**RĪGAS TEHNISKĀ UNIVERSITĀTE**

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

Mākslīgā intelekta un sistēmu inženierijas katedra

**Roberts Čīčis**

**Akadēmiskās** bakalaura studiju programmas „Datorsistēmas”

students, stud. apl. nr. 131RDB057

**CART bāzētu modeļu sarežģītības samazināšanas metožu novērtējums sintētiskiem un dabiskiem datiem**

**Bakalaura darbs**

Zinātniskā vadītāja

Dr.sc.ing., **V. Šakele**

Rīga 2021

DARBA IZPILDES UN NOVĒRTĒJUMA LAPA

Bakalaura darbs izstrādāts ***Mākslīgā intelekta un sistēmu inženierijas katedrā.***

Ar parakstu apliecinu, ka visi izmantotie materiāli ir norādīti literatūras sarakstā un iesniegtais darbs ir oriģināls.

Darba autors:

stud. **Roberts Čīčis** ...…………………………........…………………………......

(paraksts, datums)

Bakalaura darbs ieteikts aizstāvēšanai:

Zinātniskais vadītājs:

Dr.sc.ing., Vita Šakele…….................……………………………...

(paraksts, datums)

Bakalaura darbs pielaists aizstāvēšanai:

Bakalaura akadēmiskās studiju programmas “Datorsistēmas” direktors:

Dr.sc.ing., asoc. prof. **E.Lavendelis**...............……………………………...

(paraksts, datums)

Bakalaura darbs aizstāvēts Lietišķo datorsistēmu institūta Gala pārbaudījumu komisijas …...gada…….....…sēdē un novērtēts ar atzīmi ( )….....………..

(gads) (datums, mēnesis)

Lietišķo datorsistēmu institūta Gala pārbaudījumu komisijas sekretāre …….....................……

(uzvārds, paraksts)

ANOTĀCIJA

CART, lēmumu koki, izskaidrojamība, sarežģītība

Bakalaura darba tips: 1. tips: Moderno risinājumu izpēte

Mākslīgā intelekta sistēmu risinājumu ieviešana ir saskārusies ar problēmu, ka noteiktās nozarēs pats svarīgākais kritērijs nav veiktspēja, bet modeļa izskaidrojamība. CART (Classification and regression trees) modeļiem piemīt veiktspējas un izskaidrojamības kompromiss. Darbs analizē, vai eksistē metodes, ar kurām CART modeļi ir spējīgi saglabāt izskaidrojamību, nezaudējot veiktspēju, vai saglabāt veiktspēju, iegūstot paaugstinātu izskaidrojamību.

Darba mērķis ir eksperimentāli novērtēt mainīgo svarīguma metožu efektivitāti CART modeļiem un definēt metožu piemērotību dažādām situācijām.

Darba pamattekstā ir 60 lappuses, 28 attēli, 3 tabulas, 24 nosaukumu informācijas avoti un 2 pielikumi.

Abstract

CART, decision trees, explainability, complexity

[Short description of the contents of the bachelor thesis......]

The thesis contains 60 pages, 28 figures, 3 tables, 24 information sources and 2 appendixes.

anotācija vēl kādā svešvalodā

[ATSLĒGVĀRDI]

[īss darba satura apraksts........]

Darba pamattekstā ir 60 lappuses, 28 attēli, 3 tabulas, 24 nosaukumu informācijas avoti un 2 pielikumi.

IEVADS

Mākslīgā intelekta (MI) sistēmu ātrais progress un ievērojamā veiktspēja ir radījusi spiedienu daudzām nozarēm ieviest mašīnmācīšanos risinājumus. Risinājumu ieviešana ir saskārusies ar problēmu, ka dažās nozarēs pats svarīgākais kritērijs nav veiktspēja, bet modeļa izskaidrojamība un spēja dot zināšanas par modelējamo procesu. Problēmas apzināšanās un juridiski likumi (General data protection regulation) ir radījuši izskaidrojamības nozares “atdzimšanu” un spiedienu ieviest izskaidrojamas MI sistēmas, nevis “melnās kastes” sistēmas. CART (Classification and regression trees) modeļi ir vieni no visplašāk izplatītajām MI sistēmām. Starp CART modeļiem tikai sekliem modeļiem (koka dziļums mazāks par 4) piemīt izskaidrojamība, sekli lēmumu koki ievērojami upurē veiktspēju. Turpretī sarežģītākiem CART modeļiem (lēmumu koku izlase, gradienta stiprinoša lēmumu koku izlase) piemīt augsta veiktspēja, bet zema izskaidrojamība. Darbs analizē, vai pastāv metodes, kuras spēj vienkāršus un izskaidrojamus CART modeļus padarīt veiktspējīgākus vai sarežģītākus modeļus (lēmumu koku izlase, gradienta stiprinoša lēmumu koku izlase) padarīt izskaidrojamākus. Darbā tiek definētas 4 stratēģijas modeļu sarežģītības samazināšanai, bet eksperimentāli novērtēta tikai viena stratēģija – mainīga svarīguma metožu izmantošana.

Darba mērķis ir eksperimentāli novērtēt mainīgo svarīguma metožu efektivitāti CART modeļiem un definēt metožu piemērotību dažādām situācijām.

Lai sasniegtu darba mērķi, ir izvirzīti darba uzdevumi:

1. apkopot informāciju par CART modeļu sarežģītības samazināšanas stratēģijām;
2. analizēt, kādas metodes ir sarežģītības samazināšanas stratēģijai – mainīga svarīguma metodes;
3. eksperimentāli novērtēt mainīgo svarīguma metožu efektivitāti CART modeļiem;
4. eksperimentāli noteikt mainīgo svarīguma metožu ātrdarbību.

Darba 1. nodaļā ir apskatīti CART modeļu darbības pamatprincipi un sarežģītības samazināšanas stratēģijas. Raksturoti izskaidrojamības un sarežģītības jēdzieni, izklāstīti iemesli izskaidrojamības metožu nesenajai popularitātei. Tiek argumentēts par izskaidrojamu sistēmu nepieciešamību un parādīti vēsturiski un aktuāli piemēri par to.

SATURA RĀDĪTĀJS

[1 CART modeļu sarežgītība un izskaidrojamība 3](#__RefHeading___Toc50247_2182101473)

[1.1 CART modeļi un to sarežģītības samazināšana 3](#__RefHeading___Toc1884_3387015501)

[1.1 Izskaidrojamības jēdziens 6](#__RefHeading___Toc50249_2182101473)

[1.2 Sarežģītības jēdziens 8](#__RefHeading___Toc50251_2182101473)

[1.2 Anskomba kvartets 9](#__RefHeading___Toc1953_3387015501)

[2 Mainīgo svarīguma metodes 13](#__RefHeading___Toc50255_2182101473)

[2.1 Filtra metodes 13](#__RefHeading___Toc10763_4027635924)

[2.1.1 Pīrsona korelācijas koeficients 14](#__RefHeading___Toc10765_4027635924)

[2.1.1 Spīrmena korelācijas koeficients 16](#__RefHeading___Toc12892_4027635924)

[2.1.2 Kendala rangu korelācjas koeficients 17](#__RefHeading___Toc5993_1926664601)

[2.1.3 Paredzošā spēka mērs 18](#__RefHeading___Toc5995_1926664601)

[2.1.4 Abpusējas informācijas mērs 19](#__RefHeading___Toc6003_1926664601)

[2.1.5 F-tests 20](#__RefHeading___Toc8327_1926664601)

[2.2 Ietvertās metodes 21](#__RefHeading___Toc8329_1926664601)

[1.2.1 Kritērija guvuma mērs 21](#__RefHeading___Toc8331_1926664601)

[2.3 Aptinuma metodes 22](#__RefHeading___Toc8333_1926664601)

[1.2.2 Permutācijas mainīgo svarīgums 22](#__RefHeading___Toc8335_1926664601)

[1.2.3 TreeSHAP mainīgo svarīgums 23](#__RefHeading___Toc1907_3704354823)

[2.4 Hibrīdmetodes 26](#__RefHeading___Toc1911_3704354823)

[1.2.4 Rekursīva mainīgo izslēgšana 26](#__RefHeading___Toc1913_3704354823)

[3 Eksperimenti mainīgo svarīguma Metožu EfektIvitātes noteikšanai 27](#__RefHeading___Toc50261_2182101473)

[3.1 Eksperimentā izmantotie CART algortimi 27](#__RefHeading___Toc2287_3409115181)

[1.2.5 Lēmumu koki 27](#__RefHeading___Toc2289_3409115181)

[1.2.6 Lēmumu koku izlase 29](#__RefHeading___Toc1955_3387015501)

[1.2.7 Gradienta stiprinoša lēmumu koku izlase 30](#__RefHeading___Toc8292_3387015501)

[3.2 Izmantotās datu kopas un to pirmsapstrāde 31](#__RefHeading___Toc50263_2182101473)

[3.3 Eksperiments ar dabiskiem datiem 33](#__RefHeading___Toc9610_667262781)

[3.4 Eksperiments ar sintētiskiem datiem 39](#__RefHeading___Toc1807_2551015736)

[3.5 Mainīgo svarīguma metožu novērtējums 45](#__RefHeading___Toc9612_667262781)

[3.6 Dabisko datu mainīgo koalīciju analīze 46](#__RefHeading___Toc15441_3955443528)

[1.2.8 Optimālu koalīciju inferences un mācību ātruma guvuma analīze 46](#__RefHeading___Toc8294_3387015501)

[1.2.9 Vienkāršu modeļu paritātes analīze 46](#__RefHeading___Toc8296_3387015501)

# CART modeļu sarežgītība un izskaidrojamība

CART (Classification and regression trees) bāzēti modeļi ir kļuvuši par vieniem no visplašāk izmantotajiem modeļiem mašīnmācīšanās nozarē. Izskaidrojamības nozares “atdzimšana” ir atjaunojusi interesi par CART izmantošanu modelēšanas kontekstos, kuros maksimāli svarīga ir izskaidrojamība, nevis veiktspēja. CART modeļos ir veiktspējas un izskaidrojamības kompromiss, kur mazākas sarežgītības modeļiem (lēmumu kokiem) ir augstāka izskaidrojamība, bet zemāka veiktspēja nekā augstākas sarežģītības modeļiem (lēmumu koku izlasēm). Nodaļā tiek apskatīti veidi, kā samazināt CART modeļu sarežģītību, lai mazinātu veiktspējas un izskaidrojamības kompromisa ietekmi uz mašīnmācīšanās procesu, tāpat ir veikts apkopojums par izskaidrojamības aktualitāti un lomu modelēšanas procesā.

## CART modeļi un to sarežģītības samazināšana

CART (Breiman et al., 1983) ir mašīnmācīšanās algoritms, kas ļauj modelēt regresijas un klasifikācijas uzdevumus. CART modeļi rekursīvi sadala datus, izvēloties noteiktas mainīgā robežvērtības katrā šķēlumā. CART modeļu šķēlumi (split) tiek veidoti alkatīgā veidā, apsverot tikai nākamo šķēlumu. Tas veido globāli neoptimālus lēmumu kokus (Sreerama et al., 1995). Neoptimāli lēmumu koki ir neprecīzāki un grūtāk izskaidrojami par optimāliem kokiem (Bertsimas et al., 2017). Optimālu šķēlumu atrašanai modeļa līmenī nav efektīva algoritma, problēmu raksturo vismaz nedeterministiska polinoma (NP-hard) izskaitļošanas laiks (Hyafil et al,. 1976). Pastāv nealkatīgi koka izveides algoritmi, kas veido globāli optimālu koku modeļus, bet to izskaitļošanas laiks ierobežo, kuras datu kopas ir iespējams modelēt – datu kopas, kuru izmērs ir zem 10000 (Hu et al., 2019; Bertsimas, et al., 2017; Lin, et al., 2020).

CART alkatīgā daba rada pārāk sarežģītus un nestabilus modeļus; minmālas izmaiņas mācību datos var radīt pilnīgi citu koku (Molnar, 2019). Lai CART spētu modelētu stabilas un vispārināmas attiecības, nepieciešams samazināt CART modeļu sarežģītību. CART modeļu sarežģītības samazināšanai pastāv 4 stratēģijas – nealkatīgs koka izveides algoritms, hiperparametru optimizācija, datu pētnieciskā analīze un globālas izskaidrojamības metodes.

**Nealkatīgs koka izveides algoritms** – pateicoties algoritmu izpētei un palielinātajai datoru skaitļošanas jaudai, ar mūsdienu risinājumiem ir iespējams izveidot optimālas koku struktūras saprātīgā izskaitļošanas laikā datu kopām, kuru izmērs ir mērāms tūkstošos (zem 10000 instancēm) (Hu et al., 2019; Bertsimas et al., 2017; Lin et al., 2020).

**Hiperparametru optimizācija** – hiperparametru optimizācijas (hyperparameter optimization) galvenā ideja ir optimizēt modeļa parametrus, piemēram, koka dziļums, apmācības ātrums, koku skaits u.c. Modeļa parametri spēj būt pārāk pielāgoti (overfitting) vai nepietiekami pielāgoti (underfitting). Hiperparametru optimizācija atrod modeļa parametrus, kas atrod kompromisu starp pārāk pielāgotu un nepietiekami pielāgotu modeli. Hiperparametru optimizāciju var veikt automātiski, izmantojot tādus ietvarus kā Auto-Weka (Thornton et al., 2013), Optuna (Takuya. 2019), BOHB (Falkner et al., 2018), Hyperband (Lisha et al., 2017) vai mašīnmācīšanās praktiķim manuāli mainot modeļa parametrus un salīdzinot parametra izmaiņas ietekmi uz modeļa optimizācijas kritērija izmaiņu.  
 **Datu pētnieciskā analīze** – datu pētnieciskā analīze (exploratory data analysis, tālāk EDA) (Tukey, 1977) ir viens no veidiem, kā cīnīties ar pārāk lielu sarežģītību un liekiem mainīgajiem modelī. Vizualizējot datus, grafikos ir iespējams noteikt, vai mainīgajam nav tādas problēmas kā zema dispersija (variance), daudz trūkstošo vērtību, augsta mainīgo savstarpējā korelācija. Vizualizācija palīdz samazināt modeļa sarežģītību, atmetot liekus mainīgos vai mainīgos, kuru iekļaušana modelī ir vairāk apgrūtinājums nekā ieguvums modelēšanas procesam.

EDA ir arī trūkumi. EDA ir manuāls process, kas prasa subjektīvu eksperta izvērtējumu. EDA piemīt tādas problēmas kā Simpsona paradokss (Simpson’s paradox), kad attiecības, kas piemērotas konkrētām datu apakšgrupām, pazūd, kad šīs apakšgrupas tiek apvienotas (Simpson, 1951).

