
Causal Inference Libraries: What They Do and What I'd Like Them to Do
Kevin Klein, QuantCo

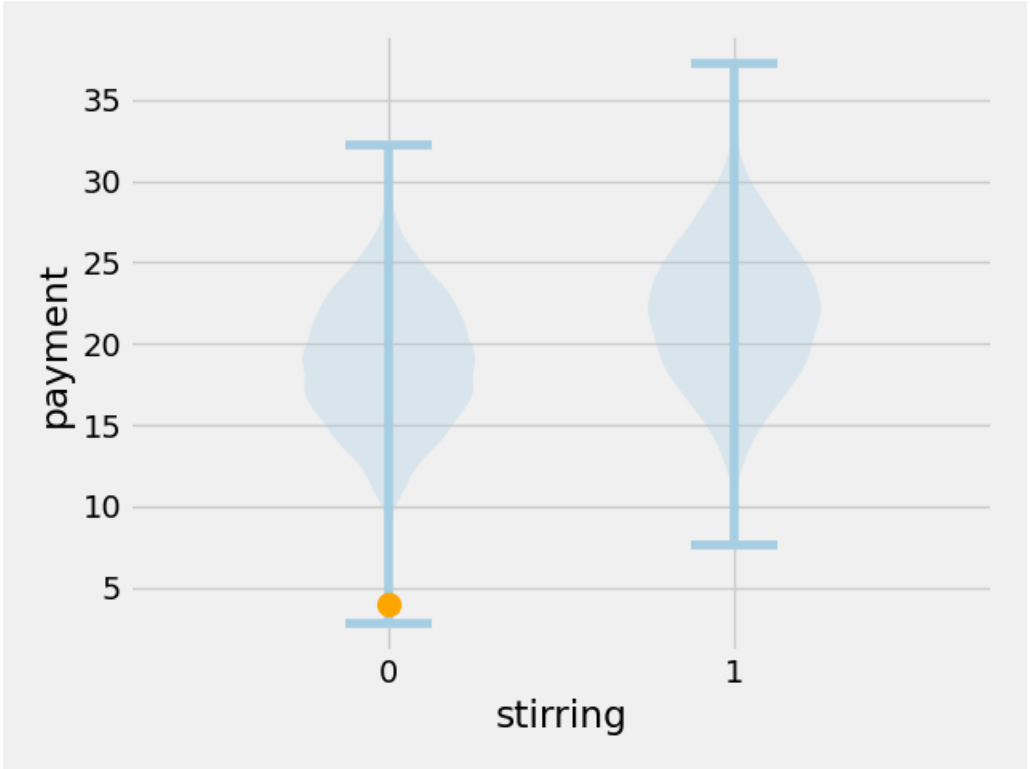
Agenda

1. Why care about Causal Inference?
2. Why care about heterogeneity?
3. How can we estimate heterogeneous treatment effects on paper?
4. How can we estimate heterogeneous treatment effects in practice?
5. What am I missing from `EconML` and `CausalML`?

