KOFETKE.

09 / 10

VODNIKI O AI

Lokalni modeli, storitve v oblaku in stroški opreme

Kaj dejansko kupujete, ko izberete model ali ponudnika?



OPREMA IN PONUDNIKI

Primerjava naročnin na aplikacije, API-jev, gostovanih modelov z javno dostopnimi utežmi in lokalnega izvajanja.

SLOVENSKA IZDAJA

september 2026
Praktična uporaba AI

Viri preverjeni 21. septembra 2026.

Nakup zmogljive grafične kartice, najem grafičnega procesorja (GPU) v oblaku, plačevanje naročnine na klepetalno storitev in klicanje API-ja so štiri različne izbire. Vse lahko podjetju omogočijo uporabo AI, vendar stroške, ravnanje s podatki, vzdrževanje in nadzor nalagajo različnim ljudem. Uporabna odločitev se začne pri dejanskem opravilu, ne pri imenu družine modelov ali lestvici najboljših.

Model je naučena komponenta. Ponudnik ga upravlja ali distribuira. Aplikacija je zaslon oziroma delovni tok, ki ga ljudje uporabljajo. GPU je strojna oprema, ki lahko vzporedno izvaja veliko izračunov. API je vmesnik z obračunom po uporabi, prek katerega programska oprema ponudnika zaprosi za zagon modela. Mesečna naročnina na klepetalno storitev običajno pokriva aplikacijo za ljudi z lastnimi omejitvami uporabe; ni samodejno dobroimetje za API. Preden predpostavite, da eno plačilo pokriva drugo storitev, preberite pogoje izdelka.

Družine modelov lahko izvirajo iz podjetij v Združenih državah, na Kitajskem, v Evropi ali drugod. Izvor je lahko pomemben za pogodbeno pristojnost, gostovanje, izvozni nadzor, podporo in jezikovno pokritost. Zaradi tega model ni sam po sebi bolj ali manj zaupanja vreden, zaseben ali primeren. Ocenite točno različico, licenco, ponudnika, kraj gostovanja, pogoje glede podatkov in delovni tok. Nacionalni stereotipi so slab nadomestek za skrbni pregled.

Lokalna pot

Lokalno izvajanje modela z javno dostopnimi utežmi pomeni nalaganje njegovih uteži na delovno postajo ali strežnik v vaših prostorih. Zagon na najetem računalniku, ki ga upravljate sami, je samostojno upravljan strežnik v oblaku, ne lokalno izvajanje: nadzorujete programsko opremo, strojna oprema in njena infrastruktura pa ostajata pri dobavitelju. Lokalno izvajanje lahko zmanjša rutinsko pošiljanje navodil ponudniku API-ja za modele, vendar prinaša operativne obveznosti. Računalnik potrebuje posodobitve, nadzor dostopa, po potrebi šifriranje diska, varnostne kopije, dnevnike, fizično zaščito in načrt za odpoved. Lokalni vmesnik v brskalniku z izpostavljenimi vrati lahko postane uhajanje podatkov, če je dosegljiv iz napačnega omrežja.

Grafični pomnilnik, običajno imenovan VRAM, je pogosto takojšnja omejitev. V pomnilniku mora biti dovolj prostora za uteži modela, delovni pomnilnik za aktivni pogovor, vmesne rezultate izračunov in dodatne pomnilniške potrebe programske opreme. Dolžina konteksta in število sočasnih zahtev lahko zato nastavitev, ki deluje pri kratkem preskusu, pri običajni uporabi spremenita v počasno ali nedelujočo. Dokumentacija vLLM opisuje tako dolžino konteksta modela, vključno z navodilom in izhodom, kot velikost predpomnilnika ključ–vrednost na posamezen GPU.[7] To so izvedbeni izrazi, vendar je poslovni pomen preprost: preizkusite celotno predvideno obremenitev.

Kvantizacija shranjuje številske vrednosti z manj biti. Tako lahko model porabi manj pomnilnika, običajno s kompromisi, ki so odvisni od modela, naloge in metode. Hugging Face dokumentira, da lahko 8-bitna metoda kvantizacije prepolovi

porabo pomnilnika modela, in kvantizacijo opisuje tudi kot način dela z omejenimi računalniškimi viri.[6] To ne pomeni, da bo GPU s »pol manj« pomnilnika izvajal vsako obremenitev. Ne upošteva celotnega pomnilnika med izvajanjem, dolgih kontekstov, slikovnih vhodov, sočasnosti ali režijskih stroškov posamezne aplikacije. To je tehnika za zmogljivost, ne jamstvo za kakovost.