Cilvēki var salīdzinoši vienkārši izskaidrot modeli, kuram ir tikai 1 mainīgais, piemēram, attēlojot attiecību starp mainīgo un mērķi izkliedes grafikā, bet, ja modelī ir vairāk nekā viens mainīgais, tad modeļa darbības izskaidrošana ar vizualizācijas palīdzību kļūst problemātiska. Pirmā problēma sarežģītu sistēmu interpretācijā ir tīri cilvēciska. Psihologs Džordžs Millers (G. A. Miller, 1956) aprakstīja cilvēku prāta spēju ierobežojumus, aprakstot slaveno Millera likumu, ka cilvēki īstermiņa atmiņā spēj rīkoties ar 5-9 objektiem (7+/-2). Otrā problēma ir emerģences īpašība (aprakstīta nodaļā 1.2), ja eksistē modelis y = f(x1, x2), tad ir salīdzinoši viegli izskaidrot, kāda ir attiecība starp (y, x1) un (y, x2), piemēram, izmantojot izkliedes grafikus, bet ir praktiski neiespējami paredzēt, kā sistēma darbojas kā kopums y = f(x1, x2), t.i. mainīgo kopai {x1, x2} savstarpējas iedarbības rezultātā, modelējot y, piemīt lielāka sarežģītība nekā mainīgajiem, atsevišķi modelējot y (emerģences īpašība, apskatīta nodaļā 1.2). Tas nozīmē, ka EDA metode ir piemērota datu kopām ar mazu mainīgo skaitu, jo lielās datu kopās tiks pārkāpts Millera likums un pārslogotas cilvēka spējas izskaidrot datus ar EDA palīdzību.

**Globālas izskaidrojamības metodes** – globālas izskaidrojamības metodes (global interpretability methods) jeb mainīgo izvēles metodes (feature selection) ir veids kā izvēlēties “lietderīgu” mainīgo apakškopas. Jāpiemin, ka vārds “globāls” šajā kontekstā nozīmē nevis “visaptverošs”, bet globāls modeļa vai datu līmenī. Termins “globālas izskaidrojamības metodes” ir balstīts izskaidrojamības taksonomiskā sadalījuma 5. kritērijā, kas apskatīts nodaļā **1.2**. Globālas izskaidrojamības metodes iedala 4 kategorijās (Guyon, 2003) – filtra (filter), aptinuma (wrapper), ietvertās (embedded) un hibrīdmetodēs (Guyon et al., 2004).

Filtra metodes (Guyon, 2003) izvēlas labāko mainīgo apakškopu, analizējot ieejas datu īpašības, piemēram, mainīgā vērtību dispersiju, mainīgo savstarpējo korelāciju, mainīgā neaizpildīto vērtību skaitu u.c. Filtra metožu galvenā priekšrocība ir ātrs skaitļošanas laiks, salīdzinājumā ar aptinuma metodēm.

Aptinuma metodes popularizēja Kohavi (Kohavi, 1997). Aptinuma metodes galvenā ideja ir uzskatīt modeli par “melno kasti”. Aptinuma metodes izmanto mašīnmācīšanās modeļa mērķa mēru, lai aproksimētu mainīgo apakškopas lietderību. Teorētiski ir iespējams aprēķināt visu mainīgo apakškopu “lietderību”, bet praksē šādas metodes izskaitļošana aizņem pārāk daudz laika (Amaldi, 1998). Aptinuma metodes ir “modeļu neatkarīgas” (model agnostic) – tās spēj jebkuram modelim noteikt “lietderīgu” mainīgo apakškopu, ja ir pietiekams skaitļošanas resursu daudzums.

Ietvertās metodes izmanto modeļa veidošanas mehānismu, apmācības laikā aprēķinot mainīgā “lietderību”. Ietvertās metodes piemērs ir lēmumu koku izlasē mainīgo svarīguma aprēķins – katram mainīgajam tiek saskaitīts, cik liels ir summārais modeļa kritērija guvums no visiem šķēlumiem, kur mainīgais ir izmantots (Breiman, 2004)

Hibrīdmetodes (Guyon et al., 2004) apvieno vairākas no iepriekš minētajām metodēm vienā metodē, nodrošinot labākus rezultātus un augstāku modeļa precizitāti nekā atsevišķām metodēm. Īpaša uzmanība jāpievērš hibrīdmetodei – rekursīva mainīgo izslēgšana (recursive feature elimination), kas apvieno filtra un aptinuma metodes, lai veiktu efektīvu “lietderīgāko” mainīgo atrašanu.

Globālas izskaidrojamības metodes ir izvēlēts kā darba galvenais pētījuma priekšmets, jo globālas izskaidrojamības metodēm ir visstiprākā saikne ar modeļa struktūru, izskaidrojamību un vispārināmību.

Nealkatīgi koka izveides algoritmi netika izvēlēti kā galvenais priekšmets, jo tā ir salīdzinoši jauna nozare (pirmā praktiskā implementācija veikta tikai 2017. gadā). Nealkatīgu koka izveides algoritmu jaunas metodes sintēzei vai esošo risinājumu novērtēšanai pietrūkst zinātniskās literatūras.

Hiperparametru optimizācija netika izvēlēta kā galvenais priekšmets, jo tā ir dziļi izpētīta, bieži izmantota metode mašīnmācīšanās nozarē, hiperparametru optimizācija tikai netiešā veidā kontrolē modeļa struktūru un sarežģītību.

Datu pētnieciskā analīze netika izvēlēta kā galvenais priekšmets, jo tā ir pakļauta cilvēku subjektīvai analīzei un nav labi mērogojama datu kopām ar daudziem mainīgajiem. Nākamajā nodaļā ir apskatīts izskaidrojamības jēdziens, motivācija izskaidrojamības nepieciešamībai mākslīgā intelekta (tālāk MI) sistēmās, izskaidrojamības taksonomisks sadalījums, izskaidrojamības vēsture.

## Izskaidrojamības jēdziens

MI sistēmas ir kļuvušas plaši izplatītas tādās nozarēs kā autonoms transports, medicīna, apdrošināšana, finanšu pakalpojumi un tiesu sistēma (Doshi‐Velez & Kim, 2017). MI sistēmu praktiska pielietošana ir radījusi nepieciešamību pēc modeļiem, kas optimizē ne tikai MI sistēmas veiktspēju, bet arī citus kritērijus, piemēram, drošību, nediskrimināciju, cilvēces eksistenciālu draudu novēršanu (Bostrom, 2014). Avots (Confalonieri, et al., 2021) apgalvo, ka kopš 2020. gada izskaidrojamība (explainability) ir identificēta kā viens no galvenajiem faktoriem MI sistēmu ieviešanai. Kā praktisku piemēru avots min GDPR (General Data Protection regulation) definēto tiesību iegūt “jēgpilnu informāciju par loģiku, kas tiek izmantota MI lēmumiem”. Šī definīcija bieži tiek interpretēta kā tiesības uz “izskaidrojumu”, kad lēmumu pieņem automātiska sistēma (Parliament and Council of the European Union, 2016/679).

Izskaidrojamībai nav skaidras un vienotas definīcijas (Molnar, 2019). Taču, ja nepastāv, skaidras un vienotas definīcijas izskaidrojamībai, iedvesmojoties no sociālajam zinātnēm, Millers (Miller, 2019) piedāvā izskaidrojamības definīciju: “Izskaidrojamība ir pakāpe, ar kādu cilvēks var saprast cēloņus MI sistēmas pieņemtam lēmumam”. Avots (Been, et al., 2016) piedāvā alternatīvu definīciju: “Izskaidrojamība ir cilvēka spēja patstāvīgi paredzēt MI sistēmas prognozes”. Savukārt Doši-Velezs un Kima (Doshi‐Velez & Kim, 2017) piedāvā visus kritērijus, kas nav saistīti ar modeļa veiktspēju, apvienot visaptverošā terminā – izskaidrojamība.

Taksonomiski Kristofs Molnars (Molnar, 2019, nodaļa 2.2) piedāvā izskaidrojamības metodes iedalīt pēc šādiem kritērijiem:

1. ietvertā (intrinsic) vai modeli analizējoša (post-hoc) izskaidrojamības metode.

Kritērijs raksturo, vai izskaidrojamība tiek iegūta samazinot modeļa sarežģītību vai izmantojot metodes pēc modeļa apmācības;

1. izskaidrojamības metodes rezultāts;
   1. mainīgā vispārīgs mērs – izskaidrojamības metodes, kas katram mainīgajam veido apkopojošu statistiku;
   2. modeļa iekšējais stāvoklis – izskaidrojamu modeļu iekšējais stāvoklis, piemēram, lineārās regresijas gadījumā mainīgo koeficienti, lēmumu koku vizuāla reprezentācija;
   3. piemēros bāzētas metodes – visas metodes, kas atgriež konkrētus datu piemērus (jau esošus vai uzģenerētus);
   4. surogātveida izskaidrojams modelis – vienkāršāks, izskaidrojams modelis, kas aproksimē oriģinālā modeļa darbību;
2. modelim specifiska (model specific) vai modeļa neatkarīga (model agnostic) metode.  
   Kritērijs raksturo, vai metode ir pielietojama visiem modeļiem (modeļa neatkarīga) vai metode ir pielietojama tikai konkrētu modeļu izskaidrošanai (model specific);
3. vietēja vai globāla izskaidrojamība.

Kritērijs raksturo, vai metode izskaidro tikai vienu datu punktu vai visa modeļa (datu kopas) darbību.

Vēsturiski izskaidrojamības popularitāte ir epizodiska. Millers ar kolēģiem (Mueller, et al., 2019), balstoties uz zinātnisko rakstu kopu, apgalvo, ka eksistē 4 epizodes – pirmā paaudze (1977-1983), otrā paaudze (1984-1995), trešā paaudze (2011-līdz šim) un izskaidrojamības “ziema” (1996-2010). Pirmās un otrās paaudzes laikā pētījumi tika veikti aktīvi, diemžēl, nozarē izveidojās liela skepse, jo ekspertu sistēmas nespēja attaisnot augstās cerības uz tām (Buchanan & Shortliffe, 1984; Wick & Thompson, 1992). Pēc pirmās un otrās paaudzes sekoja izskaidrojamības “ziema”, kad praktiski netika pētīta izskaidrojamība, sekojot pirmās un otrās paaudzes neveiksmēm.

Trešā paaudze ir raksturojama kā izskaidrojamības atdzimšana. Iepriekš jau tika minēts, ka, pateicoties MI sistēmu veiksmēm dažādās nozares, ir atdzimusi interese par izskaidrojamību, radot pētījumu pieaugumu izskaidrojamības nozarē.

Nākamajā nodaļā ir apskatīts sarežģītības jēdziens, lai skaidrotu, kādas īpašības piemīt optimālas sarežģītības modeļiem un izskaidrojumiem.

## Sarežģītības jēdziens

Modeļu izvēle bieži tiek izskaidrota kā kompromisa atrašana starp modeļa spēju pielāgoties datiem un modeļa sarežģītību. Modeļa apmācība nodrošina gan spēju reprezentēt esošos datus, gan spēju precīzi vispārināt nākotnes datus (Guthke, 2017). Pastāv skaidras definīcijas, kas raksturo modeļa apmācības kvalitāti, diemžēl modeļa sarežģītībai neeksistē skaidras definīcijas. Tāpēc nepieciešama dziļāka izpēte modeļa sarežģītības terminam.

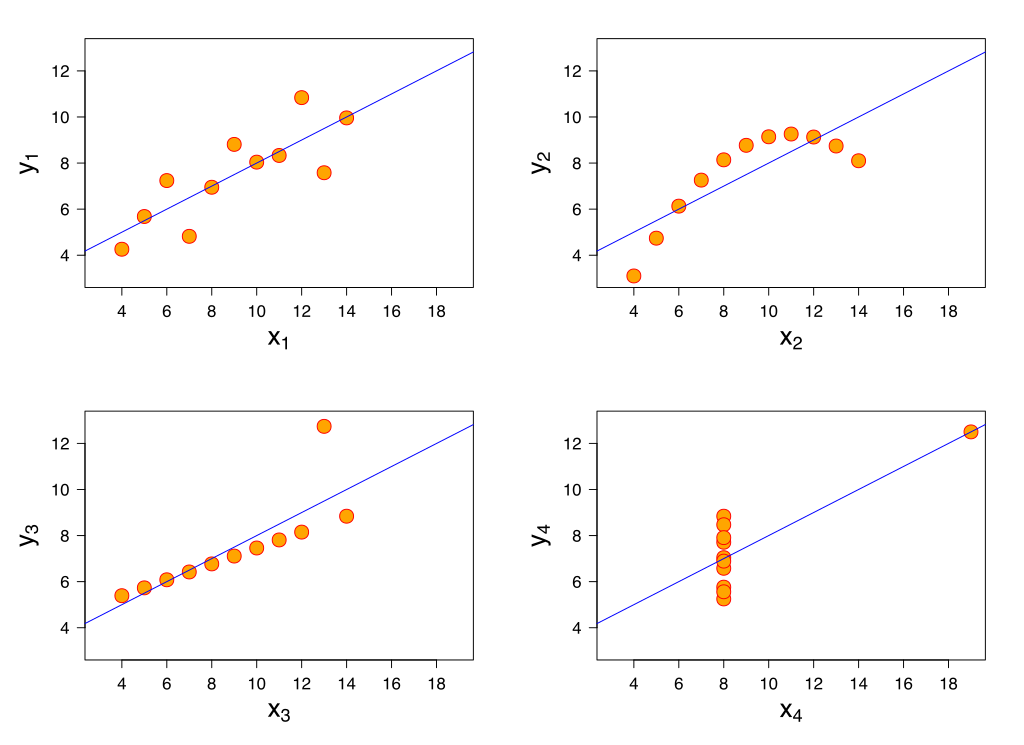
Sarežģītība ir netverama (elusive) īpašība (Van Emden, 1971). Lai gan intuitīvi ir viegli izprast, vai kaut kas ir sarežģīts vai nē, nepastāv skaidras definīcijas vai iespējas kvantitatīvi raksturot sarežģītību. Sarežģītība ir saistīta ar emerģenci (angl. emergence). Emerģence (Prokopenko, 2009) ir īpašība, kad vairāku identisku objektu kopai piemīt lielāka sarežģītība nekā objektiem individuāli. Balstoties uz emerģences īpašību, var pieņemt, ka mainīgo pievienošana modelim palielinās modeļa sarežģītību.

Modeļi ar vairākiem parametriem un mainīgajiem ne vienmēr ir labāki par modeļiem ar mazāku parametru un mainīgo daudzumu (Perrin et al., 2001). Optimālais modelis nav nedz pārāk sarežģīts (Myung, 2000), nedz arī pārāk vienkāršs (Mendoza et al., 2015) konkrētajam modelēšanas uzdevumam.

Nākamajā nodaļā ir apskatīti praktiski piemēri MI sistēmu izstrādes vēsturē, kad MI praktiķi ir koncentrējušies uz MI sistēmu veiktspējas mēriem, nevis izskaidrojamību un sistēmas vērtības guvumu. Ir paskaidrots, kādi principi jāievēro, lai izveidotu vispārināmas, netrauslas MI sistēmas.

