VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Magistro baigiamasis darbas

Hiperparametrų optimizavimo metodai virpesinės spektrometrijos duomenų analizėje

Atliko:
Brendonas Stakauskas

Vadovas:
Tomas Raila

Vilnius
2021

Turinys

Santrauka

Spektrometrijos duomenyse dėl įvairių priežasčių gali atsirasti nepageidaujamų savybių, apsunkinančių duomenų analizę. Norint pasiekti geresnį rezultatą, nekokybiškų duomenų sutvarkymui galima pasitelkti tam tikrus metodus, dalies jų reguliavimui naudojami parametrai. Dėl spektrometrinių duomenų požymių kolinearumo svarbu atrinkti reikiamas spektrines savybes, parinkti tinkamą mašininio mokymo modelį. Visa tai sudaro ištisą procesų grandinę, kurios parametrai, galimos metodų pozicijų grandinėje variacijos, sukelia kombinatorinį sprogimą. Darbo tikslas – paruošti automatizuotos hiperparametrų paieškos metodiką, skirtą virpesinės spektrometrijos duomenų analizei.

Magistro darbe aptariamos metodikos paremtos genetiniu optimizavimu bei atsitiktiniu neuroninių tinklų architektūros sudarymu. Tyrimo pagrindas – spektrometrinių duomenų apdorojimo ir analizės metodai. Toliau ši aibė papildyta bendrai mašininiame mokyme naudojamais metodais.

Sudarytos metodikos validuotos naudojantis išorinėmis duomenų aibėmis, apimančiomis NIR, Ramano ir MIR spektrometrijas. Atrinkti duomenys aptariami publikuotuose straipsniuose, tad gaunami rezultatai galėjo būti tikslingai palyginti su realiais panaudojimo atvejais.

Rezultatyviausios metodikos pagrindą sudaro spektrinių požymių atrinkimas bei genetinės paieškos įrankio *TPOT* panaudojimas tolimesnei paieškai. Tokiu būdu sudaryti modeliai du kartus iš trijų veikia geriau nei autorių siūlomi modeliai. Metodiką papildžius dar vienu neprivalomu žingsniu, gaunamas rezultatyvesnis – trečiasis – modelis.

Summary

Hyperparameter Optimization Methods for Vibrational Spectroscopy Data Analysis

Data analysis of vibrational spectroscopy requires a deep understanding of both spectroscopy and data analysis fields. Spectroscopy data may contain unwanted properties (e.g. noise, data scattering). This characteristic makes it harder to conduct data analysis experiments for the dataset. To clean out the data of those unwanted attributes, one can use various methods that may require additional parameters. Spectroscopic data contains many collinear properties so to properly use this kind of data for analysis one must pick important features and machine learning model properly. Data preprocessing, important variables selection, and machine learning models make up the whole data analysis pipeline. The pipeline parameters – method combinations, methods place in the pipeline, method parameters – can cause a combinatorial explosion, which makes it hard to find a sufficient pipeline for the given task. The aim of this master's thesis – to find a method that is suitable for automatic hyperparameter search of analysis models for vibrational spectroscopy data.

Methods discussed in this master's thesis are based on genetic optimization (*TPOT*) and random search of neural network architecture (*AutoKeras*). The main focus of this work was methods that are used in vibrational spectroscopy data analysis. Optimization tasks were built by using various combinations of these methods and tweaking the genetic search task parameters as well. Later research was conducted by using more generic machine learning models (e.g. decision trees, k-NN) as a subset for the pipeline search. This search was conducted not on the whole dataset but only on the features that were kept after applying variable selection algorithm. The last piece of research was carried out on neural networks – by training some simple CNN model and comparing it with the one random search can find.

The datasets used in this work were picked from published articles, which allows for meaningful result comparison. Datasets included MIR (FT-IR) spectra of fruit purees (classification – 0.9350 accuracy), Raman spectra of tablets (regression – 0.56 RMSE), and NIR spectra of frozen and thawed chicken (classification – 0.8760 accuracy).

It was found that using uninformative variable elimination algorithm and *TPOT* (using a search space of basic machine learning methods) can lead to building better models (purees, 0.9573 accuracy, tablets, 0.2769 RMSE). An optional step has been discovered which allows building a good pipeline for the chicken dataset (0.9333 accuracy). Compared with *TPOT*, the results obtained with the *AutoKeras* tool are poor or negligible.

Although the computing time of the search was not evaluated in this work, more complex models were not considered due to higher training times. The feasibility of search parallelization should be explored. Successful parallelization could lead to the applicable inclusion of more complex machine learning methods in the search space.

Įvadas

Klasikiniai medžiagų analizės metodai gali būti lėti, destruktyvūs, tyrimus būtina vykdyti laboratorijoje. Vis labiau populiarėja virpesinės spektrometrijos duomenų analizė, ši pasitelkiama kaip pakaitalas sudėtingiems metodams. Tokia analizė, dėl savo greičio bei lankstumo, gali pakeisti standartinius tyrimų metodus [33, 72]. Norint pasiekti kaip įmanoma tikslesnį galutinį duomenų analizės rezultatą, svarbu išmanyti reikalingus duomenų apdorojimo ir duomenų analizės metodus.

Dėl mėginio savybių ar matavimo sąlygų, virpesinės spektrometrijos duomenyse gali atsirasti įvairių neinformatyvių duomenų variacijų, triukšmų. Dėl šios savybės, duomenų analizės metodai gali veikti netiksliai [16]. Siekiant sutvarkyti duomenis, galima naudotis praktikoje taikomais normalizavimo, bazinės linijos korekcijos, triukšmo sumažinimo ar išvestinių metodais, kurių naudojimas daro reikšmingą įtaką galutiniam modelio rezultatyvumui [16, 25, 48].

Spektrometrijos duomenų aibę sudaro daug kolinearių požymių. Nors tokių duomenų analizei yra sukurta gretimų požymių koreliacijai nejautrių metodų [70], požymių atrinkimo žingsnis gali žymiai pagerinti sudaryto modelio rezultatyvumą [71, 55]. Šis žingsnis svarbus, jį praleidžiant apribojame mašininio mokymo modelių pasirinkimų aibę, todėl dar ir dabar taikymuose naudojama dalinių mažiausių kvadratų regresija, ar pagrindinių komponenčių analizė [59, 54].

Duomenų apdorojimo, požymių parinkimo bei regresijos / klasifikacijos modeliai sudaro ištisą procesų grandinę (angl. *pipeline*) (sudaryt procesų grandinė magistro darbe gali būti vadinama modeliu). Dalis metodų gali būti reguliuojami parametrų, keisti savo vietą grandinėje, todėl natūraliai kyla klausimas – kaip turėtų būti sudaromos šios grandinės. Šį klausimą nagrinėjo ne vienas autorius [16, 23, 57], aprašyti sprendimai priima išankstines prielaidas apie metodų išsidėstymą [16, 23], kuris šaltiniuose pateikiamas skirtingai [72, 23], ar optimizuoja perrenkant visą metodų aibę [23]. Nepaisant siūlomų metodų, apdorojimo grandinės būna sudaromos stebint spektro struktūros pokyčius ar perrenkant kelis galimus duomenų paruošimo metodus ir tikrinant galutinio modelio rezultatyvumą (įprastu atveju – dalinių mažiausių kvadratų regresijos) [47, 19, 26].

Darbo tikslas: paruošti automatizuotos hiperparametrų paieškos metodiką, skirtą virpesinės spektrometrijos duomenų analizei.

Darbo uždaviniai:

1. Peržvelgti virpesinės spektrometrijos duomenų analizėje naudojamus metodus, jų parametrus bei reikšmingumą;
2. Išnagrinėti mašininio mokymo procesus, naudojamus modelius;
3. Aptarti hiperparametrų paieškos metodus;
4. Surasti įrankius, leidžiančius automatizuoti virpesinės spektrometrijos duomenų analizės modelių sudarymą;
5. Paruošti duomenų aibes, kurios buvo naudojamos sprendžiant realius uždavinius;
6. Sudaryti hiperparametrų paieškos metodų ir parametrų aibes;
7. Parengti galimas paieškos metodikas;
8. Parengti bandymų aibę, leidžiančią tiksliai palyginti metodikų gaunamus rezultatus;

9. Apžvelgti gautus rezultatus, palyginant su duomenų tyrimuose gautais rezultatais, identifikuoti metodikų trūkumus.

Darbe trumpai aptariami virpesinės spektrometrijos metodai, supažindinama su spektrometrinių duomenų korekcijos būdais. Kitame skyriuje pristatomi mašininio mokymo eksperimentų vykdymo procesai bei modeliai, naudojami spektrometrinių duomenų analizei. Paskutiniame teorinės dalies skyriuje pateikiama informacija apie hiperparametrų optimizavimo metodus, kurie naudojami šiame darbe. Eksperimentinėje dalyje pristatomos duomenų aibės, bandymų aprašymai, pateikiami magistro tiriamosios dalies darbo rezultatai, jie palyginami su straipsniuose pateikiamais rezultatais.

Šiame darbe yra dalių, paimtų iš praeito semestro MTDP darbo:

- dalis santraukos (lietuvių ir anglų kalbomis);
- 1 skyriaus dalis apie virpesinės spektrometrijos metodus;
- 1.3 skyrius apie virpesinės spektrometrijos duomenų apdorojimo metodus (išskyrus iliustracijas);
- 2.1.2, 2.1.3 poskyriai apie pagrindinių komponenčių bei dalinių mažiausių kvadratų regresijas;
- 3 skyrius apie hiperparametrų paieškos metodus.

1. Virpesinė spektrometrija

Virpesinė spektrometrija – spektrometrijos rūšis, naudojanti elektromagnetinę spinduliuotę, sukeliančią dalelių virpesius. Virpesinės spektrometrijos analizė tampa vis populiaresnė tiriant įvairias medžiagos savybes, klasifikuojant skenuojamas medžiagas. Tokia analizė populiari dėl savo nedestruktyvaus pobūdžio ir tik minimalaus (arba jokio) mėginių paruošimo reikalaujančių tyrimų. Virpesinės spektrometrijos analizė gali būti taikoma daugelyje sričių, pvz.: tiriant mineralus [32], sprogmenis [34] ar bendrai taikoma kriminaliniams tyrimams [54] bei ypač populiarus taikymas maisto pramonėje [68, 49, 69, 73]. Tokia analizės sritis, kai cheminės savybės nustatomos netiesiogiai, o išmatuojant cheminės sistemos požymius (pvz.: infraraudonųjų spindulių sugerties spektras, Ramano spektras) bei pasitelkiant matematinius / statistinius modelius, dar vadinama chemometrija (angl. *chemometrics*).

1.1. Infraraudonųjų spindulių spektrometrija

Plačiausiai naudojama virpesinės spektrometrijos rūšis – infraraudonųjų spindulių spektrometrija (angl. *infrared spectroscopy* – IR). IR spektras dažniausiai gaunamas tiriant mėginio sugertį – infraraudonosios spinduliuotės šaltinio paveikta medžiaga sugeria dalį šios spinduliuotės. Spektrometras užfiksuoja, kiek spinduliuotės buvo sugerta, taip sudaromas sugerties spektras, simbolizuojantis tiriamos medžiagos molekulių virpesius. Gautą spektrą sudaro sugerties kiekis (reliatyvus vienetas) duotajame bangos ilgyje (bangos numeryje).

IR spinduliuotė apima spektrą, kurio bangos ilgiai yra intervale nuo 700 nm iki 1 mm, spektruose įprastai naudojami dažnio vienetai – atvirkštiniai centimetrai. Išreiškus šiais vienetais IR apibrėžtas intervale 14000 cm^{-1}– 10 cm^{-1}. Priklausomai nuo aparatinės įrangos, spektras fiksuojamas skirtinguose bangų ruožuose. IR spektrometrija, pagal apimamus bangų ruožus, gali būti skirstoma į tris dalis:

- artimosios infraraudonosios spinduliuotės spektrometrija (angl. *Near-infrared spectroscopy*– NIR). Apima bangų ruožą maždaug nuo 14000 cm^{-1} iki 4000 cm^{-1};
- viduriniosios infraraudonosios spinduliuotės spektrometrija (angl. *Mid-infrared spectroscopy*–MIR). Apima bangų ruožą maždaug nuo 4000 cm^{-1} iki 400 cm^{-1};
- tolimosios infraraudonosios spinduliuotės spektrometrija (angl. *Far-infrared spectroscopy*– FIR). Apima bangų ruožą maždaug nuo 400 cm^{-1} iki 10 cm^{-1}.

Nors pradžioje IR spektrometrija buvo naudojama tik kaip kokybinis metodas, tobulėjant matavimo prietaisams bei duomenų apdorojimo metodams, IR spektrometrija imta naudoti ir kaip kiekybinis metodas. Kiekybinės analizės pagrindą sudaro Beer-Lambert dėsnis, teigiantis, kad sugerties koeficientas tirpaluose tiesiškai priklauso nuo spinduliuotę sugeriančios medžiagos koncentracijos. Tiriant mėginius, keliama prielaida, kad Beer-Lambert dėsnis yra sudėtinis mišinio komponentų atžvilgiu [27].

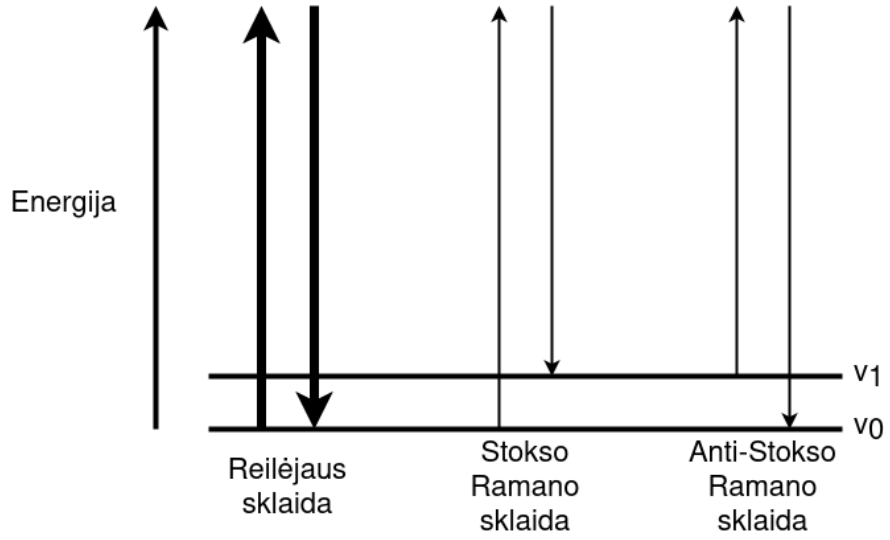
Iš minėtų regionų dažniausiai naudojamas MIR. MIR regionas apima 1500–650 cm^{-1} regioną, kuris laikomas molekuliniu pirštu atspaudo regionu. Tai leidžia atlikti palyginamąją analizę, nustatinėjant medžiagos kilmę. NIR regionas naudojamas atliekant medžiagos kiekybinę analizę. FIR regionas naudojamas tiriant medžiagos struktūrą ir gardelių dinamiką [27].

Norint atlikti IR spektrometrijos tyrimą, molekulės vibracijos turi būti aktyvios IR regione. Taip pat, dėl stiprios vandens infraraudonųjų bangų sugerties, įprasti IR spektrometrijos metodai apriboti atliekant vandeningų mėginių analizę.

1.2. Ramano spektrometrija

Ramano spektrometrija remiasi Chandrashekhar Venkat Raman 1928 metais aptiktu sklaidos efektu, pavadintu jo garbei. Skirtumas tarp IR ir Ramano spektrometrijos – pastaruoju atveju elektromagnetinė spinduliuotė nėra sugeriama mėginio, o dėl dalelių virpesio – išsklaidoma. Ramano sklaida – monochromatinės šviesos išsklaidymas medžiagoje, kurio metu dėl šviesos kvantų ir medžiagos molekulių sąveikos pakinta spinduliuotės dažnis.

Ramano spektrui gauti naudojamas didelės energijos šviesos šaltinis, kurio pagalba mėginio molekulės sužadinamos į aukštesnį energijos lygmenį. Molekulėms grįžus iš šio lygmens, vyksta sklaidos efektas. Jeigu išsklaidyta šviesa grįžta į pradinį energijos lygmenį – stebime Reilėjaus (angl. *Rayleigh*) sklaidą, jei grįžta į aukštesnį lygmenį – Stokso (angl. *Stokes*) sklaidą, jei pasiekė žemesnį lygmenį nei pradinį – anti-Stokso (angl. *anti-Stokes*) sklaidą. Gautas spektras rodo Ramano sklaidos intensyvumą šviesos dažnio (reliatyvaus naudojamam šviesos šaltiniui), išreikšto bangos numeriais, atžvilgiu. 0 bangos numeris simbolizuoja Reilėjaus, neigiami numeriai – anti-Stokso, o teigiami – Stokso sklaidas. Įprastu atveju spektrometras fiksuoja Stokso sklaidą.



1 pav. Energijos lygių diagrama

Ramano spektrometrijai naudojamas šviesos šaltinis (lazeris), kuris skirtingai nuo IR spektrometrijos, naudoja tik vieną (labai precizišką) bangos ilgį. Naudojamas įvairus lazerio bangos ilgis – nuo ultravioletinio ruožo, regimosios šviesos, iki artimosios infraraudonosios spinduliuotės. Ramano signalo intensyvumas atvirkščiai proporcingas lazerio ilgio ketvirtam laipsniui, todėl stipriausias signalas gaunamas naudojant trumpesnio bangos ilgio lazerius. Tokiu atveju mėginiuose gali pasireikšti fluorescencija, užgožianti silpną Ramano signalą [63].

Ramano spektrometrija patraukli, nes beveik nereikalauja mėginio paruošimo, gali būti skenuojami bet kokios agregatinės būsenos mėginiai. Ramano spektrometrija pasitelkiama narkotikų [15], sprogių medžiagų aptikimui [34], be to įrodyta, kad ši spektrometrija tinka ir organinių junginių identifikavimui [73].

Ramano spektrometriją galima naudoti kartu su IR, taip apeinant abiems metodams aktualias problemas. Kai kurie virpesiai, neegzistuojantys naudojant IR, gali būti stebimi Ramano spektrometrijos bandymuose ir atvirkščiai. Taip pat Ramano spektrometrija nėra jautri vandeningiems mėginiams bei geriau susitvarko su kietos būsenos mėginiais [37]. Atsižvelgiant į tai, kad IR problemų nekelia mėginių fluorescencija [62], šie metodai puikiai vienas kitą papildo.

1.3. Virpesinės spektrometrijos duomenų apdorojimas

Dėl įvairių mėginio fizikinių savybių, spektriniuose duomenyse gali pasireikšti nepageidaujamos variacijos, nesusijusios su cheminėmis medžiagos savybėmis. Tokios duomenų variacijos gali apsunkinti tinkamą regresijos, klasifikacijos modelių veikimą. Prieš atliekant duomenų analizę, nereikalingas variacijas būtina pašalinti. Metodų, gebančių pašalinti nereikalingus duomenų artefaktus, parinkimas gali skirtingai paveikti duomenų analizės rezultatus [25, 48]. Siekiant pagerinti galutinį rezultatą, reikalingos žinios, kaip šiuos metodus panaudoti tinkamai.

1.3.1. Normalizavimas

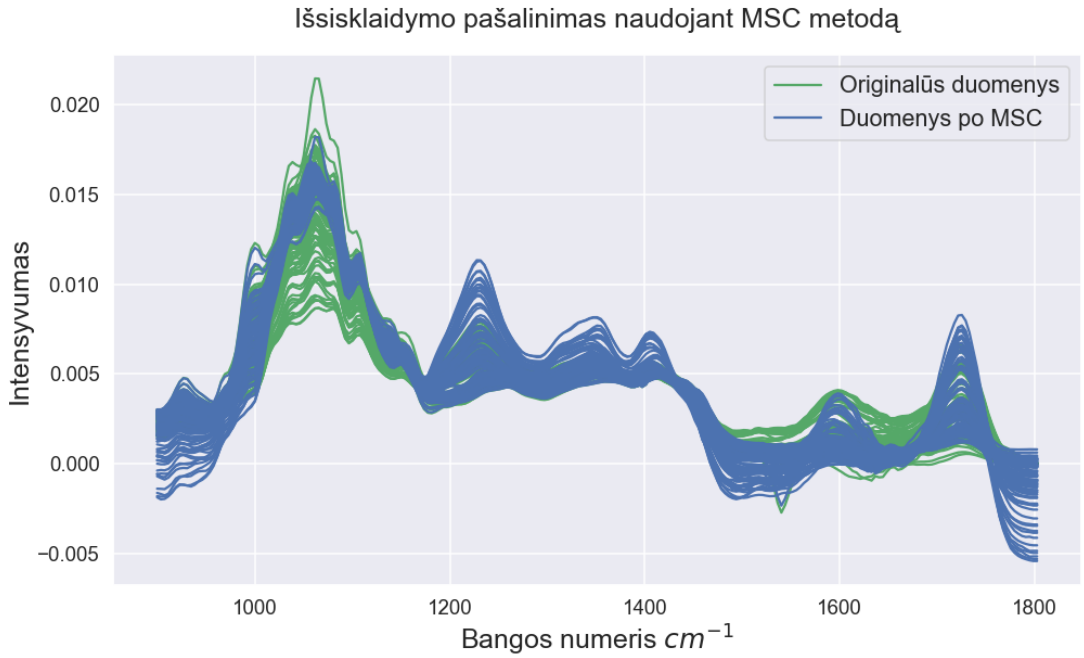
Stebint to paties mėginio spektrus, galima gauti kelis skirtingus rezultatus, kuriuose formos iš esmės nesiskirs. Tačiau gali skirtis viršūnių aukščiai, todėl būtina tokius spektrus normalizuoti, taip juos suvedant į panašią skalę.

Daugybinio išsibarstymo pataisymas (angl. *Multiplicative Scatter Correction–MSC*) naudoja spektrų vidurkį arba spektro etaloną, taip sumažindamas variacijas spektruose. X_{ref} – spektras, pagal kurį bus tvarkomi duomenys. Tuomet kiekvienam spektrui X_i bandomos rasti reikšmės a_i ir b_i, tokios, kad:

$$X_i \approx a_i + b_i X_{ref} \tag{1.1}$$

Kitaip tariant, šiuo metodu teigiama, kad bet kuris spektras sudarytas iš spektro etalono tiesinės sandaugos. Tai reiškia, kad spektras bus pataisytas, jeigu:

$$X_i^{MSC} = (X_i - a_i)/b_i \tag{1.2}$$



2 pav. Duomenys (iš 4.1.3 skyriaus) sutvarkyti naudojant MSC metodą.

Panašiai veikia standartinės normaliosios variacijos (angl. *Standard Normal Variate–SNV*) transformacija. Šis metodas paprastesnis – jam nereikalingas etalonas. Iš duoto spektro X_i atimamas jo vidurkis $\bar{X}_i$. Rezultatas padalinamas iš šio spektro reikšmių standartinio nuokrypio,

gaunama:

$$X_i^{SNV} = (X_i - \bar{X}_i)/\sigma_i \tag{1.3}$$

SNV ir MSC metodai pateikia labai panašų rezultatą – įrodyta, kad šių metodų rezultatus sieja tiesinė transformacija [17].

Taip pat gali būti naudojami vektorinio normalizavimo arba minmax normalizavimo metodai. Vektorinio normalizavimo metodas transformuoja duomenis taip, kad spektro reikšmių vektorinis ilgis būtų lygus 1.

$$X_i^{vector} = \frac{X_i}{|X|}, i \in \mathbb{N} \tag{1.4}$$

Čia $|X|$ – Euklidinis spektro vektoriaus ilgis, N – spektro reikšmių kiekis ir X_i – i-toji spektro reikšmė. Metodas kartojamas kiekvienam spektrui. Minmax metodo veikimo principas – spektrų reikšmių mastelis pakeičiamas į iš anksto apibrėžtą intervalą (dažniausiai intervalas yra [0,1]). min ir max apibrėžiami kaip intervalo rėžiai, o X_{min} ir X_{max} atitinkamai kaip spektro mažiausia ir didžiausia vertės.

$$X'_{minmax} = \frac{X - X_{min}}{X_{min} - X_{min}} \tag{1.5}$$

$$X^{minmax} = X'_{minmax} * (max - min) + min \tag{1.6}$$

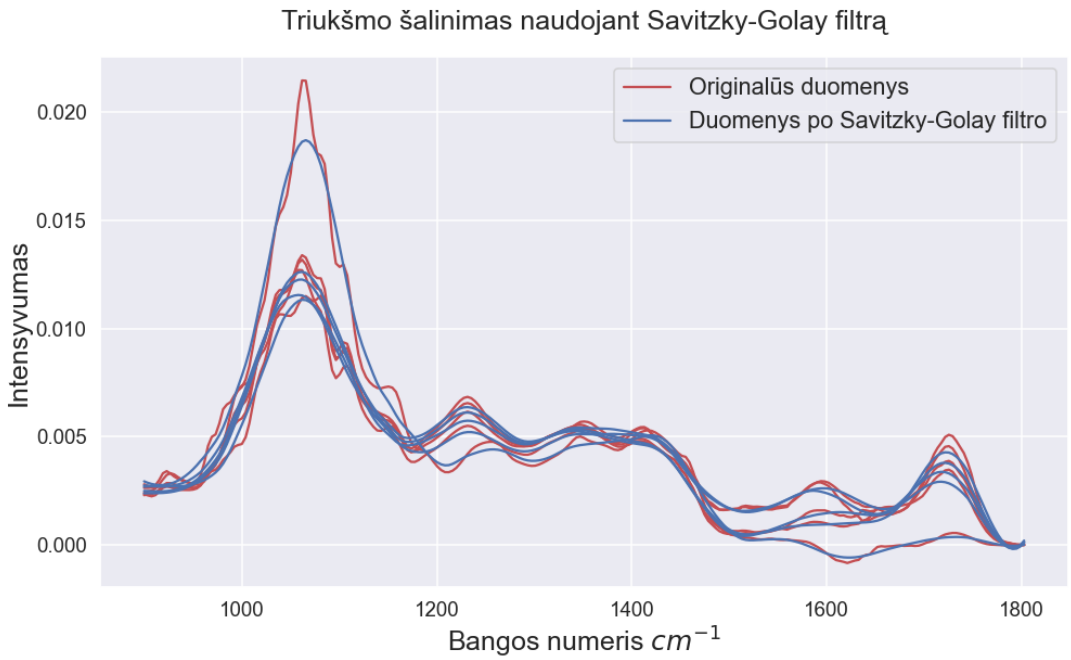
Metodas kartojamas kiekvienam spektrui.

1.3.2. Triukšmo pašalinimas

Spektriniuose duomenyse dažnai atsiranda triukšmas. Šios duomenų variacijos gali atsirasti dėl įvairių prietaiso ar mėginio savybių. Savitzky-Golay filtras [53] gali padėti jį pašalinti. Šio filtro veikimo principas:

1. Parenkamas n dydžio langas;
2. Parenkamas m laipsnio polinomas, labiausiai aproksimuojantis šį langą;
3. Spektriniai duomenys sukeičiami su polinomo duomenimis.

