

Datordomāšana 2.0

Datordomāšana (DT) ir kļuvusi par datorzinātņu izglītības stūrakmeni. Mati Tedre un viņa komanda nesen ieviesa CT2.0^[1], lai palīdzētu studentiem atšķirt tradicionālās uz noteikumiem balstītās problēmu risināšanas pieejas (CT1.0) no datu vadītajām pieejām (CT2.0), ko izmanto mākslīgā intelekta sistēmas. Tā kā sistēmās arvien vairāk ir iekļauti gan uz noteikumiem balstīti, gan uz datiem balstīti elementi, ir svarīgi, lai studenti izprastu atšķirības un spētu strādāt ar abām paradigmām.

☰ Summary >

Galvenie jēdzieni

Problēmu risināšana

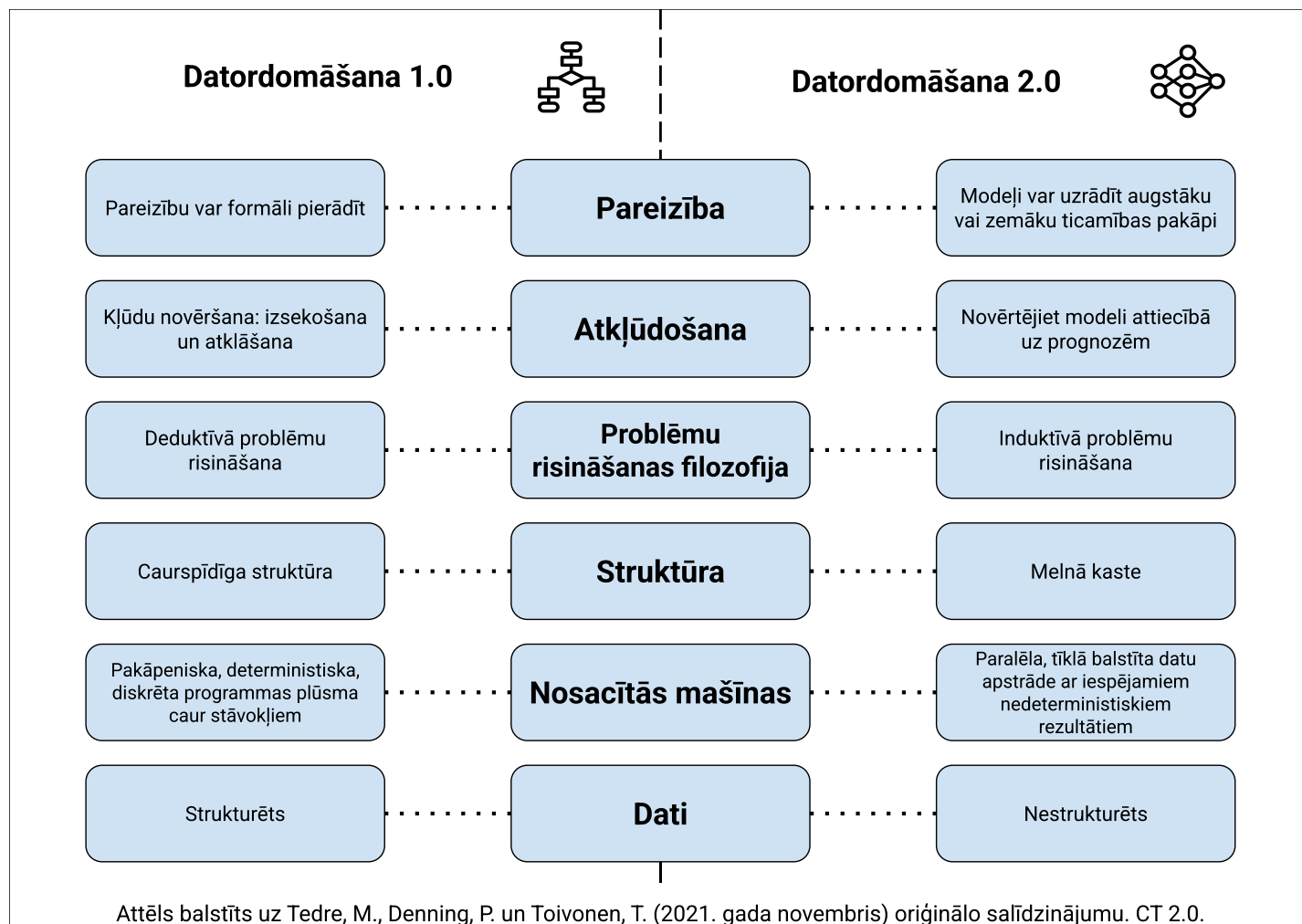
- CT1.0 problēmu risināšanā izmanto uz noteikumiem balstītas pieejas, piemēram, tās, kas tiek izmantotas Scratch un Python.
- CT2.0 piedāvā pāreju uz uz datiem balstītu pieeju problēmu risināšanā.
- Modeļa novērtēšana, datu kvalitāte un neobjektivitāte kļūst svarīga CT2.0, jo kļūdaini dati var izraisīt negodīgus rezultātus.