## Anskomba kvartets

Mūsdienu mašīnmācīšanās aizsācēji bija statistiķi, mēģinot ar matemātikas palīdzību modelēt procesus un iegūt zināšanas no modelēšanas procesa. Statistikas nozarei attīstoties, izveidojās mīti (Anscombe, 1973), ka “skaitliskie aprēķini ir precīzi, bet grafiki ir aptuveni”, “datus var pareizi izanalizēt tikai vienā veidā”, “veikt sarežģītus aprēķinus ir pareizi, bet vizualizēt datus ir šmaukšanās”. Statistiķi pārāk koncentrējās uz mēriem un ne uz to, kādu vērtību un zināšanas ir iespējams izgūt no modelēšanas procesa. Lai apgāztu šos mītus, Anskombs izveidoja Anskomba kvartetu. Anskomba kvartets ir 4 datu kopu apkopojums, kuriem piemīt identiski aprakstošie mēri (descriptive measures) – y vidējā vērtība, x vidējā vērtība, regresijas koeficients, regresijas līnijas vienādojums, kvadrātu summa, standartkļūda, determinācijas koeficients, taču tajā pat laikā ļoti atšķirīgi vizuālie sadalījumi. Anskomba kvartets ir attēlots att. 1.1

1.1 att Anskomba kvartets [aizgūts no Wikipedia]

Anskomba kvartets parāda, cik svarīgi ir nepaļauties uz modeļa mēriem (aprakstoša statistika), lai novērtētu risinājumus. Līdzīga pieeja ir vērojama mašīnmācīšanās praksē – modeļa optimizācijas funkcijas zaudējums un modeļu novērtēšanas mēri tiek izmantoti ar reduktīvu pieeju. Tas var novest pie neparedzētiem izaicinājumiem un pat neveiksmēm, mēģinot ieviest MI modeļus produkcijas vidē (D'Amour, A. et al., 2020).

Gadījumus, kad starpnieka mērs (proxy measure) nespēj precīzi attēlot produkcijas vidi vai īsto darbības efektivitāti, raksturo Gudharta likums (Goodhart’s law). Gudharta orģinālā definīcija ir: “Jebkura statistiska likumsakarība mēdz sabrukt, kad uz to tiek izdarīts spiediens kontroles nolūkos” (Goodhart, 1984). Strathens (Strathern, 1997) piedāvā vispārināmāku definīciju: “Kad mērs kļūst par mērķi, tas pārstāj būt par labu mēru”.

Gudharta likumu taksonomiski var iedalīt 4 kategorijās (Manheim & Garrabrant, 2018):

1. regresijas Gudharta likums.

Izvēloties aproksimējošo mēru, izvēlas ne tikai patieso mērķi, bet arī starpību starp aproksimējošo mēru un mērķi. Piemērs: garums ir saistīts ar basketbolista spējām un faktiski tieši palīdz, bet labākais spēlētājs ir tikai 190 cm garš un nejauša 213 cm gara persona 20 gadu vecumā, visticamāk, nebūs tik laba;

1. ekstremāls Gudharta likums.

Telpas, kurās aproksimējošam mēram ir ekstremāla vērtība, var ļoti atšķirties no parastajām pasaulēm, kurās tiek novērota korelācija starp aproksimējošo mēru un mērķi. Piemērs: cilvēki evolūcijas gaitā attīstīja patiku pret cukuru, jo cukurā ir daudz kaloriju. Cilvēku evolūcijas vidē cukura prioritizēšana bija lietderīga, bet mūsdienu vidē izraisa veselības problēmas;

1. cēlonisks Gudharta likums.

Ja starp aproksimējošu mēru un mērķi pastāv cēloņsakarība, maiņa aproksimējošajā mērā var nemainīt mērķa mainīgo. Piemērs: kāds, kurš vēlas būt garāks, varētu novērot, ka augums ir saistīts ar basketbola prasmēm, un nolemt sākt praktizēt basketbolu, lai kļūtu garāks;

1. pretiniecisks Gudharta likums.

Optimizējot aproksimējošo mēru, pretinieciskiem aģentiem ir iniciatīva mazināt modeļa aproksimējošā mēra paredzēšanas spēku, tādējādi iznīcinot korelāciju ar modeļa mērķi. Piemērs: Britu Indijā valdība vēlējas apkarot kobru populāciju, piedāvājot naudas atlīdzību par katru beigto kobru. Laika gaitā iedzīvotāji sāka pavairot kobras. Valdība atcēla programmu, bet iedzīvotāja izlaida pavairotās kobras savvaļā, tādējādi galarezultātā indīgo čūsku populācijas problēmai pasliktinoties.

Izskaidrojamības metodes var tikt izmantotas, lai izprastu, kādi modeļi no relatīvi līdzīgu modeļu apkopojuma ir vislabāk piemēroti konkrētajam uzdevumam. Praktisks piemērs izskaidrojamības nepieciešamībai ir COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) programmatūra, kura tiek lietota ASV tiesās, lai paredzētu noziegumu recidīvisma gadījumus. COMPAS sastāv no 137 jautājumu aptaujas; to izstrādāja privāta kompānija, tāpēc tā modeļa struktūra tiek turēta noslēpumā, un nav zināms, kā aptaujas jautājumi ļauj paredzēt cilvēku uzvedību. Dotā programmatūra tiek asi kritizēta; ProPublica žurnālistu pētījums (Angwin et al., 2016) apgalvo, ka programmatūras modelim piemīt rasistiski aizspriedumi un tas nav drošs un precīzs veids, kā paredzēt vardarbīgas noziedzības un recidīvisma gadījumus.

Pētījums (Julia & Farid, 2018) atklāja, ka programmatūra COMPAS nav ievērojami precīzāka par “indivīdiem bez vai ar minimālu pieredzi kriminālistikā”. Salīdzinājumam: COMPAS modeļa precizitāte (accuracy) ir 65%, bet indivīdu precizitāte (accuracy) ir 63%. Likumsakarīgi, ka tas raisīja bažas par šāda modeļa lietderību un drošību (Rudin, et al., 2018a), it īpaši programmatūras COMPAS tieksme atzīmēt indivīdus ar garu un vērā ņemamu noziegumu vēsturi kā zema riska grupu, kā arī neskaidra privātuma politika.

(Rudin, 2018b) argumentē, ka “melnās kastes” modeļi nav labs risinājums “augsta riska” lēmumiem, tā vietā vajadzētu izmantot sākotnēji izskaidrojamus modeļus. Šis pats avots skaidro, ka “melnās kastes” neizskaidrojamība tiek izmantota peļņas gūšanai no intelektuālā īpašuma. Programmatūras COMPAS alternatīva ir CORELS (Certifiably Optimal Rule Lists) (Angelino, et al., 2017) modelis, kas ir tikpat akurāts kā COMPAS, bet izmanto vien 3 likumu sistēmu un ir pilnībā caurskatāms:

JA indivīda vecums ir no 18 – 20 un dzimums ir vīriešu,

TAD tiek paredzēts arests (2 gadu laikā),

CITĀDI JA vecums ir no 21 – 23 un indivīdam bijuši 2 – 3 kriminālās sodāmības,

TAD tiek paredzēts arests,

CITĀDI JA indivīdam ir vairāk nekā 3 kriminālās sodāmības,

TAD tiek paredzēts arests, CITĀDI arests netiek paredzēts.

Nākamajā nodaļā ir apskatītas konkrētas filtra, aptinuma, ietvertās, hibrīd globālās izskaidrojamības metodes ar mērķi raksturot to darbības principus un izvēlēties reprezentatīvu metožu kopu efektivitātes novērtēšanai.

# Mainīgo svarīguma metodes

Konkrētas globālas izskaidrojamības metodes ir izvēlētas:

1. pētot ietvara Scikitlearn (versija 0.24.0) pieejamās metodes (Pīrsona, Spīrmena, Kendala korelācijas, kritērija guvums, permutācijas mainīgo svarīgums, abpusējas informācijas mērs, F-tests);
2. atrodot metodes literatūras analīzes procesā (TreeSHAP mainīgo svarīgums, paredzošā spēka mērs).

Darbā tika apsvērts analizēt arī metodi Boruta (Kursa, 2010), bet algoritmiskas nesaderības (ar CART algoritmiem) un zemās ātrdarbības dēļ metodes analīze netika veikta. Netiek apskatītas lokālas izskaidrojamības metodes, piemēram, parciālatkarības grafiki (partial dependence plot), individuālas nosacītas vērtības (individual conditional expectation), akumulētie lokālie efekti (accumulated local effects), lokāli izskaidrojamu modeļu neatkarīgi skaidrojumi (local interpretable model-agnostic explanations, LIME), jo šīs metodes dod lokālu, nevis globālu izskaidrojamību (nodaļa 1.2 izskaidrojamības metožu taksonomisks sadalījums, 5 kritērijs). Dziļāku izpratni par metodēm, kuras netika apskatītas, var gūt avotā (Molnar, 2019).

## Filtra metodes

Filtra metodes ir visvienkāršākās un senākas metodes. Filtra metodes tiek pielietotas ne tikai mašīnmācīšanās nozarē, bet arī statistikā un datizracē. Tiks apskatīta tikai apakškopa no visām filtra metodēm. Filtra metodes var iedalīt 2 kategorijās – robežvērtības un globālas izskaidrojamības metodēs. Netiks apskatītas robežvērtības filtra metodes (dispersijas, savstarpējās korelācijas, trūkstošo vērtību filtrs), jo tās nespēj ranžēt mainīgos pēc to svarīguma, un ir jāizvēlas subjektīva robežvērtība.

### Pīrsona korelācijas koeficients

Pīrsona korelācijas koeficients (Boslaugh & Watters, 2008*)* (dažreiz to dēvē par produkta un momenta korelācijas koeficientu no angļ. product-moment correlation coefficient) ir lineārā saistība starp diviem mainīgajiem. Pīrsona korelācijas koeficienta diapazons ir (−1, 1), kur 0 norāda, ka nav saistības starp mainīgajiem, un lielākās absolūtās vērtības norāda uz ciešāku pozitīvu vai negatīvu saistību starp mainīgajiem. Pīrsona korelācijas koeficients var būt maldinošs, ja starp datiem ir nelineāras attiecības.

Matemātiski Pīrsona korelācijas koeficients ir definēts formulās 1.1. un 1.2.

, (1.1)

kur SSx - kvadrātu summa x

SSx - kvadrātu summa y

SSxy- kvadrātu summa x un y

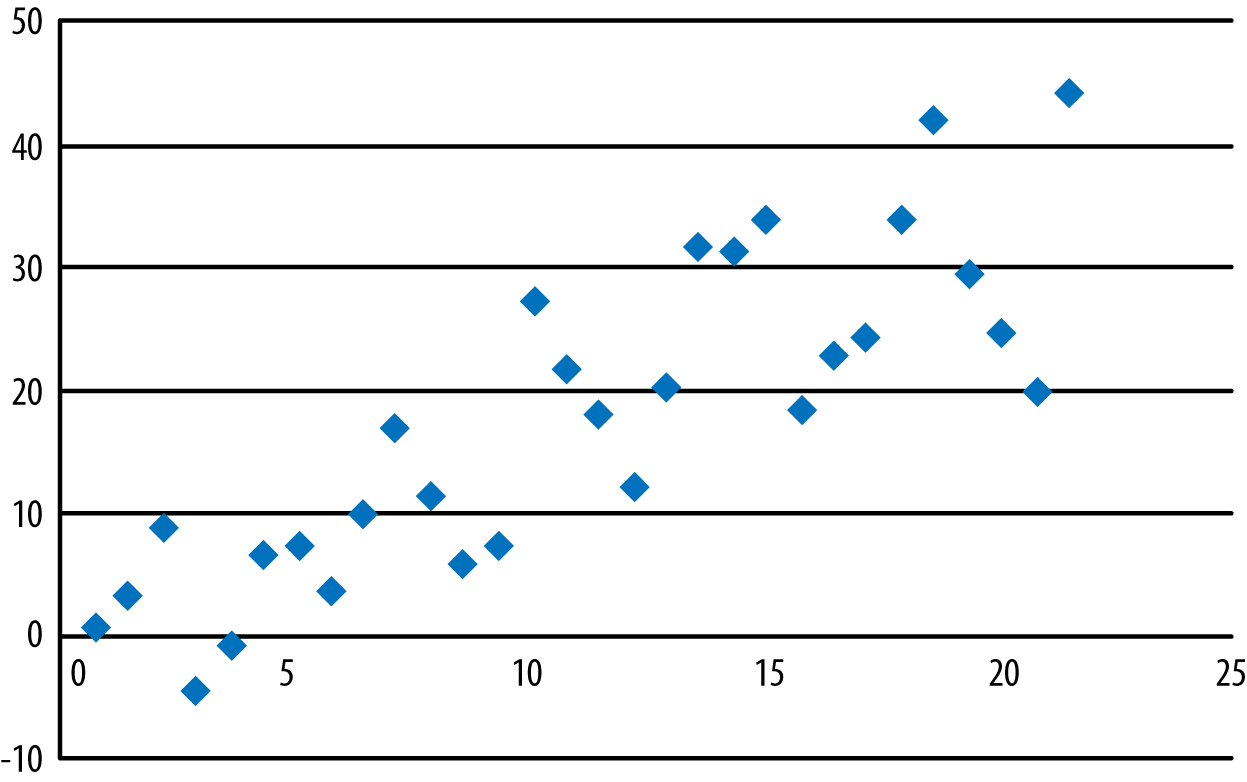
, (1.2)

kur xi – konkrēta x vērtība

– parauga vidējā vērtība

n – parauga izmērs

Praktisku piemēru Pīrsona koeficientam var redzēt attēlā (3.1)

Att. 3.1. Izkliedes grafiks mainīgajiem ar Pīrsona koeficientu 0.84 [pagaidu attēls]

Kopumā Pīrsona koeficients tiek izmantots, lai izvērtētu lineāru monotonu attiecību spēku starp 2 mainīgajiem; koeficients ir simetrisks.

### Spīrmena korelācijas koeficients

Spīrmena rangu korelācijas koeficientam (Corder & Foreman, 2009) piemīt praktiski identiskas īpašības kā Pīrsona korelācijas koeficientam, ar vienu būtisku atšķirību – Spīrmena koeficients spēj precīzi attēlot nelineāras attiecības un ir balstīts uz datu vērtību ranžēšanu, nevis datu vērtībām.

Matemātiski Spīrmena koeficients ir definēts formulā 1.1

, (1.1)

kur

, (1.2)

, (1.3)

n – ranga pāru skaits

Di – starpība starp rangu pāri

g – vienādu grupu skaits datos

ti – vienādu vērtību skaits grupā

Kopumā Spīrmena koeficients tiek izmantots, lai izvērtētu nelineāru vai lineāru monotonu dilstošu vai augošu attiecību spēku starp 2 mainīgajiem; koeficients ir simetrisks. Eksperimentā tiek izmantota absolūtā vērtība starp katru mainīgo un mērķi, lai noteiktu, cik svarīgs ir mainīgais.

### Kendala rangu korelācjas koeficients

Kendala koeficients (*Prokohorov, 2001)* izmēra divu datu kopu līdzību, salīdzinot ranžētu objektu pārus. Koeficients atkarīgs no pāru saskanības un nesaskanības.

Matemātiski Kendala koeficients ir definēts formulā 1.1:

, (1.1)

kur

, (1.2)

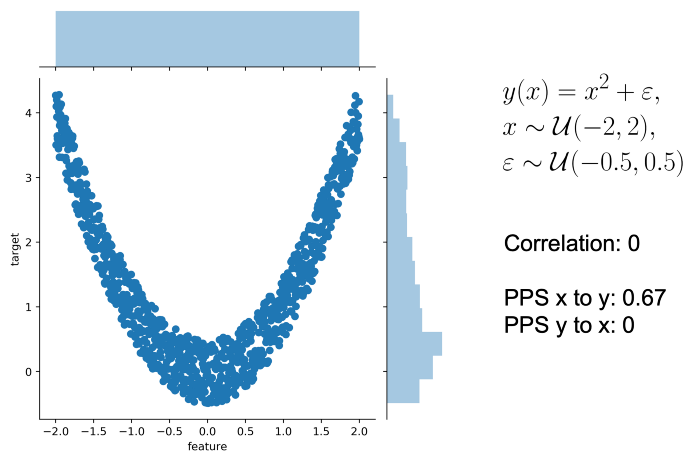
ri  rangs y no (x, y) pāra, kuram x rangs ir vienāds ar i

N – elementu skaits kur j > i un rj > ri

Kopumā Kendala rangu koeficientu izmanto, lai izvērtētu lineāru vai nelineāru monotonu dilstošu vai augošu attiecību spēku starp 2 mainīgajiem. Šis koeficients ir simetrisks.