Šis metodas gali būti naudojamas nurodant išvestinės laipsnį (dažniausiai naudojamos pirmo arba antro laipsnio išvestinės). Išvestinės padeda išvengti bazinės linijos postūmių sukeliamų problemų, tačiau apsunkina vizualų duomenų analizavimą.



3 pav. Duomenys (iš 4.1.3 skyriaus) sutvarkyti naudojant Savitzky-Golay filtrą.

1.3.3. Bazinės linijos korekcija

Spektrus gali būti sunku palyginti dėl bazinės linijos postūmių, kurie neretai (ypač Ramano spektrometrijoje) gali būti netiesiniai. Šiems reiškiniams kompensuoti naudojami tokie metodai:

- funkcijos pritaikymu paremti metodai

  Šie metodai paremti kokios nors funkcijos pritaikymu duotam įvesties spektro vektoriui X. Metodo sprendinys – funkcijos kreivė, kurios klaidos kvadratas (tarp gautos kreivės ir pradinio spektro) yra minimalus. Koreguotas spektras gaunamas iš spektro atėmus pritaikytą funkcijos kreivę;

- guminės juostos (angl. *Rubberband*) [50] metodas

  Metodo esmė – rasti spektrą gaubiantį daugiakampį (angl. *convex hull*) . Tuomet daugiakampio ir spektro sąlyčio taškuose atliekama interpoliacija splainais – taip gaunama bazinė linija. Metodui nereikia jokių parametrų, tačiau metodas tinkamiausias naudoti su spektrais, kurių galai yra „išgaubti“;

- asimetrinių mažiausių kvadratų metodas [24]

  Asimetrinių mažiausių kvadratų (angl. *Asymmetric Least Squares*) metodo (toliau ALS) esmė – surasti kreivę, kuri su mažiausia paklaida gali aproksimuoti įvesties duomenis, bet kartu kreivė turi būti glodi. Tokie reikalavimai susiveda į funkciją:

$$S = \sum_i w_i (y_i - z_i)^2 + \lambda \sum_i (\delta^2 z_i)^2 \tag{1.7}$$

$$\delta^2 z_i = z_i - 2z_{i-1} + z_{i-2} \tag{1.8}$$

  Čia y_i ir z_i – atitinkamos spektro ir bazinės linijos reikšmės. Pirmasis (1.7 formulės) sumos dėmuo parodo, kiek kreivė panaši į duomenų vektorių, o antrasis dėmuo – baudos funkcija,

parodanti, ar bazinė linija glodi. λ parametras skirtas subalansuoti 1.7 formulės dėmenis. w yra svorių vektorius, kurio sudarymui reikalingas parametras p:

$$w_i = \begin{cases} p, & \text{jei } y_i > z_i \\ 1-p, & \text{kitu atveju} \end{cases} \tag{1.9}$$

1.7 formulės minimizavimo problema priveda prie lygčių sistemos:

$$(W + \lambda D'D)z = Wy) \tag{1.10}$$

Čia $W = diag(w)$ – svorių matricos įstrižainė, D – skirtumų matrica, kur $Dz = \delta^2 z$. Algoritmas vykdomas nurodytą iteracijų skaičių, kiekvienoje iteracijoje svoriai atnaujinami. ALS algoritmas turi du parametrus: p ir λ, kuriuos keičiant kinta ir bazinės linijos forma. Didinant parametro p reikšmę, bazinė linija panašėja į patį spektrą, o didinant λ reikšmę – panašėja į tiesę. Dažniausiai p reikšmės yra parenkamos iš intervalo $0.001 \leq p \leq 0.1$, o λ reikšmės – $10^2 \leq \lambda \leq 10^9$;

- adaptyvus iteratyviai atsvertų penalizuotų mažiausių kvadratų (angl. *adaptive iteratively reweighted penalized least squares*) metodas [76]

  Adaptyvus iteratyviai atsvertų mažiausių kvadratų metodas (toliau AIRPLS) iš esmės panašus į ALS. AIRPLS naudoja kitokį svorių priskyrimo būdą bei kitokią baudos reikšmę, kurios pagalba kontroliuojamas bazinės linijos glodumas [9]. Svorių vektorius w laiko momentu t yra randamas iteratyviai:

$$w_i = \begin{cases} 0, & \text{jei } y_i \geq z_i \\ e^{\frac{t(y_i - z_i}{|d|}}, & \text{jei} y_i < z_i \end{cases} \tag{1.11}$$

  Čia d sudarytas iš $y-z$ elementų. Iteracijos vykdomos, kol pasiekiamas maksimalus galimas iteracijų skaičius arba kol įgyvendinama 1.12 sąlyga:

$$|d| < 0.001 \times |y| \tag{1.12}$$

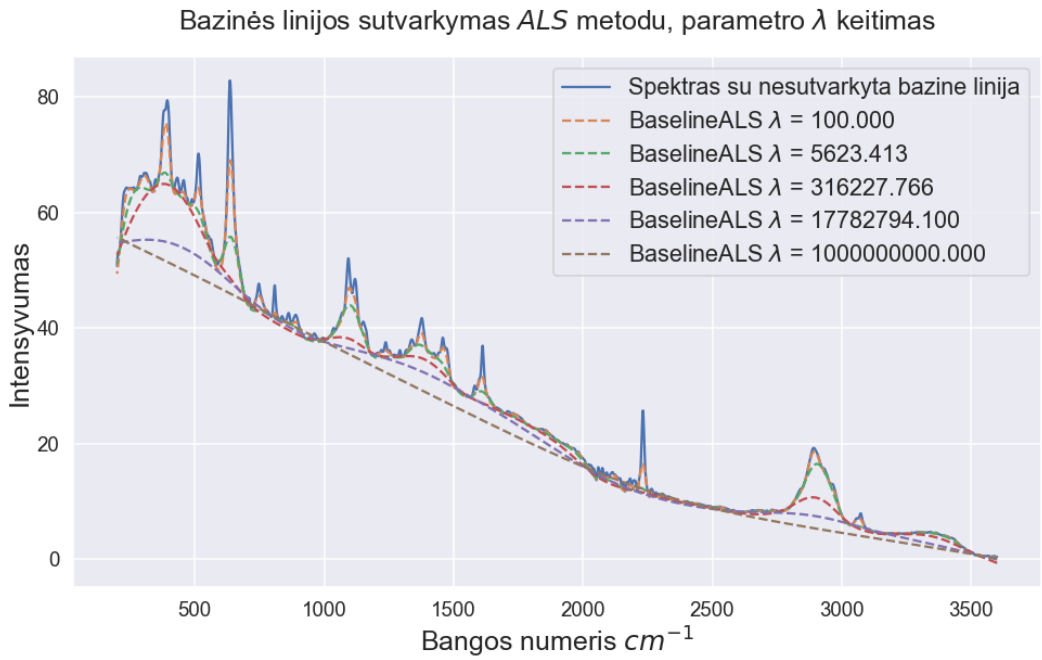
- asimetriškai atsvertų penalizuotų mažiausių kvadratų (angl. *asymmetricaly reweighted penalized least squares*) metodas [8]

  Asimetriškai atsvertų penalizuotų mažiausių kvadratų (toliau ARPLS) metodas taip pat panašus į ALS ir AIRPLS, šiuo atveju skiriasi svorių vektoriaus w sudarymo metodas:

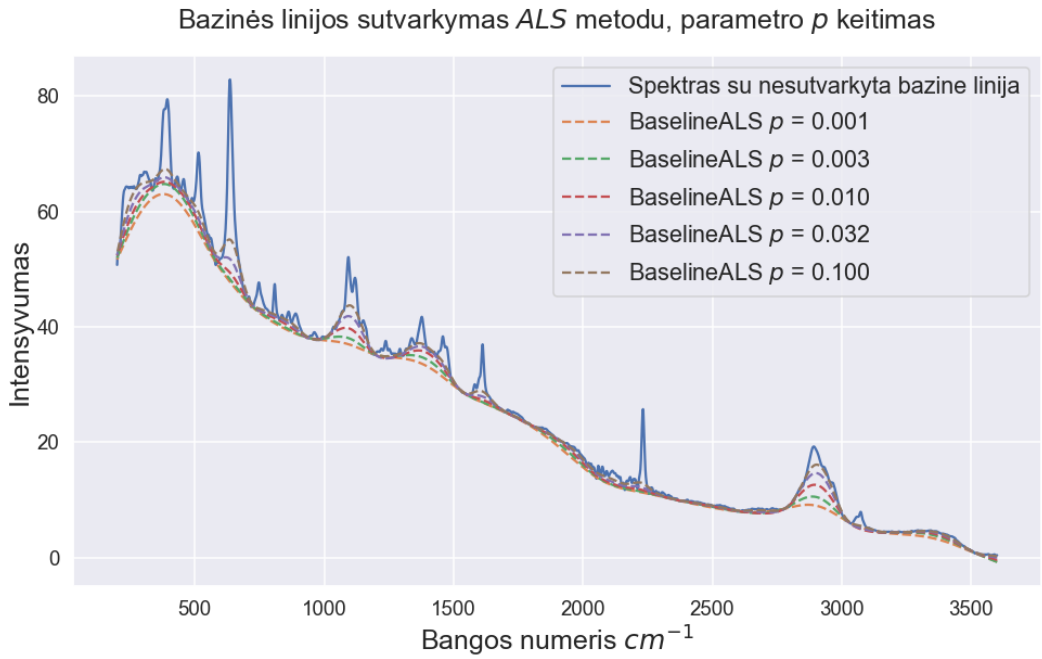
$$w_i = \begin{cases} logistic(y_i - z_i, m_{d'}, \sigma_{d'}), & \text{jei } y_i \geq z_i \\ 1, & \text{jei } y_i \leq z_i \end{cases} \tag{1.13}$$

  Čia $m_{d'}$, $\sigma_{d'}$ yra atitinkamai d' vidurkis ir standartinis nuokrypis. $d = y - z, d'$ yra d dalis, apibrėžta regione, kur $y_i < z_i$. Funkcija $logistic$ yra apibrėžta:

$$logistic(d, m, \sigma) = \frac{1}{1 + e^{2(\frac{d-(-m+2\sigma)}{\sigma})}} \tag{1.14}$$



(a) λ parametro įtaka algoritmo veikimui



(b) p parametro įtaka algoritmo veikimui

4 pav. Duomenys (iš 4.1.1 skyriaus) sutvarkyti naudojant skirtingus ALS bazinės linijos korekcijos algoritmo parametrus.

1.3.4. Spektrinių požymių atrinkimas

Spektrinių požymių atrinkimas – svarbus žingsnis, galintis nulemti galutinio analizės rezultato tikslumą. Xiaobo et al. 2012 metų spektrinių požymių atrinkimo metodų apžvalgoje [71] teigia, kad spektrinių požymių atrinkimas svarbus patikimo statistinio modelio sudarymui. Šis žingsnis panaikina gretimų požymių koreliacijas, kolinearumą. Nors naudojant kai kuriuos modelius

duomenų analizę galima atlikti ir visam spektrui, požymių išrinkimas pagerina ir kolineariems duomenims nejautrių modelių rezultatą [55].

Duomenis tiriantis žmogus turėtų žinoti, kurie požymiai yra reikšmingi uždaviniui. Naują užduotį sprendžiantis žmogus gali nežinoti, kuriuos požymius reikia pasirinkti. Dėl šios priežasties sukurtas ne vienas metodas, galintis pateikti informaciją apie spektrinių požymių svarbą duotajam uždaviniui. Xiaobo apžvalgoje mini aibę metodų, besiremiančių spektro padalinimu į atskirus langus. Tokie metodai gali apspręsti, kurie spektro intervalai svarbesni, turi daugiau esminės informacijos. Dažniausias tokių metodų veikimo principas:

1. Spektras suskaldomas į n (vienodo arba nevienodo dydžio) langų;
2. Vienam arba keliems langams yra įvertinamas PLS (2.1.3 skyrius) modelio rezultatyvumas;
3. Pagal gautą rezultatyvumą atrenkami tik geriausi regionai / regionų kombinacijos.

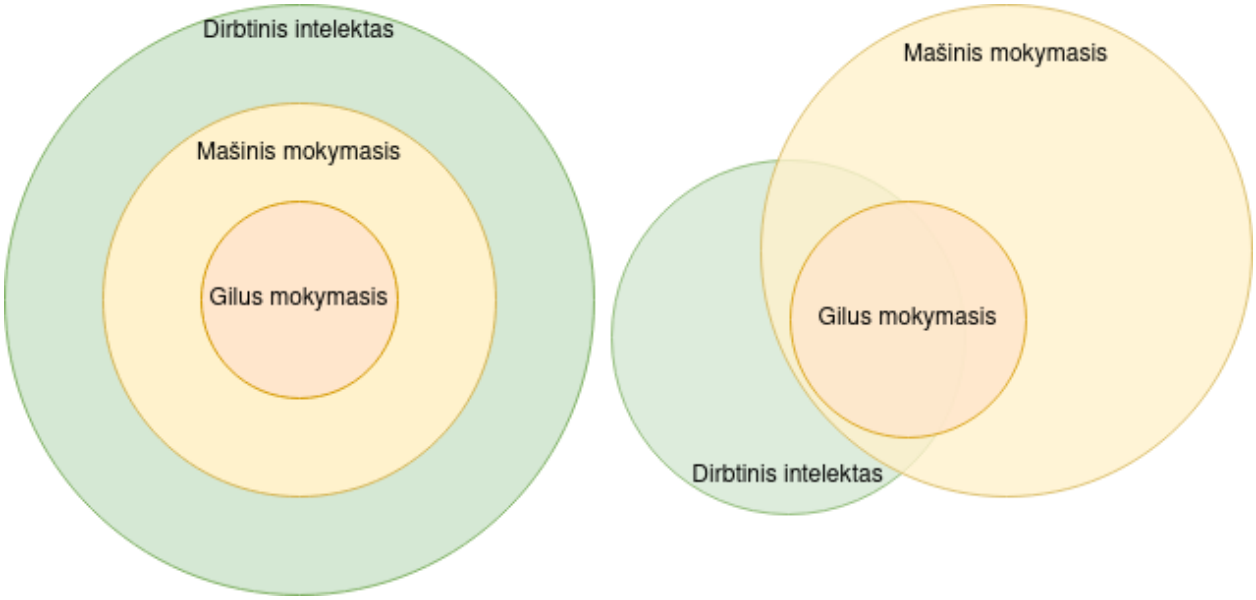
Taip pat požymių atrinkimui galima naudoti metodus, sutinkamus optimizavimo uždaviniuose. Vienas iš tokių metodų – genetinis algoritmas (3.3) – jis taip pat yra minimas Xiaobo apžvalgoje. Genetinio algoritmo panaudojimas spektrų ruožų aptikimui bus aptartas 3.3.2 skyriuje. Genetinis algoritmas įprastai remiasi spektro dalinimo į atskirus langus idėja.

Dar viena apžvalgoje siūloma idėja – neinformatyvių kintamųjų atmetimas (angl. *uninformative variable elimination – UVE*). Metodo esmė – sugeneruojamas atsitiktinis triukšmas, jis pridedamas į PLS apmokymo aibę. Tuomet sudaromas modelis ir užfiksuojami spektriniai ruožai, kurie yra mažiau informatyvūs nei atsitiktinis triukšmas. Vėliau procesas kartojamas, kol pasiekiamas tam tikras apibrėžtas kriterijus [12]. Dėl atsitiktinai atliekamų bandymų, šis metodas dar vadinamas Monte Karlo neinformatyvių kintamųjų atmetimu (angl. *Monte Carlo uninformative variable elimination – MCUVE*)

2. Mašininis mokymasis

Dažnai įprastų programų veikimo principus apibūdina pats programuotojas – įvertinęs galimas vartotojų įvestis, sudaro galimų išvesčių aibę. Mašininio mokymosi uždavinys – sudaryti programą, kuriai, norint sudaryti išvesties taisykles, nereikalingas programuotojas. Tokios programos mokosi naudodamos iš anksto paruoštas duomenų įvesties ir išvesties poras, pagal kurias sudaromos taisyklės, kaip ateityje vertinti įvestus duomenis.

Mašininio mokymosi apibrėžimas literatūroje randamas jau nuo 1959 metų, šio termino autoriumi laikomas Arthur L. Samuel, savo studijoje aprašęs mašininio mokymosi panaudojimo galimybes žaidžiant šaškėmis [51]. Autorius patvirtina, kad programa gali būti sudaryta taip, kad kompiuteris gali išmokti šaškėmis žaisti geriau nei buvo užprogramuotas. Mašininis mokymas gana sena sąvoka, tačiau iki šiol dar nesutariama, ar visas mašininis mokymas yra dirbtinio intelekto poaibis [46, 22].



5 pav. Du skirtingi požiūriai į mašininio mokymosi išsidėstymą dirbtinio intelekto kontekste. Pritaikyta pagal [46, 22].

Mašininio mokymosi pagrindą sudaro modeliai, kurie pateikus įvestį grąžina kažkokį rezultatą. Įprastai mašininio mokymosi modeliai skirstomi į tris pagrindines kategorijas:

- **prižiūrimas mokymasis** (angl. *supervised learning*). Šio tipo modeliams pateikiamos įvesties – požymių (angl. *features*) ir išvesties – žymių (angl. *labels*) aibė. Tokio tipo modeliai gali spręsti klasifikacijos (kai žymės yra diskrečios, pvz.: ar pateiktas el. laiškas yra brukalas) ir regresijos (kai žymės yra tolydžios, pvz.: numatoma namo kaina) uždavinius.

  Vienas paprasčiausių regresijos modelių – tiesinė regresija, bandoma rasti $y_i = \alpha + \beta x_i + \epsilon$ lygties α, β reikšmes, minimizuojant paklaidos ϵ reikšmę.

  Sprendimų medis yra klasikinis, lengvai suprantamas klasifikavimo atvejis, kai atrenkami požymių rėžiai, leidžiantys suskirstyti duomenų aibę. Taip sukuriamos savotiškos taisyklės, leidžiančios nustatyti, kokiai klasei reikia priskirti mėginį.

- **neprižiūrimas mokymasis** (angl. *unsupervised learning*). Šio tipo modeliai skirti ne nuspėti kažkokią reikšmingą žymę, bet surasti užslėptas duomenų požymių struktūras, taip pagilinant žinias apie nagrinėjamą objektą.

Naudojant neprižiūrimojo mokymosi modelius, galima spręsti klasterizavimo uždavinius, kai duomenų požymiai pagal tam tikras taisykles gali būti suskirstomi į grupes. Vienas paprasčiausių šio tipo modelių – *K-Means* – jis duomenų vektorius suskirsto aplink *k* skaičių centrų, pagal iš anksto nurodytą atstumo metriką.

Kitas šio tipo modelių sprendžiamas uždavinys – dimensijų mažinimas, kai požymių aibėje identifikuojami svarbiausi elementai, kurie gali būti atvaizduoti žemesnėje dimensijoje. Vienas tokių modelių jau minėtas PCA modelis, užfiksuojantis didžiausios duomenų variacijos kryptis, pastarosios suformuoja naujos dimensijos ašis.

- **skatinamasis mokymasis** (angl. *reinforcement learning*). Šio tipo modelių užduotis – pagal tam tikrą būseną įvertinti geriausią atsaką (kaip veiksmą). Apmokant tokio tipo modelius pateikiamas scenarijus, kurio metu agentas turi priimti sprendimą. Už priimto sprendimo reikšmingumą agentas apdovanojamas. Vienas tokių modelių yra AlphaGo [1], įveikęs geriausius *Go* žaidėjus pasaulyje. Tam, kad šis modelis būtų apmokytas, kūrėjai jį supažindino su dideliu skaičiumi žaidimų. Kai modelis jau gebėjo žaisti žaidimą pats, jis buvo suporuojamas su savo ankstesne versija, taip mokydamasis iš savo klaidų.

2.1. Mašininio mokymo modeliai

Šiame skyriuje aptariami metodai, naudojami spektrometrinių duomenų analizėje. Klasikinį chemometrijos mašininių mokymo metodų pagrindą sudaro tokie modeliai kaip: daugianarė tiesinė regresija, pagrindinių komponenčių regresija ar dalinių mažiausių kvadratų regresija, kurie jau ilgą laiką naudojami nustatinėjant įvairias mėginių savybes [35, 14, 70]. Toliau aptariami metodai, atrinkti iš komercinių spektrometrijos analizės įrankių [2, 5] ir / arba atliktų tyrimų [7, 30, 44, 60, 10, 58]. Atrinkti santykinai (mašininio mokymo metodų kontekste) paprastesni metodai (su dirbtinių neuroninių tinklų išimtimi), taip palengvinant tolesniuose skyriuose aptariamus bandymus.

2.1.1. Tiesiniai modeliai

Tiesiniai modeliai ieškomą reikšmę nustato pagal tiesinę nurodytų požymių kombinaciją:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \cdots + w_p x_p \tag{2.1}$$

Čia w – tiesinės lygties koeficientai. Tokia regresija, kai spėjama reikšmė yra priklausoma nuo kelių kintamųjų, vadinama daugianare tiesine regresija (angl. *multiple linear regression – MLR*). Tiesiniai modeliai iš esmės skiriasi pagal optimizavimo uždavinį:

- *tiesinė regresija*
$$\min_w ||Xw - y||_2^2 \tag{2.2}$$
- *ridge regresija*
$$\min_w ||Xw - y||_2^2 + \alpha||w||_2^2 \tag{2.3}$$
- *lasso regresija*
$$\min_w \frac{1}{2n}||Xw - y||_2^2 + \alpha||w||_1 \tag{2.4}$$

- *elastinio tinklo regresija*

$$\min_w \frac{1}{2n}||Xw - y||_2^2 + \alpha\rho||w||_1 + \frac{\alpha(1-\rho)}{2}||w||_2^2 \tag{2.5}$$

Čia $||x||_1$ ir $||x||_2$ atitinkamai žymi pirmą ir antrą vektoriaus normas (Manhateno ir Euklido vektoriaus ilgiai), w – tiesės funkcijos svorių vektorius, X – požymių vektorius, y – ieškoma reikšmė, kitos reikšmės – reguliuojami parametrai.

2.1.2. Pagrindinių komponenčių regresija

Pagrindinių komponenčių analizė (angl. *Principal component analysis*) – PCA – gali sutvarkyti problemas, atsiradusias daugianarėje tiesinėje regresijoje. PCA metodo esmė – atlikti tiesinę duomenų projekciją į žemesnę dimensiją, išsaugant kuo didesnį originalių duomenų variacijos kiekį. Tam, kad būtų atlikta tokia projekcija, reikia žinoti požymių koreliacijos arba kovariacijos matricas, tuomet šias matricas išskaidyti tikrinėmis reikšmėmis (angl. *eigenvalue decomposition*).

Atlikus duomenų projekciją ir išsaugojus tik svarbias komponentes, galima atlikti tiesinę regresiją. Toks metodas (komponenčių radimas ir regresija) vadinamas pagrindinių komponenčių regresija (angl. *principal components regression*) – PCR. Pagrindinė šio PCR problema – PCA metodas neužtikrina, kad išsaugotos komponentės atsako už duomenų variacijas Y matricoje, nes PCA tikslas – tik išsaugoti komponentes, kurios paaiškina didžiausias X variacijas. Kitaip tariant – didžiausios X variacijos nebūtinai turės sąryšį su Y matrica.

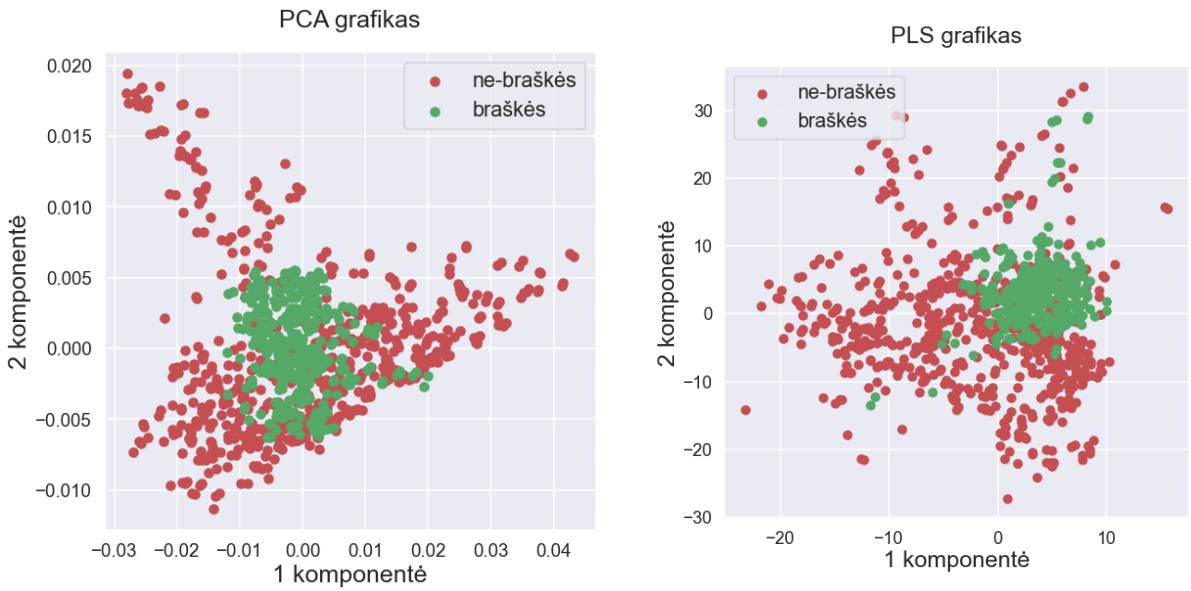
2.1.3. Dalinių mažiausių kvadratų regresija

Vienas pagrindinių chemometrijoje naudojamų mašininio mokymosi modelių – PLSR – dalinė mažiausių kvadratų regresija (angl. *partial least squares regression*). Šio metodo sėkmingumą, naudojant spektrometrijos duomenis, lemia metodo gebėjimas nagrinėti didelį kiekį triukšmingų, koreliuojančių, kolinearių ar net trūkstamų savybių [70]. PLSR metodas iš esmės apjungia PCA ir daugianarės tiesinės regresijos savybes – metodo naudojimo metu randama ne tik Y priklausomybė nuo X, šis metodas taip pat sumodeliuoja šių matricų struktūras (kaip PCA). Pagrindinis PLSR modelis sudarytas iš dviejų lygčių:

$$\begin{aligned} X &= TP^T + E \\ Y &= UQ^T + F \end{aligned} \tag{2.6}$$

X– $n \times m$ dydžio požymių matrica, Y – $n \times p$ dydžio priklausomų reikšmių matrica, T, U – $n \times l$ dydžio X ir Y projekcijos, P, Q – $m \times l, p \times l$ dydžio pakrovimo (angl. *loading*) matricos, E, F – paklaidos. X ir Y turi būti suskaldomi taip, kad kovariacija tarp T ir U būtų maksimali. T, U, P, Q reikšmėms rasti egzistuoja skirtingi metodai.

