Primljen: 3.2.2026. Stručni rad
Prihvaćen: 16.5.2026. UDK: 004.85:004.41

AGENTNO AI PROGRAMIRANJE ZA ZADATKE STROJNOG UČENJA

AGENTIC AI PROGRAMMING FOR MACHINE LEARNING TASKS

Darko Andročec

Sveučilište u Zagrebu Fakultet organizacije i informatike

Pavlinska 2, HR 42000 Varaždin, Hrvatska

E-mail: dandrocec@foi.unizg.hr

Sažetak:

Pojava agentnih razvojnih alata utemeljenih na velikim jezičnim modelima (engl. Large Language Models, LLM) mijenja način na koji razvojni inženjeri i analitičari podataka rade s kodom. Za razliku od asistenata koji samo predlažu dovršetke koda, agentni alati poput Claude Code-a samostalno čitaju repozitorij, izvršavaju naredbe u terminalu, uređuju datoteke, planiraju nizove koraka i delegiraju potposlove specijaliziranim podagentima. Rad istražuje na koji način se takav pristup može produktivno primijeniti na zadatke strojnog učenja, konkretno na razvoj prediktivnih modela u okruženju Kaggle natjecanja. Empirijski dio rada temelji se na dvije studije slučaja: binarnoj klasifikaciji potrebe za navodnjavanjem i regresiji vremena kruga utrke u Formuli 1. Za svaki slučaj prikazani su konfiguracija projekta, tipični tijek rada s alatom, korišteni obrasci agentne suradnje te promjene metričke izvedbe modela kroz iteracije. Rad zaključuje raspravom o prednostima, ograničenjima i prenosivosti uočenih obrazaca u poslovno okruženje, posebice u kontekstu detekcije prijevara, upravljanja odljevom korisnika i drugih klasifikacijskih problema koji dijele metodološku jezgru s razmatranim natjecanjima.

Ključne riječi: *agentno AI programiranje, Claude Code, strojno učenje, Kaggle, prediktivni modeli, ansambli.*

Summary:

The emergence of agentic AI development tools based on large language models is changing the way developers and data analysts work with code. Unlike assistants that merely suggest code completions, agentic tools such as Claude Code can autonomously read repositories, execute shell commands, edit files, plan multi-step sequences, and delegate subtasks to specialised subagents. This paper examines how such an approach can be productively applied to machine learning tasks, specifically the development of predictive models in the Kaggle competition setting. The empirical part is based on two case studies: a binary classification task for predicting irrigation need and a regression task for Formula 1 lap-time prediction. For each case, the paper presents the project configuration (the CLAUDE.md file), the typical workflow with the tool, the agentic collaboration patterns used, and the changes in model performance across iterations. The paper concludes with a discussion of the benefits, limitations, and transferability of the observed patterns to a business setting — particularly in the context of fraud detection, customer churn prediction, and other classification problems that share a methodological core with the competitions discussed.

Key words: *agentic AI programming, Claude Code, machine learning, Kaggle, predictive models, ensembles.*

1. Uvod

U razdoblju od 2023. do 2026. godine alati za pomoć pri programiranju utemeljeni na velikim jezičnim modelima evoluirali su od jednostavnih dopunjavača koda (*GitHub Copilot, Tabnine*) prema agentnim sustavima koji samostalno donose niz odluka, izvršavaju naredbe i međusobno se koordiniraju (Anthropic, 2025.; Liu i sur., 2026.). Među vodećim alatima u toj kategoriji nalazi se *Claude Code* – alat tvrtke Anthropic (u konzolnom ili grafičkom načinu rada) koji omogućuje korisniku da prirodnim jezikom opiše cilj, a alat zatim u nekoliko minuta pročita relevantne dijelove repozitorija, izvrši potrebne naredbe, izmijeni više datoteka i predoči rezultate na odobrenje.

Zadatci strojnog učenja posebno su zahvalan poligon za takav pristup. Razvoj prediktivnog modela u praksi rijetko završava prvom obradom podataka i prvim treniranjem – riječ je o iterativnom postupku u kojem se mijenjaju značajke, validacijske strategije, hiperparametri i strukture ansambala, a svaka iteracija proizvodi nove zapise (engl. *logs*), grafove i metrike koje treba protumačiti. *Kaggle* natjecanja taj postupak dosta realno simuliraju: fiksni skup podataka, javna metrika, jasna razlika između validacijskog rezultata (engl. *cross-validation, CV*) i javne ljestvice (engl. *public leaderboard, LB*).