Lokalna postavitev ima lahko dobre razloge. Urednik bo morda moral preiskovati zaupen arhiv, ne da bi besedilo pošiljal splošnemu API-ju modela. Delavnica ima lahko nestalno povezavo. Skupina s stalno in predvidljivo uporabo lahko nadzor ceni bolj kot najeto storitev po urah. Ista postavitev je slab prvi izbor, če je nihče ne more vzdrževati, če je obremenitev neenakomerna, če mora model hkrati služiti mnogim ljudem ali če je opravilo odvisno od zmožnosti, ki jo nudi samo ponudnik.

Ocena pomnilnika, ne priporočilo za strojno opremo

Za namenoma poenostavljen primer predpostavimo, da ima model natanko osem milijard parametrov in da vsak parameter zaseda štiri bite. Samo uteži bi potrebovale **8,000,000,000 × 4 ÷ 8 = 4,000,000,000 bytes**: 4 GB v decimalnih enotah oziroma približno 3,73 GiB. To je aritmetika za predpostavljeno predstavitev, ne izmerjena poraba pomnilnika določene izdaje.

Prava datoteka lahko vsebuje tudi skalirne faktorje kvantizacije, metapodatke in vrednosti, hranjene pri večji natančnosti. Delujoča aplikacija potrebuje dodaten prostor. Grafična kartica s pomnilnikom, enakim izračunani velikosti uteži, zato ni smiselno priporočilo za nakup. Izračun tudi ne napoveduje hitrosti. Uporaben preizkus strojne opreme naloži točno obliko modela, pošlje reprezentativne dokumente, zažene predvideno število

sočasnih zahtev ter zabeleži največjo porabo pomnilnika in odzivni čas. Tako se ocena zmogljivosti spremeni v dokaz. VRAM GPU-ja, sistemski RAM in zmogljivost diska naj bodo ločena polja v delovnem listu. Del uteži je mogoče prenesti na CPU in hraniti v sistemskem pomnilniku, vendar lahko prenosi med CPU-jem in GPU-jem postanejo ozko grlo; dokumentacija vLLM izrecno opozarja, da njena pot prenosa zahteva hitro povezavo.[7] Zabeležite točno revizijo in kvantizacijo modela, nato pa preizkusite nalaganje iz hladnega stanja, primer z dolgim navodilom in izhodom ter največjo sočasnost. Za vsakega beležite porabo pomnilnika, pretečeni čas in napake, namesto da samo preverite, ali se odpre okno za klepet.

Pot v oblaku

Najem GPU-ja v oblaku prenese nabavo strojne opreme in velik del upravljanja infrastrukture na ponudnika. Uporaben je za preizkuse, sunkovite obremenitve, začasne pakete videoposnetkov ali slik ter skupine, ki potrebujejo več zmogljivosti, kot jih nudi en pisarniški računalnik. AWS-jeva stran s cenami On-Demand opisuje obračunavanje po urah ali sekundah brez dolgoročne zaveze ter razlikuje primere glede na operacijski sistem: na strani so instance s SUSE Linux Enterprise Server izrecno navedene kot obračunane po celih urah.[51] Preverite pravila obračunavanja izbrane konfiguracije, namesto da predpostavite, da se vsaka instanca v oblaku obračunava enako. Čas računanja je le del celotnega računa za storitev AI.

Resna ocena stroškov oblaka vključuje čas GPU-ja, shranjevanje, prenos podatkov, posnetke stanja, upravljane podatkovne zbirke, spremljanje, javni IP ali prehodne storitve, čas inženirjev ter

neuspele poskuse. Nižja cena na uro je lahko nepomembna, če instanca čez noč ostane zagnana ali če zahteva preveč dela za zaščito. Prekinljiva oziroma spot zmogljivost lahko stane manj, vendar lahko izgine; dolg prenos modela ali neshranjen paket potrebujeta načrt za ponovni zagon. Trenutne cenovne strani so odvisne od regije in konfiguracije, zato rokopis ne sme ohraniti ene urne številke kot trajne primerjave.