Lengva pastebėti, kuo šis metodas panašus ir kuo pranašesnis už PCA rastų komponenčių regresiją. Čia, atliekant projekciją, atsižvelgiama tiek į požymių, tiek į priklausomų reikšmių matricas, todėl išsaugomos komponentės paaiškina ne didžiausią variaciją X matricoje, bet kovariaciją tarp X ir Y matricų.



(a) PCA transformacija

(b) PLS transformacija

6 pav. PLS ir PCA skirtumas 4.1.3 skyriaus duomenų aibei. PCA didžiausi regimi komponenčių skirtumai ortogonalūs.

2.1.4. Sprendimų medžiai

Sprendimų medžiai išveda paprastas, lengvai interpretuojamas taisykles. Pats klasifikatorius veikia medžio principu – kiekviena šaka yra taisyklė, pagal jas pasiekiami lapai, pažymėti atitinkamomis klasėmis. Medis sudaromas randant savybę bei jos reikšmę, kuri geriausiai atskiria duomenų klases. Toks viršūnių skirstymas gali būti kartojamas, kol neišpildoma viena iš galimų sustojimo sąlygų:

- visos viršūnės išgrynintos;
- visos šakos pasiekusios maksimalų galimą gylį;
- pasiektas maksimalus viršūnių skaičius;
- nebeįmanoma plėsti medžio (dėl kitų reikalavimų).

Viršūnė tampa gryna, kai apmokymo metu į šią viršūnę patenka tik vienos klasės duomenys, tokios viršūnės Gini priemaiša (angl. *Gini impurity*), kurios skaičiavimas pateikiamas 2.7 formulėje, bus 0:

$$I_G(p) = \sum_{i=1}^{J}(p_i \sum_{k \neq i}^{J} p_k) \tag{2.7}$$

Čia J – klasių skaičius, $i \in \{1, 2, \cdots, J\}$ ir p_i yra dalis mokymo aibės elementų, pažymėtų kaip i klasė.

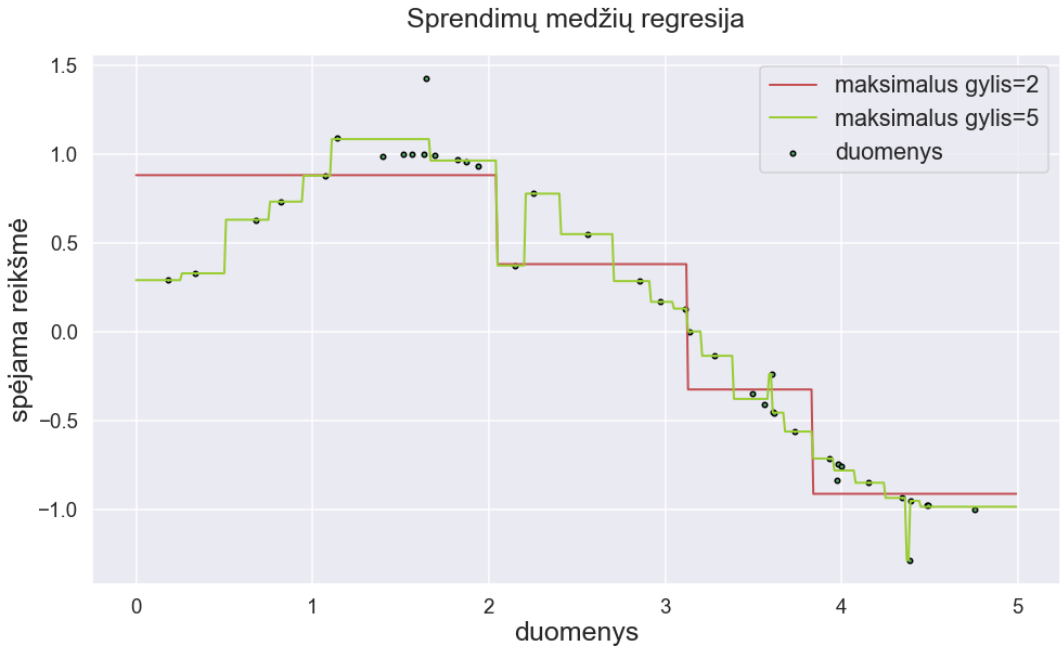
Neribojant medžio gylio, maksimalaus viršūnių skaičiaus, minimalaus kiekio lapo sudarymui ar kitų parametrų, galime gauti medį, kurio visi lapai bus išgryninti, tačiau kiekvieną lapą sudarys po vieną duomenų elementą. Tokiu atveju iškyla didelė persimokymo (angl. *overfitting*) rizika.

Vienas iš reikšmingiausių modelio privalumų – lengvas interpretavimas. Atvaizdavus sprendimų medį, galima identifikuoti modelio sudarytas taisykles ir įgauti įžvalgų apie duomenis. Tačiau tokia modelio analizė įmanoma tik turint santykinai tvarkingą, per daug neišsiplėtusį, medį.

Sprendimų medis tinka ir regresijai. Atliekant regresijos analizę, šakos dažniausiai formuojamos pagal dispersijos mažinimo kriterijų, kai padalijimas parenkamas su mažiausia σ^2 reikšme:

$$\sigma^2 = \frac{\sum(\chi - \mu)^2}{N} \tag{2.8}$$

Atliekant regresijos analizę, lape esančių reikšmių vidurkis bus priskirtas įvestam duomenų vektoriui (žr. 7 paveikslą).

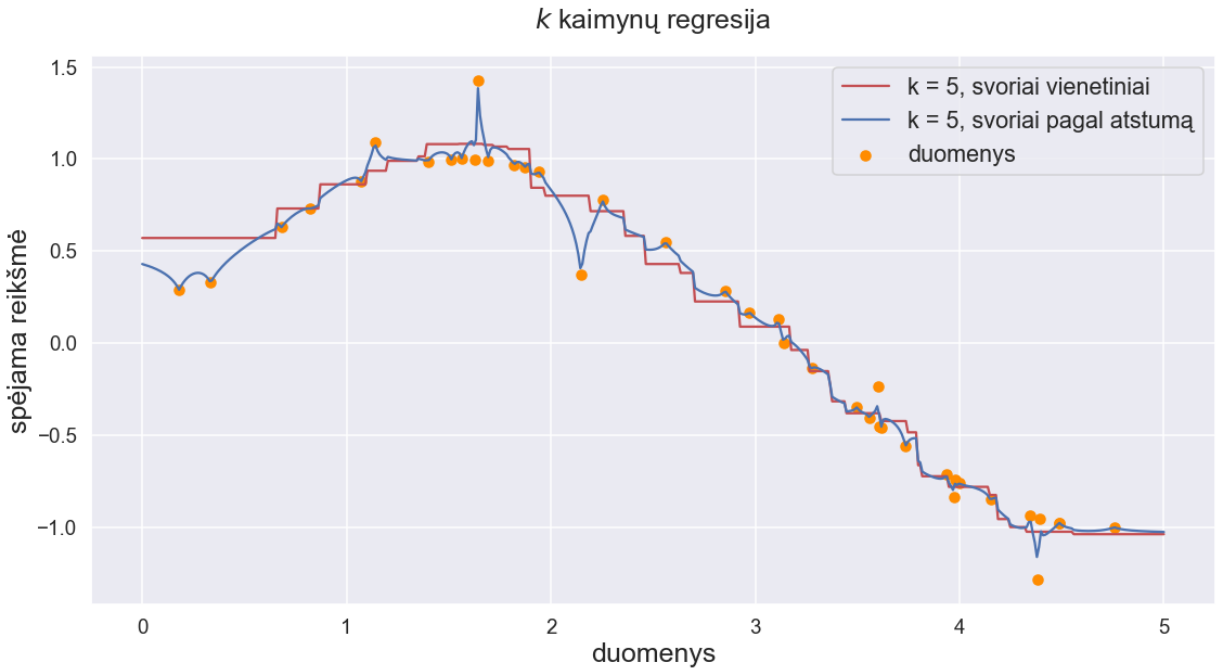


7 pav. Sprendimų medžių regresija atsitiktinei duomenų aibei. Pritaikyta pagal [3].

2.1.5. *k* artimiausių kaimynų modeliai

k artimiausių kaimynų modelis – dar vienas nesudėtingas duomenų analizės modelis, kurį naudojant galima spręsti tiek klasifikavimo, tiek regresijos problemas. Iš esmės, šis modelis nesimoko, o saugo apmokymo aibės taškus. Pateikus klasifikacijos užklausą, modelis klasę parenka pagal tai, kokioms klasėms priklauso *k*, pagal tam tikrą metriką arčiausių, taškų. Artimiausi kaimynai įprastai turi po vieną balsą, tačiau balsavimo vertę galima nustatyti ir pagal atstumą iki taško.

Regresijos atveju modelis veikia labai panašiai, tačiau čia reikšmė būna prilyginama artimiausių taškų vidurkiui. Taip pat, kaip ir klasifikavimo atveju, šios funkcijos svorius galima pareguliuoti pagal atstumus nuo taškų.



8 pav. k artimiausių kaimynų regresija atsitiktinei duomenų aibei iš 7 paveikslo. Pritaikyta pagal [4].

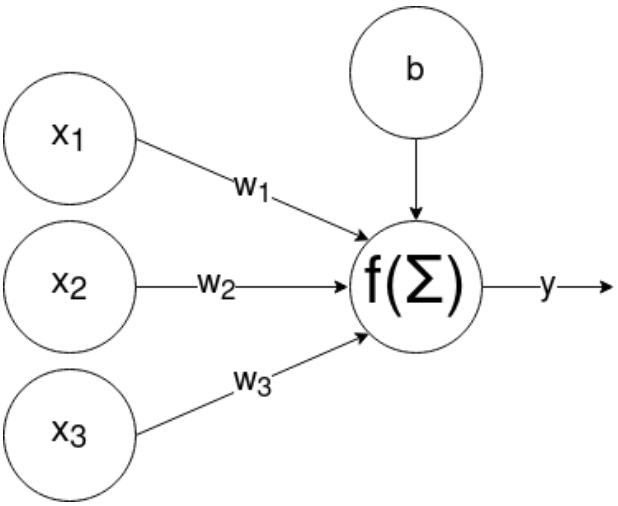
2.1.6. Dirbtiniai neuroniniai tinklai

Dirbtiniai neuroniniai tinklai (DNT) (angl. *artificial neural networks*) – skaičiavimų modelis, taikomas sprendžiant tiek regresinius, tiek klasifikavimo uždavinius. Šio tipo skaičiavimai paremti smegenyse vykstančiais procesais, kai skirtingai stimuliuojami neuronai perduoda signalą kitiems neuronams. DNT modeliai sėkmingai pritaikyti sprendžiant ir virpesinės spektrometrijos problemas [7, 30, 44].

Paprasčiausias tokio tipo modelis – perceptronas (žr. 9 paveikslą) – kurį galima aprašyti taip:

$$y = f(\sum_{i=1} w_i x_i + b) \tag{2.9}$$

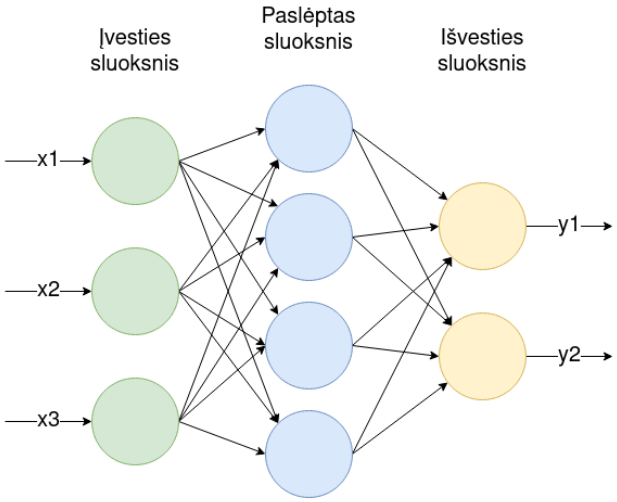
Čia f – tam tikra netiesinė funkcija (dar vadinama aktyvacijos funkcija), w – perceptronų svoriai, x – įvesties reikšmės, o b – nepriklausoma reikšmė. Pradžioje buvo naudojamos paprastos aktyvacijos funkcijos, leidžiančios spręsti tik binarinio klasifikavimo uždavinius, po to aktyvavimo funkcijų aibė vis plėtėsi [39].



9 pav. Perceptronas su trimis įvesties reikšmėmis

Perceptrono idėja gali būti išplėsta apjungus kelis perceptroną primenančius neuronus į vieną sluoksnį ir juos apjungiant gaunamas daugiasluoksnis perceptronas (angl. *multilayer perceptron*), kuris yra vienas populiariausių ir dažniausiai naudojamų neuroninių tinklų modelių [43]. Sluoksniai, sudarantys daugiasluoksnį perceptroną, dar vadinami pilnai sujungtais (angl. *fully connected*) sluoksniais.

Daugiasluoksnis perceptronas priklauso tiesioginio sklidimo (angl. *feed-forward*) neuroninių tinklų tipui. Tokiuose tinkluose duomenys sklinda tik į vieną pusę – iš įvesties sluoksnio į išvesties. Daugiasluoksnį perceptroną gali sudaryti daug sluoksnių – sluoksniai, kurie nėra nei įvesties, nei išvesties sluoksniai, vadinami paslėptais (angl. *hidden*) sluoksniais. Kiekvienas neuronas yra sujungtas su sekančio sluoksnio neuronais. Tiesioginio sklidimo neuroninių tinklų mokymas vykdomas naudojantis atgalinio klaidos skleidimo (angl. *backpropagation*) algoritmu. Atgalinio klaidos sklidimo idėja – įvertinti klaidos išvestines tinklo svorių atžvilgiu. Šiuo būdu galima nustatyti svorių įtaką paklaidos kitimo greičiui, taip nustatoma, kaip turi būti perdėlioti tinklo svoriai.



10 pav. Daugiasluoksnis perceptronas

Daugiasluoksnio perceptrono (ar kito DNT modelio) panaudojimo atvejai gali skirtis nuo parenkamų aktyvacijos funkcijų [39]. Vienos dažniausiai naudojamų aktyvacijos funkcijų:

- sigmoido funkcija:

$$f(x) = (\frac{1}{1+e^{-x}}) \tag{2.10}$$

- hiperbolinis tangentas (Tanh):

$$f(x) = (\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}) \tag{2.11}$$

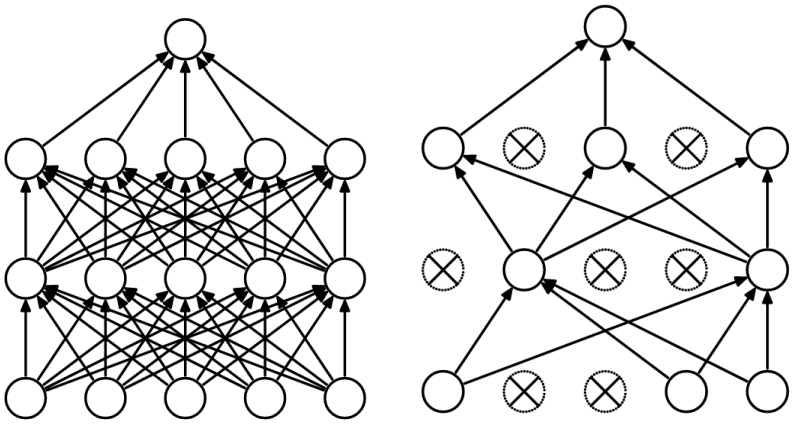
- Softmax:

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{j}} \tag{2.12}$$

- ReLU:

$$f(x) = max(0, x) \tag{2.13}$$

DNT gali būti sudaromas ir su kitokio tipo sluoksniais. Atmetimo (angl. *dropout*) sluoksnis naudojamas norint generalizuoti neuroninį tinklą. Atmetimo sluoksnio esmė – apmokymo metu atjungti kai kuriuos neuronus. Naudojimo metu atmetimo sluoksniai neatjunginėja neuronų, todėl svoriai bus didesni, nei apmokant tinklą. Dėl šios priežasties, prieš užbaigiant tinklo mokymą, svoriai atitinkamai apdorojami. Tokio sluoksnio naudojimas tinkle gali suteikti geresnį efektą nei kitokie reguliarizacijos būdai [56].

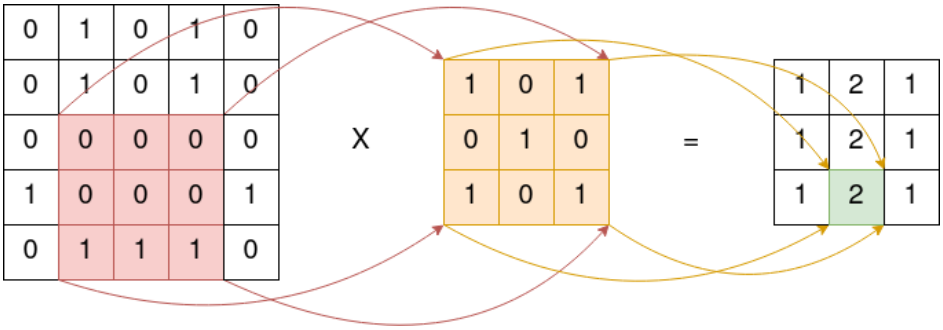


(a) Standartinis DNT (b) DNT su atmetimo sluoksniu

11 pav. DNT modelis, naudojant atmetimo sluoksnį [56]

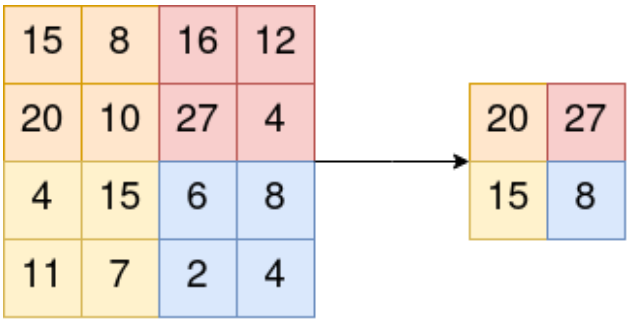
Dar vienas populiarus sluoksnio tipas – konvoliucinis (angl. *convolutional*) sluoksnis (žr. 12 paveikslą). Šie sluoksniai naudojami sudaryti konvoliucinius tinklus, ypač gerai veikiančius vaizdų atpažinimo, virpesinės spektrometrijos uždaviniams spręsti [7, 32, 74].

Konvoliucinį sluoksnį sudaro vienas arba keli kvadrato formos filtrai, dar vadinami branduoliais (angl. *kernels*). Tinklo apmokymo metu šie filtrai slenka per įvestį ir skaičiuoja filtro bei įvesties sandaugos sumas. Konvoliucinis sluoksnis skirtas vaizdinės medžiagos požymių atpažinimui – pirmieji sluoksniai gali atpažinti kontūrus, tolimesniuose sluoksniuose sudedamos sudėtingesnės detalės.



12 pav. Konvoliucinio sluoksnio veikimas.

Po konvoliucinio sluoksnio dažniausiai naudojamas sujungimo (angl. *pooling*) sluoksnis (žr. 13 paveikslą), kurio tikslas – apjungiant požymius sumažinti įvesties dimensiją. Panašiai kaip ir konvoliuciniame sluoksnyje, čia pasitelkiamas filtras, kai naudojant tam tikrą funkciją (dažnu atveju maksimumo) į filtrą patekę duomenys apjungiami.



13 pav. Sujungimo sluoksnio veikimas, filtro dydis 2, žingsnio dydis 2.

2.2. Mašininio mokymo procesų eiga

Nors mašininio mokymo modelio sukūrimo procesą galima sudaryti įvairiai, pagrindinė esmė susiveda į 7 vystymo žingsnius [75]:

1. **Duomenų surinkimas**. Vienas svarbiausių proceso žingsnių – atrinktų duomenų kokybė atspindės modelio rezultatyvumą. Šiame žingsnyje duomenys suvedami į vienodą formatą;

2. **Duomenų paruošimas**. Įvertinama duomenų kokybė, užpildomi tušti požymiai, pritaikomi duomenų normalizavimo ar kiti reikalingi algoritmai, atliekamos pirminės duomenų vizualizacijos, įvertinamas duomenų poreikis. Taip pat šiame žingsnyje duomenys suskirstomos į apmokymo bei testavimo (ir validacijos) aibes. Šis žingsnis taps reikšmingu kituose žingsniuose. Dažniausiai duomenys padalinami 80 %, 20 % proporcijomis. Plačiau apie duomenų dalinimą 2.2.2 poskyryje;

3. **Modelio tipo parinkimas**. Atsižvelgiant į duomenų rūšį, dimensijų ar mėginių skaičių ir kitus faktorius, reikia parinkti tinkamiausią modelį (arba neuroninio tinklo architektūrą);

4. **Modelio apmokymas**. Šio žingsnio vykdymas priklauso nuo taikomo modelio. Modelio apmokymas dažnai vyksta iteratyviai, kas žingsnį įvertinant paklaidą ir pagal tai atnaujinant modelio svorius;

5. **Modelio įvertinimas**. Šiam žingsniui reikalinga antrajame žingsnyje atidėta testavimo duomenų aibė. Modelio įvertinimui, priklausomai nuo užduoties tipo, galime naudoti kelias skirtingas metrikas (jos aptariamos 2.2.1 poskyryje). Dažnai šiame žingsnyje atliekama ir kryžminė patikra, ji plačiau aptariama 2.2.2 poskyryje;

6. **Hiperparametrų derinimas**. Jeigu įvertinimo metrika netenkina, siekiant pagerinti norimą rezultatą, galima bandyti suderinti modelio hiperparametrus. Hiperparametrai priklauso nuo modelio rūšies, jų skaičius gali skirtis. Parametrų parinkimas gali užimti didžiąją modelio sudarymo dalį, nes kiekvieną kartą parinkus parametrų aibę būtina vėl įvertinti modelį. Pastarasis žingsnis šio darbo kontekste yra esminis, todėl hiperparametrų paieška plačiau aptariama 3 skyriuje;

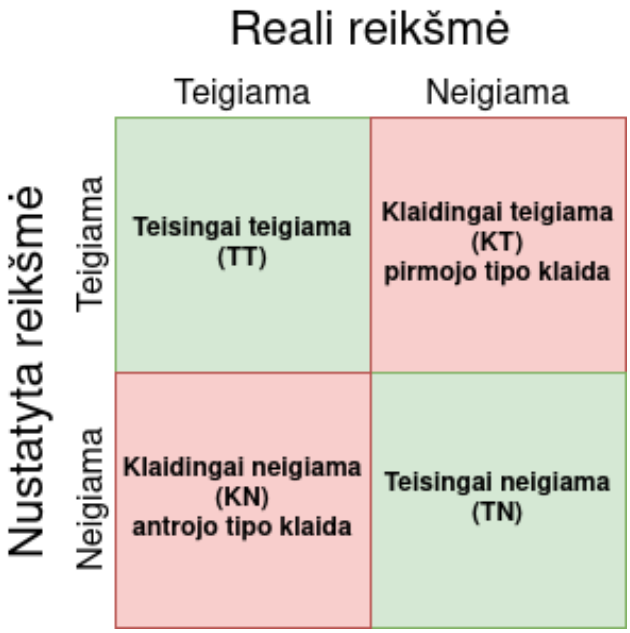
7. **Modelio naudojimas**. Modelis kuriamas norint išspręsti realų duomenų analizės uždavinį, todėl galutinė siekiamybė turėtų būti realus modelio panaudojimas.

2.2.1. Vertinimo metrikos

Vertinimo metrikos, priklausomai nuo uždavinio, naudojamos įvertinti kaip gerai veikia modelis. Metrikos dažnai parenkamos pagal uždavinio ypatybes, kai yra žinoma, kokia modelio savybė svarbiausia (pvz.: nėra labai svarbu, jei modelis kartais nefiksuos kažkurios klasės). Nors rezultato įvertinimas gali būti atliekamas įvairiais būdais, čia aptariamos tik šiame darbe svarbios funkcijos.

Klasifikavimas

Norint apibrėžt klasifikavimo metrikas, svarbu identifikuoti klaidų ir teisingų sprendimų rezultatus (binariniam bandymui, kurio atsakymas gali būti teigiamas arba neigiamas):



14 pav. Klaidų tipai binariniam uždaviniui. Šiuo principu atvaizduojami rezultatai vadinami painiavos matricomis (angl. *confusion matrix*).

Tikslumas (angl. *accuracy*) pasako, kiek atvejų buvo parinkta teisingai. Ši metrika geriausiai veikia, kai egzistuoja geras klasių balansas. Vyraujant klasių disbalansui, ši metrika pasidaro nebereikšminga. Pvz.: jei duomenų aibę sudaro 99 teigiamai pažymėti mėginiai ir 1 neigiamai pažymėtas mėginys, o modelis visada, nepriklausomai nuo duomenų įvesties, atsako, kad duomenys

priklauso teigiamai klasei, tai tokio modelio tikslumas bus 99 %.

$$\text{tikslumas} = \frac{TT + TN}{TT + KT + KN + TN} \tag{2.14}$$

Preciziškumas (angl. *precision*) nusako, kiek iš teigiamai klasei priskirtų mėginių iš tiesų priklauso teigiamai klasei. Aukštas preciziškumas – geras rodiklis tais atvejais, kai norima tiksliai įvertinti visas teigiamas klases, tačiau mažai reikšminga, jei teigiama klasė bus priskirta neigiamai. Pvz.: el. pašto filtrai, kurie kartais praleidžia šlamštą į dėžutę (KN), tačiau stengiasi kuo rečiau įprastus laiškus pažymėti kaip brukalą (KT).

$$\text{preciziškumas} = \frac{TT}{TT + KT} \tag{2.15}$$

Jautrumas (angl. *sensitivity*) nusako, kiek teigiamų reikšmių modelis klasifikavo kaip teigiamas. Aukštas jautrumas – geras rodiklis tais atvejais, kai nenorima gauti klaidingai neigiamų rezultatų. Pvz.: virusinės infekcijos testai, jeigu testas nebūtų jautrus, sergantys žmonės būtų klasifikuojami kaip sveiki. Taip sukeliama rizika, kad tokia klaida nulems didesnį užsikrėtimo kiekį.

$$\text{jautrumas} = \frac{TT}{TT + KN} \tag{2.16}$$

Regresija

Vidutinė kvadratinė paklaida (angl. *mean square error – MSE*) – viena paprasčiausių regresijos metrikų. Ši metrika skaičiuojama susumavus tikrų ir spėjamų reikšmių skirtumų kvadratus bei išvedant sumos vidurkį:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \text{ čia, ir toliau } y_i \text{ – spėjama reikšmė, } \hat{y}_i \text{– tikroji reikšmė} \tag{2.17}$$

Šios metrikos rezultatas nėra standartizuotas, metrikos įvertis priklauso nuo spėjamos reikšmės rėžių (priklausomai nuo nustatinėjamo požymio reikšmės amplitudės, geru modeliu gali būti laikomas modelis, kurio MSE reikšmė bus 1000, tačiau sprendžiant kitą uždavinį, MSE = 1 gali būti prastas rezultatas). MSE metrika, dėl naudojamo kėlimo kvadratu funkcijos, griežtai baudžia modelius su didesne paklaida.