### Paredzošā spēka mērs

Paredzošā spēka mērs jeb PPS (angl. Predictive power score, tālāk PPS) (Wetschoreck et al., 2020) ir asimetriska, no datu tipa neatkarīga metode, kas spēj noteikt lineāras vai nelineāras attiecības starp diviem mainīgajiem. Rezultāts ir no 0 (nespēj prognozēt) līdz 1 (perfekti prognozē). Galvenā ideja PPS ir veidot lēmumu kokus ar 1 mainīgo, lai paredzētu modelēšanas mērķi, izmantojot 4-kārtīgas krusteniskās validācijas, klasifikācijas uzdevumos nejauši stratificējot (stratification) testa un apmācības datus pēc mērķa mainīgā, lai pārbaudītu rezultātu uzticamību. Izskaidrojums pieejams saitē: <https://www.sr-sv.com/the-predictive-power-score/>

Att 1.1 PPS aprēķina piemērs [pagaidu attēls]

Attēlā x.x redzams, ka x un y korelācija ir 0.67, bet y un x korelācija ir 0. X var labi paredzēt y, jo ir deterministiska kvadrātiska attiecība, bet Y nespēj paredzēt X, jo, teiksim, kad Y ir 4, nav iespējams noteikt, vai X ir aptuveni -2 vai 2 (vienam Y punktam pieder 2 X punkti, kas izraisa nepārliecību par Y vērtību).

Kopumā PPS tiek izmantots, lai izvērtētu nelineāru vai lineāru monotonu dilstošu vai augošu attiecību spēku starp 2 mainīgajiem; koeficients ir asimetrisks. Eksperimentā tiek izmantota absolūtā vērtība starp katru mainīgo un mērķi, lai noteiktu, cik svarīgs ir mainīgais.

### Abpusējas informācijas mērs

Abpusējas informācijas mērs (mutual information) (Cover & Thomas, 1991) mēra “informācijas” daudzumu, kurš pieder vienam mainīgajam par otru mainīgo. To arī var aprakstīt kā nenoteiktības (uncertainty) mazinājumu vienam mainīgajam par otru zināšanu dēļ. Matemātiski abpusējas informācijas mēru aprakstu formula 1.1:

(1.1)

Viena no galvenajām priekšrocībām abpusējas informācijas mēram ir, ka tas spēj modelēt nemonotonas, nelineāras attiecības (atšķirībā no Pīrsona korelācijas koeficienta, kas spēj modelēt tikai monotonas lineāras attiecības).

### F-tests

Klasifikācijas uzdevumos ir iespējams izmantot viena faktora ANOVA (angl. **AN**alysis **O**f **Va**riance) (McDonald, 2014) metodi, lai noteiktu, cik labi mainīgais spēj atdalīt datus starp mērķa klasēm. Galvenā ideja ir aprēķināt vidējo vērtību katrā grupā un salīdzināt katras grupas vidējās vērtības dispersiju ar vidējo dispersiju katrā grupā. Matemātiski process ir definēts formulās 1.1-1.1:

(1.1)

(1.1)

(1.1)

kur,

F – viena faktora ANOVA F vērtības

MST – vidējā kvadrātiskā kļūda starp grupām

MSE – vidējā kvadrātiskā kļūda

Yij – ij datu punkts (novērojums)

Ti – grupas kopējā vērtība

G – ir datu punktu (novērojumu) kopējais skaits

ni – punktu skaits grupā i

n – kopējais datu punktu skaits

k – kopējais grupu skaits

## Ietvertās metodes

Ietvertās metodes izmanto modeļa veidošanas mehānismu, apmācības laikā aprēķinot mainīgā “lietderību” modelim. Lēmumu kokiem pastāv tikai viena ietvertā metode – kritērija guvuma mērs. Kritērija guvuma mēra metodei pastāv daudzas variācijas, piemēram, XGBoost (Chen & Guestrin, 2016) piedāvā saskaitīt, cik reizes mainīgais ir izmantots, nevis tā modeļa kritērija guvumu. Variācijas netiek apskatītas, jo visi lēmumu koki neatbalsta metodes variāciju aprēķinus.

### Kritērija guvuma mērs

Viens no veidiem, kā noteikt mainīgo globālo svarīgumu, ir izmantot koku modeļa katra šķēluma kritērija vērtības guvumu (Trevor Hastie, 2009). Šo guvumu apkopo pa visiem kokiem un sarēķina katram mainīgajam atsevišķi. Breimans (Breiman, 2001) definēja, kā izvērtēt mainīgo svarīgumu lēmumu kokos. Mainīgajam Xj paredzot Y, summējot netīrības mazinājumu p(t) delta i(st, t) katram koka zaram, kur Xj ir izmantots, ņemot vidējo vērtību pa visiem kokiem mežu izlasē (random forest). Matemātiska definīcija formulā 1.1

, (1.1)

kur

p(t) – proporcija Nt / N paraugu skaits, kas sasniedz t

jt – identifikators mainīgajam, kas izmantots, lai sadalītu koku zaru t

## Aptinuma metodes

Aptinuma metodes ir “modeļu neatkarīgas” metodes; tas nozīmē, ka vienu metodi var izmantot visiem MI modeļiem. Aptinuma metodes uzskata modeli par “melno kasti” un veic mainīgo lietderības izvērtējumu, balstoties uz modeļa kritērija guvumu vai zaudējumu, izmainot modeļa datus (piemēram, aizvietojot mainīga ieejas datus ar nejaušiem datiem). Aptinuma metožu trūkums ir zemā ātrdarbība.

### Permutācijas mainīgo svarīgums

Breimans (Breiman, 2004) piedāvāja arī alternatīvu veidu, kā aprēķināt mainīgo svarīgumu koku modeļos. Permutācijas mainīgo svarīgums (Permutation feature importance), saukts arī par Vidējo precizitātes zaudējumu (Mean Decrease Accuracy) vai Vidējo kļūdas palielinājumu (Mean Increase Error), balstās uz modeļa precizitātes zaudējuma (vai kļūdas palielinājuma) mērīšanu, nejauši sajaucot mainīgā vērtības (Louppe, 2014, pp. 123-125).

(1.1)

kur

xj – mainīgais j

π sub j (%LAMBDA) – replika xj vērtībām, kur veikta nejauša sajaukšana

mk1...mkM-1 – indeksi kokiem, kas izveidoti no atkārtotas paraugu ņemšanas (bootstrap), kurā nav (xi, yi)

### TreeSHAP mainīgo svarīgums

Lai izprastu TreeSHAP (Molnar, 2019) algoritma darbību, nepieciešams raksturot:

1. Šaplija vertības;
2. Summāra mainīgo attiecinājuma (attribution) metodi;
3. TreeSHAP darbības principu;
4. TreeSHAP mainīgo svarīgumu aprēķinu.

Šaplija vērtības – metode no koalīciju spēļu teorijas – ļauj “taisnīgi” sadalīt “izmaksu” starp spēlētājiem (mainīgajiem) (Shapley, 1951). Modeļa lēmumus var izskaidrot, pieņemot, ka katrs mainīgais ir “spēlētājs”, kur katra spēlētāja “izmaksa” ir modeļa paredzējums. Matemātiski Šaplija vērtības aprēķinu ir definēts formulā 1.1

(1.1)

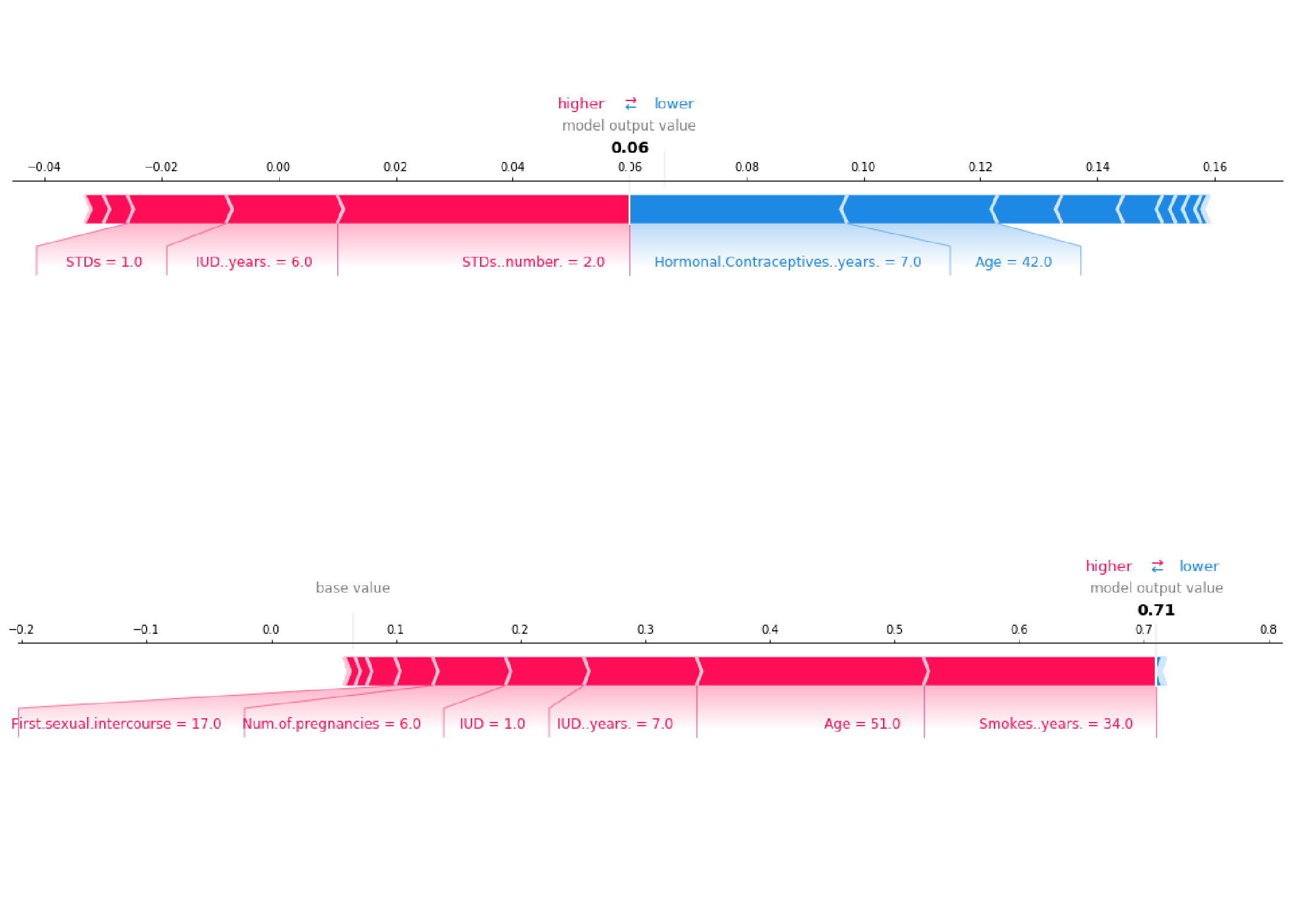
kur

phi i – Šaplija vertībā spēlētājam i

S – spēlētāju koalīcija

fx(s) – koalīcijas vērtība spēlētājiem sadarbojoties

Summāra mainīgo attiecinājuma (attribution) metodes mērķis ir izskaidrot katras instances paredzējumu x, aprēķinot mainīgā “ieguldījumu” (contribution) paredzējumā. Izskaidrošanas metode aprēķina Šaplija vērtības no spēļu teorijas, un katras instances mainīgo vērtības uzvedas kā spēlētāji koalīcijā. Šaplija vērtības izskaidrojums tiek reprezentēts kā summārs mainīgo attiecinājums vizuālas reprezentācijas principam 1.1.att.

1.1 att Summāra mainīgo attiecinājuma metodes vizualizācija [pagaidu attēls]

Attēlā redzams, kā dzemdes kakla vēža paredzēšanas modelis ar summāru mainīgo attiecinājuma metodi raksturo mainīgo vērtību ietekmi uz dzemdes kakla vēža varbūtību; kā redzams 1.1. att risku mazina faktori zilā krāsā un palielina faktori sarkanā krāsā, sasummējot visus risku palielinošus un samazinošus faktorus, iegūst modeļa paredzējumu.

Matemātiski šo procesu definē formula 1.1:

(1.1)

kur

g - izskaidrojuma modelis

z’ {0, 1}M – koalīcijas vektors

M – maksimālais koalīcijas izmērs

phi j – Šaplija vērtībās mainīgā j vērtībām

phi o – modeļa paredzējums ar tukšu koalīciju (visi mainīgo vērtības ir “trūkstošas”)

Galvenā motivācija TreeSHAP lietošanai ir tā, ka Šaplija vērtību aprēķins ir nepraktisks, jo tā sliktākā gadījuma skaitļošanas sarežģītība ir 2N (eksponenciāla). Dziļāku izpratni par to, kā Šaplija vērtības tiek aprēķinātas koalīcijās, var iegūt avotā Grabisch & Roubens, 1999.

Lai spētu praktiski pielietot Šaplija vērtības modeļu darbības izskaidrošanā, tās tiek deterministiski aproksimētas lēmumu kokos bāzētos modeļos ar TreeSHAP algoritma palīdzību, kas samazina sliktākā gadījuma skaitļošana sarežģītību uz polinomiālu laiku (eksponenciāla vietā). Precīzs TreeSHAP algoritma skaitļošanas laiks raksturots formulā 1.1:

O(TLD2) (1.1)

kur

T ir lēmumu koku skaits,

L ir maksimālais koka galotņu skaits (leaves) jebkurā kokā,

D ir maksimālais dziļums jebkuram kokam

Pseidoalgoritma aprakstu TreeSHAP ir iespējams atrast (Lundberg, 2018) 3.2 nodaļā

TreeSHAP mainīgā svarīgums tiek aprēķinats izmantojot formulu 1.1:

(1.1)

kur

Ij – mainīgā j svarīguma vērtība

phi j pow i – mainīgā j Šaplija vērtības i mainīgā vērtībai

Būtībā aprēķins ir ļoti vienkāršs – TreeSHAP algoritma visas iegūtās absolūtās Šaplija vērtības tiek sasumētas katram mainīgajam un tas rada mainīgā svarīguma mēru.

## Hibrīdmetodes

Hibrīdmetodes ir vairāku metožu (filtra, ietverto, aptinuma) apvienojums, lai veiktu sarežģītākus mainīgo izvēles aprēķinus.

### Rekursīva mainīgo izslēgšana

Rekursīva mainīgo izslēgšana (tālāk RFE) (Kuhn, 2013) ir hibrīdmetode, kas tiek izmantota ar aptinuma, filtra, ietverto algoritmu, lai ranžētu mainīgos pēc to svarīguma. Pašu nesvarīgāko mainīgo atmet no mainīgo kopas, un modeli apmāca ar mainīgo apakškopu (kopa bez visnesvarīgākā mainīgā). Process turpinās līdz a) nav mainīgo, kurus izslēgt; b) ir sasniegta norādīta mainīgo izslēgšanas robeža.