Tudi oblak ni ena sama nastavitev zasebnosti. Najeti navidezni računalnik lahko upravlja stranka, medtem ko ima upravljana končna točka AI svoje nadzore ponudnika, vloge pri obdelavi in beleženje. Vprašanje je: kam gredo navodila modelu, izhodi, naložene datoteke, dnevniki, varnostne kopije in telemetrija ter kdo lahko do njih dostopa? OpenAI navaja, da se podatki API-ja ne uporabljajo za učenje modelov, razen če stranka izrecno privoli; ista dokumentacija navaja, da se dnevniki spremljanja zlorab privzeto ustvarijo in se lahko hranijo do 30 dni.**[5]** Te izjave obravnavajte kot kontrolni seznam za dejansko storitev, ne kot splošen odgovor za vsakega ponudnika.

Uteži, licence in ponudniki

Javno dostopne uteži pomenijo, da so parametri modela na voljo za prenos pod določenimi pogoji. Ne dokazujejo, da je celoten sistem odprtokoden, da je komercialna redistribucija neomejena ali da lahko podjetje prezre pravila uporabe. OSI-jeva opredelitev odprtokodne AI ločuje podatke, kodo in parametre ter navaja, da model AI vključuje arhitekturo, parametre in kodo za sklepanje.**[4]**

Razlika je lahko konkretna. Licenca skupnosti za Meta Llama 3.1 zahteva, da nekateri distributerji ali storitve, ki vsebujejo njeno gradivo, prikažejo »Built with Llama«, njeni pogoji pa vključujejo ločen prag obsega za organizacije z več kot 700 milijoni mesečno aktivnih uporabnikov.[9] Pogoji za Google Gemma zahtevajo, da se določene omejitve uporabe vključijo v nekatere nadaljnje pogodbe, in vključujejo pravilnik o prepovedani uporabi.[11] To so primeri, ne splošna pravila. Pokažejo, zakaj mora nabavni kontrolni seznam beležiti točno različico modela, besedilo licence, predvideno uporabo, načrt distribucije in vse pogoje, značilne za ponudnika. Datoteka, ki jo je mogoče prenesti, ni pravno mnenje.

Gostovani ponudniki lahko iste ali podobne družine modelov ponujajo po različnih pogodbah. Ena pot je lahko API razvijalca modela, druga pa tržnica v oblaku, kjer podjetju račun izda ponudnik oblaka. Anthropicova dokumentacija o cenah razlikuje stroške vhodnih žetonov, zapisovanja v predpomnilnik, branja iz predpomnilnika in izhodnih žetonov ter pojasnjuje, da lahko partnerske platforme v oblaku račun izstavijo prek ponudnika oblaka.[10] Pot vpliva na evidence obračunavanja, podporo, regionalne možnosti in pogoje glede podatkov. Primerjajte pot, ki jo dejansko kupujete.

Qwen in DeepSeek na praktičnem ožjem seznamu

Qwenova dokumentacija opisuje iskanje in prenos njegovih modelov v namizni aplikaciji LM Studio.[49] To neprogramerju ponuja mogoč način za lokalno eksperimentiranje. Izbrani model mora še vedno ustrezati računalniku, njegovo licenco je še vedno treba prebrati, aplikacija z zaslonom za klepet pa samodejno ne postane povezan pisarniški pomočnik.

Dokumentacija za API DeepSeek opisuje način sklepanja, ki podpira klice orodij.[50] Klic orodja je strukturirana zahteva, naj okoliška programska oprema izvede funkcijo; model ne pridobi vašega koledarja ali bančnega računa že samo zato, ker ta zmožnost obstaja. Za delovni tok razporejanja bi morala aplikacija zagotoviti funkcijo za koledar, poverilnice in dovoljene operacije. Enako razlikovanje velja za modele drugih ponudnikov, ki kličejo orodja.

Za ožji seznam vodite ločene stolpce za točno izdajo, pot dostopa, licenco, pogoje glede podatkov, podporo orodjem in rezultate pri lastnih nalogah. Ne združujte vseh modelov, ki jih je mogoče prenesti, kot šibkejše kopije gostovanega izdelka niti vseh dragih gostovanih modelov kot pravilne izbire za vsak korak. Skromen model lahko ustrezno opravi ozko nalogo razvrščanja, medtem ko drug model obravnava zahtevno razlago dokumentov. To je hipoteza za preizkus, ne tu zatrjevana razvrstitev. Zlasti slovensko pisanje in izluščanje potrebujeta lastne primere; prepričljiv prikaz v angleščini ne dokazuje niti prvega niti drugega.