Risotto

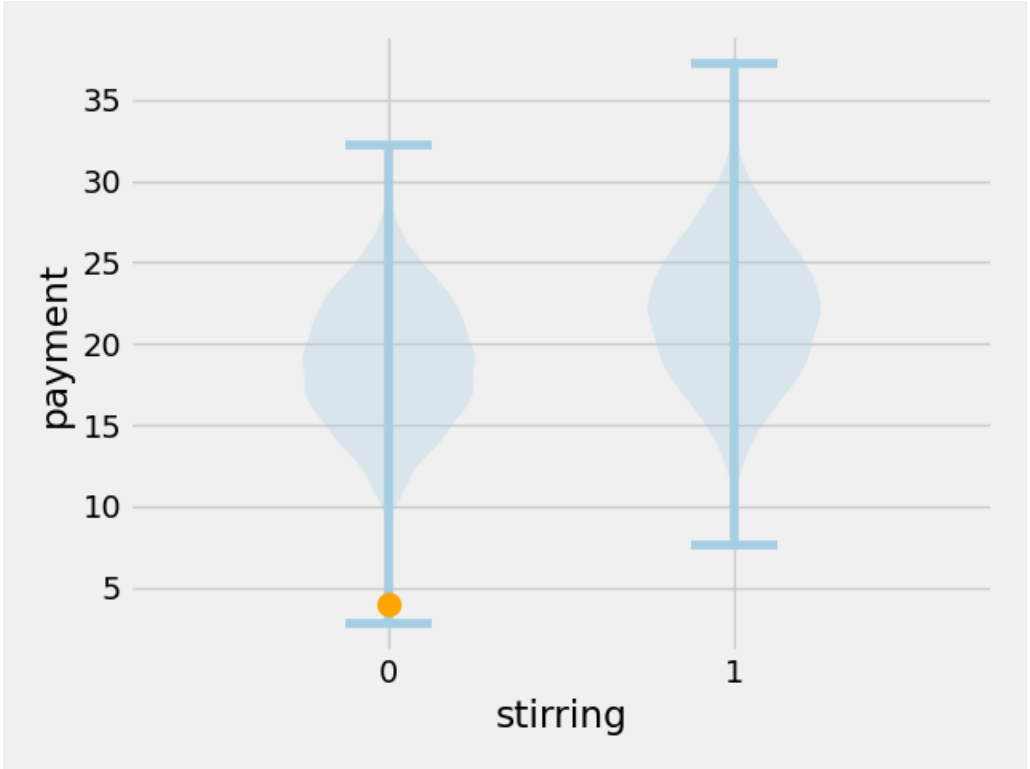
- Risotto can either be prepared
  - in a laborous and delicate fashion, involving a lot of **stirring** or
  - in a cut-throat, cantine style fashion, **not** involving a lot of **stirring**
- Consumers of risotto are **free to decide how much they pay** for their risotto.
- Naturally we wonder: **should we be stirring**?