Siekiant suvesti paklaidos vienetus į tą pačią skalę, dažnai naudojama MSE modifikacija – **vidutinės kvadratinės paklaidos šaknis** (angl. *root mean square error – RMSE*), pasitelkiant tokią modifikaciją paprasčiau interpretuoti rezultatus:

$$RMSE = \sqrt{MSE} \tag{2.18}$$

Vidutinė absoliuti paklaida (angl. *mean absolute error – MAE*) panaši į jau minėtą MSE, tačiau šiuo atveju, vietoje kvadratinės, skaičiuojama absoliuti paklaida:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.19}$$

Iš esmės, MAE metrika vidutinę paklaidą reprezentuoja geriau nei MSE ar RMSE – ši metrika visas paklaidas vertina vienodai.

2.2.2. Kryžminė patikra ir duomenų dalinimas

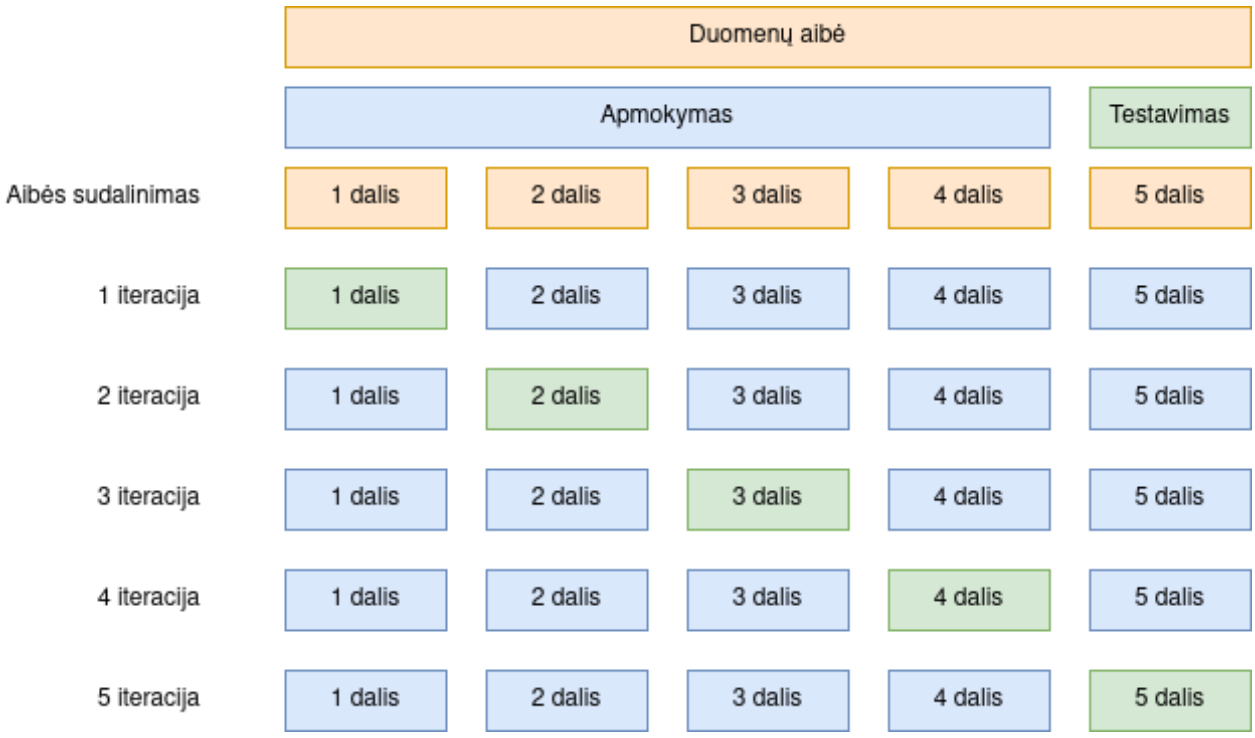
Norint įvertinti rasto modelio rezultatyvumą, modeliui pateikiamos reikšmės, kurių išvestys iš anksto žinomos. Taip šį modelį galima įvertinti pagal kurią nors apibrėžtą metriką. Šiame žingsnyje naudoti duomenis, jau naudotus modelio apmokymui, yra netinkama idėja, nes tai neparodo tikėtino modelio patikimumo nematytiems duomenims. Norint tikslingai įvertinti modelio rezultatyvumą, reikalinga pasiruošti atskirą duomenų aibę, kuri nebus naudojama modelio apmokymui. Ši aibė vadinama testavimo aibe. Dažna praktika, kad pradinė duomenų aibė būtų padalinama į 80 % apmokymo ir 20 % testavimo duomenų aibes.

Situacija, kai modelio rezultatas yra ypač geras su apmokymo aibės duomenimis, tačiau labai prastas su nematytais duomenimis, vadinama persimokymu. Persimokymo atveju įprastai atliekama tinkamesnių modelio parametrų paieška. Jos metu apmokomas modelių su skirtingais parametrais rinkinys, po to, remiantis surinktomis testavimo metrikomis, atrenkamas geriausias parametrų rinkinys. Čia susiduriama su kita problema – modelis buvo parinktas pagal testavimo rezultatą. Toks sprendimas gali nulemti, kad modelio parametrai buvo pritaikyti tik šiai (testavimo) aibei ir realus modelio rezultatyvumas bus daug prastesnis. Šios problemos sprendimui galima atskirti dar vieną duomenų aibę. Tokia aibė vadinama validavimo (arba patikros). Tačiau čia dažnai (turint santykinai mažą duomenų aibę) susiduriama su duomenų kiekio problema – skirstant duomenų aibę į tris dalis, kažkuri iš jų gali tapti nereprezentatyvi.

Norint užtikrinti gerą testavimo aibės reprezentaciją ir pakankamą apmokymo aibės dydį, atliekama procedūra, vadinama kryžmine patikra (angl. *cross-validation*). Vienas paprasčiausių kryžminės patikros variantų – *k*-dalinimų (angl. *k-fold*) kryžminė patikra. Metodo esmė – duomenų aibė padalinama į *k* dalių (žr. 15 paveikslą), tuomet šio metodo veikimą galima apibūdinti žingsniais, kurie kartojami *k* iteracijų:

1. Modelis apmokomas naudojantis *k*-1 duomenų dalių (žr. 15 paveikslą);
2. Modelis validuojamas naudojant atmestą duomenų dalį.

Surinkus kiekvieno padalinimo validacijos rezultatus, išvedamas vidurkis, parodantis kryžminės patikros rezultatą.



15 pav. 5-dalinimų kryžminė patikra.

Kiti kryžminės patikros metodai naudojasi panašiomis idėjomis, dažniausiai kinta tik dalinimo strategijos, tokių strategijų pvz.:

- vieno atmetimo kryžminė patikra (angl. *leave-one-out cross-validation*), kurios metu duomenų aibė suskirstoma taip, kad kiekvienas aibės elementas po kartą patektų į testavimo aibę;
- p atmetimų kryžminė patikra (angl. *leave-p-out cross-validation*) veikia panašiai kaip ir vieno atmetimo kryžminė patikra, tačiau čia atmetama p mėginių visais galimais būdais, taip gaunama C_p^n padalinimų (n – duomenų aibės dydis). Šis kryžminės patikros metodas, parinkus net santykinai nedidelę p reikšmę, gali tapti visiškai nepanaudojamas.

Dalinant duomenų aibes, galima (ir patartina) atsižvelgti į duomenų pasiskirstymą (kad duomenų klasės numatomo požymio reikšmės būtų vienodai reprezentuotos abejose aibėse). Dar reikalinga atsižvelgti ir į grupes, kurios galėjo susidaryti duomenų surinkimo metu (pvz.: tiriant mėginį buvo surenkami keli matavimai, todėl norint tikslesnės testavimo metrikos, šie matavimai turėtų būti interpretuojami kaip atskira grupė. Tokie duomenys turi būti tik apmokymo arba tik testavimo aibėje).

3. Automatinis mašininis mokymasis

Automatinis mašininis mokymasis (angl. *automated machine learning*) (toliau AutoML) – mašininio mokymo sritis, leidžianti duomenis analizuoti be gilaus mašininio mokymo suvokimo. Pagrindinė AutoML užduotis – optimizuoti hiperparametrus taip, kad galutinis mašininio mokymo modelis pasiektų gerą (ar optimalų užduočiai) rezultatą. Svarbus tokios optimizacijos bruožas – optimizuojama funkcija iš esmės nėra žinoma, ji veikia „juodosios dėžės“ (angl. *black box*) principu, kai funkcijos negalima išreikšti analitine formule. Tokiu atveju negalima naudoti optimizacijos metodų, kuriems reikalingos funkcijų Hesiano matricos ar jų gradientai. Automatinio mašininio mokymo sritis nėra nauja, jau sudaryta aibė įrankių ir metodų, leidžiančių be didelių pastangų (reguliuojant įrankių parametrus) gauti tinkamus rezultatus [21]. Šiame darbe susifokusuojama į genetinę paiešką, šio metodo efektyvumas jau įrodytas vykdant spektrometrinių duomenų modelio paieškas [16].

3.1. Tinklelio paieška

Pats paprasčiausias, nereikalaujantis jokių žinių, parametrų paieškos metodas – tinklelio paieška (angl. *grid search*). Šis metodas – visiškas brutalios jėgos metodas: suformuojama tikslo funkcijos f parametrų aibė P (parametrai – diskrečios reikšmės) ir atliekami perrinkimai taip, kad rastas parametrų rinkinys p būtų geriausias apibrėžtai funkcijai, tai yra:

$$f(p) \geq f(x), \forall x \in P \tag{3.1}$$

Duomenų analizės atveju, f būtų kažkokio modelio rezultato įverčio funkcija (modelio tikslumo metrika) duotam parametrų, metodų ir duomenų rinkiniui.

Metodas garantuotai ras geriausią atsakymą duotai parametrų aibei. Nors procesų išlygiagretinimas yra gana paprastas, laiko atžvilgiu šis metodas vis tiek tinkamas tik nedidelėms parametrų dimensijoms – didėjanti parametrų aibė sukelia kombinatorinį sprogimą. Parametrų aibę turi sudaryti tik diskrečios reikšmės, tai gali reikšti, kad optimalios reikšmės bus praleidžiamos.

3.2. Atsitiktinė paieška

Atsitiktinė paieška ištaiso tinklelio paieškos trūkumus. Nors išbandomi ne visi galimi parametrų rinkiniai, atsitiktinės paieškos atveju parametrai gali būti išreikšti skirstiniais. Esant tokiai sąlygai, kiekvienos iteracijos metu parametrai bus atrenkami ne iš iš anksto apibrėžto sąrašo, bet atsitiktinai atrenkami iš apibrėžto skirstinio. Toks paieškos būdas apima didesnę parametrų erdvę.

Bergstra et al. [11] 2012 metų straipsnyje palygino atsitiktinės ir tinklelio paieškos rezultatus mašininio mokymo hiperparametrų optimizavime. Autoriai pastebi, kad dažnai atsitiktinė paieška pasiekia geresnį rezultatą per trumpesnį vykdymo laiką. Tiesa, pastebima, kad atsitiktinė paieška neatsižvelgia į jau pasiektus rezultatus, todėl tinklelio paieškos ir rankinio parametrų parinkimo kombinacija gali veikti geriau nei atsitiktinė paieška.

3.3. Genetinis algoritmas

Genetinis algoritmas – paieškos metodas, kai paieška atliekama vykdant atsitiktinius pakeitimus, tačiau tuo pat metu išsaugant informaciją apie geras parametrų kombinacijas.

3.3.1. Apibrėžimas

Genetinis algoritmas [52] – tai euristinis paieškos metodas, paremtas Charles Darwin evoliucijos teorijos idėja. Naudojant natūraliosios atrankos įkvėptus operatorius, tokius kaip mutacija, kryžminimas ir selektyvus atrinkimas, gaunami nauji sprendiniai. Jie pagal idėją turėtų būti neprastesni už prieš tai buvusius sprendinius.

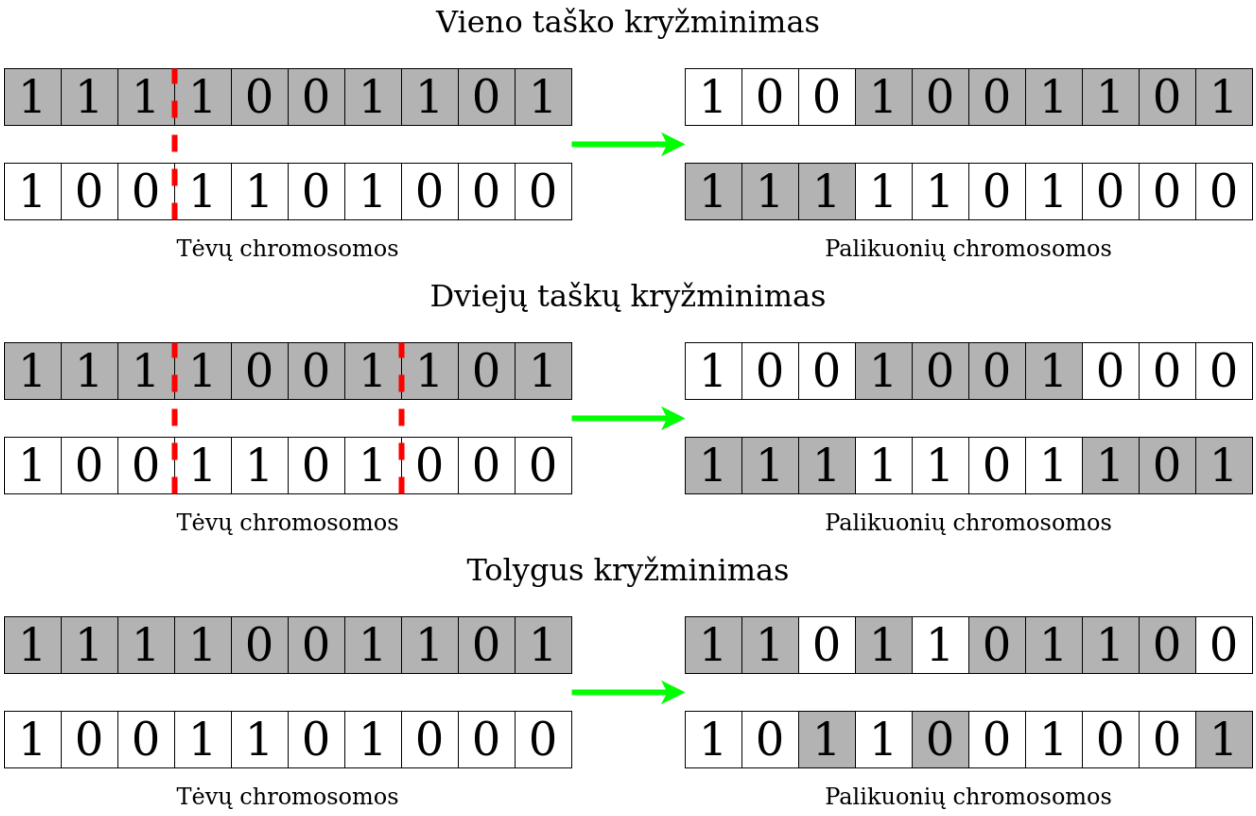
Tipinį genetinį algoritmą sudaro:

- **individas**. Potencialus genetinio algoritmo sprendinys;
- **chromosomos**. Individo požymių aibė. Dažniausiai šie požymiai išreiškiami bitų eilutėmis;
- **genas**. Dalis chromosomos – tam tikras individo požymis;
- **alelis**. Geno reikšmė;
- **populiacija**. Individų aibė;
- **generacija**. Duotos iteracijos populiacija;
- **fitneso funkcija**. Metrika, matuojanti individo tinkamumą genetinio algoritmo sprendžiamai problemai;
- **kryžminimo (rekombinacijos) operatorius**. Parenkami genai, kurių aleliai chromosomose bus sukeisti, taip gaunami nauji chromosomų rinkiniai – palikuonys, jie savyje turi informacijos iš abiejų tėvų. Taip paieška savotiškai „įsimena“ gerus sprendimus ir paieškos istorija daro įtaka dabartinei generacijai. Yra daug būdų, kaip galima atlikti kryžminimą. Keli primityviausi būdai pavaizduoti 16 paveiksle;
- **mutacijos operatorius**. Atsitiktinai pakeičia alelio reikšmę. Reikalingas tam, kad algoritmas neužstrigtų (jeigu tėvų kryžminamų genų aleliai būtų vienodi arba, jeigu visą populiaciją sudarantys individai būtų su vienodomis chromosomomis).

Tipinio genetinio algoritmo veikimą galima suvesti į tokius žingsnius:

1. **Inicializavimas**. Genetinis algoritmas inicializuojamas sukuriant atsitiktinę (arba iš karto apibrėžtą) individų aibę;
2. **Įvertinimas**. Kiekvienas generacijos individas įvertinamas pagal fitneso funkciją.
3. **Atrinkimas kryžminimui**. Pagal fitneso reikšmes kryžminimui atrenkami potencialūs individai. Individai, turintys aukštesnę fitneso reikšmę, turi didesnę tikimybę būti atrinkti.
4. **Kryžminimas**. Individai kryžminami, taip gaunant naujus individus.
5. **Mutacija**. Naujiems individams atliekama mutacijos operacija.
6. **Naujos generacijos sudarymas**. Su naujai sukurtais individais sukuriama nauja generacija. Ši generacija gali būti su visiškai nauja populiacija (individai iš praeitos populiacijos bus pašalinami) arba dalis naujos populiacijos gali pakeisti dalį senosios.

2–6 žingsniai atliekami tol, kol įgyvendinamas iš anksto apibrėžtas uždavinio reikalavimas (arba paieška pasiekia maksimalų generacijų skaičių).



16 pav. Keletas primityviausių kryžminimo genų parinkimo metodų.

3.3.2. Genetinis algoritmas virpesinės spektrometrijos duomenų analizėje

Genetinio algoritmo panaudojamumas virpesinės spektrometrijos duomenų analizės automatizavime yra jau ne kartą įrodytas. Vienas populiariausiu uždavinių – svarbiausių spektrinių ruožų atrinkimas. Genetinio algoritmo panaudojimą, norint atrasti svarbius spektro regionus, įrodo Soh et al. 2005 metų straipsnyje [55]. Šioje studijoje bandoma PLS modelį apmokyti nustatyti atitinkamas žmogaus serumo albumino (angl. *human serum albumin*), γ-globulino ir gliukozės reikšmes mėginiuose. Kadangi NIR spektruose reikšmės gali būti labai persidengusios, genetinis algoritmas buvo paruoštas taip, kad gebėtų aptikti spektrinius regionus, o ne atskirus ruožus. Ruožai buvo suskirstyti po 6, į atskirus langus, taip palengvinant genetinio algoritmo darbą. Genetinio algoritmo rezultatas gali priklausyti nuo atsitiktinės būsenos, todėl apmokymo aibė buvo padalinta į 5 dalis. Tokiu atveju galima gauti 5 skirtingus rezultatus, kuriuos suagregavus galima gauti tikslesnį rezultatą. Rezultate regimas modelių spėjimo klaidos sumažėjimas visoms trims sudedamosioms mėginių dalims, kai modeliui apmokyti naudojami genetinio algoritmo rekomenduojami požymiai.

Genetinis algoritmas tinka ir Ramano spektrų analizėje, panašų kaip Soh [55] metodą pateikė ir Duraipandian et al. 2011 metais [18]. Šiame straipsnyje buvo tiriami normalūs ir priešvėžiniai gimdos kaklelio audiniai. Genetinis algoritmas sėkmingai nustatė svarbiausius požymius, identifikuojančius susirgimą – aptikti požymiai susiję su baltymais, nukleorūgštimis ir lipidais audinyje.

Genetinis algoritmas gali būti panaudotas ir platesnei paieškai. Devos et al. 2013 metų straipsnyje [16] siūlo metodą, kurį vadina GENOPT-SVM. Jo tikslas – pasitelkiant genetinį algoritmą, optimizuoti SVM modelio ir spektrų paruošimo kombinacijų rinkinį. Užduotis genetiniam algoritmui sudaryta taip, kad galimi paruošimo algoritmai bei jų parametrai (taip pat kaip ir SVM parametrai) yra užkoduoti dvejetainėje eilutėje. Pastaroji apibūdina populiacijos individą, taip galima daryti įvairias kombinacijas ir įvertinti bendrą (spektro paruošimo ir modelio) rezultatą.

Šis tyrimas parodo automatizavimo potencialą spektrinių duomenų analizėje. Nors buvo pasiekti geresni rezultatai nei naudojant PLS metodą su eksperto parinktu spektro paruošimu, atliekant šį tyrimą nebuvo įtraukiamas požymių atrinkimas. Straipsnio autoriai pastebi, kad toks žingsnis galėjo sumažinti modelių persimokymo tikimybę ir pagerinti jų tikslumą.

4. Magistro darbo tiriamoji dalis

Norint išbandyti ir validuoti šiame darbe siūlomą metodiką, gaunamus rezultatus reikšminga palyginti su rezultatais, gautais naudojat kitus metodus. Jeigu magistro darbe siūloma metodika būtų lyginama su įprastais metodais gaunamais rezultatais, tai gali įnešti šališkumo išvadose, todėl turėtų būti lyginama su kitų šioje srityje dirbančių mokslininkų rezultatais. Tokiam palyginimui buvo surinkta duomenų aibė, kurią sudaro moksliniuose tyrimuose nagrinėti duomenys. Darbų, kuriuose pasiekiamas 100 % tikslumas, palyginimas gali būti beprasmis (jeigu siūlomos metodikos rezultatai būtų neprastesni). Dėl pastarosios priežasties, surinkti duomenys straipsniuose buvo naudojami spręsti regresijos problemas arba originaliame straipsnyje klasifikavimo problema nebuvo išspręsta iki galo (pasiektas mažesnis nei 100 % tikslumas). Tokių duomenų naudojimas leis nusistatyti siekiamų rezultatų rėžius. Surinktų duomenų ir šiuos duomenis nagrinėjančių straipsnių aibė veikia ne tik kaip bazinė tikslumo riba, tačiau kaip ir žinių aibė, pagal kurią galima papildyti, pataisyti siūlomų metodų aibę. Aptarta algoritmo modifikacija gali užtikrinti metodikos stabilumą ir patikimumą.

4.1. Duomenų aibės

4.1.1. Aktyviosios medžiagos kiekio nustatymas vaistų tabletėse

Dyrby et al. sudarė antidepresanto *Escitalopram* (gamintojai – *Lundbeck*) NIR ir Ramano spektrometrijos duomenų rinkinį [19]. Tyrimui buvo naudojamos keturių (5, 10, 15 ir 20 mg veikliosios medžiagos, tablečių svoriai atitinkamai 90, 125, 188, 250 mg) skirtingų stiprumų vaistų tabletės. Duomenys MATLAB ir Unscrambler formatais prieinami Kopenhagos universiteto chemometrijos puslapyje[1]. Autoriai, norėdami išplėsti kalibracijos rėžius, dirbtinai praplėtė mėginių aibę (pagamindami naujų tablečių aibę), tačiau dėl vientisumo (ir dėl rezultatų prieinamumo) bus naudojami tik originalūs duomenys (duomenų lentelėje žymimi atributu „Scale“ = 1).

Ramano spektrui fiksuoti tyrėjai naudojo Perkin-Elmer System 2000 NIR FT-Raman spektrometrą. Naudojamo lazerio bangos ilgis 1064.4 nm, tai atitinka NIR regioną. Surinktas Ramano spektras apėmė 200–3600 cm^{-1} spektrą, spektrinė rezoliucija 8 cm^{-1}, fiksuojamas buvo 64 spektrų vidurkis. NIR spektrų analizei buvo naudojamas ABB Bomem FT-NIR MB-160 spektrometras. Surinkti spektro duomenys apėmė 4000–14000 cm^{-1} rėžį, skenavimo rezoliucija 16 cm^{-1}, spektras fiksuotas iš 128 vidurkintų skenavimų.

Duomenų aibę sudarė 120 mėginių (12 tablečių partijų, kiekvieną partiją sudaro po 10 tablečių). Validacija atliekama suskirstant aibę į 10 dalių, kiekvieną dalį sudarė skirtingų partijų tabletės.

Autoriai naudojo PLS regresijos modelį. Norėdami patikslinti analizės rezultatus, tyrėjai išbandė ir įvairius duomenų apdorojimo modelius bei svarbių spektrinių ruožų parinkimo būdus. Geriausi rezultatai (žr. 1, 2 lenteles) buvo pasiekti naudojant antro laipsnio išvestinės NIR duomenis kartu su intervaliniu PLS algoritmu (atrinktas ruožas: 8625–9173 cm^{-1}) .

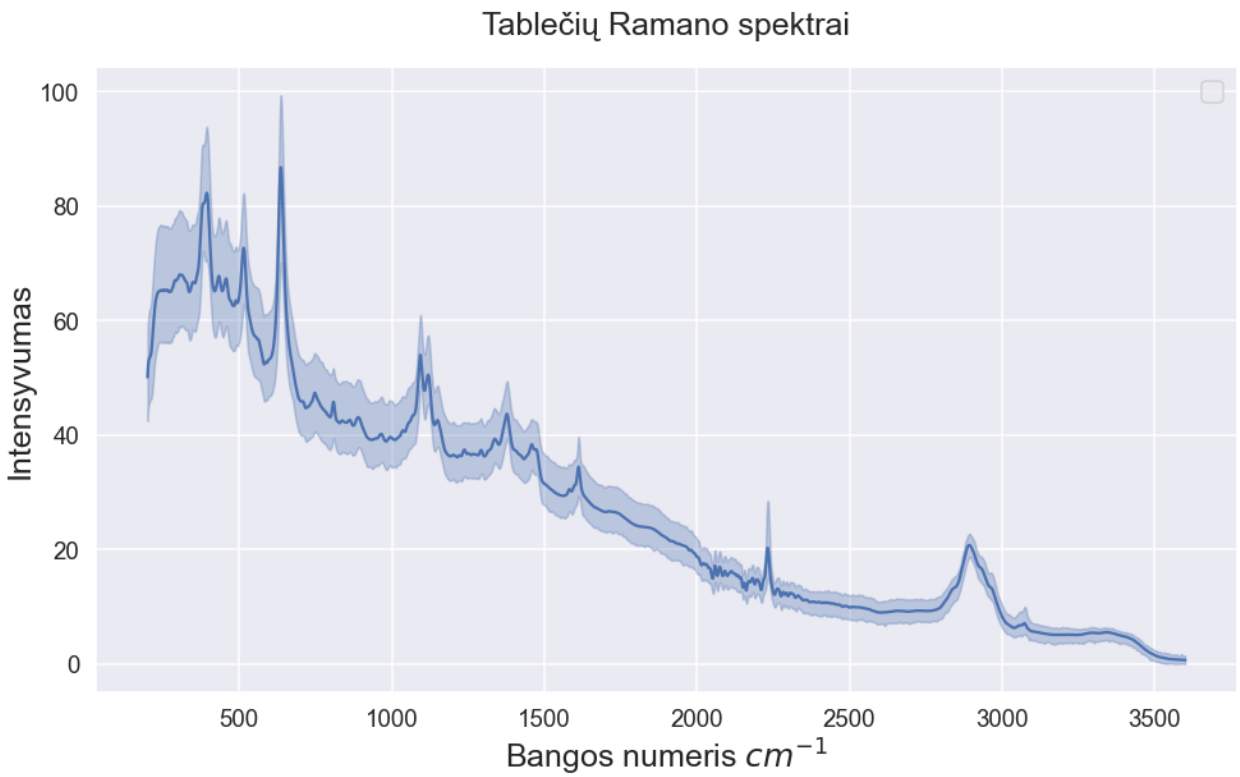
[1] http://www.models.life.ku.dk/Tablets [Žiūrėta:2021-01-04]

Duomenų paruošimo metodas	Komponenčių skaičius	RMSE
-	5	0.63
MSC	6	0.59
Pirmojo laipsnio išvestinė	5	0.56
Antrojo laipsnio išvestinė	5	0.56
SNV	6	0.59
SNV + antrojo laipsnio išvestinė	5	0.65

1 lentelė. Dyrby et al. [19] rezultatai, nustatinėjant tabletėje esančios aktyviosios medžiagos kiekį pagal Ramano spektrometrinius duomenis.

