# Fine Tuning: tutorial

Georgy Kalashnov

# What is finetuning

- Large language model is loosely speaking a **regression model** predicting the **next word**, let us call it $f_\theta(X)$, where $X$ is text so far
- Large language model are fitted with gradient descent, which is **numerical optimization** trying to find a local optimum of logit loss.
- We can try to use it as a **starting point** and fit $Y' = g(f_\theta(X')$, for our own dataset $(Y', X')$.
- So, **finetuning** is just running a regression on text

# Three ways:

- Using OpenAI API
- Using huggingface.io Python library, computing on:
  - Rented hardware at vast.ai
  - Using your own laptop (can be feasible using a small model for a small task I demonstrate)

# Why a CS person may want to do it

- **Condense** a more complex model output to a more cheap one
- Finetune the **format/sentiment** of the output
- Sometimes **improve** on few-shot prompts

# How can it be useful to an economist

- Text as output.
  - We can test if the texts are **different** at all between treatment and control group (Ludwig et al., 2017)
  - With a more structural approach we can say even more Modarressi et al. (2025)
- Text as a control variable: we can fit a **propensity score** model $\hat{e}(X)$ with text or do outcome **regression adjustment** with it $\hat{m}(X)$. Or combine in an AIPW approach.

# Example: Failures in Contingent Reasoning

- This is an experiment on contingent reasoning Martínez-Marquina et al. (2019).
- There were two treatment groups. In the end of experiment the subjects were asked to give an advice to future subjects.
- I will try to answer if the treatment had any effect on text

| Treatment | Advice length |
|---|---|
| Deterministic | 168.2 |
| Probabilistic | 256.3 |

# Fitting and using the model

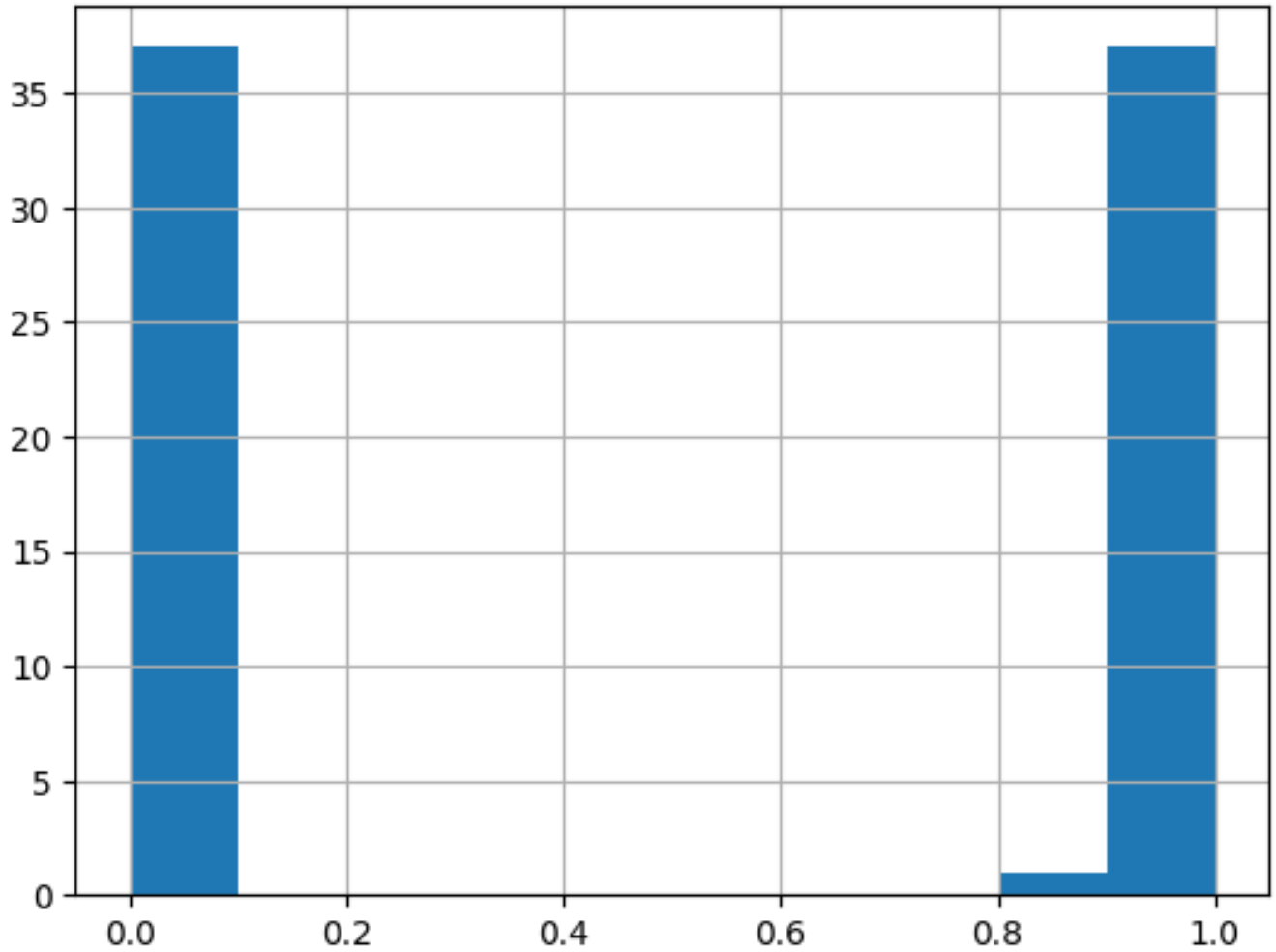
We prepare dataset:

- X: The advice.
- Also useful to use include a **preamble to a prompt**, e.g. "You are a classifier. Given the following advice text, predict whether it is 'Deterministic' or 'Probabilistic'."
- Y: Either a word 'Deterministic/Probabilistic', or a binary 0/1

Then we fit and predict the model on holdout dataset
Note on priming the model with labels or preamble: we need to make sure the propensity model fits to data, rather than to their prior, of what "Deterministic" is. However setting a prior may improve quality of the fit.

# Result: distribution of scores

Probability of correct classification: 0.84



# The use of the model

As an outcome:

- On a hold out set, test if the probability of correct classification is greater than $\mathbb{P}[W]$ Ludwig et al. (2017)

If using as a propensity score model:

- It is a good idea to calibrate the model on a hold out set ˆ $\mathrm{E}_{holdout}[\hat{f}_\theta(X_i)]$: as we see, the model tend to overfit

- **Note**: if we calibrate we could use few shot prompt engineering instead, the only problem with prompt engineering is poor calibration

# Examples of misclassified outputs

Some outputs, where the model was not that sure:

- "No advice, no idea what I'm doing."
- "I would always choose the lowest price because you don't want to overpay."

# Further reading

- Code: `https://github.com/kalashnov/text_in_econ`
- Open AI fine tuning documentation and advice: `https://platform.openai.com/docs/guides/fine-tuning`
- A comprehensive guide for fine-tuning at large scale: `https://huggingface.co/spaces/nanotron/ultrascale-playbook?section=high_level_overview`

*Ludwig Jens, Mullainathan Sendhil, Spiess Jann*. Machine-learning tests for effects on multiple outcomes // arXiv preprint arXiv:1707.01473. 2017.

*Martínez-Marquina Alejandro, Niederle Muriel, Vespa Emanuel*. Failures in contingent reasoning: The role of uncertainty // American Economic Review. 2019. 109, 10. 3437–3474.

*Modarressi Iman, Spiess Jann, Venugopal Amar*. Causal Inference on Outcomes Learned from Text // arXiv preprint arXiv:2503.00725. 2025.