Izmišljen delovni tok: pomočnik za poizvedbe o računih

Predstavljajte si izmišljeno arhitekturno prakso Cedar Steps z dvema zaposlenima. Stranke pogosto vprašajo, ali se zaračunana postavka ujema s sprejeto ponudbo. Praksa želi pomočnika, ki poišče ustrezno ponudbo in račun, pripravi osnutek odgovora ter označi manjkajoče dokumente. Nikoli ne sme izdati vračila, spremeniti računa ali poslati sporočila brez odobritve.

Prva zamisel je API, saj je obremenitev majhna in neenakomerna. Cedar Steps vzpostavi zasebno mapo za pridobivanje informacij, ki vsebuje le podpisane ponudbe in končne račune, mape strank pa so ločene s pravili dostopa. Zaposleni postavi vprašanje v aplikaciji; aplikacija pridobi izseke izvornega gradiva, pošlje izbranemu API-ju najmanj potrebno besedilo in pripravi osnutek s citati iz izvirnih datotek. Lastnik odgovor preveri in ga pošlje ročno.

Slabo optično prebran dokument lahko na kateri koli od teh poti povzroči napačno branje zneska; članek 03 obravnava to vhodno težavo. Ta primerjava uvedbe pri tem ohranja iste odobrene preskusne dokumente in pravila pregleda, medtem ko se spreminja infrastruktura. En preizkus uporablja model na lastni infrastrukturi, eden strežnik v oblaku, ki ga upravlja stranka, in eden upravljano končno točko. Za vsakega praksa zabeleži, kam gredo besedilo in dnevniki, kdo vzdržuje storitev, kaj se zgodi ob prekinitvi omrežja ter strošek sprejetega dela, vključno s človeškim pregledom. To je predlagana primerjava, ne poročan rezultat preizkusa.

Po treh mesecih praksa pregleda dejansko uporabo. Če strošek API-ja ostaja majhen, se prihrani čas zaposlenih in je pot podatkov sprejemljiva, lahko storitev v oblaku ostane smiselna. Če pogodba s stranko zahteva, da mora vse izvorno besedilo ostati na upravljanem računalniku, lahko praksa preizkusi lokalni model z javno dostopnimi utežmi na strogo zaščiteni delovni postaji. Ne bi smela najprej kupiti strojne opreme in šele pozneje upati, da bo odkrila primeren delovni tok.

Primerjava stroškov in zmogljivosti

Izbira	
Naročnina na klepetalno storitev	
Začetni strošek	Nizek
Spremenljivi strošek	Običajno fiksen paket z omejitvami
Operativno breme	Nizko
Najprimernejše za	Ena oseba, ki pripravlja osnutke in raziskuje
Glavna past	Predpostavka, da daje pravice do API-ja ali avtomatizacije
Gostovani API	
Začetni strošek	Nizek
Spremenljivi strošek	Žetoni, orodja in včasih predpomnjenje
Operativno breme	Nizko do srednje
Najprimernejše za	Majhni, neenakomerni, povezani delovni tokovi
Glavna past	Ocenjevanje stroška na zahtevo namesto vhodnega in izhodnega obsega
GPU v oblaku	
Začetni strošek	Nizek do srednji
Spremenljivi strošek	Čas izvajanja ter shranjevanje in storitve
Operativno breme	Srednje
Najprimernejše za	Občasne večje obremenitve ali začasni preizkusi
Glavna past	Stroji ali podatkovne storitve ostanejo zagnani

Lokalni GPU

Začetni strošek	Visok
Spremenljivi strošek	Elektrika, vzdrževanje in nadgradnje
Operativno breme	Srednje do visoko
Najprimernejše za	Stalna uporaba ali strožji nadzor nad uvedbo
Glavna past	Nakup VRAM-a samo za uteži ob neupoštevanju konteksta in sočasnosti

Upravljana platforma modelov

Začetni strošek	Nizek do srednji
Spremenljivi strošek	Merjenje, značilno za ponudnika
Operativno breme	Srednje
Najprimernejše za	Skupine, ki želijo nadzore in integracije v oblaku
Glavna past	Obravnavanje platforme kot enake neposredni storitvi razvijalca modela

Obračunavanje API-ja pogosto ločuje vhod in izhod. Obsežen pravilnik podjetja, poslan ob vsaki zahtevi, lahko stane več kot kratek odgovor, ki ga vidi stranka; obsežno ustvarjeno poročilo lahko razmerje obrne. Popusti za predpomnjenje lahko pomagajo pri ponavljajočem se gradivu, vendar imata zapisovanje v predpomnilnik in branje iz njega lastne pogoje. Anthropicova dokumentacija izrecno navaja, da se žetoni za zapis v predpomnilnik zaračunajo, ko je vsebina prvič shranjena.**[10]** Napoved pripravite iz reprezentativnih navodil, pričakovanih

dolžin odgovorov, ponovnih poskusov, klicev orodij in števila dejanskih opravil na mesec. Nato dodajte omejitev porabe in prvo obračunsko obdobje primerjajte z napovedjo.