Duomenų paruošimo metodas	Parinkta požymių aibė (cm^{-1})	Komponenčių skaičius	RMSE
-	7400–10500	3	0.35
su iteratyviu PLS	8286–8995	3	0.34
MSC	7400–10500	3	0.32
su iteratyviu PLS	8293–9119	3	0.31
Pirmojo laipsnio išvestinė	7400–10500	2	0.36
su iteratyviu PLS	8432–9273	3	0.33
Antrojo laipsnio išvestinė	7400–10500	2	0.39
su iteratyviu PLS	8625–9173	3	0.30

2 lentelė. Dyrby et al. [19] rezultatai, nustatinėjant tabletėje esančios aktyviosios medžiagos kiekį pagal NIR spektrometrinius duomenis.



17 pav. Tablečių spektrai. Vientisa linija žymi spektrinių duomenų vidurkį, o linijos fonas žymi duomenų pasiskirstymą (įvertinta iš viršaus $vidurkis + min(std(X), X)$ ir iš apačios $vidurkis - max(std(X), X)$)

4.1.2. Vištienos filė autentiškumo nustatymas

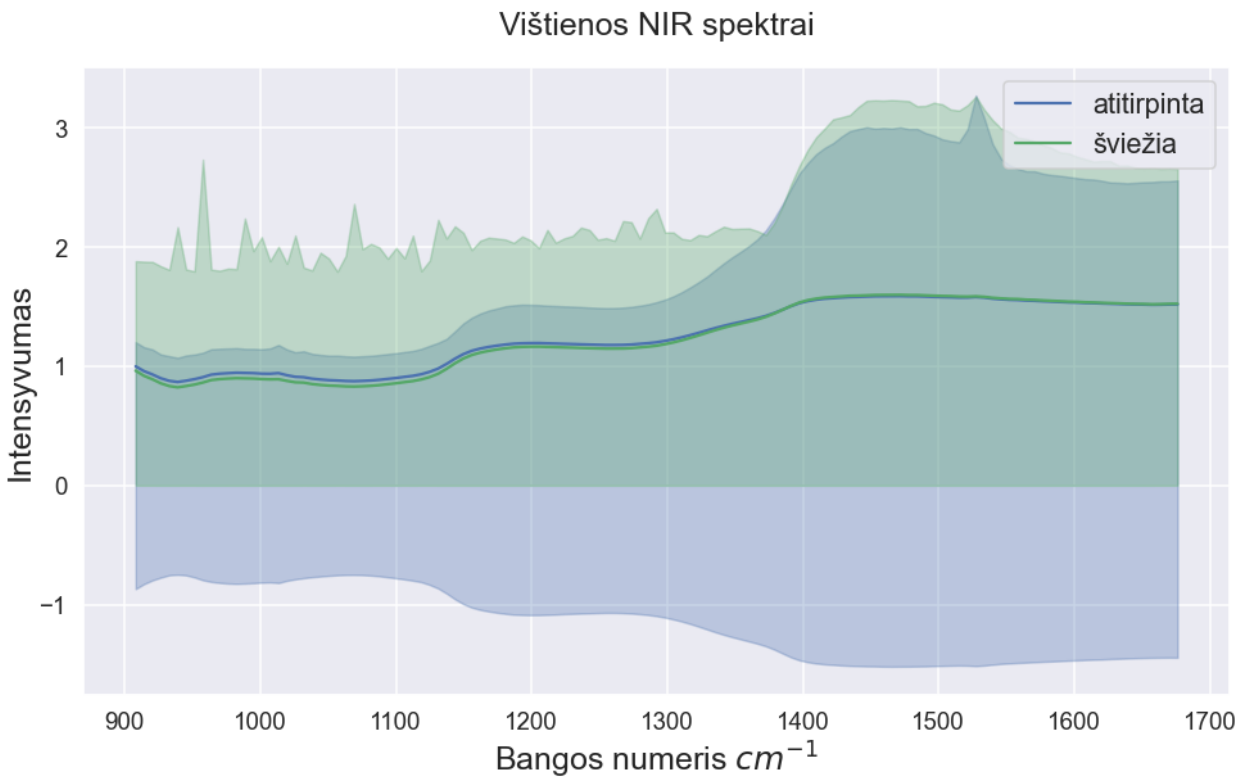
Parastar et al. [40] sudarė 153 skirtingų šviežios vištienos mėginių NIR spektrometrinių duomenų aibę. Duomenys surinkti naudojant MicroNIR PRO (Viavi Solutions, Milpitas, CA, USA) rankinį spektrometrą. Duomenų aibė apėmė 908–1676 nm spektrinį regioną, duomenų eilutę sudarė 125 lygiai spektriniame regione pasiskirstę požymiai. NIR matavimai buvo atliekami trimis būdais: skenuojant mėsą tiesiogiai (M), per pakuotę (PP), per pakuotę iš apačios (PA). Kiekvienam matavimo būdui buvo surenkami 5 spektrai, nuskenavus šviežią vištieną ši buvo užšaldoma ir vėliau atitirpinama, NIR tyrimai pakartojami. Viso duomenų aibę sudaro 4590 NIR spektrų. Duomenys prieinami Mendeley[2] puslapyje [65].

Duomenų analizei tyrėjai naudojo RSDE (*random subspace discriminant ensemble*), SVM, PLS-DA, ANN modelius, taip pat naudojant validacijos aibę buvo parenkami ir keičiami spektrų apdorojimo žingsniai. Pastarieji duomenų apdorojimo žingsniai buvo parenkami pagal Gerretzen et al. 2015 metais pasiūlytą metodiką [23]. Modeliai naudoti nustatinėjant ar mėsa buvo šviežia, ar atitirpinta, taip pat atskiriant vištų auginimo sąlygas. Autoriai atliko vienos klasės atmetimo kryžminę patikrą. Metodikų palyginimui parenkamas tik mėsos būklės uždavinys, tai daroma, nes vištienos auginimo sąlygos, pateiktos duomenyse, nėra suskirstytos pagal bendrą standartą (dėl skirtingų kilmės šalių). Taip pat bandymai buvo atliekami kombinuojant duomenų aibes (PP / PA ir M / PP / PA matavimai), tačiau paprastumo dėlei bus nagrinėjami tik M matavimų rezultatai.

[2] https://data.mendeley.com/datasets/cp2hdhkys3/4 [Žiūrėta:2021-01-04]

		Apmokymas	Kryžminė patikra	Testavimas
	Tikslumas	90.2	87.6	85.2
Šviežia	*Jautrumas*	90.1	89.1	88.4
	Preciziškumas	90.3	89.4	87.1
Atitirpinta	*Jautrumas*	89.4	86.3	84.0
	Preciziškumas	90.0	87.2	85.2

3 lentelė. Parastar et al. [40] klasifikavimo rezultatai %. Uždavinys: atskirti, vištiena buvo šviežia ar šaldyta ir atitirpinta. Naudojami tik M tipo duomenys.



18 pav. Vištienos spektrai. Bendrą duomenų aibę iškraipė keli nekokybiški spektrai. Vientisa linija žymi spektrinių duomenų vidurkį, o linijos fonas žymi duomenų pasiskirstymą (įvertinta iš viršaus $vidurkis+min(std(X), X)$ ir iš apačios $vidurkis-max(std(X), X)$). Šviežios vištienos užpildytas fonas sufleruoja tuščių spektrų vyravimą duomenyse.

4.1.3. Tyrelės tipo nustatymas

Holland et al. [26] 1998 metų straipsnyje aprašo FT-IR (MIR sritis) spektrometrijos galimybes, nustatinėjant maisto padirbinėjimo atvejus, susijusius su braškių tyre. Autoriai duomenų surinkimui naudojo Spectra-Tech MonitIR FT-IR spektrometrą. Duomenų aibę sudarė reikšmės iš 899–1802 cm^{-1} spektrinio ruožo (spektrinė rezoliucija 8 cm^{-1} , viso 235 reikšmės). Norint užtikrinti tyrių autentiškumą, šios buvo ruošiamos laboratorijoje, tam naudojant šviežius ar šaldytus vaisius iš 1993–1995 metų derliaus. Tyrės ruoštos jas spaudžiant per metalinį tinklelį arba sutrinant elektriniu rankiniu trintuvu. Straipsnio autoriai paruošė braškių, aviečių, obuolių, juodųjų serbentų, gervuogių, slyvų, vyšnių, abrikosų tyrių ir vynuogių sulčių mėginius. Braškių bei aviečių tyrių mėginiai buvo maišomi su kitokios rūšies vaisių tyrėmis, taip pat su raudonųjų vynuogių sultimis

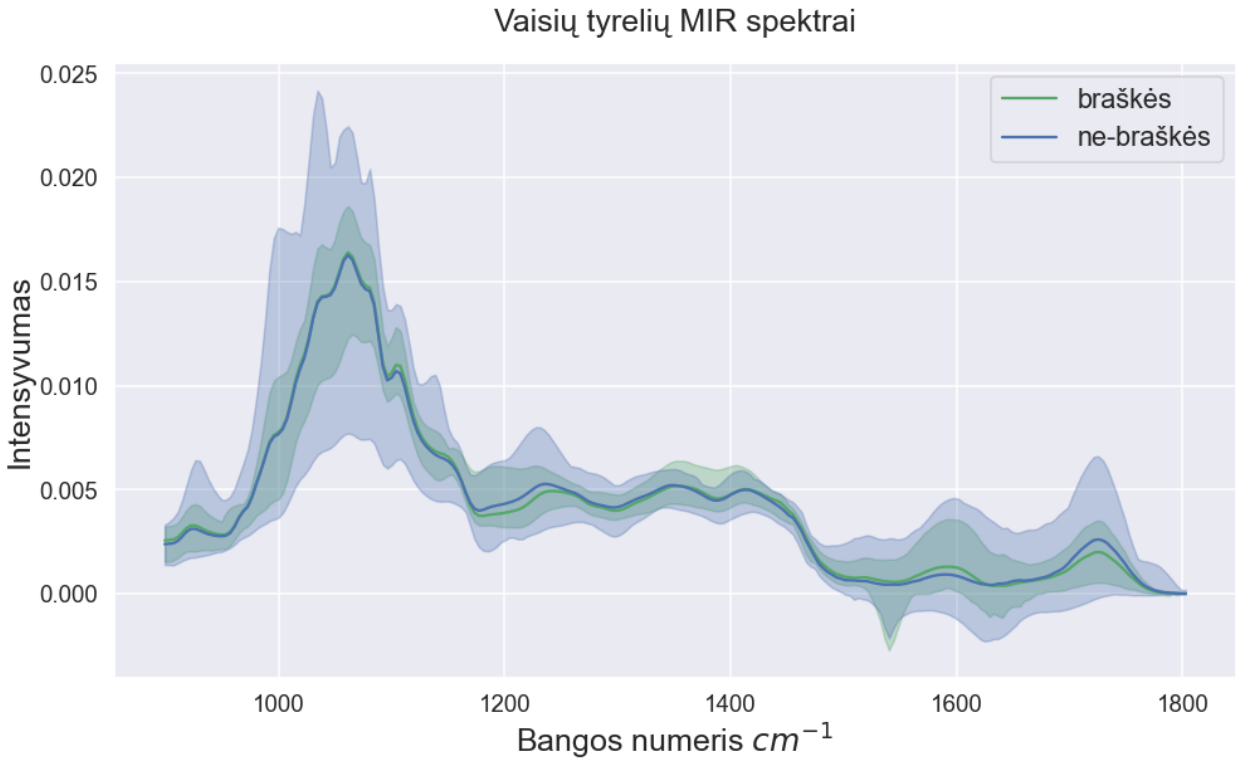
bei rabarbarų kompotu, tokiu būdu sudaryta ir užterštų mėginių bazė. Iš viso mėginių bazę sudarė 983 elementai, jie buvo suskirstyti į apmokymo, validavimo (autorių pavadinta derinimo (angl. *tuning*)) ir testavimo aibes (pateiktame duomenų rinkinyje nėra išreikšto požymio, lei/-džian/-čio tai pakartoti). Duomenys prienami Quadram instituto puslapyje[3].

Tyrėjai duomenų transformacijai naudojo vieno taško (atliktas tiesinis postūmis pagal vieną požymį ties 1802 cm^{-1} ruožu) bazinės linijos korekciją, po to duomenys normalizuoti.

Duomenų analizei buvo naudojamas PLS regresorius (regresijai naudotas pseudo-požymis, kuris po to buvo paverčiamas į klasę). Bandymui perrinkta 40 skirtingų PLS modelių (su skirtingu kiekiu komponenčių), geriausiai veikė modelis, naudojantis 14 komponenčių.

	Neteisingai klasifikuota		% teisingai klasifikuota		
	Braškės	Ne braškės	Braškės	Ne braškės	Viso
Apmokymas	3 (iš 119)	10 (iš 218)	97.5	95.4	96.5
Validavimas	8 (iš 117)	13 (iš 218)	93.2	93.8	93.5
Testavimas	6 (iš 115)	12 (iš 218)	94.8	94.1	94.3

4 lentelė. Holland et al. [26] pasiekti rezultatai



19 pav. Vištienos spektrai. Vientisa linija žymi spektrinių duomenų vidurkį, o linijos fonas žymi duomenų pasiskirstymą (įvertinta iš viršaus $vidurkis + min(std(X), X)$ ir iš apačios $vidurkis - max(std(X), X)$)

[3] https://csr.quadram.ac.uk/example-datasets-for-download/ [Žiūrėta:2021-01-04]

4.2. Programinė ir kompiuterinė įranga

- *python* 3.8 [66]. Tyrimams naudojama programavimo kalba.
  Nuoroda: https://www.python.org/
- *ipykernel* 5.3.4 [42]. Interaktyvus python branduolys.
  Nuoroda: https://ipython.readthedocs.io
- *Jupyter Notebook* 6.1.4 [61]. Tyrimams naudojama aplinka.
  Nuoroda: https://jupyter.org/
- *pandas* 1.1.4 [36]. Duomenų manipuliacijos.
  Nuoroda: https://pandas.pydata.org/
- *numpy* 1.18.5 [64]. Duomenų manipuliacijos.
  Nuoroda: https://numpy.org/
- *scikit-learn* 0.23.2 [41]. Duomenų apdorojimas ir duomenų analizės algoritmai.
  Nuoroda: https://scikit-learn.org/
- *scipy* 1.5.4 [67]. Duomenų apdorojimas.
  Nuoroda: https://www.scipy.org/
- *TPOT* 0.11.6 [31]. Genetinė procesų grandinių paieška.
  Nuoroda: https://epistasislab.github.io/tpot/
- *tensorflow* 2.3.1 [6]. Gilaus mokymo biblioteka.
  Nuoroda: https://www.tensorflow.org/
- *Keras* 2.4.3 [13]. *tensorflow* sąsaja.
  Nuoroda: https://keras.io/
- *AutoKeras* 1.0.11 [29]. Automatinė DNT paieška.
  Nuoroda: https://autokeras.com/
- *rampy* 0.4. Bazinės linijos korekcijos algoritmai.
  Nuoroda: https://github.com/charlesll/rampy
- *matplotlib* 3.3.3 [28]. Iliustracijos.
  Nuoroda: https://matplotlib.org/
- *R* 3.6.1 [45]. Tyrimams naudojama programavimo kalba.
  Nuoroda: https://www.r-project.org/
- *plsVarSel* 0.9.6 [38]. Požymių atrinkimas.
  Nuoroda: https://cran.rstudio.com/web/packages/plsVarSel/index.html
- *rpy2* 3.3.6. *python* sąsaja *R* kalbai.
  Nuoroda: https://rpy2.github.io/
- *graphviz* 2.43.0 [20]. Iliustracijos.
  Nuoroda: https://graphviz.org/

TPOT privalumas – vartotojas nebūtinai turi apibrėžti, kaip turi būti išsidėstę metodai (pvz.: du duomenų transformatoriai ir regresorius), vietoje to algoritmas ieško ir pačių metodų išsidėstymo formos, dėl to rasta grandinė gali būti ne ištisa, o sudaryta iš kelių atskirų modelių, priimančių skirtingai apdirbtus duomenis (viena tokių grandinių 28 paveiksle), taip formuojamas savotiškas modelių ansamblis (angl. *ensemble*). Tokia specifika leidžia atrasti naujas algoritmų kombinacijas, kurių tyrėjai galėjo neįvertinti. Taip rastos procesų grandinės vėliau gali būti optimizuojamos atskirai, naudojant tikslesnę hiperparametrų paiešką (pvz.: tinklelio paiešką).

Neuroninių tinklų architektūros paieškai buvo naudojamas *AutoKeras* modulis. Šis modulis leidžia be išankstinių žinių apie gilųjį mokymą ar net apie duomenų aibę, sudaryti neuroniniais tinklais paremtą duomenų analizės modelį. *AutoKeras* modelį sudaro blokai, kurie pagal tam tikrus nurodymus sudėlioja reikiamus neuroninio tinklo sluoksnius. Šių sluoksnių bei tinklo apmokymo parametrai sudaro paieškos parametrų aibę, joje ieškoma tinkamiausio varianto.

Bandymai buvo vykdomi asmeniniame kompiuteryje, jo parametrai:

- OS: Ubuntu 20.04.1 LTS;
- CPU: Intel Core i5-8500 3.00GHz. 6 branduoliai, 6 gijos;
- GPU: GeForce GTX 1080 Ti, 11GB;
- RAM: 16GB.

4.3. Bandymų sudarymas bei vykdymo eiga

Viso darbo tikslas – sudaryti metodiką, skirtą virpesinės spektrometrijos duomenų analizės modelių suradimui, todėl daugiausia dėmesio skiriama spektrometrijoje naudojamiems metodams. Šių metodų aibę sudarė 1.3 skyriuje aptarti spektrinių duomenų apdorojimo algoritmai bei PLS modelis. Paieška buvo atlikta pasitelkiant *TPOT*, detalesnė vykdymo eiga aptariama 4.3.1 skyriuje.

Bandymų metu iškelta hipotezė dėl kitokio tipo modelių ir metodų tinkamumo uždavinio sprendimui. Remiantis 2.1 skyriuje aprašytais metodais, sudaryta papildoma duomenų analizės modelių aibė. Iškyla klausimas, ar sudaryta nauja aibė bus sujungta su prieš tai naudotu rinkiniu, ar tai bus naujas tyrimas. Priimtas sprendimas atskirti šiuos tyrimus ir paruošti naują duomenų apdorojimo ir analizės aibę. Išnagrinėjus *TPOT* siūlomus metodus, pasirinkta viena iš numatytųjų metodų ir parametrų aibių – *TPOT-light*[4] (konfigūracija pateikiama A priede). Šį rinkinį sudaro gana paprasti analizės ir duomenų apdorojimo modeliai, esminiai mašininio mokymo modeliai aptariami 2.1 skyriuje. Ieškant geriausio *TPOT-light* rinkinio, iškilo problema – nepaisant aibėje esančių požymių atrinkimo metodų, didelis dimensijų skaičius bei požymių kolinearumas apsunkina mokymo procesą, tai metodiką iš karto paverčia nepraktiška bet kokiam panaudojimui. Norint pašalinti šią problemą, prieš paiešką buvo naudojamas spektrinių požymių atrinkimo metodas MCUVE-PLS.

Taip pat, dėl DNT panaudojamumo vykdant virpesinės spektrometrinės duomenų analizės uždavinius [32, 30, 74, 44], priimtas sprendimas palyginti tokių modelių veikimą su *TPOT* įrankiu rastais rezultatais. Atsižvelgus į darbo tikslą – sudaryti automatinės paieškos metodiką – DNT sudarymui pasitelktas automatizuotas įrankis *AutoKeras*.

Apibendrinant, bandymai buvo suskirstyti į tris tipus:

1. Spektrometrijos duomenų analizėje naudojamų metodų bandymai;

[4] https://github.com/EpistasisLab/tpot/blob/master/tpot/config/classifier_light.py, https://github.com/EpistasisLab/tpot/blob/master/tpot/config/regressor_light.py [Žiūrėta:2021-01-04]

2. *TPOT-light* bandymai;

3. Neuroninių tinklų bandymai.

Bandymuose Nr. 1 ir Nr. 2 bus naudojamas genetinio optimizavimo algoritmas, neuroninių tinklų architektūros optimizuotos atsitiktinai. Visi tyrimai atlikti *Jupyter Notebook* aplinkoje, naudojant *python* (3.8) kalbą. Magistro baigiamasis darbas neapima paieškos laiko stebėjimo, tačiau vykdymo laikus galima rasti B priede.

Genetiniam optimizavimo algoritmui nurodyta atlikti 200 generacijų. Kiekviena iš jų buvo sudaryta iš 300 individų. Genetinio algoritmo vidinė patikra vykdyta 5-padalinimų principu. Suradus geriausią modelį, atlikta kryžminė patikra: tabletėms ir vištienai vieno mėginio atmetimo principu (atsižvelgiant į mėginių grupes), tyrelės atveju – atmetant trečdalį duomenų[5]. Architektūrų paieška plačiau aptariama 4.3.3 skyriuje.

4.3.1. Spektrometrijos duomenų analizės metodų bandymai

Bandymų uždavinys: naudojantis genetiniu algoritmu, rasti optimalią duomenų apdorojimo, analizės metodų parametrų rinkinius (bei šių metodų seką). Šią atliktų bandymų aibę sudaro keli atskiri tyrimai:

- bazinis spektrometrijoje naudojamų metodų tyrimas – SM;
- SM tyrimas su padvigubintu populiacijos dydžiu – SMP;
- SM tyrimas su padvigubintu generacijų skaičiumi – SMG;
- SM tyrimas, naudojant bazinės linijos pataisymo funkcijas – SMB;
- SM tyrimas, naudojant MCUVE požymių atrinkimo metodiką – SM-MCUVE;
- SM tyrimas, naudojant genetinio algoritmo požymių atrinkimo metodiką – SM-GA;

Metodas	**Parametrai**	
	Parametras	**Galimos reikšmės**
PLS (regresorius arba klasifikatorius)	komponentų skaičius	$2 \leq x \leq 14$, žingsnio dydis 2
	ar duomenys bus papildomai normalizuojami	loginis kintamasis
SG filtras	lango ilgis	$5 \leq x \leq 29$, žingsnio dydis 2
	polinomo laipsnis	2,3
	išvestinės laipsnis	0,1,2
MSC	ar duomenys bus centruojami	loginis kintamasis
tiesinės funkcijos korekcija	–	–
normalizavimas	normalizavimo metodas	minmax, vektorinis, standartinis (SNV)
regionų kaukė	regionų skaičius	10
	regionų kaukė	$2^0 \leq x \leq 2^{10} - 1$, žingsnio dydis 1

5 lentelė. SM metodų ir parametrų aibė

[5] Tyrelės validavimo aibė, dėl informacijos trūkumo, gali nesutapti su tyrėjų validavimo aibe.

Naudojami metodai daugiausia paimti iš *python* kalbai skirtos mašininio mokymo bibliotekos *scikit-learn* [41].

SMG bei SMP bandymams naudota ta pati parametrų ir metodų aibė kaip SM, tačiau čia atitinkamai generacijų bei populiacijos dydžio skaičius buvo dvigubinamas (SMG buvo vykdomas 400 generacijų, populiacijos dydis – 300; SMP buvo vykdomas 200 generacijų, populiacijos dydis – 600).

SMB metodų aibė papildyta bazinės linijos poslinkius koreguojančiomis funkcijomis, esančiomis 6 lentelėje. Taip pat šiame žingsnyje, dėl prasto rezultatyvumo, buvo atmestas požymių atrinkimo žingsnis (regionų kaukė).

Metodas	Parametrai	
	Parametras	Galimos reikšmės
spektro korekcija polinomine funkcija	polinomo laipsnis	$1 \leq x \leq 4$, žingsnio dydis 1
spektro korekcija guminės juostos metodu	–	–
spektro korekcija ALS metodu	λ	$10^x, 2 \leq x \leq 9$, žingsnio dydis 1
	p	$10^x, -3 \leq x \leq -1$, žingsnio dydis -1
spektro korekcija ARPLS metodu	λ	$10^x, 2 \leq x \leq 9$, žingsnio dydis 1

6 lentelė. Bazinės linijos korekcijos metodų ir parametrų aibė

Bazinės linijos korekcijos metodai paimti iš *python* Ramano spektrometrijos analizei specializuotos bibliotekos *rampy*. Šie metodai transformuoti taip, kad atitiktų *scikit-learn* transformatorių klasių struktūrą.

Atlikus SM bandymus, gautos duomenų apdorojimo procesų grandinės, jų rezultatas perduodamas klasifikatoriui (arba regresoriui). Šios grandinės papildytos duomenų požymio atrinkimo žingsniais (svarbiausi požymiai atrinkti prieš duomenų transformavimą), taip gautos naujos duomenų aibės. Svarbiausių spektrinių požymių atrinkimui naudoti du metodai – MCUVE (SM MCUVE) bei genetinis algoritmas (SM GA). Spektrinių požymių atrinkimui naudotas R programinės kalbos modulis plsVarSel [38], modulis buvo naudojamas per *python* sąsają. Dėl skirtingo gautų rezultatų pobūdžio, kiekvienai duomenų aibei šie metodai pritaikyti skirtingai (žr. 22 paveikslą):

- vištienos būsenos nustatymo uždaviniui transformuoti požymiai atrenkami prieš klasifikatoriaus apmokymą;
- tyrelės rūšies nustatymo uždaviniui duomenų atrinkimas buvo naudotas dukart (dėl išsišakojusio duomenų transformavimo);
- veikliosios medžiagos nustatymo tabletėje uždaviniui MCUVE ir GA požymių atrinkimo algoritmai nebuvo taikomi (dėl jau optimizavimo metu atrinktų požymių).