Cilj ovoga rada jest opis i kritička ocjena načina na koji se *Claude Code* može primijeniti u takvom kontekstu. Empirijski dio temelji se na dvije studije slučaja iz *Kaggle* ekosustava – binarnoj klasifikaciji potrebe za navodnjavanjem te regresiji vremena kruga u Formuli 1 – koje su provedene tijekom 2025. i 2026. godine. Studije su odabrane jer pokrivaju dvije različite vrste problema (klasifikacija na strukturiranim podatcima u poljoprivredi te regresija na sportskim podatcima) i jer omogućuju usporedbu istih obrazaca agentne suradnje u različitim domenama.

Rad ne pretendira na rigorozno znanstveno mjerenje produktivnosti – to bi zahtijevalo kontroliran eksperiment s usporednom skupinom razvojnih inženjera koji rade bez agentnog alata. Riječ je o stručnom prikazu s opisom razvojnog ciklusa, primjerima upita upućenih alatu, tabličnim prikazima napretka modela i interpretativnom raspravom. Takav je opseg primjeren stručnom članku te slijedi obrazac sličnih radova u srodnim hrvatskim zbornicima. Cilj rada bio je ostaviti *Claude Code* da samostalno predlaže rješavanje problema. Sam alat često traži od korisnika odobrenje da se potvrdi pojedina akcija, ili predlaže više akcija koje su moguće s listom redoslijeda tih akcija koje sam alat predlaže. U radu se slijedilo da su ti njegovi prijedlozi prihvaćeni. Također nisu korišteni Kaggle forumi gdje tijekom natjecanja korisnici znaju dijeliti ideje za rješavanje problema definiranih natjecanja. U pristupu je alatu prepušteno da samostalno rješava problem bez većih ljudskih intervencija ili prijedloga o načinu rješavanja. Time se što objektivnije sagledava sposobnost alata da riješi zadani problem.

Tekst je strukturiran kako slijedi. Poglavlje 2 daje teorijski okvir o agentnom AI programiranju i opisuje ključne arhitektonske elemente *Claude Code* alata. Poglavlje 3 obrazlaže metodologiju i opisuje opće tijekove rada. Poglavlja 4 i 5 prikazuju dvije studije slučaja s ostvarenim rezultatima i karakterističnim isječcima upita. Poglavlje 6 sintetizira

zapažanja iz oba slučaja u skupinu metodoloških lekcija primjenjivih i u poslovnom okruženju. Rad završava zaključkom i popisom literature.

2. Teorijski okvir

2.1. Od pomoći pri kodiranju prema agentnom programiranju

Pomoć pri kodiranju utemeljena na strojnom učenju prošla je u kratkom razdoblju tri generacije. Prva je ponudila statističko dovršavanje koda na temelju lokalnog konteksta. Druga, koju obilježavaju proizvodi kao što su *GitHub Copilot* i *Tabnine* od 2021. nadalje, oslonila se na velike jezične modele za prijedlog cijelih funkcija. Treća, agentna generacija – u kojoj se nalaze *Claude Code, OpenAI Codex CLI, Aider, Cursor Agent* i drugi – uvodi sposobnost samostalnog izvršavanja niza koraka u pravom okruženju korisnika: čitanje datoteka, pisanje izmjena, pokretanje terminalnih naredbi, instalacija ovisnosti, pokretanje testova, pa i otvaranje zahtjeva za povlačenjem (engl. *pull request*) (Liu i sur., 2026.).

Pojam agentnog programiranja ovdje se koristi u užem značenju: razvojnog tijeka u kojem programer prirodnim jezikom postavlja cilj, a alat ga ostvaruje višestrukim pozivima alata unutar omeđenog skupa dozvola. Kvaliteta takvog tijeka uvelike ovisi o trima čimbenicima – kvaliteti samog modela, kvaliteti konteksta koji mu se isporuči i kvaliteti dozvola i ograničenja koja se postave na alate kojima raspolaže.

2.2. Claude Code: ključni elementi arhitekture

Claude Code je alat naredbenog retka (CLI) tvrtke Anthropic, objavljen u veljači 2025., koji se isporučuje kao npm paket (@anthropic-ai/claude-code) ili kao samostalna instalacijska datoteka. Pokreće se iz mape projekta naredbom *claude* i otvara interaktivnu sesiju u kojoj korisnik komunicira s modelima serije Claude (*Sonnet, Opus* ili *Haiku*, ovisno o pretplati i konfiguraciji). U trenutku pisanja rada u uporabi su modeli iz Claude 4 obitelji. Kod izrade ovog rada korištena je *Claude Pro* preplata i odabran je u ovom trenutnu najmoderniji model, *Opus 4.7*.