Praktično vrednotenje pred zavezo

Uporabite majhno primerjavo, ne javne lestvice najboljših. Cedar Steps lahko pripravi 20 anonimiziranih poizvedb: jasna ujemanja, manjkajoča ponudba, podvojen račun, slabo optično prebran dokument, trk imen strank, mešana slovenska in angleška opomba ter zahteva, ki skuša preglasiti navodila. Iste primere zaženite skozi izbrano aplikacijo s predvidenimi nastavitvami pridobivanja informacij in odobritve.

Zabeležite različico modela in ponudnika, po potrebi regijo uvedbe, predlogo navodila, omejitev konteksta, kvantizacijo pri lokalni izvedbi, pretečeni čas, pridobljene izvorne dokumente, izhod, odločitev pregledovalca in oceno stroška. Ocena mora ločevati dejansko pravilnost od koristne zadržanosti. Sistem, ki reče »Iz tega optično prebranega dokumenta ne morem preveriti zneska,« je lahko varnejši od sistema, ki vedno napiše gladek odgovor.

Preskusno vprašanje	Dokazi, ki jih je treba ohraniti	Pravilo za uspeh
Ali je izbral pravilne dokumente stranke?	Pridobljena imena datotek in izseki	Brez razkritja izvornega gradiva med strankami

Preskusno vprašanje	Dokazi, ki jih je treba ohraniti	Pravilo za uspeh
Ali je pravilno navedel znesek?	Slika, besedilo OCR in končni osnutek	Točno ujemanje ali izrecna negotovost
Ali je spoštoval pristojnosti?	Dnevniki orodij in evidenca odobritve	Brez pošiljanja, vračil ali sprememb evidenc
Ali ustreza zmogljivosti?	Največji sočasni zagon, kontekst in napake	Dokonča predvideno obremenitev brez napake pomnilnika
Ali je strošek razložljiv?	Evidence žetonov, orodij in infrastrukture	Dejanski mesečni strošek ostane pod dogovorjeno mejo

NIST-jevo gradivo za upravljanje tveganj AI umešča preskušanje, vrednotenje, preverjanje in validacijo skozi celoten življenjski cikel ter poziva k stalnemu spremljanju med delovanjem.[12] To je združljivo s prakso malega podjetja: vzdržujte preskusno mapo, vzorčite dejanske izhode, beležite incidente, znova preskusite po spremembi ponudnika ali modela in ohranite možnost za izhod. Najuporabnejša izbira infrastrukture je tista, ki daje pregledno delo ob sprejemljivem celotnem strošku in ne razširja tiho pristojnosti prek naloge.

Viri

Številke virov sledijo skupnemu seznamu. Povezave vodijo do izvirnih dokumentov. Pogoji uporabe in razpoložljivost storitev se lahko spremenijo.

[4] **OSI: Open Source AI Definition**
https://opensource.org/ai/open-source-ai-definition

[5] **OpenAI: API data controls**
https://developers.openai.com/api/docs/guides/your-data

[6] **Hugging Face: bitsandbytes quantisation**
https://huggingface.co/docs/transformers/quantization/bitsandbytes

[7] **vLLM: engine arguments**
https://docs.vllm.ai/en/stable/configuration/engine_args

[9] **Meta: Llama 3.1 Community License**
https://raw.githubusercontent.com/meta-llama/llama-models/main/models/llama3_1/LICENSE

[10] **Anthropic: API pricing**
https://docs.anthropic.com/en/docs/about-claude/pricing

[11] **Google: Gemma Terms of Use**
https://ai.google.dev/gemma/terms.md.txt

[12] **NIST: AI actor tasks across the lifecycle**
https://airc.nist.gov/airmf-resources/airmf/appendices/app-a-descriptions-of-ai-actor-tasks

[49] **Qwen documentation: local use with LM Studio**
https://qwen.readthedocs.io/en/latest/run_locally/lmstudio.html

[50] **DeepSeek API: tool calls**
https://api-docs.deepseek.com/guides/tool_calls

[51] **AWS EC2 On-Demand pricing and billing granularity**
https://aws.amazon.com/ec2/pricing/on-demand