4.3.2. *TPOT* metodų bandymai

Bandymų uždavinys: naudojantis genetiniu algoritmu, rasti optimalią duomenų apdorojimo, analizės metodų parametrų rinkinius (bei šių metodų seką). Metodų aibės skiriasi priklausomai nuo to, ar buvo vykdoma regresija, ar klasifikavimas. A priedo 10 lentelėje pateikiami duomenų transformavimo metodai, įterpti į paiešką nepriklausomai nuo užduoties tipo. Klasifikacijos bei regresijos metodų aibės buvo papildytos metodais, nurodytais A.11 ir A.12 lentelėse.

4.3.3. Neuroninių tinklų bandymai

Bandymų uždavinys: naudojantis atsitiktine paieška, rasti optimalią neuroninių tinklų architektūrą bei jos parametrus. Paieškai sudaryti naudota viena iš numatytųjų parinkčių, pagal šiuos nustatymus rastoje architektūroje galimi tokie sluoksniai:

- kategorijų kodavimas;
- pilnai sujungtas sluoksnis;
- rinkinio normalizavimas;
- ReLU sluoksnis;
- atmetimo sluoksnis;
- sigmoido funkcijos sluoksnis (klasifikacijai).

Atsitiktinė architektūros paieška vykdyta 1000 iteracijų. Kiekvienoje paieškos iteracijoje tinklo apmokymas vykdomas 1000 epochų (ankstyvas sustojimas vykdomas, jei per 30 epochų nepasiekiamas geresnis rezultatas). Prieš atliekant paiešką, 20 % duomenų (atsižvelgiant į mėginių grupes) buvo priskirta validacijai. Modelių paieškos metu, vidinė validacija vykdyta atrenkant 20 % apmokymo aibės duomenų. Kadangi genetinio algoritmo optimizavimo validacijai naudota kitokia strategija, norint palyginti gautus rezultatus, rastos procesų grandinės turėjo būti validuotos taip pat, kaip ir neuroninių tinklų modeliai.

AutoKeras rezultatai palyginti ir su gana paprasto konvoliucinio tinklo rezultatais. Šis modelis mokytas 1000 epochų, duomenys dalinti 60 %, 20 %, 20 % (kaip ir paieškos atveju). Modelį sudarė tokie sluoksniai:

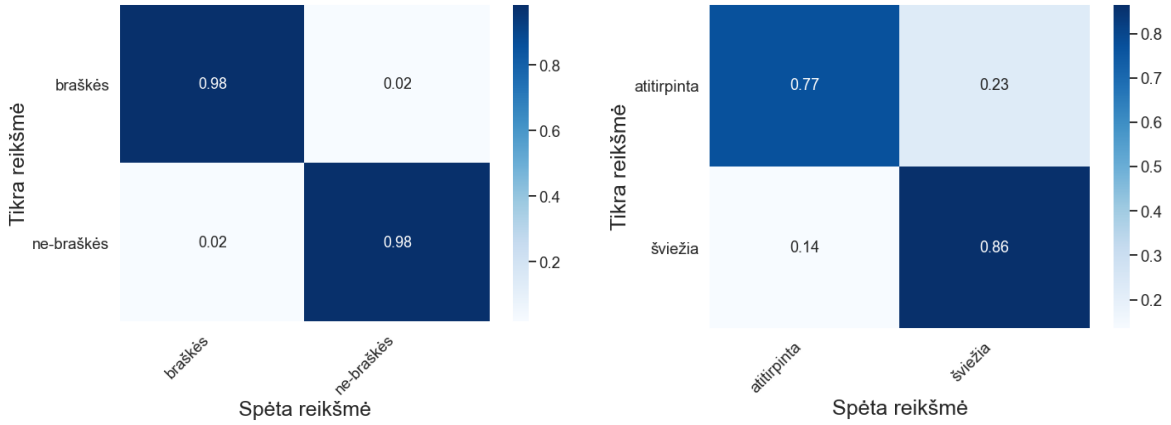
1. Įvestis;
2. 1D konvoliucija + ReLu;
3. 1D konvoliucija + ReLU;
4. Sujungimas;
5. Ištiesinimas;
6. Atmetimas;
7. ReLU;
8. Išvestis (softmax arba tiesinė funkcija).

4.4. Rezultatai bei jų aptarimas

4.4.1. Spektrometrijos duomenų analizės metodų bandymai

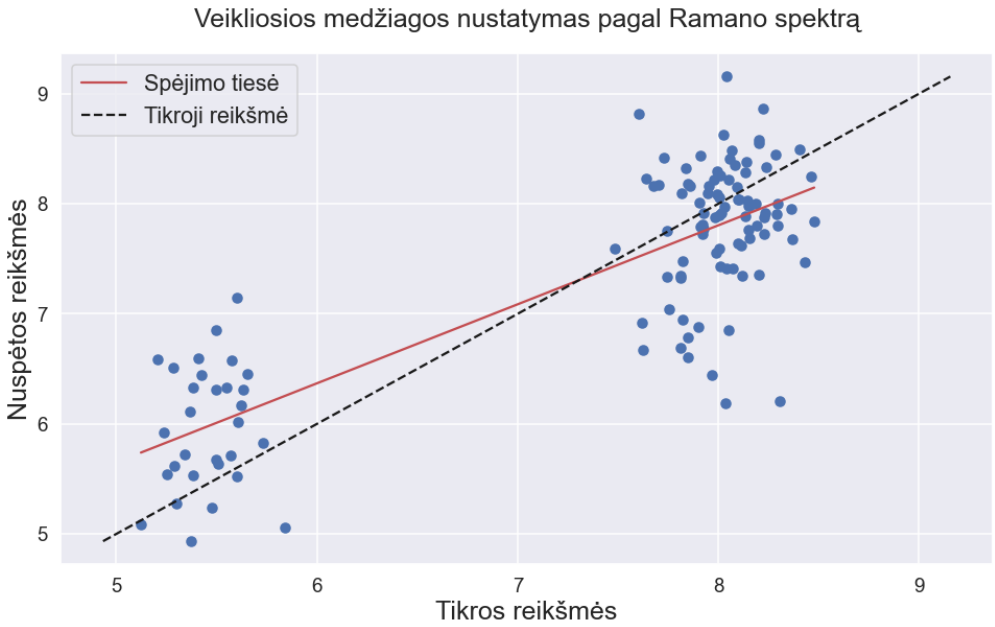
		Geriausias pasiektas rezultatas					
Duomenų rinkinys	Metrika	SM	SMP	SMG	SMB	SM-MCUVE	SM-GA
Vištiena	Tikslumas	0.8143	**0.8237**	0.8143	0.8108	0.7679	0.7342
Tyrelė	Tikslumas	**0.9817**	0.9787	**0.9817**	0.9787	0.9573	0.9512
Tabletės	RMSE	**0.5756**	0.6282	0.6306	0.7368	–	–

7 lentelė. Geriausių rastų grandinių kryžminės patikros, pagal straipsniuose pateikiamas strategijas, rezultatai.



(a) Tyrelės tipo nustatymo uždavinys SM ir SMG (b) Vištienos būsenos nustatymo uždavinys SMP

20 pav. Geriausių rastų spektrometrinių metodų modelių normalizuotos kryžminės patikros painiavos matricos.



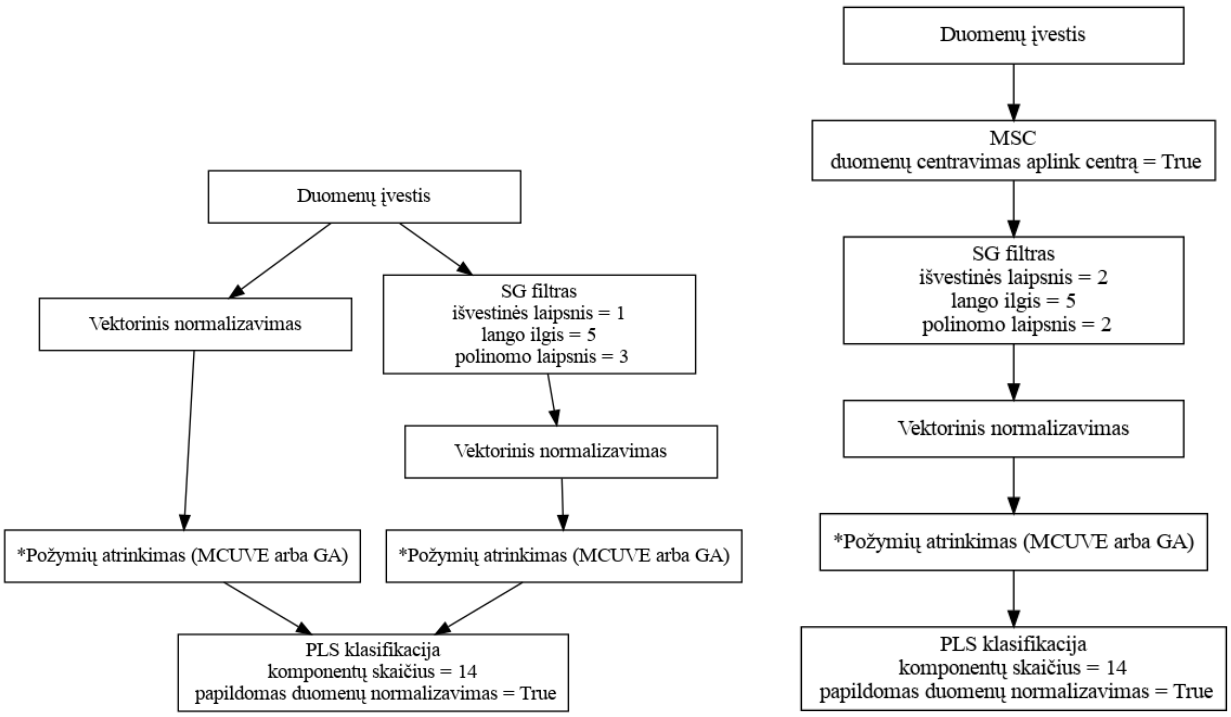
21 pav. SM geriausio rasto modelio kryžminės patikros grafikas tabletėse esančios veikliosios medžiagos kiekio nustatymo uždaviniui. Spėjimo tiesė žymi nuspėtos reikšmės priklausomybę nuo tikrosios reikšmės.

23, 24 paveiksluose pavaizduotos geriausios rastos grandinės, jų rezultatai paryškinti 7 lentelėje. Visas sudarytas grandines galima rasti C priede.

Generacijų arba populiacijos skaičiaus padvigubinimas, lyginant su SM, suveikė prasčiau vienu atveju iš trijų. Galimai parinkti generacijų ir populiacijos dydžių skaičiai nebuvo optimalūs uždaviniui.

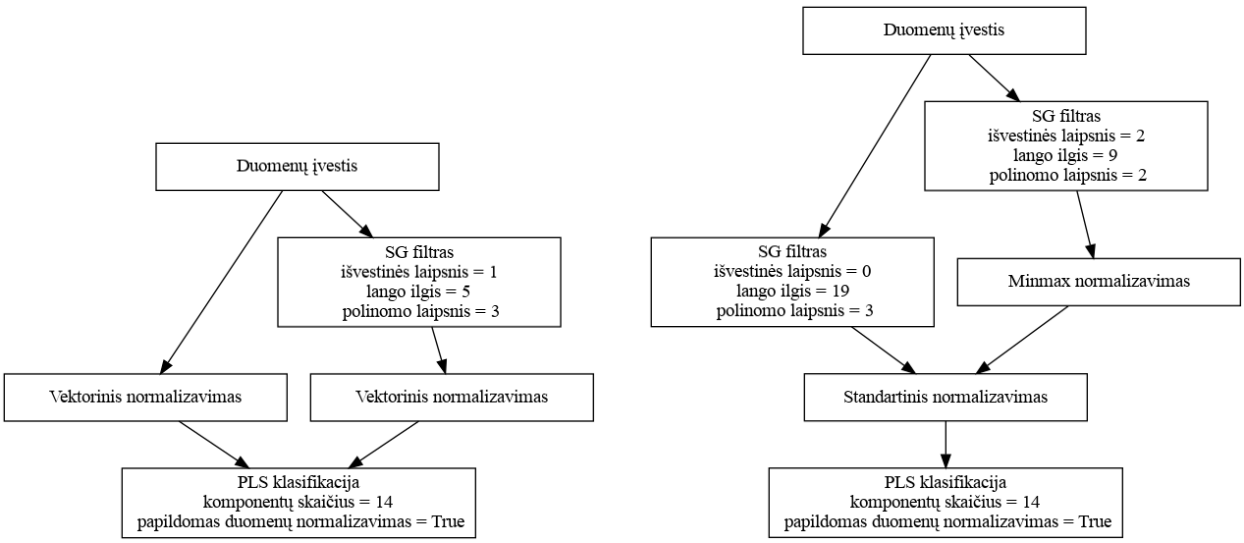
Bazinės linijos transformacijos metodų įtraukimas nedavė geresnių rezultatų. Rasti modeliai bazinės linijos korekcijos žingsniui naudojo tik polinominio laipsnio funkcijos pritaikymo algoritmus (žr. C.31, C.32 paveikslus), arba šio žingsnio visiškai neįtraukė į galutinę grandinę (žr. C.30 paveikslą).

SM atveju, duomenų parinkimo žingsnis (žr. 24 paveikslą) parinktas tik tablečių duomenų aibei (atrinkti požymiai atvaizduoti prieduose D.33 paveiksle) . Papildomas požymių atrinkimas atliktas jau rastoms tyrelės bei vištienos procesų grandinėms – PLS-MCUVE ir PLS-GA žingsniai atlikti prieš pat PLS klasifikaciją (žr. 22 paveikslą, atrinkti požymiai atvaizduoti D.34 ir D.35 paveiksluose), tuomet modelis apmokomas tik su atrinktais bruožais (prieš tai įvykdant reikiamas transformacijas pilnam spektrui). Toks žingsnis pasirodė neefektyvus.



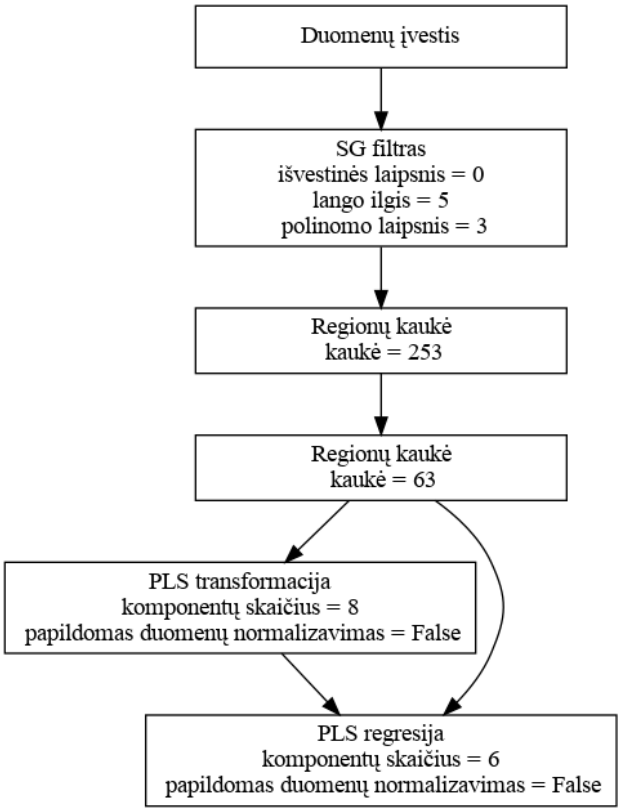
(a) tyrelės tipo nustatymas (b) vištienos filė būsenos nustatymas

22 pav. SM-MCUVE, SM-GA bandymo metu naudotos procesų grandinės klasifikacijų duomenų aibėms.



(a) tyrelės tipo nustatymas (b) vištienos filė būsenos nustatymas

23 pav. *TPOT* surastos procesų grandinės klasifikacijų duomenų aibėms.

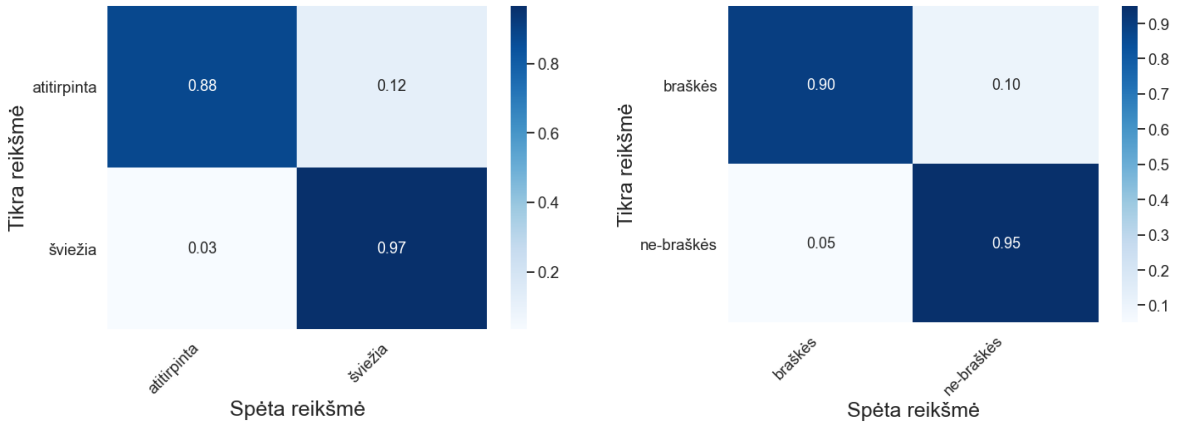


24 pav. *TPOT* surastos procesų grandinės tablečių nustatymo duomenų aibei.

4.4.2. *TPOT* numatytųjų metodų bandymai

		Geriausias pasiektas rezultatas		
Duomenų rinkinys	Metrika	*TPOT -light*	SM*	Straipsnis
Vištiena	Tikslumas	0.8347	0.8237	**0.8760**
Tyrelė	Tikslumas	0.9573	**0.9817**	0.9350
Tabletės	RMSE	**0.2769**	0.5756	0.5600

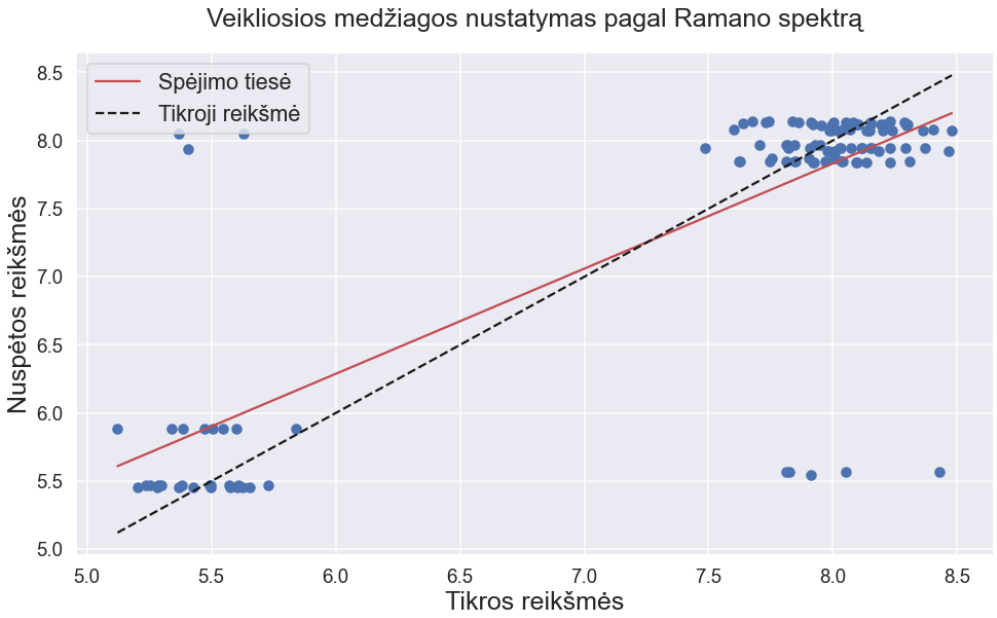
8 lentelė. Geriausių *TPOT-light* ir spektrometrinių duomenų analizės metodų (SM*) aibėse rastų grandinių kryžminės patikros, pagal straipsniuose pateikiamas strategijas, rezultatai.



(a) Višienos būsenos nustatymo uždavinys

(b) Tyrelės tipo nustatymo uždavinys

25 pav. *TPOT-light* geriausių rastų modelių normalizuotos kryžminės patikros painiavos matricos.



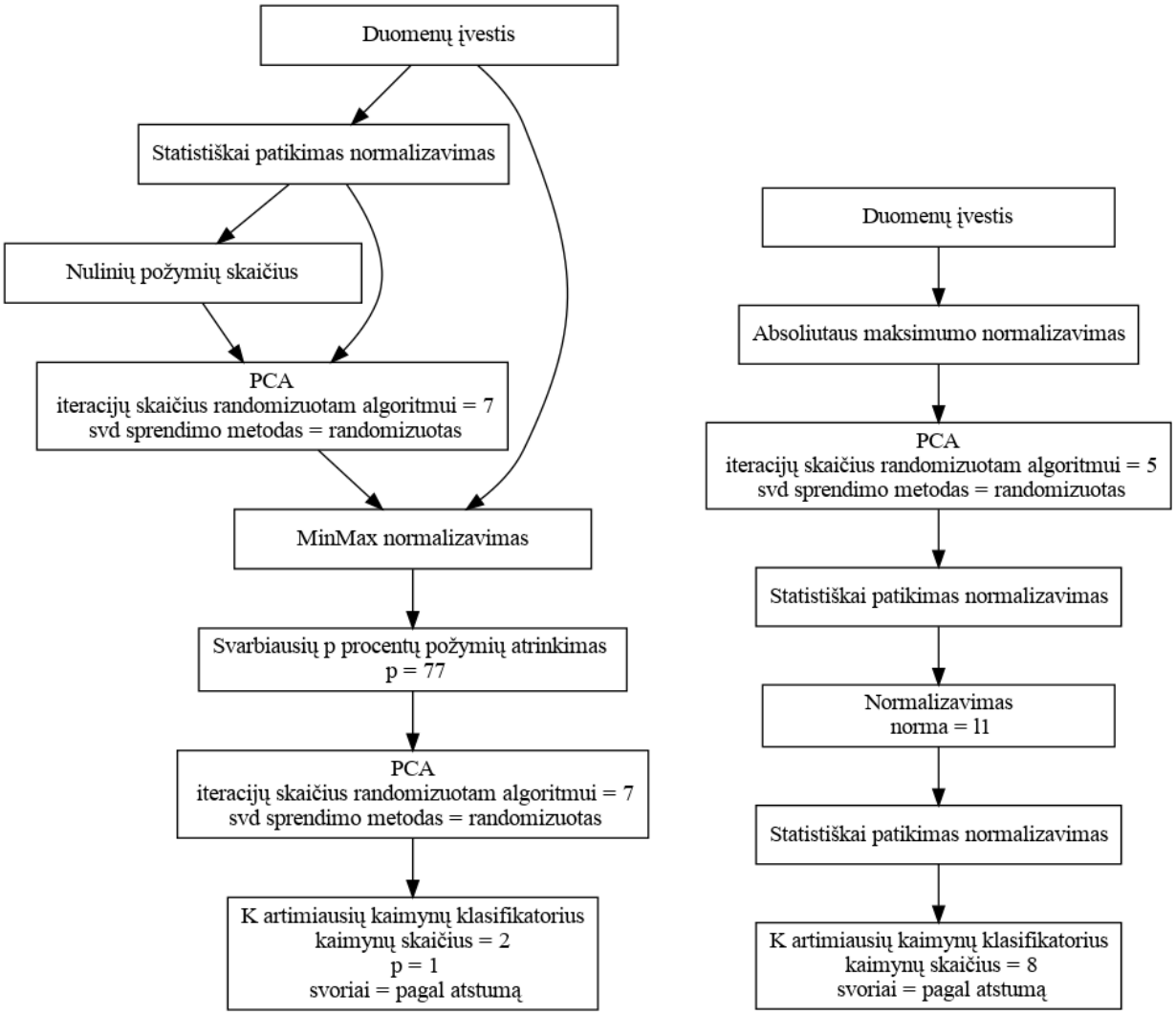
26 pav. Geriausio rasto modelio kryžminės patikros grafikas tabletėse esančios veikliosios medžiagos kiekio nustatymo uždaviniui. Spėjimo tiesė žymi nuspėtos reikšmės priklausomybę nuo tikrosios reikšmės.

TPOT-light metodų aibė pasirodė geresnė dviem atvejais iš trijų, 27, 28 paveiksluose pavaizduotos geriausios rastos procesų grandinės. Nors tiek vištienos, tiek tyrelės atvejais rezultatų skirtumai nėra ypatingi, tablečių duomenims rezultatai pranoko bet kokius lūkesčius. Šis rezultatas gali būti lyginamas su autorių atlikta NIR duomenų analize (geriausias RMSE rezultatas 0.3). Tai ne tik įrodo metodikos efektyvumą, tačiau atskleidžia ir Ramano spektrometrijos galimybes, sprendžiant veikliosios medžiagos kiekio nustatymo uždavinį.

Dėl pasitaikiusios klaidos kryžminės patikros metu, 27 paveiksle pavaizduota vištienos analizės grandinė buvo validuota klaidingai – neatrenkant duomenų požymių, kas buvo atliekama paieškos metu. Tokia klaida privedė prie 93.33 % tikslaus modelio, kuris žymiai lenkia SMP bandymą, bei straipsnio autorių gautus rezultatus. Nors šis rezultatas vertas aptarimo, jis neįtrauktas į palyginimo lentelę. Pabandžius pakartoti šį žingsnį kitoms dviems aibėms, rezultatai gavosi prastesni, todėl

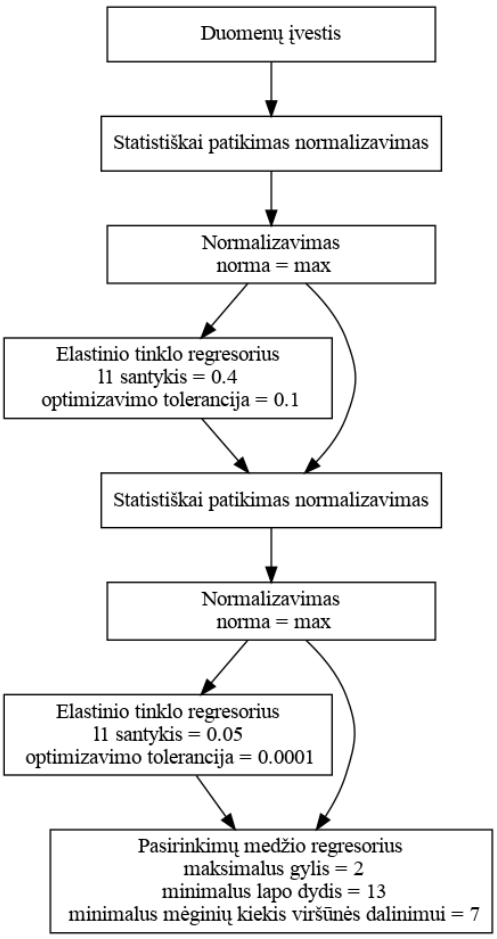
greičiausiai toks rezultatas – išimtis, o ne taisyklė. Bet kokiu atveju, toks patikrinimas jau rastai grandinei nereikalauja daug resursų, todėl galėtų būti naudojamas kaip papildomas, po paieškos atliekamas, žingsnis.