Za potrebe ovog rada važnih je nekoliko arhitektonskih elemenata:

Datoteka CLAUDE.md – Datoteka konteksta u korijenu projekta koja se učitava automatski pri svakoj sesiji i služi kao trajni kontekst (cilj projekta, konvencije, putovi do

podataka, metrika). Liu i sur. (2026.) ističu da se ona dostavlja modelu kao korisnička poruka, ne kao sistemska, što znači da je njezino korištenje u izvođenju zadataka probabilističko, ali u praksi vrlo pouzdano ako su upute jasne.

Podagenti (engl. *subagents*) – specijalizirani agentni potprocesi s vlastitim kontekstnim prozorom, sustavnim uputama i podskupom dozvoljenih alata. Glavni agent može im delegirati omeđene zadatke (npr. pokretanje unakrsne validacije za XGBoost s određenim hiperparametrima uz povrat samo prosjeka i standardne devijacije AUC-a). Time se izbjegava trošenje glavnog konteksta na pomoćne ispise i omogućuje paralelno izvođenje više različitih zadataka (Developers Digest, 2026.).

Protokol kontekstualnog modela (MCP) – otvoreni standard kojim *Claude Code* komunicira s vanjskim poslužiteljima alata: bazama podataka, sustavima upravljanja inačicama, GitHub-om, Figma-om, Kaggle API-jem putem zaobilaznice ili lokalnim datotekama. MCP poslužitelji proširuju repertoar radnji koje alat može izvesti bez ručne intervencije korisnika.

Vještine (engl. *Skills*) – paketi uputa i pomoćnih skripti koji se po potrebi učitavaju u kontekst. Korisnik može definirati vlastite vještine koje se aktiviraju na temelju opisa.

Kuke (engl. *hooks*) – skripte koje se izvršavaju prije ili poslije korištenja pojedinih alata; korisne za zapisivanje svake datoteke s predajom rješenja (engl. *submission*) u verzijski sustav ili automatsko pokretanje provjera.

Načini razmišljanja (engl. *extended thinking, plan mode*) – eksplicitne razine planiranja u kojima alat prije izvršenja iznosi plan i čeka odobrenje. Ključno za netrivijalne zadatke poput preoblikovanja validacijske strategije.

2.3. Strojno učenje na strukturiranim podatcima i Kaggle kao kontekst

Obje studije slučaja koriste se metodama nadziranog učenja na tabličnim podatcima – području u kojem desetljećima dominiraju metode utemeljene na stablima odluka, posebice algoritmi gradijentnog pojačavanja (engl. gradient boosting) poput XGBoost-a, LightGBM-a i CatBoost-a (Chen i Guestrin, 2016.; Ke i sur., 2017.). Iako se velika pažnja u medijima posvećuje dubokom učenju, na tabličnim podatcima srednjeg opsega ovi modeli ostaju referentna mjera te se njihova kvaliteta još uvijek najučinkovitije unaprjeđuje kombinacijom (engl. *ensembling, stacking*) različitih modela.

Kaggle (kaggle.com) najveća je svjetska platforma za natjecanja u strojnom učenju, s aktivnom zajednicom od više milijuna korisnika. Njegovi tjedni i mjesečni izazovi serije Playground koriste se u akademskom kontekstu kao kontrolirani razvojni okoliš jer su skupovi podataka relativno mali (do nekoliko milijuna redaka), problem je dobro definiran, a izvedba se može jednoznačno usporediti. Kaggle stoga predstavlja prikladan okvir za prikaz primjene novih alata bez utjecaja organizacijske kompleksnosti koja postoji u stvarnim poslovnim sustavima. Ovi primjeri natjecanja su uzeti jer se modeli mogu izvoditi i na običnim računalima, tj. nije potreban GPU ili TPU poslužitelj. Za potrebe ovog rada korišteno je prijenosno računalo s procesorom Intel Core Ultra 7 255H 2.00 GHz i 32 GB radne memorije.

