+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 05/04/2026

Sign up

DSPy Training: Managing LLM Costs

3 days (21 hours)

Overview

DSPy enables the design of more reliable and cost-effective LLM pipelines by replacing “intuitive” prompts with a programmable, measurable, and optimizable approach. You will learn how to reduce token costs and latency while improving quality for use cases such as RAG, structured extraction, and domain-specific assistants.

The training focuses on cost control: model selection, context reduction strategies, caching, routing, and automatic optimization of prompts/strings via DSPy modules. Each concept is linked to concrete metrics (tokens, time, error rate) and quality criteria.

100% hands-on approach: guided workshops, reproducible demos, and a mini-project. Deliverables: a repository of DSPy examples, an evaluation grid, and an optimized pipeline (baseline vs. optimized version) with a cost/quality report.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Instrument and measure the costs, latency, and quality of an LLM pipeline.
- Build DSPy modules (Predict, ChainOfThought, RAG) and combine them.
- Define datasets, metrics, and non-regression tests.
- Automatically optimize prompts and parameters using teleprompting.
- Implement caching, model routing, and context reduction.

Target Audience

- Python developers integrating LLMs into production
- Data scientists / ML engineers
- Tech leads and application architects
- Product engineers working on assistants/RAGs

Prerequisites

- Proficiency in Python (functions, classes, environments)
- Basic understanding of REST APIs and JSON formats
- Basic understanding of LLMs (tokens, context, temperature)
- Basic knowledge of Git and the command line

Technical prerequisites

- Computer with 16 GB of RAM recommended (8 GB minimum)
- Windows (WSL2 recommended), macOS, or Linux
- Python 3.10+, venv/poetry, and an editor (VS Code/PyCharm)
- Access to an LLM API (key provided by your organization) and a stable internet connection

Our DSPy training program: mastering LLM costs

[Day 1 - Morning]

DSPy Fundamentals and LLM Cost Measurement

- Understanding DSPy’s positioning: declarative programming of LLM pipelines and optimization
- Identifying cost drivers: tokens (prompt/completion), latency, error rate, retries
- Setting up instrumentation: token counting, duration, estimated cost per request and per scenario
- Define quality/cost metrics: accuracy, coverage, hallucinations, cost per success
- Hands-on workshop: Instrument a mini DSPy pipeline and generate a cost/latency/quality table.

[Day 1 - Afternoon]

Reducing tokens: signatures, compact prompts, and constrained outputs

- Design DSPy signatures (inputs/outputs) to limit ambiguity and verbosity
- Apply compression strategies: short instructions, minimal examples, noise removal
- Constrain the output: strict formats (JSON), required fields, maximum length
- Implement safeguards: output validation and targeted follow-ups rather than a complete re-prompt

- Hands-on workshop: Refactor a “long” prompt into a DSPy signature + validation, then measure the token savings.

[Day 2 - Morning]

DSPy optimization: compilation, teleprompting, and model selection

- Structuring a DSPy program: modules, chaining, separation of extraction/reasoning/writing
- Understanding DSPy compilation: objectives, datasets, metrics, and cost constraints
- Using teleprompting to improve quality at constant cost (or reduce cost at constant quality)
- Model routing strategies: small model by default, scaling up to a larger model in case failure
- Hands-on workshop: Compile a DSPy module on a dataset and compare cost/quality before and after.

[Day 2 - Afternoon]

Reducing calls: caching, batching, and retry control

- Implement a cache (key by signature + parameters + version) and manage invalidation
- Reduce calls via batching and grouping of similar requests
- Limit retries: backoff, thresholds, error classification (transient vs. logical)
- Detect and fix costly loops: unnecessary retries, over-chaining, redundant prompts
- Hands-on workshop: Add caching + retry policy and measure the impact on a high-volume flow.

[Day 3 - Morning]

Quality under constraints: evaluations, tests, and budgets

- Building a test suite: nominal cases, boundary cases, “trap” data, and acceptance criteria
- Implementing automated evaluations: scoring, thresholds, regression testing, and monitoring over time
- Introducing budgets: max cost per request, max cost per session, timeouts, and controlled degradation
- “Fail fast” approach: intermediate validations to stop non-compliant processes early
- Hands-on workshop: Create a suite of evaluations + cost budget and integrate regression testing.

[Day 3 - Afternoon]

Deployment: observability, governance, and continuous optimization

- Implement operational observability: structured logs, traces, costs per endpoint and per client
- Set up alerts: cost drift, increased latency, decreased quality, increased failure rate
- Versioning prompts/signatures/compilations: rollbacks, A/B tests, canary releases
- Continuous optimization plan: weekly review of top costs, prioritization, and quick wins
- Hands-on workshop: Deploy a “budget-aware” DSPy pipeline with a cost/quality dashboard and rollback strategy.

Target Companies

This training is intended for both individuals and companies, large or small, wishing to train their teams in new advanced IT technologies or to acquire specific business knowledge or modern methods.

Entry-level assessment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency in various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Practical Course: 60% Practical, 40% Theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the trainer, supported by examples and reflection sessions, and group work.

Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been properly acquired.

Certification

A certificate will be issued to each trainee who has completed the entire training program.