Statistiškai patikimas normalizavimas (angl. *robust scaler*) pašalina duomenų medianą (standartinis normalizavimas pašalina vidurkį) bei sutraukia duomenis į pirmo ir trečio kvartilio rėžius (minmax sutraukia į 0–1 rėžius). Tokio tipo normalizavimas priskirtas visoms duomenų analizės grandinėms. Šį metodą reikšminga įvertinti ir SM aibės rėmuose.



(a) tyrelės tipo nustatymas (b) vištienos filė būsenos nustatymas

27 pav. *TPOT* surastos procesų grandinės klasifikacijų duomenų aibėms.



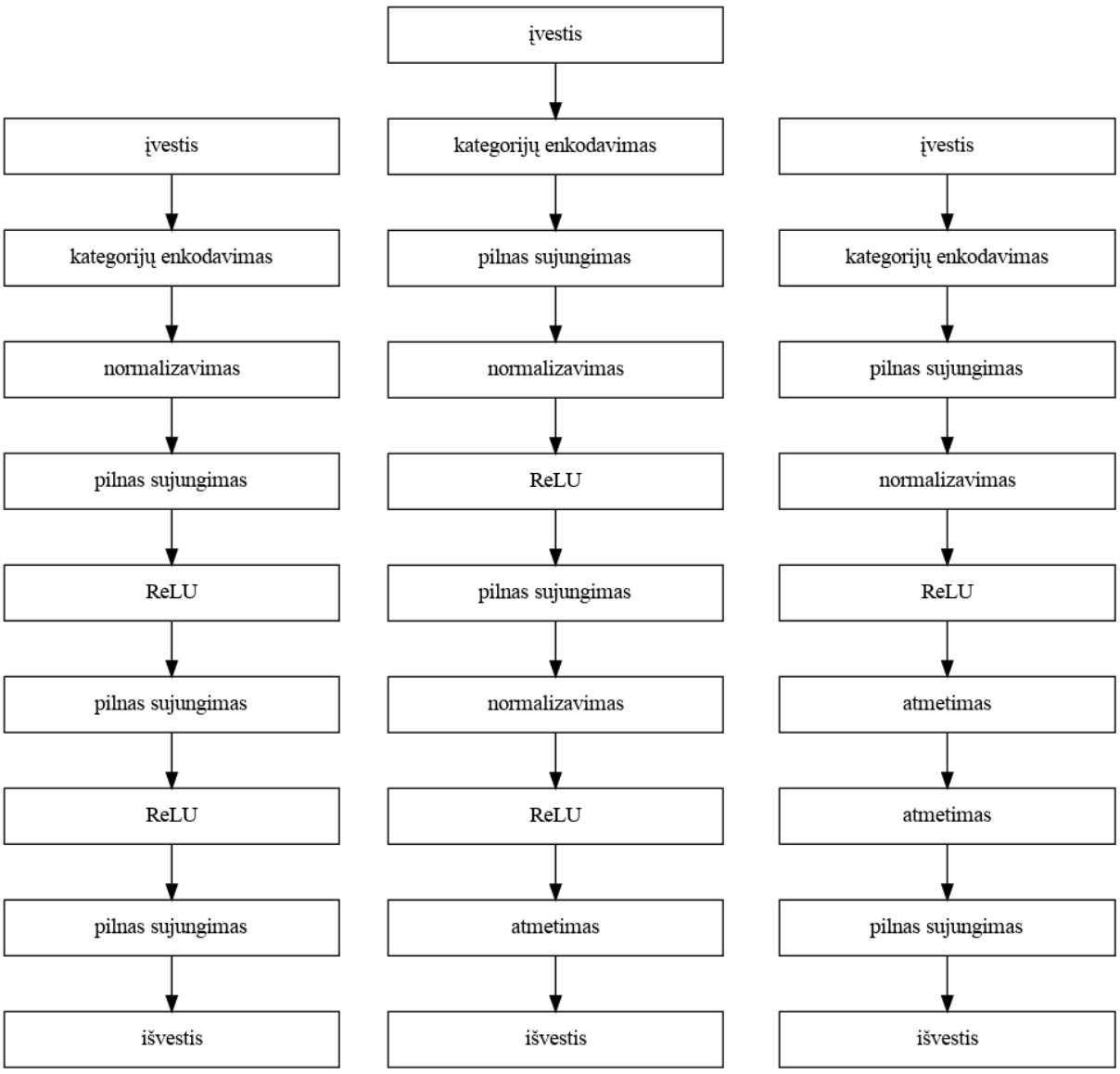
28 pav. *TPOT* surastos procesų grandinės tablečių nustatymo duomenų aibei.

4.4.3. Neuroninių tinklų bandymai

		Geriausias pasiektas rezultatas			
Duomenų rinkinys	Metrika	CNN	*AutoKeras*	SM*	*TPOT-light*
Vištiena	Tikslumas	0.7548	0.3806	0.8097	**0.8548**
Tyrelė	Tikslumas	0.9746	0.9797	0.9797	**0.9848**
Tabletės	RMSE	2.0742	6.0099	0.4874	**0.2226**

9 lentelė. Konvoliucinių tinklų, *AutoKeras* rasto tinklo ir prieš tai rastų modelių (spektrometrinių duomenų analizės metodai pažymėti SM* ir *TPOT-light*) validavimo rezultatas. Prieš tai sudaryti modeliai apmokyti ir validuoti su tais pačiais duomenimis, kaip ir neuroniniai tinklai.

Tiek *AutoKeras*, tiek bazinio CNN rezultatai nepranoko SM* ar *TPOT-light* rezultatų. Nors *AutoKeras* ir pasirodė neblogai tyrelės duomenų aibei (rezultatai geresni nei CNN ir tokie patys kaip SM), kiti rezultatai yra per prasti, kad ši metodika būtų laikoma bent kiek efektyvia ir verta dėmesio. *AutoKeras* paieška vištienos ir tyrelės duomenims veikė trumpiau nei nurodyta (įvykdyti atitinkamai 140 ir 104 bandymai). Nėra aišku, ar rezultatams pritrūko apmokymo laiko (epochų), ar sutrukdė *AutoKeras* ankstyvojo sustojimo mechanizmas. Taip pat pakenkti galėjo naudojamos numatytosios konfigūracijos. 29 paveiksle pavaizduotos rastos tinklų architektūros.



(a) tyrelės tipo nustatymas. Išvesties aktyvacijos funkcija – sigmoidas.

(b) tabletėse esančios veikliosios medžiagos kiekio nustatymas. Išvesties aktyvacijos funkcija – tiesė.

(c) vištienos filė būsenos nustatymas. Išvesties aktyvacijos funkcija – sigmoidas.

29 pav. AutoKeras surastos architektūros skirtingoms duomenų aibėms. Pilnas sujungimo sluoksnis nurodo tiesinę aktyvacijos funkciją.

Išvados ir rekomendacijos

Šiame darbe pristatytos kelios automatinio duomenų analizės modelio sudarymo metodikos, naudojamos virpesinės spektrometrijos duomenims . Įrodyta, kad įprasti spektrometrinių duomenų apdorojimo metodai gali būti nereikalingi, norint sudaryti patikimą modelį.

Prasčiausi rezultatai pasiekti naudojantis *AutoKeras* automatiniu neuroninių tinklų architektūros paieškos įrankiu. Vienu atveju iš trijų rezultatai buvo patenkinami, tačiau kitų dviejų bandymų rezultatai buvo per prasti, kad būtų galima šią metodiką vadinti efektyvia. *TPOT-light* ir spektrometrinių metodų aibėse vykdomos paieškos rezultatai nevienareikšmiški. Nors genetinės paieškos rezultatai buvo geresni tyrelės (SMP metodų aibė) ir tablečių (*TPOT-light* metodų aibė) duomenų aibėms, gauti sudarytos metodikos rezultatai nepranoko Parastar et al. [40] rezultatų vištienos duomenų aibei. Kaip jau aptarta *TPOT-light* rezultatuose, dėl įsivėlusios klaidos buvo rastas papildomas žingsnis, kurį atlikus galima gauti geresnį rezultatą ir pagerinti Parastar rezultatus. Svarbu atsižvelgti į tai, kad duotajame straipsnyje [40] duomenų analizės grandinės paieška vykdoma pasitelkiant optimalios paieškos metodiką.

Apžvelgus rezultatus, optimaliam virpesinės spektrometrijos analizės modelių sudarymui siūloma atlikti sekančius žingsnius:

1. Požymių atrinkimas MCUVE-PLS metodu;

2. *TPOT* įrankio panaudojimas su šiais parametrais: *TPOT-light* numatytieji metodai, 200 generacijų, populiacijos dydis 300, 5-dalių vidinė kryžminė patikra.

Turėtų būti atlikti tolimesni darbai, susiję su požymių reikšmingumo algoritmo parinkimu. Šiame darbe išbandyti du (įskaitant paieškos metu atliekamą maskavimą – trys) požymių atrinkimo metodai, iš kurių tik vienas buvo naudojamas su *TPOT-light* konfigūracija. Vertėtų patikrinti, ar algoritmo parinkimas turėjo didelę įtaką metodikos efektyvumui. Taip pat rasta, kad *TPOT-light* aibėje sudarytas modelis puikiai veikia ir neatrinktiems požymiams. Tokia paieška reikalauja daugiau laiko, tačiau čia vertėtų apžvelgti lygiagrečios paieškos galimybes.

Darbe naudojamos *TPOT-light* ir individualiai sudaryti SM metodų rinkiniai. *TPOT* įrankis turi ir kitokių numatytų konfigūracijų, tad atsižvelgiant į gautus rezultatus, turėtų būti išbandomos ir kitos kombinacijos (numatytieji *TPOT* rinkiniai arba SM duomenų korekcijos + *TPOT* metodų aibės). Magistro darbe nenaudotų *TPOT* aibių optimizavimas trunka ilgesnį laiką, tačiau pavykus išlygiagretinti procesus, būtų įmanoma išbandyti ir sudėtingesnius modelius.

TPOT sudaryti modeliai pasižymi savo išskirtine forma. Sumažinus parametrų kiekį, bet padidinus metodų aibę, būtų galima susikoncentruoti į pačios formos radimą. Verta ištirti tolimesnį taip sudarytų modelių optimizavimą parametrų atžvilgiu.

Literatūros šaltiniai

[1] AlphaGo: The story so far, Dec 2020. [Online; accessed 6. Dec. 2020].

[2] Chemometrics - Data Analysis Software - PLS_Toolbox - Eigenvector, Feb 2020. [Online; accessed 28. Dec. 2020].

[3] Decision Tree Regression — scikit-learn 0.24.0 documentation, Dec 2020. [Online; accessed 1. Jan. 2021].

[4] Nearest Neighbors regression — scikit-learn 0.24.0 documentation, Dec 2020. [Online; accessed 1. Jan. 2021].

[5] Unscrambler | Camo Analytics - The leader in industrial analytics, Nov 2020. [Online; accessed 28. Dec. 2020].

[6] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[7] Jacopo Acquarelli, Twan van Laarhoven, Jan Gerretzen, Thanh N. Tran, Lutgarde M. C. Buydens, and Elena Marchiori. Convolutional neural networks for vibrational spectroscopic data analysis. *Anal. Chim. Acta*, 954:22–31, Feb 2017.

[8] Sung-June Baek, Aaron Park, Young-Jin Ahn, and Jaebum Choo. Baseline correction using asymmetrically reweighted penalized least squares smoothing. *Analyst*, 140(1):250–257, Dec 2014.

[9] Sung-June Baek, Aaron Park, Young-Jin Ahn, and Jaebum Choo. Baseline correction using asymmetrically reweighted penalized least squares smoothing. *The Analyst*, 140(1):250–257, 2015.

[10] Roman M. Balabin, Ravilya Z. Safieva, and Ekaterina I. Lomakina. Gasoline classification using near infrared (NIR) spectroscopy data: Comparison of multivariate techniques. *Anal. Chim. Acta*, 671(1):27–35, Jun 2010.

[11] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012.

[12] Vítězslav Centner, Désiré-Luc Massart, Onno E. de Noord, Sijmen de Jong, Bernard M. Vandeginste, and Cécile Sterna. Elimination of Uninformative Variables for Multivariate Calibration. *Anal. Chem.*, 68(21):3851–3858, Nov 1996.

[13] François Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[14] Christophe B. Y. Cordella. PCA: The Basic Building Block of Chemometrics. In *Analytical Chemistry*. IntechOpen, Nov 2012.

[15] Ciro Augusto Fernandes de Oliveira Penido, Marcos Tadeu Tavares Pacheco, Igor K Lednev, and Landulfo Silveira Jr. Raman spectroscopy in forensic analysis: identification of cocaine and other illegal drugs of abuse. *Journal of Raman Spectroscopy*, 47:28–38, 2016.

[16] Olivier Devos, Gerard Downey, and Ludovic Duponchel. Simultaneous data pre-processing and SVM classification model selection based on a parallel genetic algorithm applied to spectroscopic data of olive oils. *Food Chemistry*, 148:124–130, April 2014.

[17] M.S. Dhanoa, S.J. Lister, R. Sanderson, and R.J. Barnes. The link between multiplicative scatter correction (MSC) and standard normal variate (SNV) transformations of NIR spectra. *Journal of Near Infrared Spectroscopy*, 2(1):43–47, January 1994.

[18] Shiyamala Duraipandian, Wei Zheng, Joseph Ng, Jeffrey J. H. Low, A. Ilancheran, and Zhiwei Huang. In vivo diagnosis of cervical precancer using raman spectroscopy and genetic algorithm techniques. *The Analyst*, 136(20):4328, 2011.

[19] M. Dyrby, S. B. Engelsen, L. Nørgaard, M. Bruhn, and L. Lundsberg-Nielsen. Chemometric quantitation of the active substance (containing c≡n) in a pharmaceutical tablet using near-infrared (NIR) transmittance and NIR FT-raman spectra. *Applied Spectroscopy*, 56(5):579–585, May 2002.

[20] John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull. Graphviz and dynagraph – static and dynamic graph drawing tools. In *GRAPH DRAWING SOFTWARE*, pages 127–148. Springer-Verlag, 2003.

[21] Hugo Jair Escalante. Automated Machine Learning – a brief review at the end of the early years. *arXiv*, Aug 2020.

[22] Dr. Michael J. Garbade. Clearing the Confusion: AI vs Machine Learning vs Deep Learning Differences. *Medium*, Sep 2018.

[23] Jan Gerretzen, Ewa Szymańska, Jeroen J. Jansen, Jacob Bart, Henk-Jan van Manen, Edwin R. van den Heuvel, and Lutgarde M. C. Buydens. Simple and Effective Way for Data Preprocessing Selection Based on Design of Experiments. *Anal. Chem.*, 87(24):12096–12103, Dec 2015.

[24] Paul H C Eilers and Hans F M Boelens. Baseline correction with asymmetric least squares smoothing. 2005.

[25] Philip Heraud, Bayden R. Wood, John Beardall, and Don McNaughton. Effects of pre-processing of raman spectra onin vivo classification of nutrient status of microalgal cells. *Journal of Chemometrics*, 20(5):193–197, May 2006.

[26] J K Holland, E K Kemsley, and R H Wilson. Use of fourier transform infrared spectroscopy and partial least squares regression for the detection of adulteration of strawberry purées. *Journal of the Science of Food and Agriculture*, 76(2):263–269, February 1998.

[27] C-P Sherman Hsu. Infrared spectroscopy. *Handbook of instrumental techniques for analytical chemistry*, 249, 1997.

[28] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[29] Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1946–1956. ACM, 2019.

[30] Afroditi Kapourani, Vasiliki Valkanioti, Konstantinos N. Kontogiannopoulos, and Panagiotis Barmpalexis. Determination of the physical state of a drug in amorphous solid dispersions using artificial neural networks and ATR-FTIR spectroscopy. *International Journal of Pharmaceutics: X*, 2:100064, Dec 2020.

[31] Trang T Le, Weixuan Fu, and Jason H Moore. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics*, 36(1):250–256, 2020.

[32] Jinchao Liu, Margarita Osadchy, Lorna Ashton, Michael Foster, Christopher Solomon, and Stuart Gibson. Deep convolutional neural networks for raman spectrum recognition: A unified solution. *The Analyst*, 142, 08 2017.

[33] Yande Liu, Yibin Ying, Haiyan Yu, and Xiaping Fu. Comparison of the HPLC method and FT-NIR analysis for quantification of glucose, fructose, and sucrose in intact apple fruits. *Journal of Agricultural and Food Chemistry*, 54(8):2810–2815, April 2006.

[34] María López-López and Carmen García-Ruiz. Infrared and raman spectroscopy techniques applied to identification of explosives. *TrAC Trends in Analytical Chemistry*, 54:36–44, 2014.

[35] Howard Mark and Jerry Workman. Chapter 21 - Calculating the Solution for Regression Techniques: Part 1—Multivariate Regression Made Simple. In *Chemometrics in Spectroscopy (Second Edition)*, pages 119–120. Academic Press, Cambridge, MA, USA, Jan 2018.

[36] Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

[37] Paul F McMillan. Raman spectroscopy in mineralogy and geochemistry. *Annual review of earth and planetary sciences*, 17(1):255–279, 1989.

[38] Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Sæbø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118:62–69, 2012.

[39] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *arXiv*, Nov 2018.

[40] Hadi Parastar, Geert van Kollenburg, Yannick Weesepoel, André van den Doel, Lutgarde Buydens, and Jeroen Jansen. Integration of handheld NIR and machine learning to "measure & monitor" chicken meat authenticity. *Food Control*, 112:107149, June 2020.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[42] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007.

[43] Marius-Constantin Popescu, Valentina E. Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7), Jul 2009.

[44] Gema Puertas and Manuel Vázquez. UV-VIS-NIR spectroscopy and artificial neural networks for the cholesterol quantification in egg yolk. *J. Food Compos. Anal.*, 86:103350, Mar 2020.

[45] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

[46] Dr. Sebastian Raschka. Chapter 1: Introduction to Machine Learning and Deep Learning, Aug 2020. [Online; accessed 6. Dec. 2020].

[47] Åsmund Rinnan. Pre-processing in vibrational spectroscopy – when, why and how. *Anal. Methods*, 6(18):7124–7129, 2014.

[48] Åsmund Rinnan, Frans van den Berg, and Søren Balling Engelsen. Review of the most common pre-processing techniques for near-infrared spectra. *TrAC Trends in Analytical Chemistry*, 28(10):1201–1222, November 2009.

[49] L. E. Rodriguez-Saona and M. E. Allendorf. Use of FTIR for Rapid Authentication and Detection of Adulteration of Food. *Annu. Rev. Food Sci. Technol.*, 2(1):467–483, Feb 2011.

[50] Christopher Rowlands and Stephen Elliott. Automated algorithm for baseline subtraction in spectra. *Journal of Raman Spectroscopy*, 42(3):363–369, 2011.

[51] A. L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM J. Res. Dev.*, 3(3):210–229, Jul 1959.

[52] Kumara Sastry, David Goldberg, and Graham Kendall. Genetic algorithms. In *Search Methodologies*, pages 97–125. Springer US.

[53] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, jul 1964.

[54] Carolina S. Silva, André Braz, Maria Fernanda Pimentel, Carolina S. Silva, André Braz, and Maria Fernanda Pimentel. Vibrational Spectroscopy and Chemometrics in Forensic Chemistry: Critical Review, Current Trends and Challenges. *J. Braz. Chem. Soc.*, 30(11):2259–2290, Nov 2019.

[55] Chit Siang Soh, Kok Meng Ong, and P. Raveendran. Variable selection using genetic algorithm for analysis of near-infrared spectral data using partial least squares. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE, 2005.

[56] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[57] Emily E. Storey and Amr S. Helmy. Optimized Preprocessing and Machine Learning for Quantitative Raman Spectroscopy in Biology. *arXiv:1904.02243 [cs, eess, q-bio, stat]*, April 2019. arXiv: 1904.02243.

[58] Seng Khoon Teh, Wei Zheng, Khek Yu Ho, Ming Teh, Khay Guan Yeoh, and Zhiwei Huang. Diagnosis of gastric cancer using near-infrared Raman spectroscopy and classification and regression tree techniques. *J. Biomed. Opt.*, 13(3):034013, May 2008.

[59] A. Margarida Teixeira and Clara Sousa. A review on the application of vibrational spectroscopy to the chemistry of nuts. *Food Chem.*, 277:713–724, Mar 2019.

[60] Ernest Teye, Charles L. Y. Amuah, Terry McGrath, and Christopher Elliott. Innovative and rapid analysis for rice authenticity using hand-held NIR spectrometry and chemometrics. *Spectrochim. Acta, Part A*, 217:147–154, Jun 2019.

[61] Kluyver Thomas, Ragan-Kelley Benjamin, Peacute;rez Fernando, Granger Brian, Bussonnier Matthias, Frederic Jonathan, Kelley Kyle, Hamrick Jessica, Grout Jason, Corlay Sylvain, and et al. Jupyter notebooks ndash; a publishing format for reproducible computational workflows. *Stand Alone*, 0(Positioning and Power in Academic Publishing: Players, Agents and Agendas):87–90, 2016.

[62] Lisbeth G. Thygesen, Mette Marie Løkke, Elisabeth Micklander, and Søren B. Engelsen. Vibrational microspectroscopy of food. Raman vs. FT-IR. *Trends Food Sci. Technol.*, 14(1):50–57, Jan 2003.

[63] David Tuschel. Selecting an excitation wavelength for raman spectroscopy. *Spectroscopy*, 2016.

[64] Stéfan van der Walt, S Chris Colbert, and Gaël Varoquaux. The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, March 2011.

[65] Geert Van Kollenburg. Data of: Integration of handheld nir and machine learning for the development of a "measure amp; monitor" technology for chicken meat authenticity, 2020.

[66] G. van Rossum. Python tutorial. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.

[67] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[68] Lu Wang, Da-Wen Sun, Hongbin Pu, and Jun-Hu Cheng. Quality analysis, classification, and authentication of liquid foods by near-infrared spectroscopy: A review of recent research developments. *Critical Reviews in Food Science and Nutrition*, 57(7):1524–1538, January 2016.

[69] R. H. Wilson and H. S. Tapp. Mid-infrared spectroscopy for food analysis: recent new applications and relevant developments in sample presentation methods. *TrAC, Trends Anal. Chem.*, 18(2):85–93, Feb 1999.

[70] Svante Wold, Michael Sjöström, and Lennart Eriksson. PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130, oct 2001.

[71] Zou Xiaobo, Zhao Jiewen, Malcolm J.W. Povey, Mel Holmes, and Mao Hanpin. Variables selection methods in near-infrared spectroscopy. *Analytica Chimica Acta*, 667(1-2):14–32, May 2010.

[72] Yi Xu, Peng Zhong, Aimin Jiang, Xing Shen, Xiangmei Li, Zhenlin Xu, Yudong Shen, Yuanming Sun, and Hongtao Lei. Raman spectroscopy coupled with chemometrics for food authentication: A review. *TrAC, Trends Anal. Chem.*, 131:116017, Oct 2020.

[73] Danting Yang and Yibin Ying. Applications of raman spectroscopy in agricultural products and food analysis: A review. *Applied Spectroscopy Reviews*, 46(7):539–560, October 2011.

[74] Jie Yang, Jinfan Xu, Xiaolei Zhang, Chiyu Wu, Tao Lin, and Yibin Ying. Deep learning for vibrational spectral analysis: Recent progress and a practical guide. *Analytica Chimica Acta*, 1081:6 – 17, 2019.

[75] G. Yufeng. The 7 Steps of Machine Learning - Towards Data Science. *Medium*, Jun 2018.

[76] Zhi-Min Zhang, Shan Chen, and Yi-Zeng Liang. Baseline correction using adaptive iteratively reweighted penalized least squares. *The Analyst*, 135(5):1138, 2010.