3. Metodologija

Rad koristi pristup višestrukih studija slučaja (engl. *multiple-case study*). Za svaki slučaj prikupljeni su sljedeći izvori podataka: konfiguracijska datoteka CLAUDE.md, dnevnik iteracija u datoteci results.md, zbirka skripti za predaju rješenja, povijest javne ljestvice na Kaggleu te slobodne autorske bilješke o tijeku rada. Pomoću tih izvora rekonstruirani su ostvareni iznosi metrike na javnoj ljestvici tijekom vremena, doprinosi pojedinih metodoloških odluka (promjena praga, prelazak na bolju validacijsku strategiju, dodavanje novog modela u ansambl) izolirani iz dnevnika iteracija te kategorija svakog upita upućenog *Claude Code* alatu (planiranje, oblikovanje značajki (engl. *feature engineering*), dijagnoza, dokumentiranje, eksperiment, predaja rješenja).

Treba naglasiti da rad ne mjeri produktivnost u smislu vremena uštede u odnosu na rad bez alata jer takvo mjerenje zahtijeva kontroliranu usporednu skupinu koja u ovom radu nije postojala. Umjesto toga, fokus je na opisu obrazaca suradnje i izolaciji onih dijelova radnog tijeka u kojima je agentni alat pružio konkretnu, evidentiranu korist.

Svi glavni koraci i izgenerirani kod može se provjeriti na javno dostupnom GitHub repozitoriju na mrežnoj adresi https://github.com/dandrocec/claude_code_kaggle. Sirovi *Kaggle* skupovi podataka i cache/trening artefakti nisu uključeni u repozitorij zbog veličina datoteke (ove datoteke ukupno imaju preko 1 Gb). Za reprodukciju rezultata potrebno je preuzeti podatke s odgovarajućih *Kaggle* natjecanja (https://www.kaggle.com/competitions/playground-series-s6e4/data i

https://www.kaggle.com/competitions/playground-series-s6e5/data) i smjestiti ih u pripadne direktorije.

3.1. Opći tijek rada s Claude Code alatom

U oba slučaja primijenjen je jednak okvir tijeka rada koji se ovdje sažeto opisuje radi izbjegavanja ponavljanja u kasnijim poglavljima. **Korak 1 – Priprema okoline:** Instalacija *Claude Code* alata, postavljanje Kaggle API ključa, preuzimanje skupa podataka i izrada projektne strukture. **Korak 2 – Izrada datoteke CLAUDE.md:** U korijenu projekta upisuju se cilj natjecanja, opis podataka, korištena metrika, validacijska strategija s fiksnim sjemenom (engl. *seed*) i nepromjenjive konvencije (npr. testni skup se ne smije koristiti za treniranje). **Korak 3 – Inicijalno istraživanje (EDA):** Upit alatu da pročita train.csv, opiše distribucije značajki, pronađe nedostajuće vrijednosti i predloži tehnike za njihovu obradu. Alat pokreće Python skripte, sažima rezultate i predlaže sljedeće korake. **Korak 4 – Osnovni model:** Brzo postavljanje LightGBM ili XGBoost modela sa zadanim hiperparametrima i validacijskim postupkom; rezultat se upisuje u *results.md* kao referentna mjera. **Korak 5 – Iterativno usavršavanje:** Kroz niz iteracija mijenjaju se značajke, hiperparametri, validacijska strategija i strukture ansambala. Za svaku iteraciju zahtijeva se eksplicitno upisivanje u *results.md* u standardiziranom formatu, čime se osigurava povijest odluka. **Korak 6 – Podagenti za usporedne eksperimente:** Za usporedbu više modela na istoj validacijskoj podjeli pokreću se podagenti – svaki rješava omeđeni zadatak i vraća samo brojčani rezultat, čime se štedi kontekst glavnog agenta. **Korak 7 – Slaganje modela i konačno rješenje:** Nakon stagnacije pojedinačnih modela kombiniraju se u slagani ansambl (engl. *stacking*) ili u ansambl miješanjem predikcija (engl. *blending*); za odabir konačnih dvaju rješenja koristi se kombinacija CV-a i javne ljestvice, uz svjesnost o rizicima precjenjivanja javnog rezultata.

3.2. Etička i metodološka napomena

Razvoj prediktivnih modela uz pomoć agentnog alata ne oslobađa korisnika od odgovornosti za točnost i ispravnost rezultata. U svim koracima koji su uključivali odluke o validacijskoj strategiji i konačnim rješenjima, autor je ručno provjerio izlaz alata.

Posebnu pozornost zahtijevali su rizici curenja podataka (engl. *data leakage*), što alat može učiniti ako nije eksplicitno upozoren u datoteci CLAUDE.md.