Eksperimentā RFE tiek izmantota kopā ar iepriekš aprakstītājām aptinuma, filtra, ietvertajām metodēm, lai noteiktu metožu efektivitāti modeļa veiktspējas saglabāšanai, atmetot liekos mainīgos.

# Eksperimenti mainīgo svarīguma Metožu EfektIvitātes noteikšanai

Eksperiments izvērtē mainīgo svarīguma metožu efektivitāti CART algoritmiem. Nodaļā 3.1 tiks aprakstīts, kādi CART algoritmi tiks izmantoti eksperimentā.

Tālāk nodaļā 3.2 tiek raksturota datu kopu izvēles un pirmsapstrādes procedūra. Nodaļās 3.3 un 3.4 tiek raksturoti dabiskie un sintētiskie eksperimenti un to izpildes process. Nodaļā 3.5 veikts globālās izskaidrojamības metožu efektivitātes novērtējums, balstoties uz eksperimentu datu analīzē iegūtajiem kritērijiem. Nodaļa 3.6 veic padziļinātu dabisko datu eksperimenta analīzi, izvērtējot kāds ir ātrdarbības un veiktspējas guvums vai zaudējums optimālu mainīgo apakškopām.

## Eksperimentā izmantotie CART algortimi

Darbā tiek izmantoti sekojošie CART algoritmi – lēmumu koki, lēmumu koku izlase, gradienta stiprinoša lēmumu koku izlase. CART algoritmi ļauj aproksimēt mainīgo svarīguma metožu efektivitāti to alkatīgās dabas dēļ (CART algoritmi vienmēr izvēlēsies pašu svarīgāko mainīgo kā koka sakni). Augstāk minētie CART algoritmi atšķiras veiktspējā, ātrdarbībā un izskaidrojamība, tāpēc ir svarīgi veikt salīdzinājumu un atrast, vai eksistē kāds optimāla kompromisa punkts.

### Lēmumu koki

Nodaļa 3.1 balstīta uz (Hastie et al., 2009, pp. 305-321) grāmatu “Elements of Statistical Learning”. CART algoritms ir lēmumu kokos bāzēta metode, lai sadalītu mainīgo “telpu” taisnstūros un taisnstūru apakštelpai izveidotu ļoti vienkāršus modeļus (konstants modelis, apakštelpai paredz konstantu rezultātu). Kur tiek veikti rekursīvi bināri lēmumi alkatīgi (greedy) izvēloties mainīgo, kas maksimizē vai minimizē izvēlēto kritēriju (parasti, klasifikācijas uzdevumos - Džinī netīrības mērs (Gini impurity), regresijas uzdevumos - vidējā kvadrātiskā kļūda). Process turpinās līdz tiek sasniegts kāds apstādināšanas kritērijs (beidzas datu punkti kurus atšķelt vai arī pārsniegts kāds no modeļa ierobežojumiem, piemēram, koka dziļums).

Turpinājumā ir apskatīti principi kā darbojas lēmumu koks regresijas gadījumā. Formula 1.1. apraksta kā no mainīgajiem x tiek iegūts modeļa rezultāts y

(1.1)

kur

y – ir modeļa paredzējums

lbrace x in {R sub m} rbrace – ir identitātes funkcija, kas atgriež 1, ja x ir apakškopā R sub m, savādāk 0

c sub m – vidējā vērtība modeļa apakštelpa (konstants modelis)

R sub m – modeļa apakštelpa m

M – modeļa apakštelpas kopa

Atrast labāko bināro apakštelpu kopu minimizējot kvadrātisko summu, ir praktiski neiespējami, tāpēc tiek implementēts alkatīgs (greedy) algoritms, kurš izskata tikai nākamo līmeni (šķēlumu), nevis mēģina uzbūvēt globāli optimālus lēmumu kokus. Lai atrastu nākamo bināro šķēlumu, tiek izvēlēts kvadrātiskās kļūdas minimums no abām apakštelpām. Tas nodrošina, ka ir iespējams ātri sarēķināt katra mainīgā šķēluma punktu visām mainīgā vērtībām, atrodot šķēluma punktu, kas naivā veidā minimizē kvadrātisko kļūdu. Atrodot labāko šķēluma punktu, process tiek turpināts visām apakštelpām līdz sasniegts beigšanas kritērijs. Formulās bāzētu dziļāku procesa aprakstu iespējams atrast avotā (Hastie, T. et al. 2009, pp. 306)

Lai algoritms strādātu klasifikācijas lēmumu kokiem vienīgā izmaiņa, kas jāveic ir šķēlumu kritērija maiņa uz džinī netīrības mēru, kas definēts formulā 1.1:

(1.1)

kur

G – Džinī netīrības mērs

C – kopējais klašu skaits mērķim

p sub i – varbūtība izvēlēties klasi i datu punktam

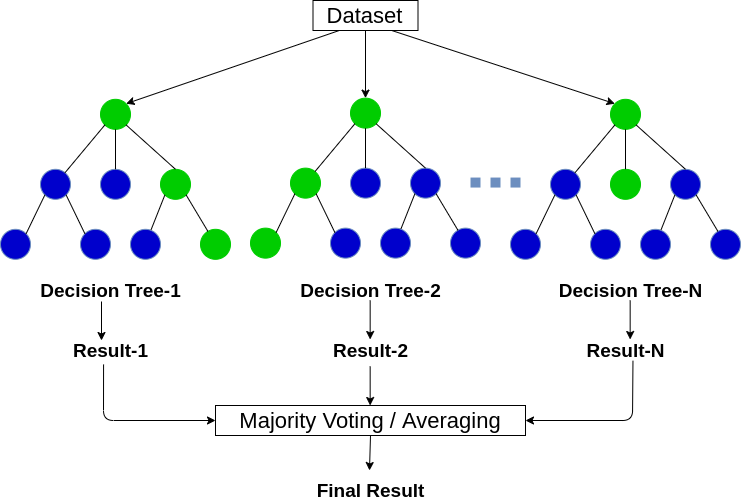
Džinī netīrības mēra vērtības atrodas skalā no 0 līdz 0.5, kur vērtība 0 raksturo šķēlumu, kas spēj ideāli sadalīt klases un vērtība 0.5 raksturo šķēlumu, kurš ir maksimāli “netīrs” t.i. nespēj sadalīt klases vispār. Kā izmantot Džinī vai modeļa kritērija guvumu mainīgo svarīguma noteikšanai, ir aprakstīts 3.1 nodaļā. Viens no lēmumu koku vislielākajiem trūkumiem lēmumu kokiem ir to pārāk lielā pielāgošanās (overfitting) apmācības datiem, praktiski, “iegaumējot” datus, nevis attēlojot vispārināmas saistības starp mainīgajiem. Šo problēmu risina lēmumu koku izlases, kuras aprakstītas nākamajā nodaļā.

### Lēmumu koku izlase

Lēmumu koku izlases metode (Breiman, 2004) veido vairākus lēmumus kokus un veic agregācijas (ensemble) veida paredzējumu (klasifikācijas uzdevumos izvēloties vispopulārāko klasi, regresijas ņemot vidējo vērtību no visu koku paredzējumiem). Viena no galvenajām atšķirībām no lēmumu kokiem, kas ļauj lēmumu koku izlasēm sasniegt optimālu “sarežģītību”, nevis pārāk stipri pielāgoties datiem kā to dara lēmumu koki ir:

* Lēmumu koku izlasēs apmācot lēmumu kokus izmanto nejauši izvēlētu mācību datu apakškopu nevis pilnu mācību datu kopu
* Lēmumu koku izlasēs apmācot lēmumu kokus katram šķēlumam izvērtē labākos šķēlumus tikai mainīgo apakškopai nevis visiem mainīgajiem (parasti kvadrātsakne no mainīgo skaita)

Vizuāla reprezentācijas lēmumu koka izlasei ir redzama attēlā 1.1. Dziļāku izpratni par matemātiskajiem principiem ir iespējams iegūt (Louppe, 2014) 4. nodaļā.



Att 1.1 Lēmumu koku izlašu agregācijas process [pagaidu attēls]

### Gradienta stiprinoša lēmumu koku izlase

Līdzīgi kā lēmumu koku izlases metode, gradienta stiprinoša lēmumu koku izlase ir balstīta uz vairāku “vāju” modeļu agregāciju vienā lielā modelī. Gradienta stiprinoši lēmumu koki (tālāk GBDT, no angl. Gradient boosted decision trees) (Natekin & Knoll, 2013) izmanto citādu koku “audzēšanas” stratēģiju. Galvenā GBDT ideja ir pievienot jaunus kokus agregētajam modelim sekvenciāli, mēģinot minimizēt iepriekšējā agregētā modeļa kļūdu (šajā kontekstā to sauc par atlikumu, angliski - residual). Šo metodi aprakstīja (Freund & Schapire, 1997) un (Friedman et al., 2000), (Friedman, 2001) tā tika saukta par gradienta stiprinošas mašīnas (Gradient boosting machines) metodi. GBDT metode ir guvusi popularitāti ar tādam programmatūras implementācijām kā CatBoost (Ostroumova et al., 2017), LightGBM (Guolin Ke et al., 2017) un XGBoost (Chen & Guestrin, 2016).

GBDT secīgi apmāca jaunus lēmuma koku modeļus, lai veidotu precīzāku aproksimāciju mērķa mainīgajam. Algoritms strādā tā, lai jaunie lēmuma koki maksimāli korelēti ar negatīvo gradientu zaudējuma funkcijai.

Vispārīgu GBDT ir iespējams definēt šādi:

Ieejas parametri:

1. ieejas dati (x, y)
2. iterāciju skaits M
3. zaudējuma funkcijai phi (y, f)
4. modeļa izvēle h(x, theta)

Algoritms:

1. Inicializē sākuma modeli f sub 0 ar konstanti (vidējā vērtība regresijai, populārākā klase klasifikācijai)
2. līdz sasniegts iterāciju limits M darīt:
   1. aprēķina negatīvo gradienta vērtību
   2. apmāca jaunu izvēlēto modeli
   3. atrod labāko gradienta nolaišanas (gradient descent) soļa izmēru
   4. atjauno funkcijas aproksimāciju ņemot vērā jaunā izvēlētā modeļa atlikumu (residual)

Dziļāku izskaidrojumu ir iespējams atrast (Friedman, 2001).

## Izmantotās datu kopas un to pirmsapstrāde

Datu kopas ir izvēlētas pēc 1) instanču skaita 2) mainīgo skaita 3) heterogenitātes (nevienādabīgas, dažādas datu kopas). Kopā ir izvēlētas 4 regresijas uzdevumu datu kopas (Diabetes, Boston housing, Crime, Ames housing) un 4 klasifikācijas datu kopas (Wine, Breast cancer, Phishing, Mushrooms).

Datu kopas tiek apstrādātas ar šādu loģiku:

1. Visus kategoriskos (categorical) mainīgos kodē ar apzīmējuma kodējumu (label encode)
2. Nepārtrauktos mainīgos ar trūkstošām vērtībām pārvēršs kategoriskajos mainīgajos ar papildus klasi – trūkstošo vērtību klasi
3. Nepārtrauktie mainīgie bez trūkstošām vērtībām tiek izmantoti modelī bez papildus darbībām
4. Dažās datu kopas neinformatīvas vai atkļudošanas informācijas kolonas ir izņemtas skatīt tabulu 4.1, lai iegūtu detalizētāku informāciju.

Tabula 4.1 Izmantotās datu kopas un to pirmsapstrāde

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Datu kopas vārds | Mainīgo skaits | Mainīgo skaits pēc apstrādes | Instanču skaits | Instanču skaits pēc apstrādes | Uzdevums | Saite uz datu kopu |
| Diabetes | 10 | 10 | 442 | 442 | Regresija | https://scikit-learn.org/stable/datasets/toy\_dataset.html |
| Boston housing | 13 | 13 | 506 | 506 | Regresija | https://scikit-learn.org/stable/datasets/toy\_dataset.html |
| Crime | 128 | 122 | 1994 | 1994 | Regresija | https://archive.ics.uci.edu/ml/datasets/communities+and+crime |
| Ames housing | 82 | 79 | 2930 | 2925 | Regresija | http://jse.amstat.org/v19n3/decock/AmesHousing.txt |
| Wine | 13 | 13 | 178 | 178 | Vairāku klašu klasifikācija | https://scikit-learn.org/stable/datasets/toy\_dataset.html |
| Breast cancer | 30 | 30 | 569 | 569 | Bināra klasifikācija | https://scikit-learn.org/stable/datasets/toy\_dataset.html |
| Phishing | 30 | 30 | 11055 | 11055 | Bināra klasifikācija | https://archive.ics.uci.edu/ml/datasets/phishing+websites |
| Mushrooms | 22 | 22 | 8124 | 8124 | Bināra klasifikācija | https://archive.ics.uci.edu/ml/datasets/mushroom |

## Eksperiments ar dabiskiem datiem

Modeļi tiek apmācīti ar bāzes hiperparametriem (skatīt ietvara Scikitlearn versijas 0.24.0 dokumentāciju). Tiek izmantota programmēšanas valodas Python ietvas Scikitlearn, lai apmācītu modeļus – lēmumu koks, lēmumu koku izlase, un Python implementācija XGBoost modulim – gradienta stiprinoša lēmumu koku izlases modelim. Eksperimenti tika veikti uz AMD Ryzen 3200g procesora. Eksperimenti aizņēma ~80 stundas. Eksperimentā datu kopas tiek nejauši sadalītas apmācības (70%) un testa datos (30%). Regresijas dati tiek vienkārši nejauši sadalīt, bet klasifikācijas dati tiek nejauši sadalīti stratificēti (stratified) saglabājot klašu balansu starp apmācības un testa datiem. Abos gadījumos nejauša datu sadalīšana tiek deterministiski kontrolēta izlases veidā (random seed).