1. Why care about Causal Inference?

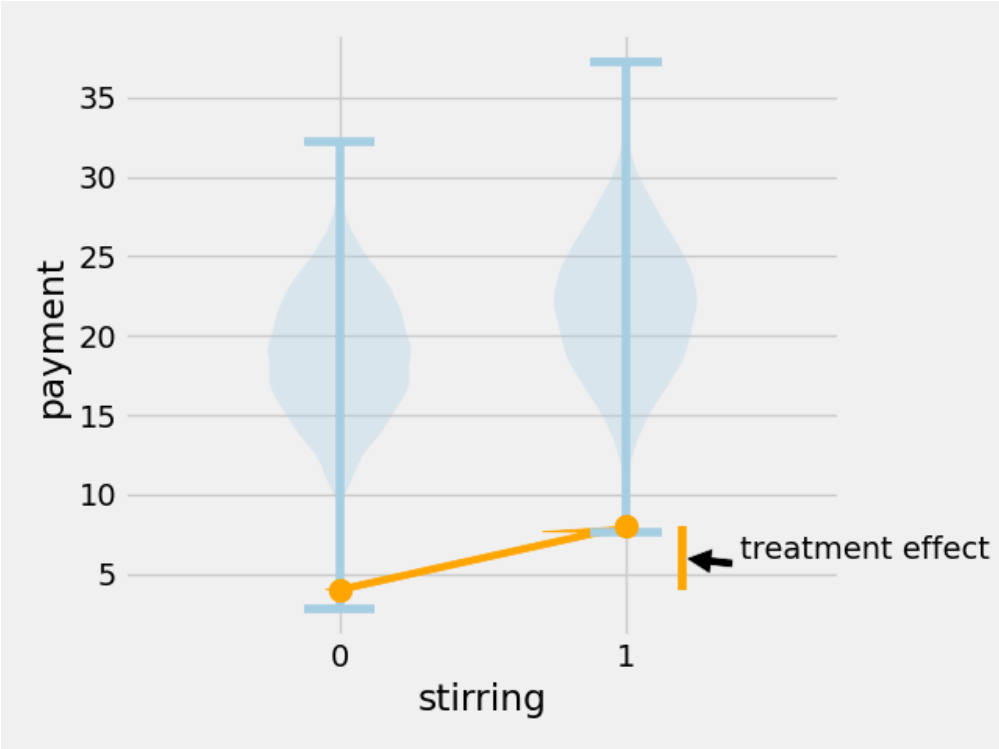

35
30
25
20
15
10
5
payment
0
1
stirring

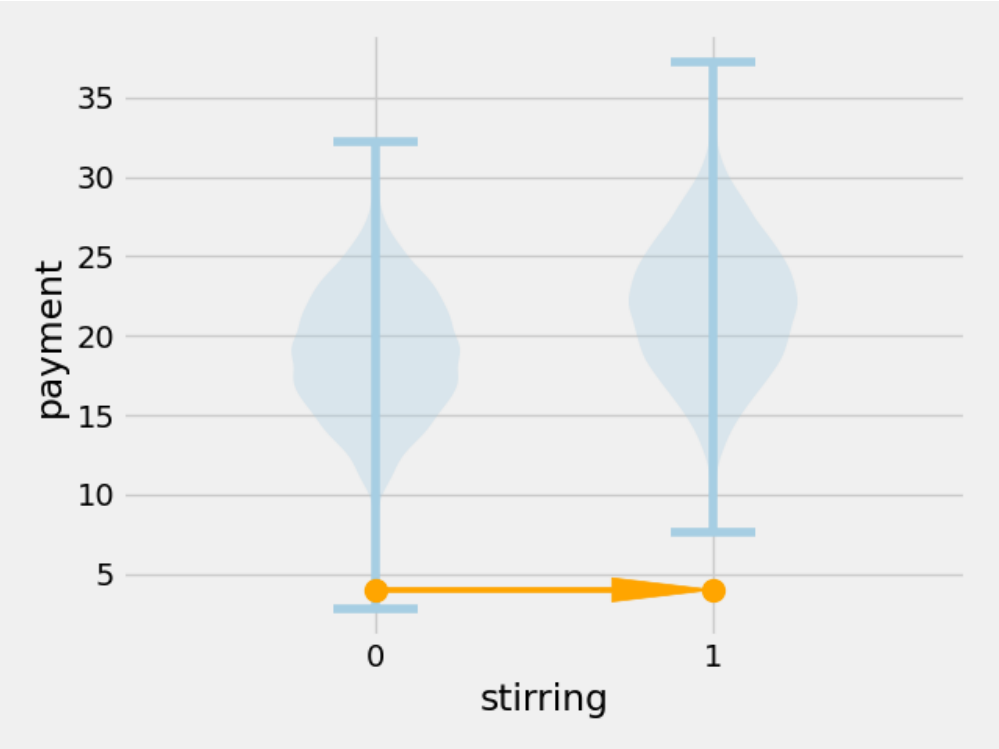
Interventions

What happens when we intervene on a data point from the left, i.e. `stirring = 0`, and now - keeping everything else unchanged - make sure that the risotto is stirred, i.e. `stirring := 1`?

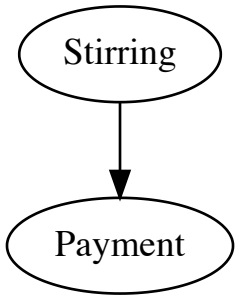


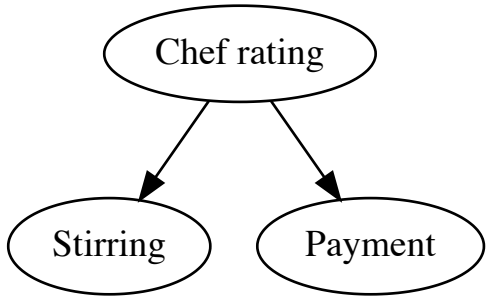
What happens if we intervene?