4. Studija slučaja 1: Predviđanje potrebe za navodnjavanjem

4.1. Opis problema

Prva studija slučaja temelji se na *Kaggle* natjecanju iz područja poljoprivrede, u kojem je cilj na temelju meteoroloških i agronomskih varijabli (temperatura, vlažnost zraka, vlažnost tla, tip kulture, faza rasta) predvidjeti binarnu oznaku potrebe za navodnjavanjem. Definirana metrika natjecanja jest točnost klasifikacije (engl. *accuracy*), s naglaskom da je razred „navodnjavanje potrebno" otprilike 1,7 puta češći od suprotnog razreda – što čini odabir praga klasifikacije netrivijalnim.

4.2. Konfiguracija i početni postav

Datoteka CLAUDE.md za ovaj projekt sadržavala je cilj, metričke ciljeve, pravila o validaciji, tehnološki stog i konvencije imenovanja. Definirani su binarna klasifikacijska zadaća s točnošću kao metrikom, polazna javna mjera od 0,8412 s ciljem od najmanje 0,8500, peterostruka stratificirana unakrsna validacija s fiksnim sjemenom te pravilo o optimizaciji praga po preklopu, a ne na cijelom skupu. Konvencije su obuhvaćale standardizirani naziv datoteke s predajom rješenja, obvezu upisa svake iteracije u *results.md* te zabranu korištenja testnog skupa za prilagođavanje modela.

4.3. Tijek iteracija i ostvareni rezultati

Razvoj se odvijao kroz devet iteracija. Tablica 1 prikazuje sažetak ključnih iteracija s napomenama o tome koji je korak u procesu obavio agentni alat samostalno, a koji uz neposredan nadzor korisnika.

Tablica 1. Sažetak iteracija — predviđanje potrebe za navodnjavanjem

Iter.	Najvažnija promjena	Public LB	Pomak	Uloga Claude Code
S001	Osnovni LightGBM, zadani hiperparametri	0,84120	—	Generiranje cjevovoda obrade (engl. pipeline), EDA sažetak
S002	XGBoost s ručno odabranim hiperparametrima	0,84298	+0,00178	Podagent za ponovnu evaluaciju
S003	Oblikovanje značajki (interakcije vlažnost × temperatura)	0,84511	+0,00213	Prijedlog i implementacija značajki
S004	Ugađanje hiperparametara (engl. tuning) alatom Optuna (100 pokušaja)	0,84803	+0,00292	Autonomno pokretanje, sažeti izvještaj
S005	Stratificirano kodiranje kategoričkih značajki	0,84850	+0,00047	Provjera curenja podataka
S006	CatBoost dodan u ansambl (prosječno miješanje predikcija)	0,84974	+0,00124	Korelacijska analiza predikcija
S007	Slaganje modela (LR meta-model nad 3 osnovna modela)	0,85029	+0,00055	Generiranje koda za slaganje modela
S008	Optimizacija praga po preklopu (umjesto 0,5)	0,85610	+0,00581	Identifikacija problema, prijedlog rješenja
S009	Konačno rješenje (S008 zamrznuta)	0,85610	0	Provjera ispravnosti i potvrda

Izvor: autorski razvojni dnevnik (results.md).

4.4. Karakteristični primjeri upita i uloga alata

Najveći jednokratni pomak ostvaren je u iteraciji S008 – optimizacijom praga klasifikacije po preklopu (engl. *fold*) umjesto korištenja standardnog praga od 0,5. Bitno je primijetiti da odluku o tom koraku nije inicirao alat samostalno, nego je on dijagnosticirao problem nakon upita korisnika u kojem je iznesena sumnja da neuravnoteženost klasa „pojede" dobitak postignut ugađanjem hiperparametara. Karakteristični upit (skraćeno):

> *CV mi je 0,861, a LB samo 0,850. Misliš li da je standardni prag 0,5 problem zbog 63/37 omjera klasa? Provjeri ovo: za svaki preklop pronađi optimalni prag na validaciji, ali bez gledanja u test skup.*

Alat je samostalno napisao pomoćnu skriptu, pokrenuo je nad predikcijama izvan preklopa (engl. *out-of-fold*), izvijestio o prosječnom optimalnom pragu (0,468) i prikazao kako se na novom pragu razlika između CV-a i LB-a smanjila. Ukupno trajanje koraka, uključujući implementaciju i validaciju, iznosilo je manje od dvadeset minuta.