Katrā iterācijā tiek:

1. Apmācīts bāzes modelis (modelis ar visiem mainīgajiem) izlases veidā (random seed) N reizes ar visiem mainīgajiem, veidojot bāzes precizitāti.
2. Sarēķināts mainīgo svarīgums izmantojot konkrētu mainīgo svarīguma metodi un apmācības datus. Mainīgie tiek saranžēti pēc svarīguma.
3. No mainīgo svarīguma aprēķina izvēlēts mainīgais ar vismazāko svarīgumu
4. Visnesvarīgakais mainīgais tiek izņemts no apmācības un testa datu kopas
5. Modelis tiek pārmācīts izlases veidā (random seed) N reizes bez visnesvarīgākā mainīgā
6. Process turpinās līdz vairs nav mainīgo kurus atmest no datu kopas

Regresijas uzdevumu precizitātes izvērtēšanai tiek izmantota vidējā kvadrātiskā kļūda, klasifikācijas uzdevumu izvērtēšanai Matteja korelācijas koeficients (Matthew correlation coefficient). Kvadrātiskā kļūda jau ir apskatīta nodaļā 3.1. Matteja korelācijas koeficientu raksturo formula 4.1:

(4.1)

kur

MCC - Matteja korelācijas koeficients

TP – patiesi pozitīvi klasificēti dati (true positives)

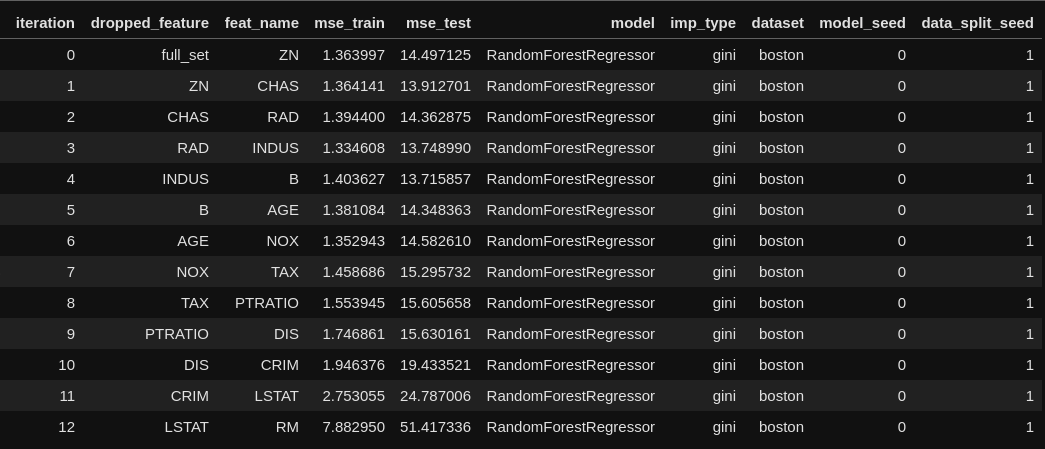
TN – patiesi negatīvi klasificēti dati (true negatives)

FP - nepatiesi pozitīvi klasificēti dati (false positives)

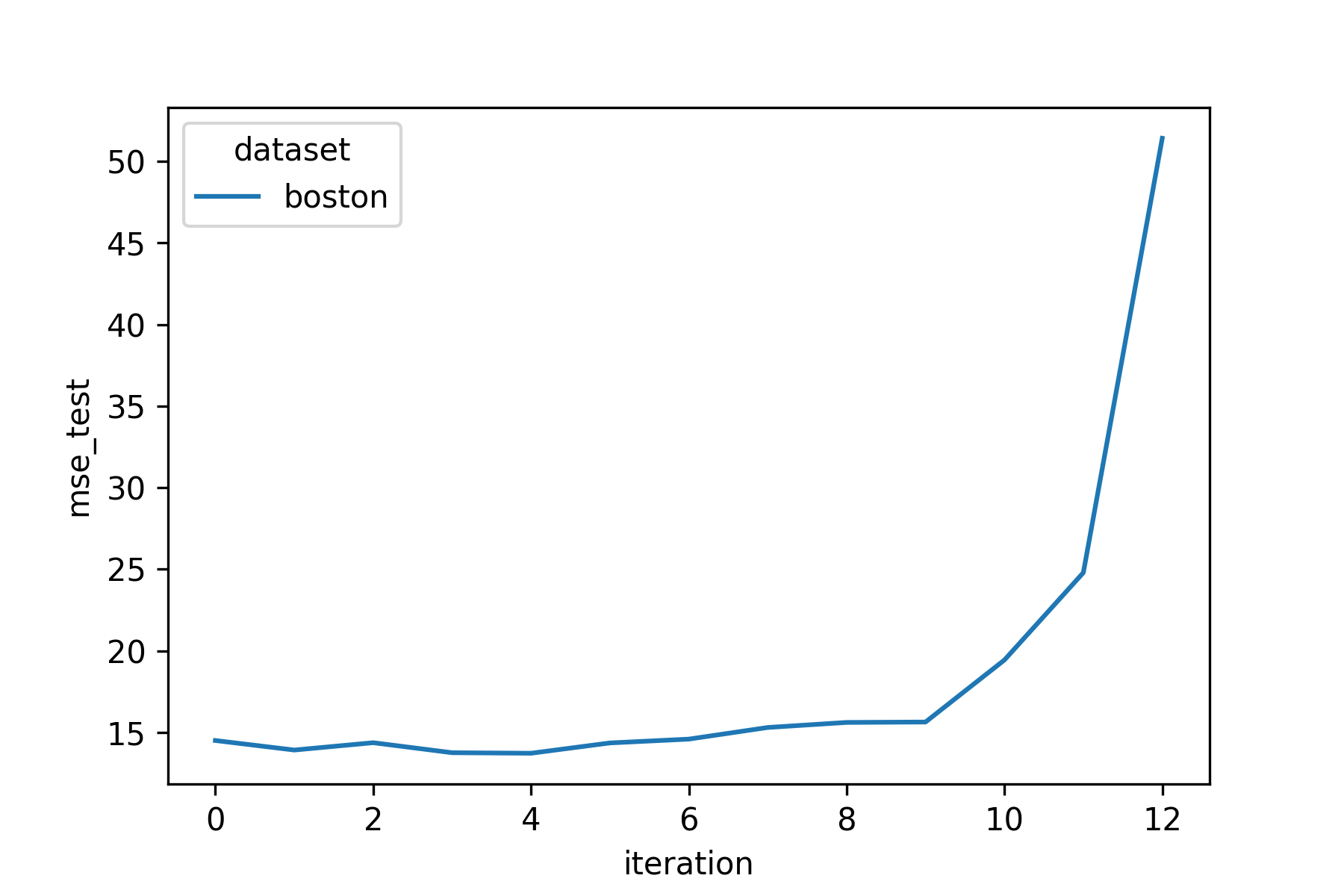
FN – nepatiesi negatīvi klasificēti dati (false negatives)

Matteja korelācijas koeficients izvēlēts kā klasifikācijas modeļu precizitātes mērs, jo tas spēj labi attēlot nebalansētu klasifikācijas uzdevumu modelēšanas precizitāti. Dziļāku izskaidrojumu ir iespējams atrast (Boughorbel et al., 2017).

Turpinājumā ir apskatīts viens no eksperimentiem. Attēlā 1.1 var redzēt kā strādā mainīga svarīguma metodes “gini” (kritērija guvums) izvērtējums datiem “boston”. Katrā iterācijā tiek nomests 1 mainīgais un modelis apmācīts no jauna.

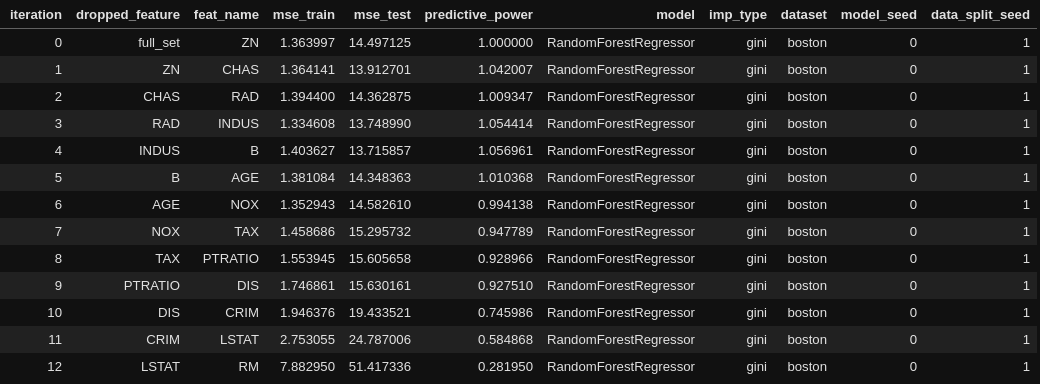
 Attēls 1.1 Metodes “gini” (kritērija guvums) izvērtējums datiem “boston”

Šo procesu var arī attēlot grafiski uzzīmējot līnijveida grafiku, kur uz x ass ir iterācija un uz y ass ir “mse\_test” (vidējā kvadrātiskā kļūda) attēlā 1.2. redzama grafiska reprezentācija eksperimentam.

 Attēls 1.2 Grafisks eksperimenta procesa attēlojums

Kā redzams attēlā 1.2 nometot pirmo visnelietderīgāko mainīgo, modeļa vidējā kvadrātiskā kļūda pat uzlabojas. Var secināt, ka mainīgais bija neinformatīvs. No 0 iterācijas līdz 9 modeļa kļūdas pieaugums ir salīdzinoši lēzens. No 10 iterācijas līdz beigām modeļa kļūdas pieaugums ir straujš. No tā var secināt, ka mainīgi, kas tika nomesti sākot no 10 iterācijas, bija svarīgi modelī.

Lai veiktu precizitātes datu agregāciju starp regresijas un klasifikācijas datu kopām nepieciešams normalizēt eksperimenta precizitāti. Dati tiek normalizēti dalot sākuma modeļa precizitātes vērtību (0 iterācija) ar modeļa precizitātes vērtību iterācijā n. Attēlā 1.3 ir parādīts aprēķina piemērs.

 Attēls 1.3 Normalizētas precizitātes koeficients (“predictive\_power”)

Normalizētas precizitātes koeficients ļauj arī daudz intuitīvāk vizualizēt modeļa precizitātes zaudējuma progresu Attēls 1.4. Lai veiktu modeļu novērtējumu eksperimentos ar dabiskiem datiem, ir izvirzīti 3 kritēriji.

Attēls 1.4 Modeļa veiktspējas degredācija

* Pirmais kritērijs “modeļa uzlabojums”. Tas raksturo vai ir iespējams atrast mainīgo apakškopu, kas uzlabo modeļa darbību.
* Otrais kritērijs “lieko mainīgo nomešana”. Tas procentuāli nosaka, cik mainīgos ir iespējams atmest bez precizitātes zaudējuma.
* Trešais kritērijs “minimālā modeļa veiktspēja”. Tas nosaka, kādu proporciju no modeļa veiktspējas ir iespējams saglabāt ar minimālu modeli (tāds modelis, kuram ir nomesti 80% no mainīgajiem).

Kritēriji ir balstīti uz praktiskiem modelēšanas scenārijiem. Pirmais kritērijs “modeļa uzlabojums” raksturo tādu darba vidi kā, piemēram, mašīnmācīšanās sacensības “Kaggle”, kur pats svarīgākais ir apmācīt modeli ar visaugstāko veiktspēju, šādos scenārijos mainīgo izvēles metodes spēj uzlabot modeļa veiktspēju un izskaidrojamību un palīdzēt modelētājam ar jaunu mainīgo izveidi, raksturojot kuri no mainīgajiem ir svarīgi modelī.

Otrais kritērijs “lieko mainīgo nomešana” raksturo praktisku darba vidi datu zinātniekam uzņēmumā, ja uzņēmums jau ir izveidojis māšīnmācīšanās risinājumus nav pieļaujams modeļa veiktspējas zaudējums. Uzņēmuma vadība, neļaus datu zinātniekam veidot mazākas veiktspējas modeli pat ja tas ir izskaidrojamāks un mazāk sarežģīts, tāpēc nepieciešams atmest maksimāli daudz liekos mainīgos, neietekmējot modeļa veiktspēju.

Trešais kritērijs “minimālā modeļa veiktspēja” ir balstīts uz praktisku scenāriju, kad vislielāko vērtību nesniedz modeļa veiktspēja, bet modeļa izskaidrojamība, piemēram, uzņēmuma mārketinga procesa modelēšanā izskaidrojamība var būt svarīgāka nekā veiktspēja. Uzņēmuma mārketinga procesa vadītāja var būt vairāk ieinteresēti, kādi faktori veido veiksmīgu mārketinga kampaņu, nevis cik liela būs konkrētas kampaņas naudiskā atdeve. Lai izveidotu izskaidrojamākus modeļus trešajā kritērijā ir izmantots Pareto princips, pieņemot, ka 20% no mainīgajiem nodrošina 80% modeļa veiktspējas.

Dabisko datu eksperimenta rezultāti redzami tabulā 1.1. Rezultātos attēlota vidējā vērtība +/- 95% pārliecības intervāla vērtības.

Tabula 1.1 Dabisko datu rezultāti

|  |  |  |  |
| --- | --- | --- | --- |
| **metode** | modeļa uzlabojums | lieko mainīgo nomešana | minimāla modeļa veiktspēja |
| Kritērija guvums | 1.31 ± 0.14 | **46.74 ± 1.92** | 91.72 ± 0.67 |
| Abpusēja informācijas mērs | **1.73 ± 0.16** | 42.69 ± 2.06 | 90.6 ± 0.73 |
| Determinācijas koeficients un ANOVA | 1.38 ± 0.2 | 38.17 ± 2.0 | 89.59 ± 0.74 |
| Permutācijas | 1.36 ± 0.15 | 45.44 ± 2.16 | **92.01 ± 0.66** |
| TreeSHAP | 1.09 ± 0.24 | 43.46 ± 2.13 | 87.69 ± 1.01 |
| Pīrsona koef. | 1.36 ± 0.19 | 34.05 ± 1.92 | 87.71 ± 0.81 |
| Spīrmana koef. | 1.41 ± 0.18 | 34.17 ± 1.86 | 87.46 ± 0.73 |
| Kendala koef. | 1.32 ± 0.17 | 32.58 ± 1.97 | 87.52 ± 0.76 |
| Spēka mērs | -0.68 ± 0.47 | 13.11 ± 1.17 | 35.19 ± 1.52 |

## Eksperiments ar sintētiskiem datiem

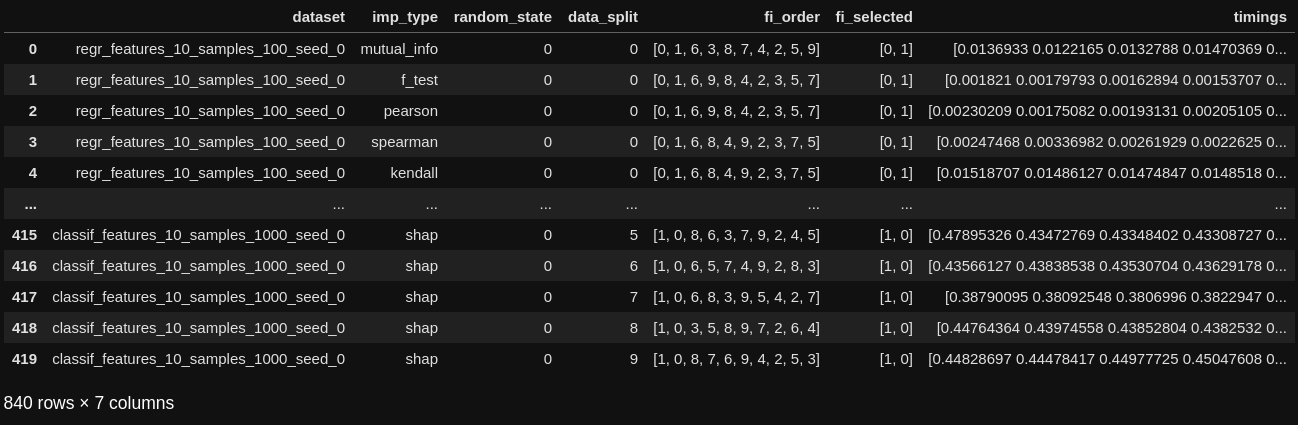
Galvenā motivācija veikt eksperimentu ar sintētiskiem datiem ir tas, ka ar sintētiskajiem datiem Ir iespējams izveidot tādu datu kopu, kurā ir skaidri zināms, kuri mainīgie ir informatīvi un kuri ir lieki. Dabisko datu kopās nepastāv šādas iespējas, jo jebkura interpretācija būs pārāk subjektīva.

