

Je agent-stack verbrandt API-credits aan een taak die een classifier van 30 miljoen parameters net zo goed zelf zou kunnen doen. Elke routeringscall, elke rerank, elke moderatiestap gaat naar een frontier-model tegen frontier-prijzen, en de latency stapelt zich op terwijl de rekening oploopt. Je bent al een tijd van plan het op te lossen. Dan vreet het volgende debat ("fine-tunen of toch maar RAG erbij?") opnieuw een sprint op, en er gaat niets live.

Het gangbare advies is om eerst ML grondig te leren: volg de cursus, leid de gradiënten af, kom terug als je vanaf nul een netwerk kunt trainen. Dat faalt, omdat je geen paper wilt publiceren. Je wilt één klein, eigen model in productie zetten naast een LLM en het werkend houden.

ML for Agent Builders is geschreven voor dat gat. Het reikt je BUILD aan (een loop, geen labproject) voor elk model waarvan je agents afhankelijk zijn: beslis bouwen vs kopen vs prompten met een matrix in plaats van een wekelijkse discussie, richt op de metric vóór de architectuur, train en evalueer als product, en inspecteer drift voordat die het systeem stilletjes laat verslechteren. Het frontier-model orkestreert; jouw smalle modellen voeren uit. BUILD is hoe je ze hun plek laat verdienen.

Je leert om:

- Bouwen vs kopen vs prompten te beslissen met één matrix, niet met drie debatten per sprint.
- Fine-tunen of RAG te kiezen op capaciteit en kosten, en precies te weten wanneer je ze combineert.
- Synthetische data snel te genereren en nóg sneller te verifiëren, met waarborgen die de faalmodi vangen.
- Een eval-harnas op te zetten dat vóór elke deploy een regressiesuite draait, met echte ship-gates.
- Een driftbudget te bepalen dat je vertelt wanneer je hertraint, wanneer je terugrolt, en wanneer je niets doet.

Dit is geen data science voor toeschouwers en geen MLOps voor een platformteam van vijftig man. Het is ML van operatorniveau: protocollen die je draait met een model in productie, een latency-budget dat pijn doet, en een agent bij de hand om te helpen labelen, genereren en testen.

Het resultaat is concreet. Je huurt geen frontier-compute meer voor klussen die een klein eigen model goedkoper, sneller en onder jouw controle doet, en je levert modellen op die je kunt verdedigen, monitoren en bewust kunt uitfasen.

Als jij productiesystemen bouwt en klaar bent met de keuze tussen retailprijs betalen voor elke inferentie en doen alsof het fine-tune-debat zichzelf wel oplost, begin dan hier.

## drs. Len P. van der Hof

Hij bouwt besturingssystemen voor het strategische zelf. Behandelt ondernemerschap, AI en machine learning, marketing, filosofie, psychologie en gezondheidsoptimalisatie als één engineeringprobleem. MSc in Strategic Entrepreneurship, Rotterdam School of Management, Erasmus Universiteit.

# ML voor Agent-Bouwers

Auto-training, fine-tuning en eval-loops voor LLM-gedreven systemen



Len P. van der Hof

Systems for the Strategic Self