4.5. Specifične prednosti agentnog pristupa u ovom slučaju

U ovom je slučaju agentni alat donio najveću korist u tri područja. Prvo, u brzom generiranju cjelovitih osnovnih implementacija (S001) koje bi inače zahtijevale otprilike sat ili dva ručnog tipkanja. Drugo, u dijagnostičkom razgovoru (S008) gdje je sposobnost alata da na temelju kratke sumnje korisnika postavi i izvede provjeru bila ključna. Treće, u održavanju discipline dokumentacije: zahtjev iz datoteke CLAUDE.md da svaka iteracija dobije redak u *results.md* praktički se svaki put automatski poštivao.

5. Studija slučaja 2: Predviđanje vremena kruga u Formuli 1

5.1. Opis problema

Druga studija slučaja odnosi se na regresijsko natjecanje u kojem je cilj predvidjeti vrijeme kruga vozača Formule 1 na temelju značajki vezanih za vozača, automobil, stazu, vremenske uvjete i strategiju guma. Definirana metrika natjecanja jest srednja apsolutna pogreška (engl. *mean absolute error*, MAE). Problem je metodološki zahtjevniji od prvog jer postoji jaka grupna struktura podataka – vremena krugova istog vozača na istoj utrci visoko su korelirana – što tradicionalni nasumični k-struki CV čini optimistički pristranim.

5.2. Konfiguracija i izazov validacije

Datoteka CLAUDE.md za ovaj projekt eksplicitno je upozoravala alat na rizik grupnog curenja podataka. Definirana je regresijska zadaća sa srednjom apsolutnom pogreškom kao metrikom, izričito upozorenje na grupnu strukturu podataka po vozaču i utrci, zabrana nasumičnog KFold-a zbog optimističkog CV-a te obvezna upotreba GroupKFold-a po race_id s pet preklopa i fiksnim sjemenom. Među konvencijama je istaknuto da svaki novi model dobiva korelacijsku analizu s postojećima te da se slaganje modela primjenjuje samo ako CV pomak iznosi 0,002 ili više.

5.3. Tijek iteracija i ostvareni rezultati

Tablica 2. Sažetak iteracija — predviđanje vremena kruga u F1

Iter.	Najvažnija promjena	CV MAE	Pomak	Uloga Claude Code
F001	LightGBM, nasumični KFold (pogrešno)	0,3214	—	Generiranje osnovice
F002	Ispravak: GroupKFold po race_id	0,4138	+0,0924*	Identifikacija pogreške, ispravak
F003	Oblikovanje značajki — relativno vrijeme guma	0,4082	−0,0056	Prijedlog značajki iz domene
F004	XGBoost dodan, ugađanje alatom Optuna	0,4011	−0,0071	Podagent za paralelno ugađanje
F005	CatBoost s rukovanjem kategorijama	0,3998	−0,0013	Podagent + korelacijska analiza
F006	Slaganje modela (Ridge meta nad 3 osnovna modela)	0,3962	−0,0036	Generiranje cjevovoda obrade za slaganje modela
F007	Revizija strategije — dvije razine validacije	0,3946	−0,0016	Predlaganje sheme nakon dijaloga
F008	Konačno rješenje (F007)	—	—	Provjera ispravnosti, dokumentacija

* Pozitivan pomak u F002 znači rast pogreške, ali se time uspostavila ispravna procjena izvedbe; F001 je davao lažno optimističnu mjeru.

Izvor: autorski razvojni dnevnik (results.md).

5.4. Karakteristični primjeri upita i uloga alata

Najznačajniji metodološki trenutak ovog projekta dogodio se u iteraciji F002. Inicijalno rješenje (F001) postiglo je nešto bolji javni rezultat nego što je CV predviđao – kombinacija koja često, paradoksalno, ukazuje na problem s validacijskom shemom. Karakteristični upit:

> *Razmisli: zašto bi LB bio bolji od CV-a? Provjeri raspodjelu race_id po preklopima u trenutnoj KFold podjeli i procijeni ima li grupnog curenja. Ako da, predloži GroupKFold po race_id i procijeni očekivanu razliku u CV-u.*

Alat je u jednom dijalogu napisao dijagnostičku skriptu, prikazao da svaki preklop sadrži primjere iz iste utrke (jasno curenje), implementirao *GroupKFold* i izvijestio o novoj, znatno manje optimističkoj mjeri. Potrebno je istaknuti dvije stvari: prvo, alat sam ne bi ovo predložio bez korisničke sumnje; drugo, kad mu je sumnja pružena, alat je u sljedećih dvadesetak minuta donio i konceptualno objašnjenje i tehničku provedbu, što je u ručnom radu često višednevni posao.