Sintētisko datu eksperiments noris līdzīgi kā dabisko datu eksperiments, bet dabisko datu vietā ar ietvaru Scikitlearn tiek izveidotas mākslīgas regresijas un klasifikācijas datu kopas. Datu kopas tiek saģenerētas pēc instanču un mainīgo skaita (skatīt tabulu 1.1). Sintētiskā eksperimenta mērķis ir 1. novērtēt mainīgo svarīguma metožu efektivitāti, aprēķinot, cik veiksmīgi metodes izvēlas informatīvus mainīgos un 2. izvērtēt metožu ātrdarbību. Metodēm ir uzlikts 60 sekunžu izpildes ierobežojums. Eksperimenti tika veikti uz AMD Ryzen 3200g procesora. Eksperimenti aizņēma ~4 stundas.

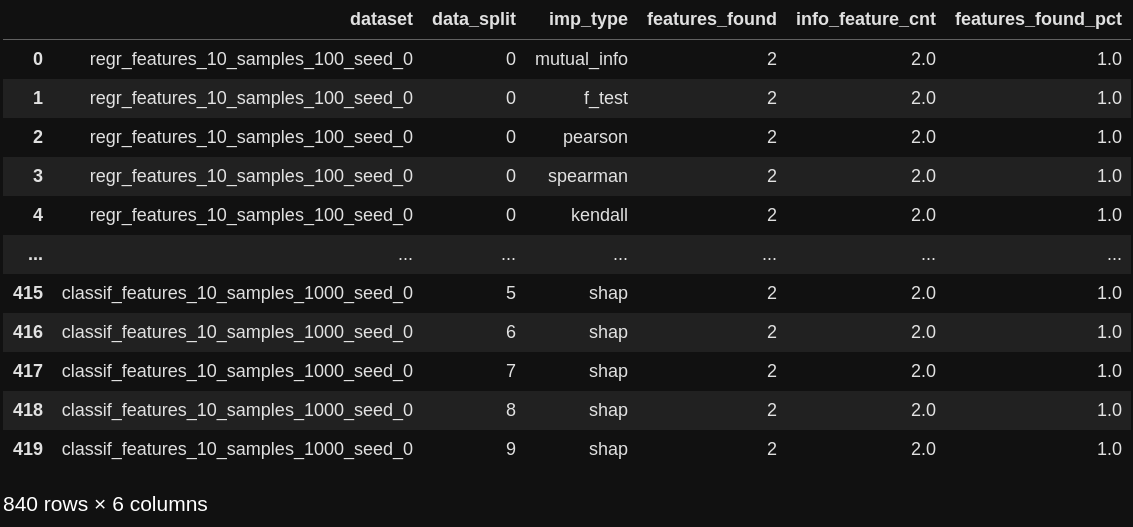
Tabula 1.1 Eksperimenta sintētiskās datu kopas

|  |  |  |  |
| --- | --- | --- | --- |
| sintētiskā datu kopa | mainīgo skaits | instanču skaits | uzdevuma tips |
| regr\_features\_10\_samples\_100\_seed\_0 | 10 | 100 | regresija |
| 'regr\_features\_10\_samples\_1000\_seed\_0' | 10 | 1000 | regresija |
| 'regr\_features\_10\_samples\_10000\_seed\_0' | 10 | 10000 | regresija |
| 'regr\_features\_100\_samples\_1000\_seed\_0' | 100 | 1000 | regresija |
| 'regr\_features\_100\_samples\_10000\_seed\_0' | 100 | 10000 | regresija |
| 'classif\_features\_10\_samples\_100\_seed\_0' | 10 | 100 | klasifikācija |
| 'classif\_features\_10\_samples\_1000\_seed\_0' | 10 | 1000 | klasifikācija |
| 'classif\_features\_10\_samples\_10000\_seed\_0' | 10 | 10000 | klasifikācija |
| 'classif\_features\_100\_samples\_1000\_seed\_0' | 100 | 1000 | klasifikācija |
| classif\_features\_100\_samples\_10000\_seed\_0' | 100 | 10000 | klasifikācija |

Sintētisko datu eksperimenta rezultātu paraugu var redzēt attēlā 1.1

Attēls 1.1 Sintētisko datu eksperimenta rezultāts

Attēlā 1.1 kolona “dataset” norāda kura no sintētiskajām datu kopām izmantota, kolona “imp\_type” (feature importance type) norāda kura mainīgo svarīguma metode ir izmantota. Kolonas “random state” un “data\_split” izlases veidā kontrolē stohastisko procesus. Kolona “fi\_order” norāda konkrētas datu kopas un mainīgo izvēles metodes mainīgo svarīguma ranžēšanu. Kolona “fi\_selected” norāda kādus mainīgos metode izvēlējās kā informatīvus. Kolona “timings” norāda kāda ir metodes ātrdarbība. Ātrdarbības pārbaude tiek veikta 10 reizes, tāpēc kolona “timings” ir attēlota kā daļskaitļu datu masīvs. Attēlā 1.1 redzama sintētisko datu eksperimenta precizitātes izvērtējums kolona “features\_found\_pct” norāda cik procentuāli konkrētājā apakšeksprimentā mainīgo svarīguma metode ir izvēlējusies informatīvus mainīgos

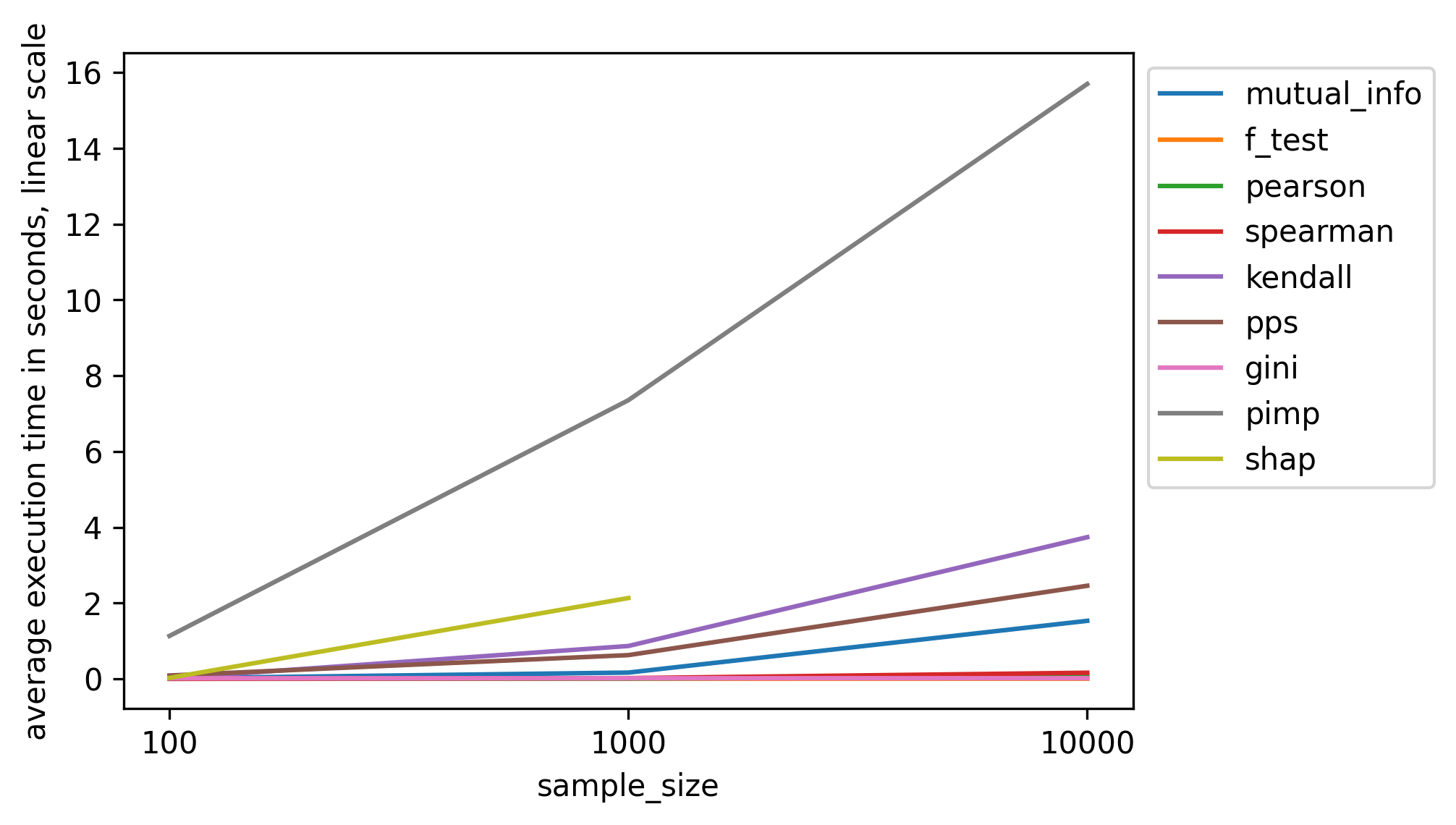
 Attēls 1.1 Sintētisko datu eksperimenta precizitātes izvērtēšana

Pēc attēlā 1.1 redzamo datu agregēšanas iegūta tabula 1.1. Attēla 1.1 datos kolona “average\_features\_found\_pct” nosaka cik veiksmīgi metode ir izvēlējusies informatīvos mainīgos, kur vērtība 1 nozīmē visos eksperimentos 100% informatīvo mainīgo izvēli un vērtība 0 nozīmē, ka metode nekad neizvēlas informatīvos mainīgos (vienmēr izvēlas liekos mainīgos, kā svarīgākos).

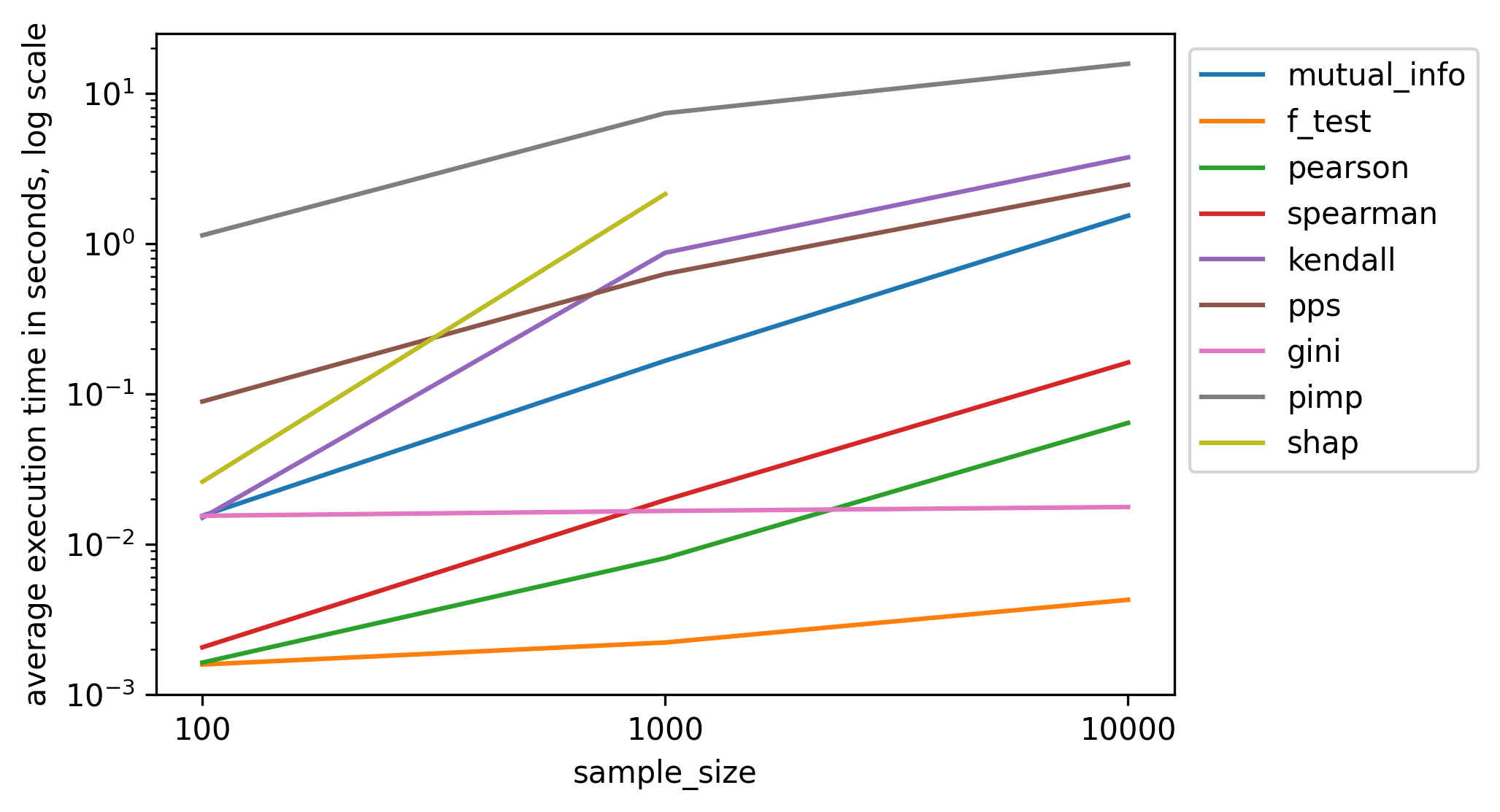
Tabula 1 Sintētisko datu eksperimenta precizitātes kritērija rezultāti

|  |  |  |
| --- | --- | --- |
| **metode** | **vidējā precizitāte** | **rangs** |
| pimp | 0.9255 | 1 |
| gini | 0.909 | 2 |
| shap | 0.875 | 3 |
| spearman | 0.748 | 4 |
| kendall | 0.7475 | 5 |
| f\_test | 0.7435 | 6 |
| pearson | 0.7435 | 6 |
| mutual\_info | 0.607 | 7 |
| pps | 0.063 | 8 |

Mainīgo izvēles metožu ātrdarbību attiecībā pret datu kopas instanču skaitu var redzēt attēlā 1.1