Priedai

A. *TPOT-light* konfigūracija

Metodas	Parametrai	
	Parametras	Galimos reikšmės
požymių binarizavimas	slenkstis	$0 \leq x \leq 1$, žingsnio dydis 0.05
minmax normalizavimas	–	–
absoliutaus maksimumo normalizavimas	–	–
normalizavimas	norma	l1,l2,max
statistiškai patikimas normalizavimas	–	–
standartinis normalizavimas (skyriasi nuo SNV);	–	–
nulinių požymių skaičiavimas	–	–
RBF transformacija	γ	$0 \leq x \leq 1$, žingsnio dydis 0.05
PCA	svd sprendimo metodas	$1 \leq x \leq 10$, žingsnio dydis 1
	iteracijų skaičius randomizuotam algoritmui	
požymių atrinkimas pagal variaciją	variacijos slenkstis	0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2
geriausių *p* požymių atrinkimas pagal jų rezultatyvumą	*p*	$1 \leq x \leq 99$, žingsnio dydis 1;

10 lentelė. Duomenų transformavimo metodai

Metodas	Parametrai	
	Parametras	Galimos reikšmės
naivus Bajeso klasifikatorius su Gauso skirstiniu	–	–
naivus Bajeso klasifikatorius su Bernulio skirstiniu	α	$10^x, -3 \leq x \leq 2$, žingsnio dydis 1
	ar įvertinti klasių pasiskirstymą	loginė reikšmė
naivus Bajeso klasifikatorius su polinominiu skirstiniu	α	$10^x, -3 \leq x \leq 2$, žingsnio dydis 1
	ar įvertinti klasių pasiskirstymą	loginė reikšmė
pasirinkimų medžio klasifikatorius	kriterijus	gini, entropija
	maksimalus gylis	$1 \leq x \leq 10$, žingsnio dydis 1
	minimalus mėginių kiekis viršūnės dalinimui	$2 \leq x \leq 20$, žingsnio dydis 1
	minimalus lapo dydis	$1 \leq x \leq 100$, žingsnio dydis 1
k kaimynų klasifikatorius	kaimynų skaičius	$1 \leq x \leq 100$, žingsnio dydis 1
	svoriai	pagal atstumą, tolygūs
	p	1,2
logistinė regresija	baudos tipas	l1, l2
	C	1^{-4}, 1^{-3}, 1^{-2}, 1^{-1}, 0.5, 1, 5, 10, 15, 20, 25
	ar sprendžiamas dualus uždavinys	loginė reikšmė
RBF branduolio aproksimacija	γ	$0 \leq x \leq 1$, žingsnio dydis 0.05

11 lentelė. Klasifikacijos metodų ir parametrų aibė

Metodas	Parametrai	
	Parametras	Galimos reikšmės
elastinio tinklo regresorius	l1 santykis	$0 \leq x \leq 1$, žingsnio dydis 0.05
	optimizavimo tolerancija	$10^x, -5 \leq x \leq -1$, žingsnio dydis 1
pasrinkimų medžio regresorius	maksimalus gylis	$1 \leq x \leq 10$, žingsnio dydis 1
	minimalus mėginių kiekis viršūnės dalinimui	$2 \leq x \leq 20$, žingsnio dydis 1
	minimalus lapo dydis	$1 \leq x \leq 20$, žingsnio dydis 1
k kaimynų regresorius	kaimynų skaičius	$1 \leq x \leq 100$, žingsnio dydis
	svoriai	pagal atstumą, tolygūs
	p	1,2
Lasso regresorius su LARS algoritmu	ar papildomai atlikti duomenų normalizavimą	loginis kintamasis
tiesinių atraminių vektorių regresorius	netekties funkcija	l1,l2
	ar spendžiamas dualus uždavinys	loginis kintamasis
	optimizavimo tolerancija	$10^x, -5 \leq x \leq -1$, žingsnio dydis 1
	C	$20 \leq x \leq 210-1$, žingsnio dydis 1
	ϵ	$10^x, -4 \leq x \leq 0$, žingsnio dydis 1
Ridge regresorius	–	–
Nystroem branduolio aproksimacija	branduolys	rbf, kosinuso, χ^2, sudėtinis χ^2, Laplaso, polinominis, tiesinis, sigmoido
	γ	$0 \leq x \leq 1$, žingsnio dydis 0.05
	komponentų skaičius	$1 \leq x \leq 10$, žingsnio dydis 1

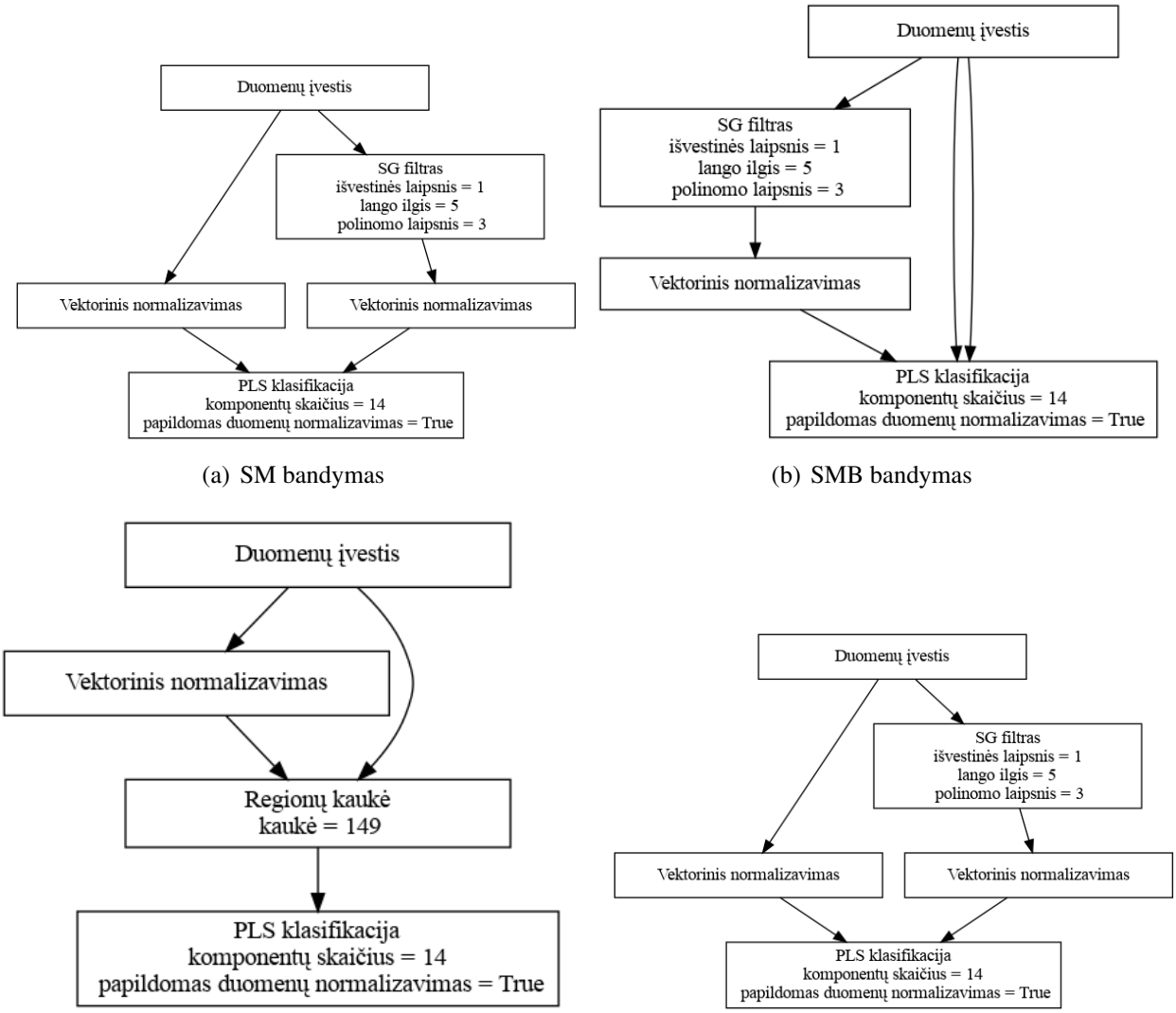
12 lentelė. Regresijos metodų ir parametrų aibė

B. Paieškos vykdymo laikas

	Vykdymo laikas (min)					
Duomenų rinkinys	SM	SMP	SMG	SMB	*TPOT-light*	AutoKeras
Vištiena	49	90	98	39	115	7
Tyrelė	37	58	82	36	54	21
Tabletės	24	48	52	72	85	760

13 lentelė. Paieškų vykdymo laikas minutėmis. Pateikti vykdymo laikai – orientaciniai – vykdymo metu galėjo nebūti užtikrintos vienodos kompiuterio apkrovos.

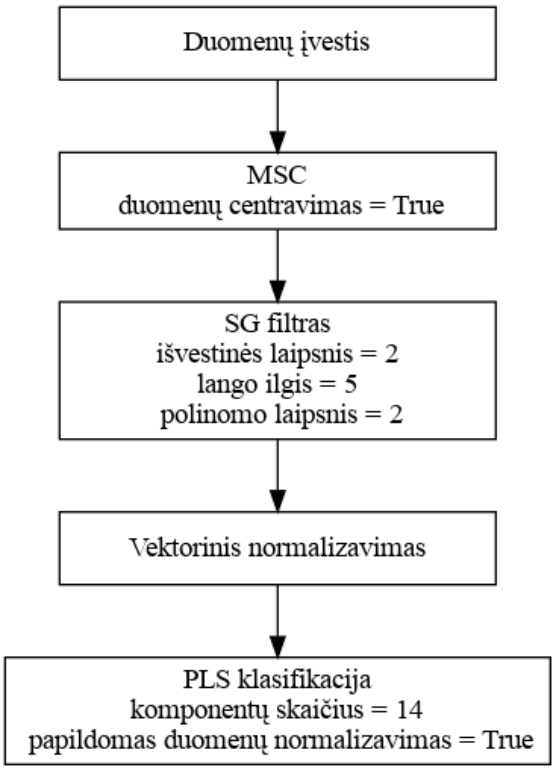
C. Spektrometrijos duomenų analizės metodų bandymai



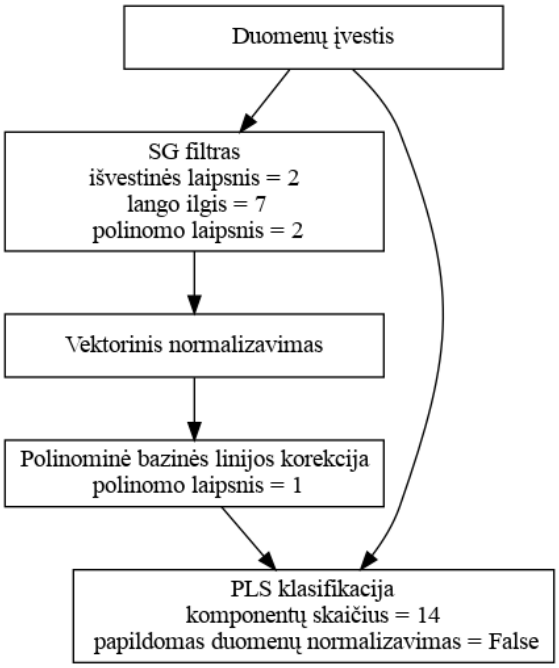
(a) SM bandymas (b) SMB bandymas

(c) SMP bandymas (d) SMG bandymas

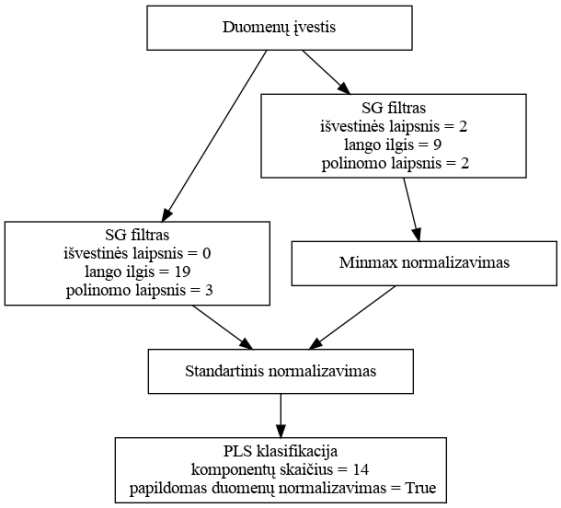
30 pav. Tyrelės tipo nustatymo uždaviniui rastos procesų grandinės



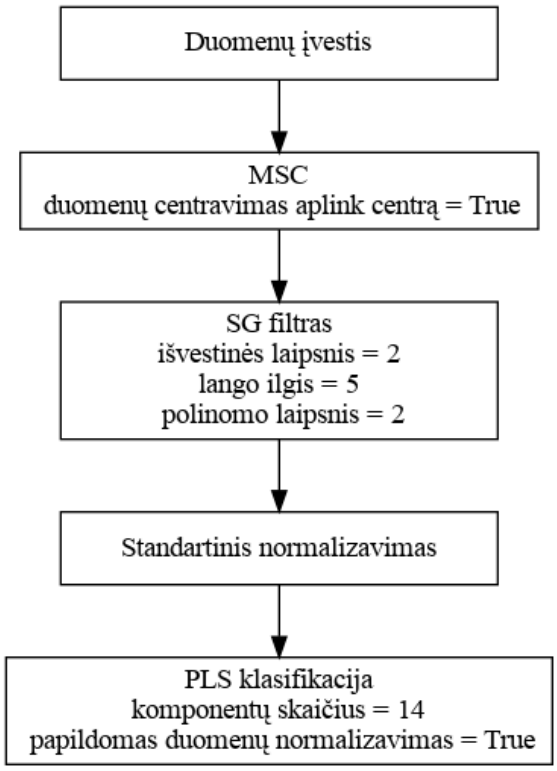
(a) SM bandymas



(b) SMB bandymas

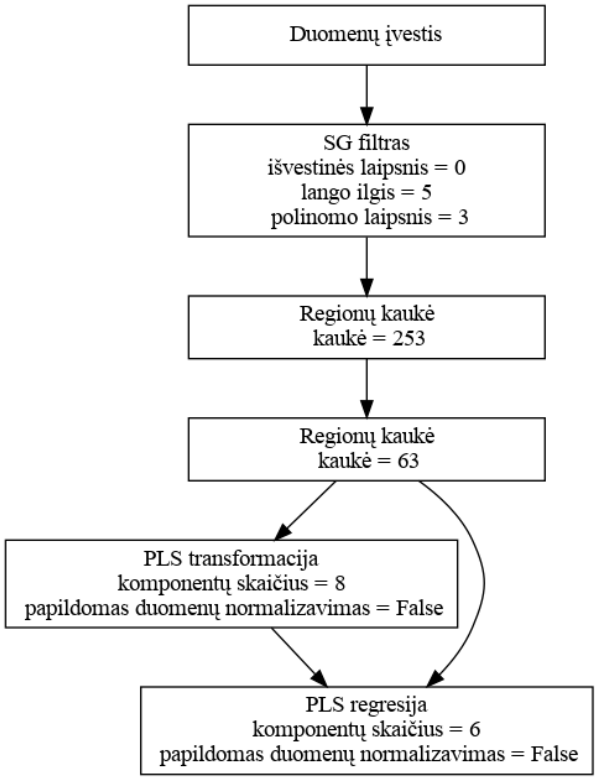


(c) SMP bandymas

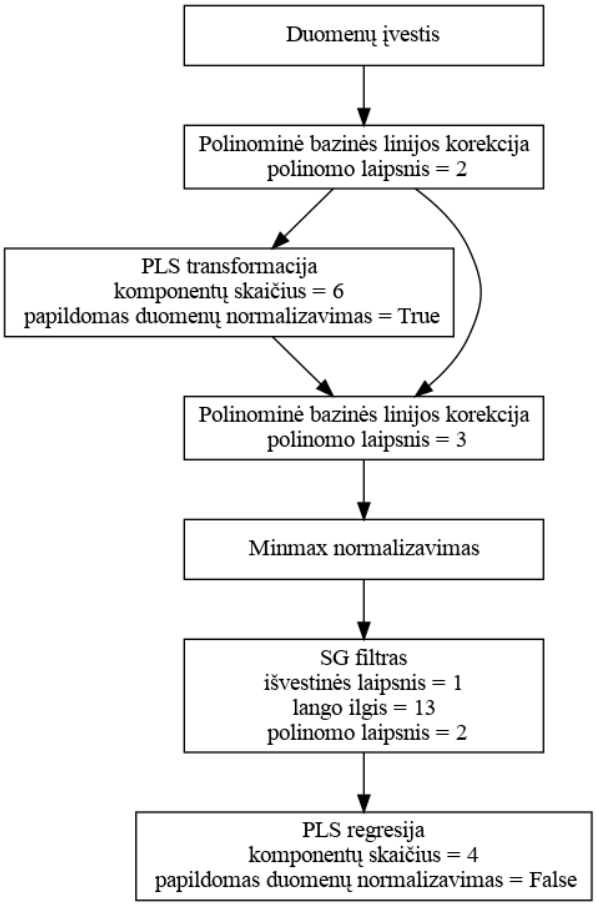


(d) SMG bandymas

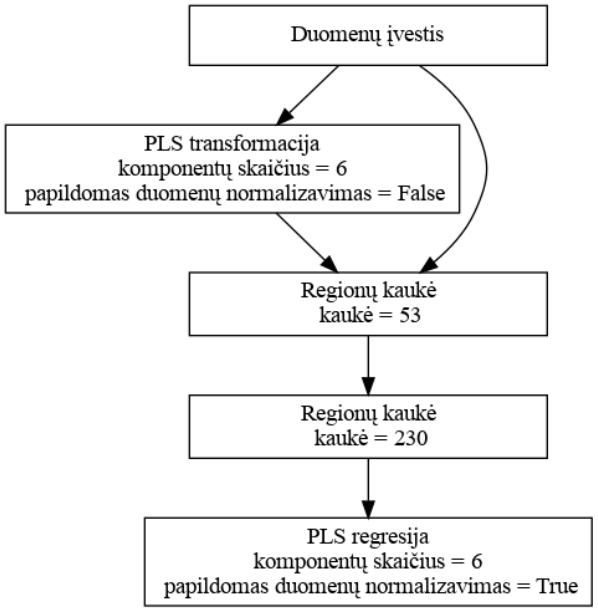
31 pav. Vištienos būsenos nustatymo uždaviniui rastos procesų grandinės.



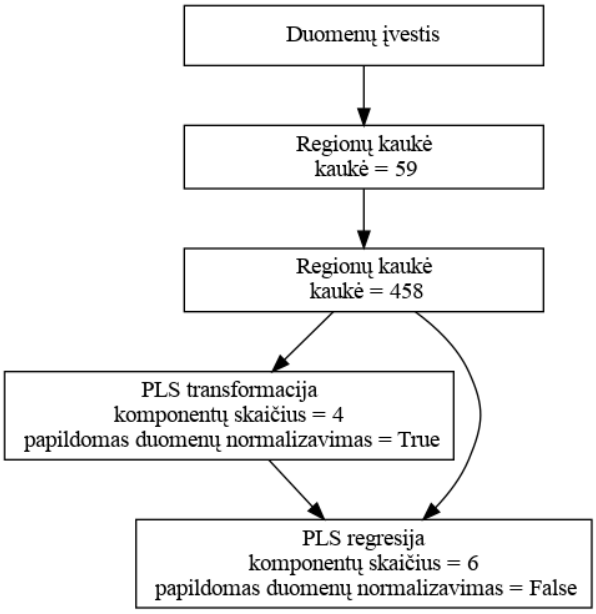
(a) SM bandymas



(b) SMB bandymas



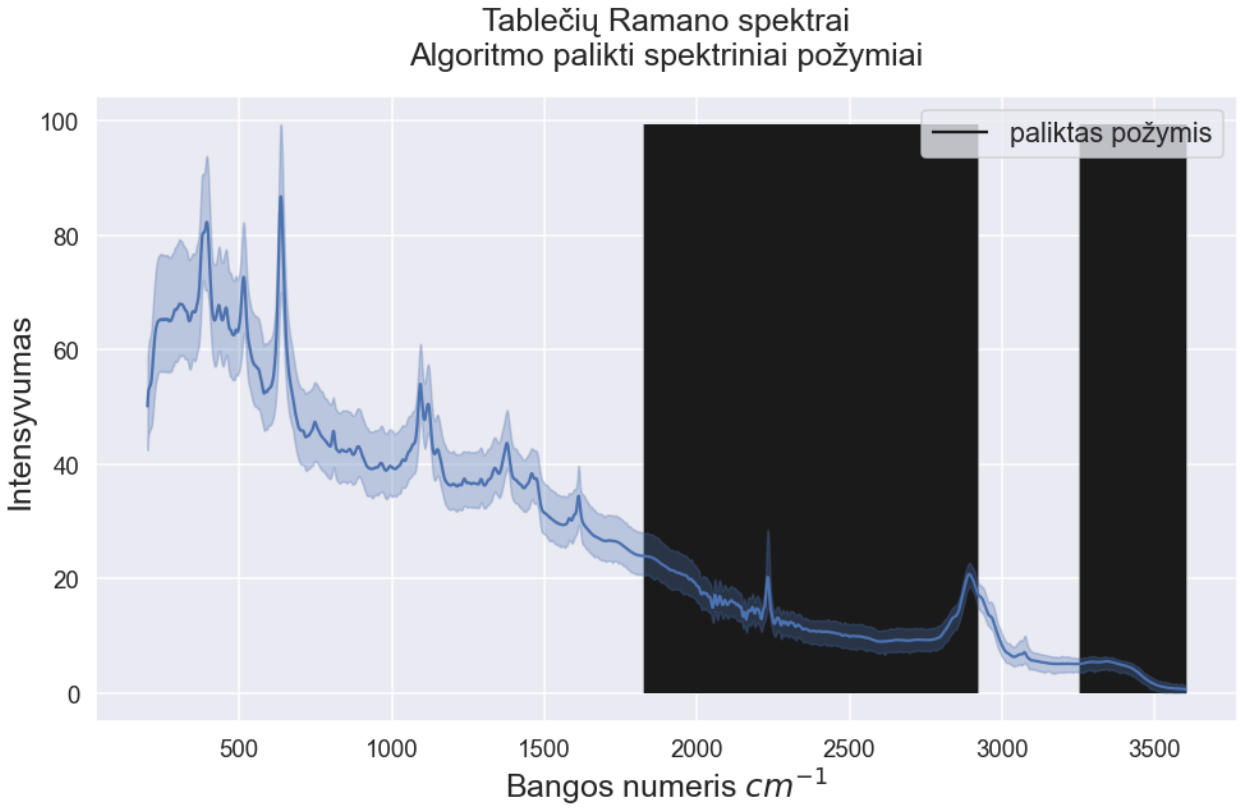
(c) SMP bandymas



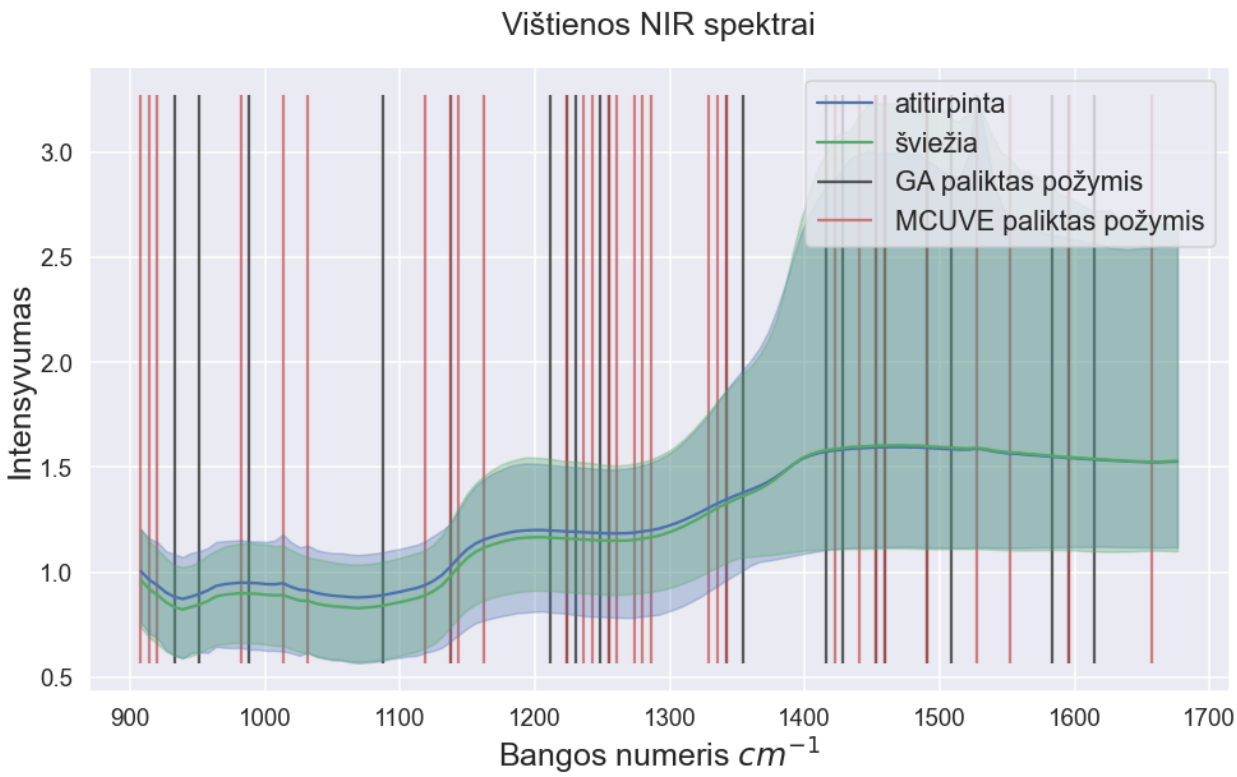
(d) SMG bandymas

32 pav. Tablečių veikliosios medžiagos kiekio nustatymo uždaviniui rastos procesų grandinės

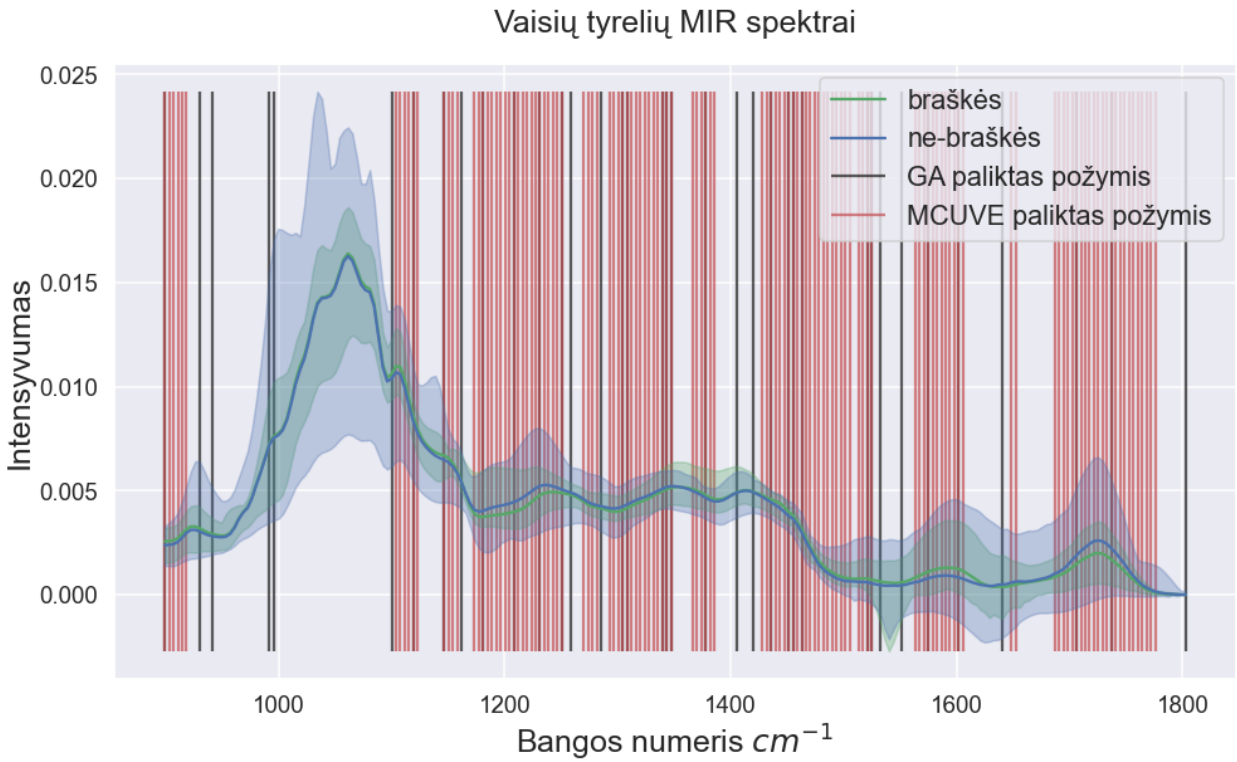
D. Spektrometrijos duomenų analizės požymių atrinkimo bandymai



33 pav. Regionų kaukės atrinkti požymiai tablečių duomenų aibei



34 pav. MCUVE ir GA atrinkti požymiai vištienos duomenų aibei



35 pav. MCUVE ir GA atrinkti požymiai tyrelės duomenų aibei