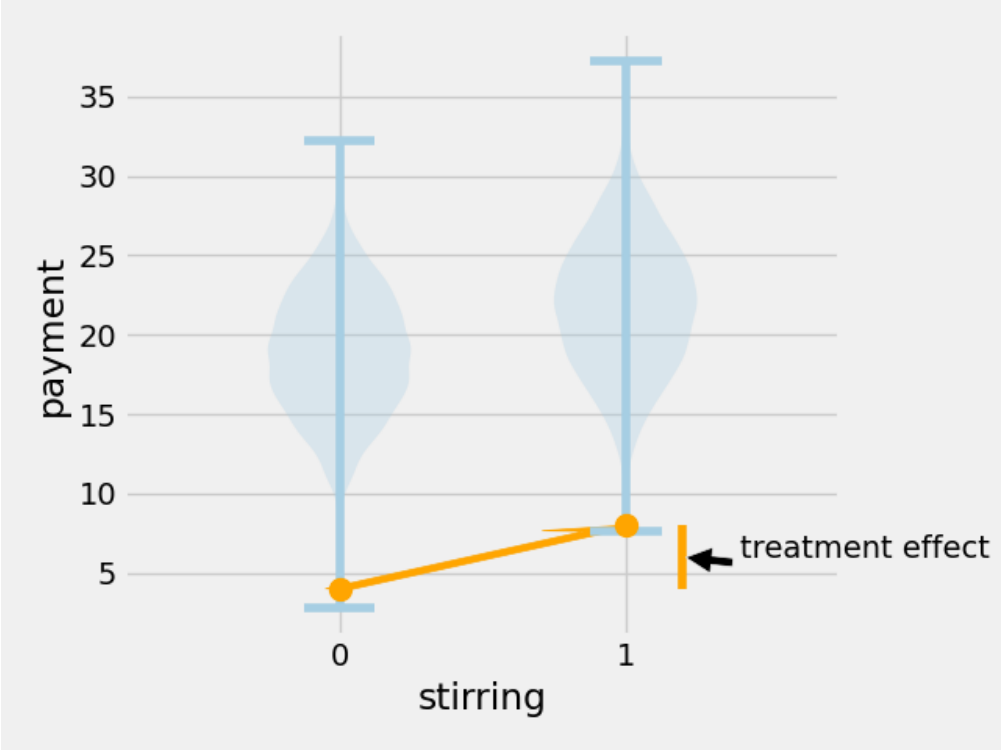


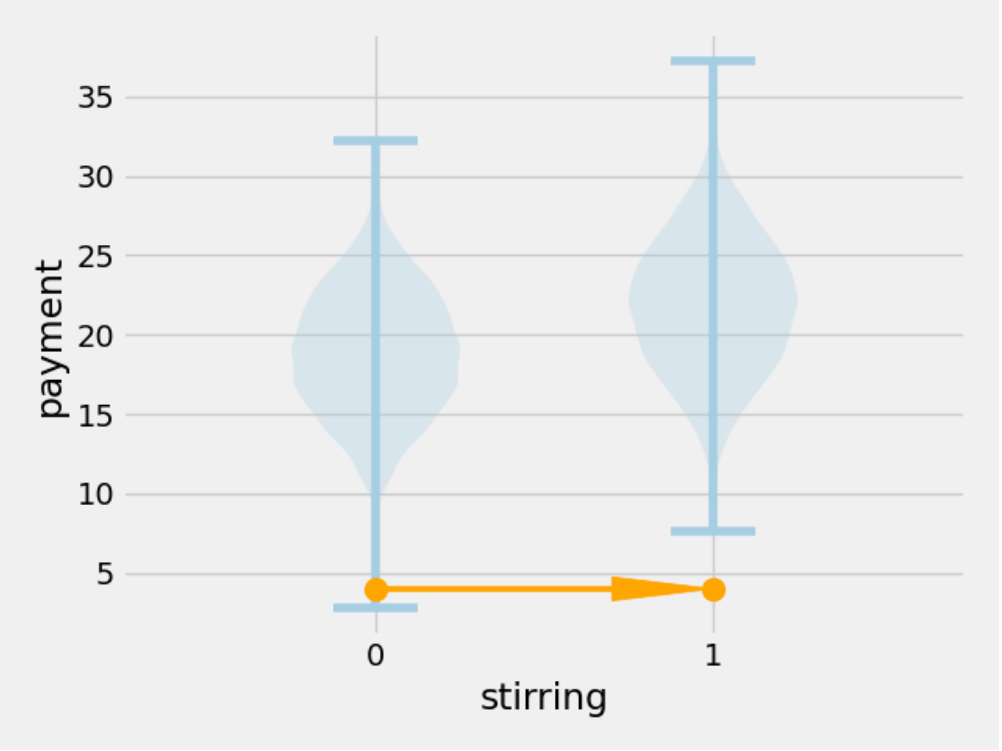


It depends

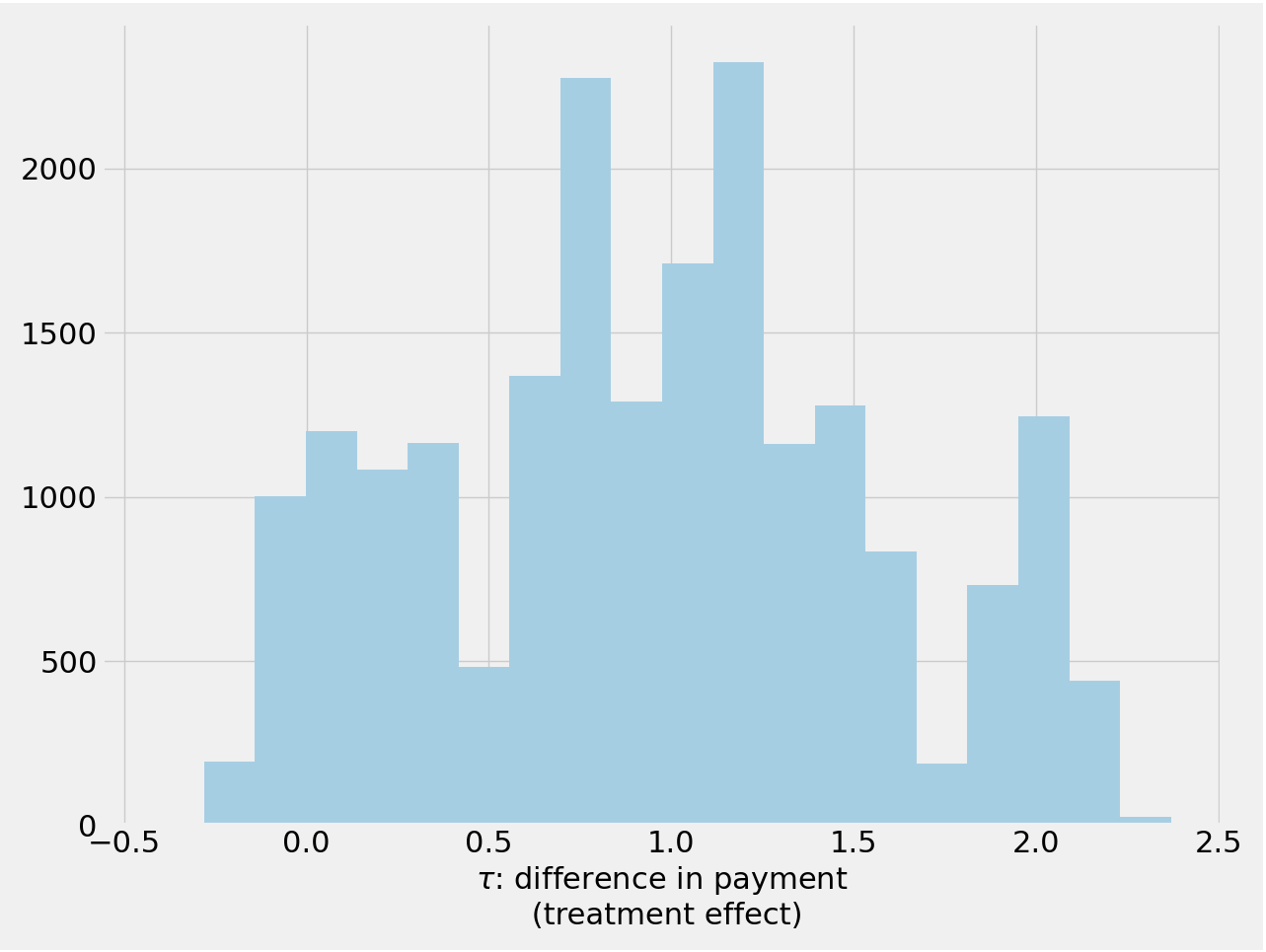

Stirring
Payment


Chef rating
Stirring
Payment


35
30
25
20
15
10
5
payment
0
1
stirring
treatment effect


35
30
25
20
15
10
5
payment
0
1
stirring

2. Why care about heterogeneity?



Because the world is heterogeneous

- Every observation of the histogram corresponds to a consumption of risotto.
- In a homogeneous setting, we would observe the same treatment effect for all observations.

From treatment effect to policy