5.5. Korelacijska analiza kao prečac do odluke o ansamblu

U iteraciji F005 alat je samostalno pokrenuo korelacijsku analizu predikcija novog modela (*CatBoost*) u odnosu na već postojeće (*LightGBM, XGBoost*). Korelacija je bila visoka (0,978), što je sugeriralo da dodavanje CatBoost-a u jednostavnu prosječnu kombinaciju vjerojatno ne bi donijelo značajan dobitak. Ova vrsta analize zatim je postala dio CLAUDE.md konvencije („svaki novi model dobiva korelacijsku analizu s postojećima") i ubrzala je odlučivanje o tome koje modele uopće isplati uključiti u slaganje modela.

6. Sinteza i metodološke lekcije

Iz oba slučaja proizlazi šest zapažanja koja se mogu generalizirati izvan konteksta *Kaggle* natjecanja i prevesti u smjernice za primjenu agentnih AI alata u poslovnom razvoju prediktivnih modela.

U oba projekta najveći broj potencijalnih grešaka (curenje podataka, nedosljedna validacijska shema, izostalo dokumentiranje iteracija) izbjegnut je upravo zato što je odgovarajuće pravilo bilo eksplicitno upisano u datoteku CLAUDE.md. Pisanje te datoteke traje petnaestak minuta, ali ako je datoteka dobro pripremljena to se vidi kroz cijelo vrijeme eksperimentiranja s modelima i rješavanja problema.

U slučaju 1 najveći jednokratni dobitak (+0,00581) nije bio nikakav novi model, nego jednostavna optimizacija praga klasifikacije po preklopu. U slučaju 2 najznačajniji metodološki pomak nije bio dodavanje novog modela, nego ispravak validacijske strategije. Oba se primjera uklapaju u poznato pravilo: u prediktivnom modeliranju najveći utjecaj često nemaju modeli, nego metodologija. Agentni alat tu ne donosi samostalne prednosti, ali ubrzava i sigurno provodi promjenu jednom kada je korisnik prepoznao problem.

Sljedeća lekcija je da podagenti mogu biti mehanizam za štednju konteksta. Treniranje i evaluacija pet pojedinačnih modela mogu unutar jedne sesije zauzeti više stotina tisuća tokena ispisom dnevničkih zapisa. Delegiranjem zadatka „treniraj model X i vrati samo prosječnu metriku i standardnu devijaciju" specijaliziranom podagentu, glavni agent ostaje fokusiran na strategiju, a kontekstni prozor se ne onečišćuje sporednim sadržajem (Developers Digest, 2026.).

Standardna praksa je dodavati nove modele u ansambl dok pomažu, no često se dodaju modeli s vrlo sličnim predikcijama što tek beznačajno pomiče metriku. Ako se uvede pravilo (preko CLAUDE.md) da svaki novi model dobiva korelacijsku analizu, alat će je provoditi automatski. Ovo nije nova metodološka spoznaja – novost je u tome što ju je sada moguće „zalijepiti" za projekt u obliku konvencije koju agent uvijek poštuje. Bez sustavnog dnevnika iteracija, nakon desetog rješenja korisnik više ne zna zašto je peto bilo bolje od šestog. Zahtjev iz CLAUDE.md datoteke da svaka iteracija dobije standardiziran redak u *results.md* praktički je uvijek ispoštovan i osigurao je rekonstrukciju čitavog razvoja – što je važno i za reprodukciju u poslovnom okruženju (npr. za reviziju i regulatornu usklađenost u financijama).

Ni u jednom slučaju alat nije postavljen na potpuno autonomni način rada (bez korisničkog odobrenja). Strateške odluke – koju značajku isprobati, kada zaustaviti ugađanje, koje je rješenje konačno – donosio je korisnik. Ulogu alata najpreciznije opisuje

izraz „disciplinirani asistent koji ne zaboravlja pravila”: on osigurava tehničku ispravnost izvedbe, a interpretaciju i odluku zadržava korisnik.

