041	6,3	3,4	5,6	2,4	Iris-virginica
-0,380	0,162	6,4	3,1	5,5	1,8	Iris-virginica
-0,257	0,154	6,0	3,0	4,8	1,8	Iris-virginica
-0,520	0,187	6,9	3,1	5,4	2,1	Iris-virginica
-0,605	0,195	6,7	3,1	5,6	2,4	Iris-virginica
-0,568	0,164	6,9	3,1	5,1	2,3	Iris-virginica

-0,217	0,329	5,8	2,7	5,1	1,9	Iris-virginica
-0,621	0,188	6,8	3,2	5,9	2,3	Iris-virginica
-0,673	0,133	6,7	3,3	5,7	2,5	Iris-virginica
-0,532	0,200	6,7	3,0	5,2	2,3	Iris-virginica
-0,260	0,365	6,3	2,5	5,0	1,9	Iris-virginica
-0,409	0,193	6,5	3,0	5,2	2,0	Iris-virginica
-0,558	0,018	6,2	3,4	5,4	2,3	Iris-virginica
-0,261	0,192	5,9	3,0	5,1	1,8	Iris-virginica