From treatment effect to policy

$$\pi(X) := \begin{cases} \text{stir} & \text{if } \hat{\tau}(X) \geq 1\text{USD} \\ \text{don't stir} & \text{otherwise} \end{cases}$$

3. How can we estimate heterogeneous treatment effects on paper?

The fundamental problem of Causal Inference: Desire

Age of consumer	...	Non-stirred outcome/payment	Stirred outcome/payment	Individual treatment effect
28	...	21	21.8	.8
10	...	12	12	0

If we had the following kind of information, everything would be nice and easy.

Unfortunately, we don't.

The fundamental problem of Causal Inference: Reality

Age of consumer	...	Non-stirred outcome/payment	Stirred outcome/payment	Individual treatment effect
28	...	21	?	?
10	...	?	12	?

What now?

- We can't know the Individual Treatment Effect (ITE).
- Yet, we can define an estimand, the Conditional Average Treatment Effect (CATE), which we can actually estimate:

$$\tau(X) := \mathbb{E}[Y_{\text{stirring}} - Y_{\text{no stirring}} | X]$$

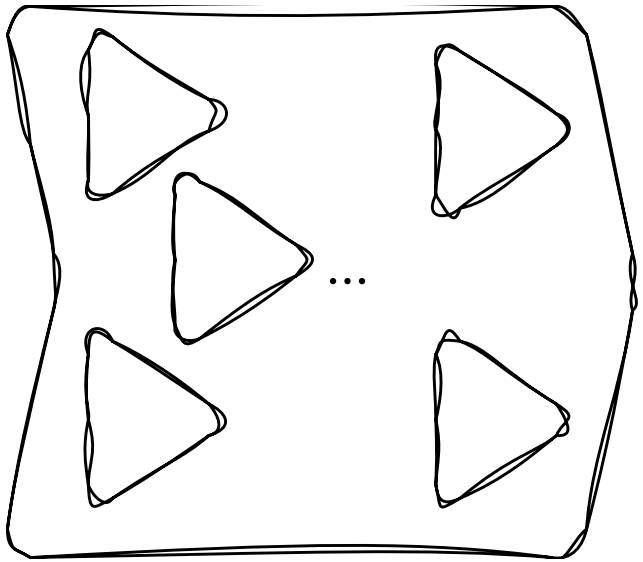
- In most of the literature 'CATE' and 'heterogeneous treatment effect' are used as synonyms.