Oba primjera, iako iz Kaggle konteksta, dijele metodološku jezgru s tipičnim klasifikacijskim i regresijskim problemima koje organizacije rješavaju u svakodnevnom radu, na primjer: detekcija prijevara u bankarstvu (binarna klasifikacija s neuravnoteženim klasama, slično slučaju 1), predviđanje odljeva kupaca u telekomunikacijama (također binarna klasifikacija s naglašenom važnošću kalibracije praga), predviđanje potrošnje energije ili vremena izvršenja procesa (regresija s grupnom strukturom, slično slučaju 2). Šest gore navedenih lekcija primjenjivo je gotovo u potpunosti, uz dodatak da poslovno okruženje obično zahtijeva i tri dodatna sloja koji u natjecanjima nisu prisutni: nadzor produkcijskog modela, mogućnost detaljnog objašnjenja predviđanja modela strojnog učenja (interpretacija) i mogućnost revizije svake odluke. Sva tri sloja imaju jasnu integraciju s *Claude Code* arhitekturom – kroz MCP poslužitelje za nadzor, vještine za interpretaciju i kuke za bilježenje svake izmjene.

7. Zaključak

Rad je prikazao kako se agentni AI alat *Claude Code* može produktivno primijeniti u tipičnom radnom tijeku strojnog učenja na tabličnim podatcima, oslanjajući se na dvije studije slučaja iz *Kaggle* ekosustava. Iz iskustva razvoja prediktivnih modela za predviđanje potrebe za navodnjavanjem (binarna klasifikacija) i predviđanje vremena kruga u Formuli 1 (regresija) izvedeno je šest metodoloških lekcija koje se izravno prenose u poslovno okruženje. Kod i rezultati eksperimenata se mogu preuzeti na javno dostupnom GitHub repozitoriju na mrežnoj adresi https://github.com/dandrocec/claude_code_kaggle.

Glavni je nalaz da se prava vrijednost agentnog programiranja u području strojnog učenja ne nalazi u zamjeni stručnjaka, nego u pouzdanoj, dosljednoj i brzoj provedbi metodoloških odluka koje stručnjak donosi. Datoteka CLAUDE.md, podagenti, MCP poslužitelji i kuke su mehanizmi koji projektnu disciplinu (validacijska shema, dokumentiranje iteracija, izbjegavanje curenja podataka) iz dobre prakse pretvaraju u nešto što alat disciplinirano provodi.

Ograničenja ovog rada – odsutnost kontrolne usporedne skupine, fokus samo na jedan agentni alat, izbor relativno omeđenih natjecateljskih problema – definiraju i moguće

smjerove za buduće istraživanje. Smisleno bi bilo provesti kontroliranu usporedbu *Claude Code, Codex CLI* i sličnih alata na istom *Kaggle* problemu, izmjeriti vrijeme do postizanja referentnog rezultata i analizirati strukturu pogrešaka u svakom od slučajeva.

Napomena

Prilikom izrade ovoga rada autor je koristio velike jezične modele (Claude od tvrtke Anthropic, posebno alat Claude Code) kao razvojni alat tijekom izrade obje studije slučaja opisane u radu – što je ujedno i predmet samog rada. Nakon korištenja ovih alata autor je pregledao i uredio sadržaj prema potrebi te preuzima punu odgovornost za sadržaj objavljenog članka.

Literatura

1. Anthropic (2025.). *Claude Code Documentation*. https://docs.claude.com/en/docs/claude-code/overview (21.05.2026.)
2. Chen, T.; Guestrin, C. (2016.). „XGBoost: A Scalable Tree Boosting System". U: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, ACM, str. 785–794.
3. Developers Digest (2026.). Claude Code Agent Teams, Subagents, and MCP: The 2026 Playbook. https://www.developersdigest.tech/blog/claude-code-agent-teams-subagents-2026 (21.05.2026.)
4. Ke, G. i sur. (2017.). „LightGBM: A Highly Efficient Gradient Boosting Decision Tree". U: *Advances in Neural Information Processing Systems*, vol. 30, str. 3149–3157.
5. Liu, J. i sur. (2026.). „Dive into Claude Code: The Design Space of Today's and Future AI Agent Systems", *arXiv preprint*, arXiv:2604.14228.
6. MindStudio (2026.). Beyond One-Shot Prompts: 5 Claude Code Workflow Patterns Explained. https://www.mindstudio.ai/blog/claude-code-agentic-workflow-patterns (21.05.2026.)
7. Towards Data Science (2026.). Claude Skills and Subagents: Escaping the Prompt Engineering Hamster Wheel. https://towardsdatascience.com/claude-skills-and-subagents-escaping-the-prompt-engineering-hamster-wheel/ (21.05.2026.)