Pareizība

- CT1.0 parasti māca pareizību kā bināru sistēmu, kur programmas ģenerē vai neģenerē pareizus rezultātus.
- CT2.0 mēra pareizību pēc pakāpes, kur ML modeļi ģenerē prognozes un ticamības līmeņus.

Atklūdošana

- CT1.0 versijā atklūdošana ir strukturēta un caurredzama. Kļūdas tiek novērstas, soli pa solim izsekojot programmas izpildi.
- CT2.0 versijā ML modeļi ir necaurredzamas melnas kastes. Problēmas tiek atklātas, analizējot ievades un izvades datus.
- Lai to izdarītu, ir jāmaina atklūdošanas domāšanas veids, koncentrējoties uz apmācības datu uzlabošanu, parametru regulēšanu un testēšanu ar dažādu datu klāstu.



Problēmu risināšana

CT ir sistēma problēmu risināšanas izpratnei, izmantojot aprēķinus. Tradicionālā CT1.0 versija radās no agrīnajām skaitļošanas koncepcijām, izmantojot uz noteikumiem balstītu pieeju, kur datorprogrammas izpilda precīzas instrukcijas: ar precīzi definētu ievadi instrukcijas tiek izpildītas soli pa solim, lai iegūtu paredzamu rezultātu. Mācot ar CT1.0, studenti iemācās sadalīt uzdevumu apakšuzdevumos un uzrakstīt skaidras instrukcijas katram solim, pirms šīs instrukcijas tiek ieviestas tādos rīkos kā Scratch un tādās valodās kā Python. Turpretī problēmu risināšana CT2.0 versijā pāriet uz datu vadītu pieeju¹. Tā vietā, lai rakstītu skaidrus norādījumus, audzēkņi mācās vākt, tīrīt, marķēt un kārtot lielus attiecīgo datu apjomus. Pēc tam studenti izmanto šos datus, lai apmācītu mašīnmācīšanās (ML) sistēmas, lai identificētu modeļus un izveidotu modeļus, kas ģenerē prognozes un risina problēmas. Piemēram, CT1.0 vidē skolēni varētu izveidot rīku, kas klasificē kaķus, izmantojot “Ja...Tad” noteikumus par ūsām un smailām ausīm. Tomēr CT2.0 versijā viņi izmantotu daudzus kaķu attēlus, lai apmācītu modeli ar pietiekamu precizitāti. Tā kā dati ir šī procesa centrālais elements, datu kvalitāte, novērtēšana un neobjektivitāte kļūst par kritiski svarīgiem jēdzieniem: kļūdaini vai neobjektīvi datu kopumi var novest pie neuzticamiem vai netaisnīgiem rezultātiem. Mūsdienu lietojumprogrammas apvieno gan uz noteikumiem balstītas, gan uz datiem balstītas pieejas — sākot ar mākslīgā intelekta ģenerētu tekstu un attēliem un beidzot ar sejas atpazīšanas programmatūru un sociālo mediju ieteikumiem. Izpratne gan par CT1.0, gan CT2.0 dod studentiem iespēju ne tikai efektīvi strādāt ar šiem rīkiem, bet arī būt aktīviem dalībniekiem un radītājiem, nevis pasīviem patērētājiem mūsu arvien vairāk uz datiem balstītajās sabiedrībās^[2].