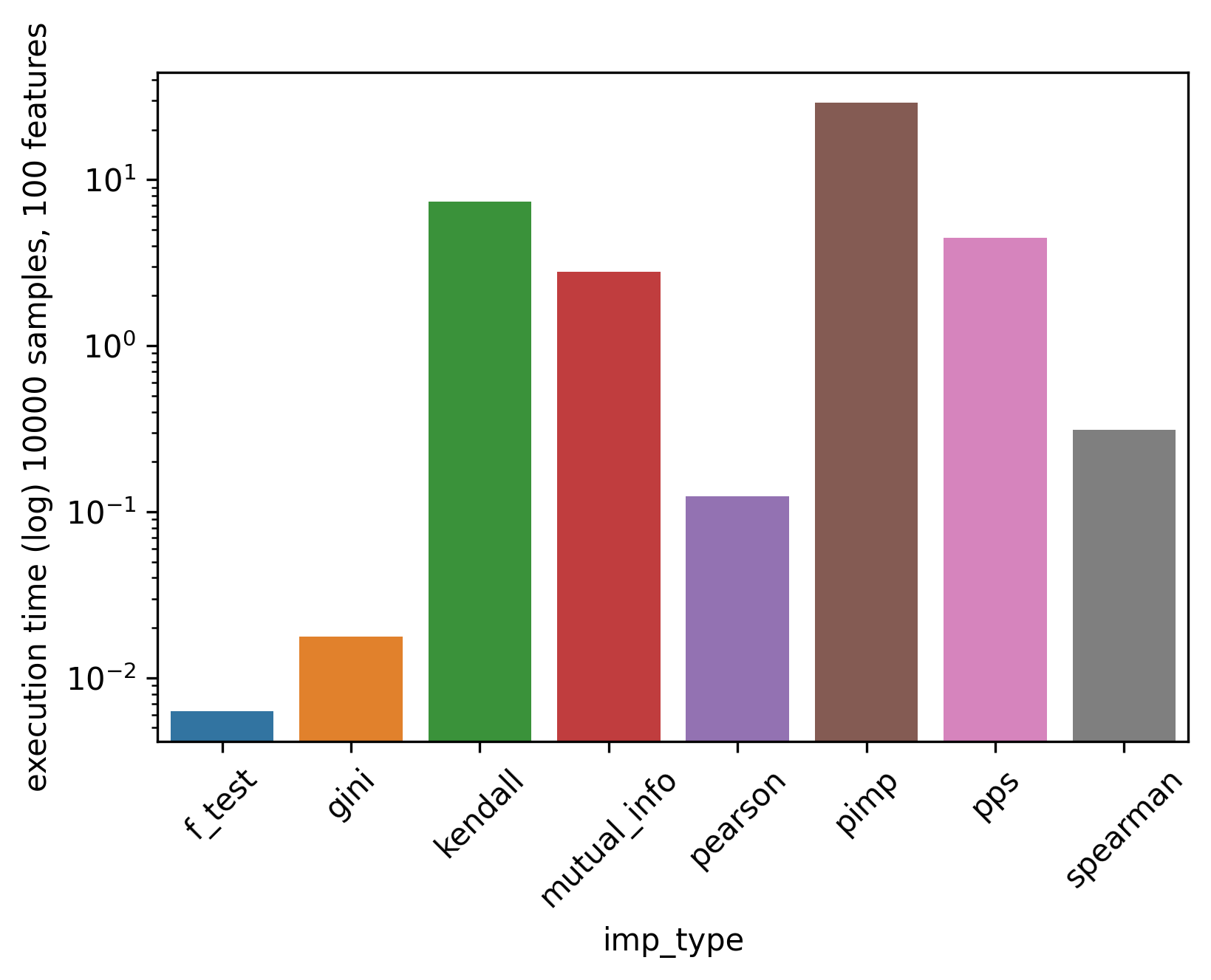
Attēls 1.1 Vidējās izpildes laika atkarība no instanču skaita (lineāra skala)

Lineāra skala grafika var radīt interpretācijas problēmas, tāpēc attēla 1.1 ātrdarbība ir attēlota logaritmiskā skalā

Att 1.1 Vidējās izpildes laika atkarība no instanču skaita (logaritmiska skala)

Sintētisko datu eksperimenta ātrdarbības kritērija noteikšanai tiek izmantota lielāka datu kopa ar 10000 instancēm un 100 mainīgajiem. Ātrdarbības kritērija galvenais mērķis ir noteikt vai metode ir piemērota lielām datu kopām, tāpēc tiek izmantota pati lielāka sintētiskā datu kopa. Metode “SHAP” nav iekļauta rezultātos, jo metode visos eksperimentos pārsniedza 60 sekunžu izpildes laika ierobežojumu. Attēlā 1.1 redzami ātrdarbības kritērija rezultāti logartimiskā skalā.

Sintētisko datu eksperimenta ātrdarbības kritērija noteikšanai tiek izmantota lielāka datu kopa ar 10000 instancēm un 100 mainīgajiem. Ātrdarbības kritērija galvenais mērķis ir noteikt vai metode ir piemērota lielām datu kopām, tāpēc tiek izmantota pati lielāka sintētiskā datu kopa. Metode “SHAP” nav iekļauta rezultātos, jo metode visos eksperimentos pārsniedza 60 sekunžu izpildes laika ierobežojumu. Attēlā 1.1 redzami ātrdarbības kritērija rezultāti logartimiskā skalā.

Figure 1: Attēls 1.1.

Tabulā 1.1 redzams ātrdarbības kritērija apkopojums un ātrdarbības kritērija metodes rangs.

Tabula 1.1 Atrdarbības kritērija metožu rangs

|  |  |  |
| --- | --- | --- |
| metode | vidējais izpildes laiks sekundēs | rangs |
| f\_test | 0.006296 | 1 |
| gini | 0.017717 | 2 |
| pearson | 0.124424 | 3 |
| spearman | 0.312715 | 4 |
| mutual\_info | 2.784574 | 5 |
| pps | 4.482986 | 6 |
| kendall | 7.398635 | 7 |
| pimp | 29.003301 | 8 |
| shap | \* | 9 |

\*shap pārsniedz 60 sekunžu izpildes limitu

## Mainīgo svarīguma metožu novērtējums

Metodes ir novērtētas pēc 5 kritērijiem (3 dabisko datu, 2 sintētisko datu).

Tabula 1.1 Mainīgo svarīguma metožu novērtējums

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **metode** | modeļa uzlabojums | lieko mainīgo nomešana | minimāla modeļa veiktspēja | sintētisko datu precizitāte | Ātrdarbība (sek) |
| Kritērija guvums | 1.31 ± 0.14 | **46.74 ± 1.92** | 91.72 ± 0.67 | 0.909 | 0.017717 |
| Abpusēja informācijas mērs | **1.73 ± 0.16** | 42.69 ± 2.06 | 90.6 ± 0.73 | 0.607 | 2.784574 |
| Determinācijas koeficients un ANOVA | 1.38 ± 0.2 | 38.17 ± 2.0 | 89.59 ± 0.74 | 0.7435 | 0.006296 |
| Permutācijas | 1.36 ± 0.15 | 45.44 ± 2.16 | **92.01 ± 0.66** | 0.9255 | 29.003301 |
| TreeSHAP | 1.09 ± 0.24 | 43.46 ± 2.13 | 87.69 ± 1.01 | 0.875 | Pārsniedza laiku |
| Pīrsona koef. | 1.36 ± 0.19 | 34.05 ± 1.92 | 87.71 ± 0.81 | 0.7435 | 0.124424 |
| Spīrmana koef. | 1.41 ± 0.18 | 34.17 ± 1.86 | 87.46 ± 0.73 | 0.748 | 0.312715 |
| Kendala koef. | 1.32 ± 0.17 | 32.58 ± 1.97 | 87.52 ± 0.76 | 0.7475 | 7.398635 |
| Spēka mērs | -0.68 ± 0.47 | 13.11 ± 1.17 | 35.19 ± 1.52 | 0.063 | 4.482986 |

## Dabisko datu mainīgo koalīciju analīze

### Optimālu koalīciju inferences un mācību ātruma guvuma analīze

### Vienkāršu modeļu paritātes analīze

Vai vienkāršiem modeļiem (lēmumu koki) ir veiktspējas paritāte ar sarežģītiem modeļeim (RF, GBDT), ja mainīgie ir labi izvēlēti?

secinājumi

Literatūra

Darbā izmantotie literatūras avoti, sakārtoti alfabētiskā secībā pēc pirmā autora uzvārda.

Kursa, M. and W. Rudnicki. “Feature Selection with the Boruta Package.” Journal of Statistical Software 36 (2010): 1-13.

Breiman, L. et al. “Classification and Regression Trees.” (1983).

Confalonieri, R. et al. “A historical perspective of explainable Artificial Intelligence.” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 11, (2021),.

Doshi-Velez, F., & Kim, B. (2017). Towards A Rigorous Science of Interpretable MachineLearning.arXiv: Machine Learning.

EU General Data Protection Regulation (GDPR): Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016, OJ 2016 L 119/1, Pieejams: https://eur-lex.europa.eu/eli/reg/2016/679/oj

Buchanan, B. and E. Shortliffe. “Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley series in artificial intelligence).” (1984).

Wick, Michael R. and W. B. Thompson. “Reconstructive Expert System Explanation.” Artif. Intell. 54 (1992): 33-70.

Miller, T.. “Explanation in Artificial Intelligence: Insights from the Social Sciences.” Artif. Intell. 267 (2019): 1-38.

Molnar, Christoph. "Interpretable machine learning. A Guide for Making Black Box Models Explainable", 2019. https://christophm.github.io/interpretable-ml-book/.

Guthke, Anneli. “Defensible Model Complexity: A Call for Data-Based and Goal-Oriented Model Choice.” Ground water 55 5 (2017): 646-650 .

Emden, Van. “An analysis of complexity.” (1971).

Prokopenko, M. et al. “An information-theoretic primer on complexity, self-organization, and emergence.” Complexity 15 (2009): 11-28.

Perrin, C., Michel, C., & Andréassian, V. (2001). Does a large number of parameters enhance model performance? Comparative assessment of common catchment model structures on 429 catchments. Journal of Hydrology, 242(3), 275–301.

Myung, I. J. (2000). The importance of complexity in model selection. Journal of Mathematical Psychology, 44(1), 190–204.

Mendoza, P. A. et al. “Are we unnecessarily constraining the agility of complex process-based models?” Water Resources Research 51 (2015): 716-728.

D'Amour, A. et al. “Underspecification Presents Challenges for Credibility in Modern Machine Learning.” ArXiv abs/2011.03395 (2020).

Goodhart, C.. “Monetary Theory and Practice.” (1984).

Strathern, Marilyn (1997). "'Improving ratings': audit in the British University system". European Review. John Wiley & Sons. 5 (3): 305–321.

Manheim, David and Scott Garrabrant. “Categorizing Variants of Goodhart's Law.” ArXiv abs/1803.04585 (2018)

Tukey, J.. “Exploratory data analysis.” Addison-Wesley series in behavioral science : quantitative methods (1977).

Simpson, E.. “The Interpretation of Interaction in Contingency Tables.” Journal of the royal statistical society series b-methodological 13 (1951): 238-241.

Miller, G. A.. “The magical number seven plus or minus two: some limits on our capacity for processing information.” Psychological review 63 2 (1956): 81-97.

Angwin J, Larson J, Mattu S, Kirchner L. Machine Bias. ProPublica; 2016. Pieejams : https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Rudin, C. et al. “The age of secrecy and unfairness in recidivism prediction.” ArXiv abs/1811.00731 (2018a).

Rudin, C.. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.” Nature Machine Intelligence 1 (2018b): 206-215.

Angelino, E. et al. “Learning Certifiably Optimal Rule Lists for Categorical Data.” J. Mach. Learn. Res. 18 (2017): 234:1-234:78.

Hastie, T. et al. “The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition.” Springer Series in Statistics (2009).

Breiman, L.. “Random Forests.” Machine Learning 45 (2004): 5-32.

Louppe, Gilles. “Understanding Random Forests: From Theory to Practice.” arXiv: Machine Learning (2014).

Natekin, Alexey and A. Knoll. “Gradient boosting machines, a tutorial.” Frontiers in Neurorobotics 7 (2013).

Freund, Y. and R. Schapire. “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.” COLT 1997 (1997).

Friedman, J.. “Special Invited Paper-Additive logistic regression: A statistical view of boosting.” Annals of Statistics 28 (2000): 374-376.

Friedman, J.. “Greedy function approximation: A gradient boosting machine.” Annals of Statistics 29 (2001): 1189-1232.

Ostroumova, L. et al. “CatBoost: unbiased boosting with categorical features.” NeurIPS (2018).

Ke, Guolin et al. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree.” NIPS (2017).

Chen, T. and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016).

Corder, Gregory W. and D. Foreman. “Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach.” (2009).

Wetschoreck, F., Krabel, T., & Krishnamurthy, S. (2020). 8080labs/ppscore: zenodo release (1.1.2) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.4091345

Shapley, L. S.. “Notes on the n-Person Game — II: The Value of an n-Person Game.” (1951).

Molnar, Christoph. "Interpretable machine learning. A Guide for Making Black Box Models Explainable", 2019, Pieejams: https://christophm.github.io/interpretable-ml-book/.

Grabisch, M. and M. Roubens. “An axiomatic approach to the concept of interaction among players in cooperative games.” International Journal of Game Theory 28 (1999): 547-565.

Lundberg, Scott M. et al. “Consistent Individualized Feature Attribution for Tree Ensembles.” ArXiv abs/1802.03888 (2018),

Cover, T. and J. Thomas. “Elements of Information Theory.” (1991).

McDonald, J.H. 2014. Handbook of Biological Statistics (3rd ed.). Sparky House Publishing,

Rudin, C. et al. “The age of secrecy and unfairness in recidivism prediction.” ArXiv abs/1811.00731 (2018a).

Rudin, C.. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.” Nature Machine Intelligence 1 (2018b): 206-215.

Boslaugh, S. and P. Watters. “Statistics in a nutshell.” (2008), O'Reilly Media.

Anscombe, F.. “Graphs in Statistical Analysis.” The American Statistician 27 (1973): 17-21.

Boughorbel, S. et al. “Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric.” PLoS ONE 12 (2017).

*Murthy, Sreerama K. and S. Salzberg. “Decision Tree Induction: How Effective is the Greedy Heuristic?” KDD (1995).*

*Hyafil, L. and R. Rivest. “Constructing Optimal Binary Decision Trees is NP-Complete.” Inf. Process. Lett. 5 (1976): 15-17.*

*Hu, Xiyang et al. “Optimal Sparse Decision Trees.”*NeurIPS *(2019).*

*Bertsimas, D. and Jack Dunn. “Optimal classification trees.”*Machine Learning*106 (2017): 1039-1082.*

*Lin, Jimmy J. et al. “Generalized and Scalable Optimal Sparse Decision Trees.”*ICML*(2020).*

*Guyon, I. and A. Elisseeff. “An Introduction to Variable and Feature Selection.”*J. Mach. Learn. Res.*3 (2003): 1157-1182.*

*Kohavi, R. and George H. John. “Wrappers for Feature Subset Selection.”*Artif. Intell.*97 (1997): 273-324.*

*Amaldi, E. and V. Kann. “On the Approximability of Minimizing Nonzero Variables or Unsatisfied Relations in Linear Systems.”*Theor. Comput. Sci.*209 (1998): 237-260.*

*Guyon, I. et al. “Gene Selection for Cancer Classification using Support Vector Machines.”*Machine Learning*46 (2004): 389-422.*

*Kim, Been et al. “Examples are not enough, learn to criticize! Criticism for Interpretability.”*NIPS*(2016).*

*Arrieta, A. et al. “Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI.”*ArXiv*abs/1910.10045 (2020)*

*Bostrom, N.. “Superintelligence: Paths, Dangers, Strategies.” (2014).*

Mueller, Shane T. et al. “Explanation in Human-AI Systems: A Literature Meta-Review, Synopsis of Key Ideas and Publications, and Bibliography for Explainable AI.” ArXiv abs/1902.01876 (2019)

Thornton, C. et al. “Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms.” Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (2013)

Akiba, Takuya et al. “Optuna: A Next-generation Hyperparameter Optimization Framework.” Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2019)

Falkner, S. et al. “BOHB: Robust and Efficient Hyperparameter Optimization at Scale.” ICML (2018).

Li, Lisha et al. “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization.” J. Mach. Learn. Res. 18 (2017): 185:1-185:52.Kuhn, M. and Kjell Johnson. “Applied Predictive Modeling.” (2013).

PIELIKUMI

1. pielikums. Pielikuma nosaukums