Forecasting and Nowcasting with Text as Data

Module 2 - Introduction

Renato Vassallo

April, 2026

Barcelona School of Economics

All views expressed here and any remaining errors are my own.

Module details

- Applied 8-hour module focused on implementation and decision-making
- Not a foundational course: emphasis on best practices, evaluation, and real-world applications
- Office hours available upon prior coordination
- Course materials and updates: course website

About me

Background

- M.A. in Economics, former DSDM student, and currently Ph.D. student in Economics
- Interests: causal inference, machine learning, text as data, and macroeconomic policy

Contact information

- : renato.vassallo@bse.eu
- : renatovassallo.github.io

Goal: share what I know and make it useful for you

What can we already do with LLMs?

- Math and analytical reasoning
- Everyday Q&A, brainstorming
- Writing and editing
- Coding and repo-level work

The constraint is no longer computation, but formulation.

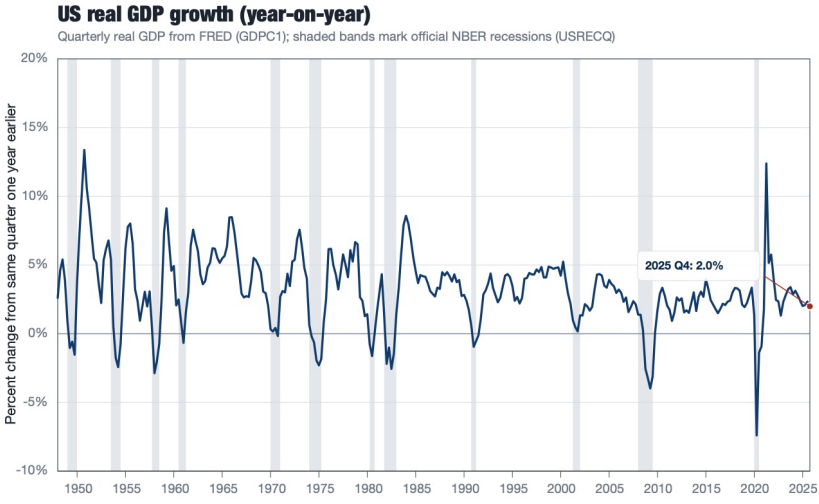
Source: Nicola Borri (2026), LinkedIn

Example 1

US GDP Growth and recessions

- Ask Codex to download US real GDP data, compute YoY growth, and plot it
- Add official NBER recession periods (Codex needs to find the recession dates)
- Produce a publication-quality figure

Output: US GDP Growth and recessions



Sources: FRED series GDPC1 and USRECQ. YoY growth = 100 x (GDPC1_t / GDPC1_t-4 - 1). Recession shading follows the official NBER chronology as distributed by FRED.

Example 2

Mincer wage equation with Python

- Data: wage1 from the wooldridge package
- Model: log(wage) on education, experience, experience squared, and a female dummy
- Regression output: `statsmodels` table saved as `mincer_table.tex`

Output: Mincer wage equation

Table 1: Results: Ordinary least squares

No. Observations:	526	Prob (F-statistic):	2.03e-56
R-squared:	0.400	Adj. R-squared:	0.395

	Coef.	Std.Err.	t	P> |t|	[0.025	0.975]
Intercept	0.3905	0.1022	3.8204	0.0001	0.1897	0.5913
educ	0.0841	0.0070	12.0941	0.0000	0.0705	0.0978
exper	0.0389	0.0048	8.0667	0.0000	0.0294	0.0484
expersq	-0.0007	0.0001	-6.3888	0.0000	-0.0009	-0.0005
female	-0.3372	0.0363	-9.2834	0.0000	-0.4085	-0.2658

Notes: OLS estimates. Standard Errors assume that the covariance matrix of the errors is correctly specified.

Paul Goldsmith-Pinkham's "Road to Enlightenment"

Adapted from Paul Goldsmith-Pinkham (2026)

Level	What it looks like
0	Copy from ChatGPT in a browser, paste into your editor
1	IDE offers inline completions and inline chat
2	"Agent mode" in the IDE: the model can read files, run tests, refactor
3	Dedicated coding agents: Claude Code, OpenAI Codex, Gemini CLI
4	Agents get tools: web search, browsing, APIs, orchestration
5	Teams of agents running for longer in containers

Most of us are still at level...

So what is left to learn?

LLMs can generate outputs, but:

- Are they reliable?
- How do we evaluate them?
- How do we scale them across countries and time?

This course focuses on:

- structured pipelines
- evaluation and metrics
- decision rules and trade-offs

Module structure

Session 1: From text to signals (2h)

- Embeddings and similarity
- Learning with limited supervision

Session 2: From signals to decisions (2:30h)

- Fine-tuning and evaluation metrics. Applications
- **In-class assignment**

Session 3: Mixed-frequency methods (2h)

- MIDAS intuition and ML extensions. Applications

Evaluation

In-class assignment (20% of the course)

- Work in groups of up to 4 members
- Duration: 45-60 minutes, followed by a brief 5-minute presentation per group
- You will have access to three text corpora
- Select one corpus, construct a text-based signal using methods from Sessions 1-2, and apply it to a specific task (event detection, classification, monitoring, or forecasting)

Important notes

- Materials are designed for teaching and illustration, not as production-ready pipelines.
- We assume limited/costly access to GPUs, cloud computing, and advanced LLMs.
- **Focus**: efficient, accessible methods and careful evaluation.
- LLMs can accelerate workflows, but do not replace rigorous modeling and validation.

Your comparative advantage is no longer coding. It is framing problems, evaluating outputs, and making decisions.