Pareizības izpratne

Pareizība ir svarīgs jēdziens skaitļošanā un nosaka, vai programma darbojas, kā paredzēts. CT1.0 vidē mēs bieži mācām studentiem, ka pareizība nozīmē, ka programma ir vai nu pareiza, vai nepareiza. Šī pieeja uzsver precizitāti, kur instrukcijām jābūt sintaktiski pareizām, loģiski uzrakstītām un tām jārada paredzētais rezultāts. CT1.0 raksturotās uz noteikumiem balstītās programmas pieņem augstu caurspīdīguma līmeni: katra instrukcija ir skaidri uzrakstīta un to var izsekot, kļūdas var precīzi noteikt, un labojumus var pārbaudīt un ieviest. CT2.0 versijā pareizība vairs nav fiksēta “pareizi” vai “nepareizi”. Daudzu mašīnmācīšanās modeļu rezultāti ir uz varbūtību balstītas prognozes ar dažādu ticamības līmeni^[1-1]. Piemēram, mašīnmācīšanās modelis varētu klasificēt kaķa attēlu ar 95 % ticamības pakāpi. Pat labi apmācīti mašīnmācīšanās modeļi, neskatoties uz to, ka tie ir apmācīti darbam ar lielu datu apjomu, var radīt kļūdas, īpaši ar jauniem ievades datiem. Piemēram, kaķa attēls varētu tikt nepareizi klasificēts kā suns ar 60% ticamības rādītāju. Izstrādātāji, projektējot un veidojot mašīnmācīšanās modeļus, nosaka pieņemamos pareizības līmeņus. Tas prasa izstrādātājiem rūpīgi pielāgot apmācības procesu un noteikt atbilstošus ticamības sliekšņus, lai noteiktu, vai prognoze ir pieņemama konkrētā kontekstā. Pedagogiem šī izpratnes maiņa par pareizību liek palīdzēt jauniešiem attīstīt kritiskās domāšanas prasmes, izmantojot uz datiem balstītus rīkus un mākslīgā intelekta sistēmas. Mēs varētu mudināt studentus uzdot dziļākus jautājumus: “Cik ticama ir šī prognoze, ņemot vērā jaunus datus?” vai “Kādas neobjektivitātes varētu būt apmācības datos?” Formulējot pareizību vai piemērotību CT2.0 kā nepārtrauktu novērtēšanas procesu un nepārtraukti pilnveidojot modeļus, lai uzlabotu to uzticamību reālās pasaules lietojumprogrammās, nevis kā fiksētu rezultātu, mēs sagatavojam studentus ne tikai mākslīgā intelekta rīku lietošanai, bet arī sistēmas ierobežojumu un sistēmas izvades radīto iespējamo kaitējumu atpazīšanai.

Atklūdošana

Atklūdošana ir vēl viena prakse, kas CT1.0 un CT2.0 versijās izpaužas dažādās formās. Piemēram, ja uz noteikumiem balstīta programma, kas ieviesta vai nu Scratch, vai Python valodā, nedarbojas, kā paredzēts, studenti var parādīt mainīgo vērtības, iestatīt pārtraukuma punktus vai izsekot kodam pa rindām, lai atrastu, kur kaut kas nogāja greizi. Tā kā šādās programmās ir augsts caurredzīguma līmenis, mēs varam izmantot sistemātiskas un strukturētas atklūdošanas prakses. Tomēr ML modeļi bieži tiek uzskatīti par melnajām kastēm^[3], un šī necaurredzamība apgrūtina atklūdošanu CT2.0. ML modeļi ir sarežģīti, savstarpēji savienoti tīkli ar miljardiem parametru, kas nosaka rezultātus un prognozes veidos, kurus nav iespējams izsekot soli pa solim. Ja attēlu klasifikators nepareizi apzīmē kaķa attēlu kā suni, apguvēji nevar vienkārši atrast koda rindiņu, kas izraisa kļūdu, jo tādas nav. Tā vietā atklūdošana CT2.0 ietver apmācības datu kvalitātes pārbaudi un uzlabošanu, mainīgo un parametru regulēšanu, kā arī testēšanu ar dažādiem ievades datiem, lai identificētu kļūdu modeļus (piemēram, kaķi ar smailām ausīm, visticamāk, tiks nepareizi klasificēti kā suņi). Tagad, lai veiktu atklūdošanu, skolotājiem ir jāpāriet no kļūdu atrašanas un kļūdu labošanas, lai koncentrētos uz to, kā izmaiņas datus un parametros var ietekmēt vispārējo veiktspēju.

Kāpēc CT2.0 ir svarīgs

Bez CT2.0 mūsdienu studenti paliks pasīvi patērētāji, nevis informēti dalībnieki pasaulē, ko arvien vairāk veido uz datiem balstītas mākslīgā intelekta tehnoloģijas. CT2.0 integrēšana ar tradicionālo skaitļošanas domāšanu sniedz studentiem precīzu izpratni par skaitļošanas sistēmām, tostarp par to, kā problēmu risināšana, pareizība un atklūdošana atšķiras datu vadītās sistēmās. Tas dos

studentiem iespēju kritiski izvērtēt mašīnmācīšanās modeļus, izprast, kā dati tiek izmantoti modeļu apmācībai, identificēt iespējamās neobjektivitātes un pat izveidot savus mašīnmācīšanās projektus. CT2.0 ieviešana padara skaitļošanu reālistiskāku un atbilstošāku reālajai pasaulei, piedāvājot studentiem ceļus ārpus tradicionālās programmēšanas un uz nākotnes karjeru, kurā mākslīgā intelekta prasme ir izšķiroša.

Atsauces

[Avota pdf fails](#)

1. Tedre, M., et al. (2021). CT 2.0. the-cc.io/qr21_1 ↩ ↪
2. Vartiainen, H., et al. (2021). Mašīnmācība vidusskolēniem: mācīšanās, izmantojot datu vadītu dizainu. the-cc.io/qr21_2 ↩
3. Hitron, T., et al. (2019). Vai bērni var saprast mašīnmācīšanās koncepcijas? Melno kastu atklāšanas efekts. the-cc.io/qr21_3 ↩