Conventional assumptions for estimating heterogeneous treatment effects

- Positivity/overlap
- Conditional ignorability/unconfoundedness
- Stable Unit Treatment Value (SUTVA)

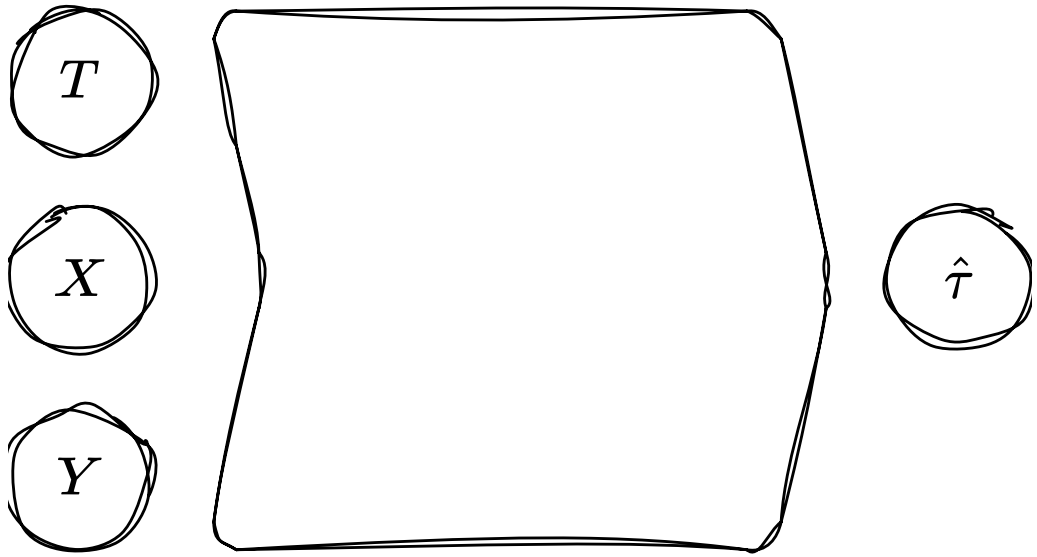
A randomized control trial usually gives us the first two for free.

For more information see e.g. Athey and Imbens, 2016.



MetaLearners

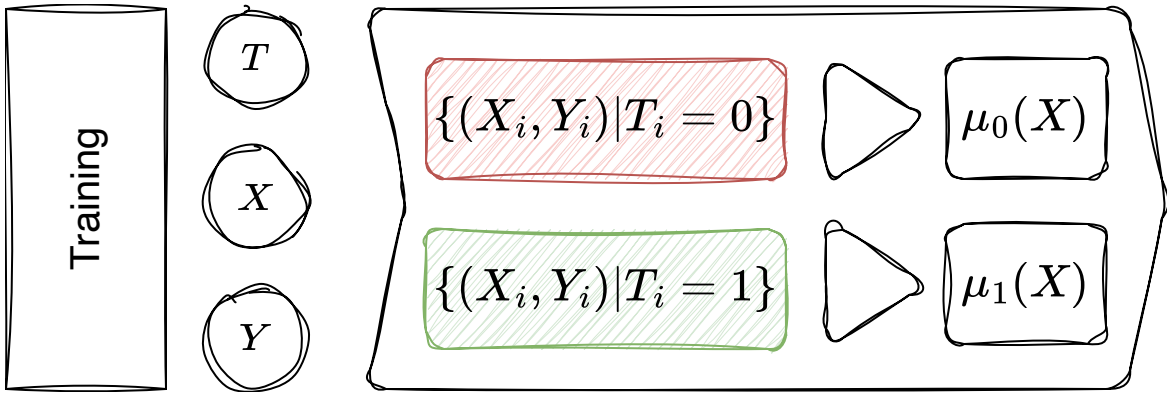
- MetaLearners are **CATE models** which rely on typical, **arbitrary machine learning estimators** (classifiers or regressors) as **components**.
- Some examples include the S-Learner, T-Learner, F-Learner, X-Learner, R-Learner, M-Learner and DR-Learner.



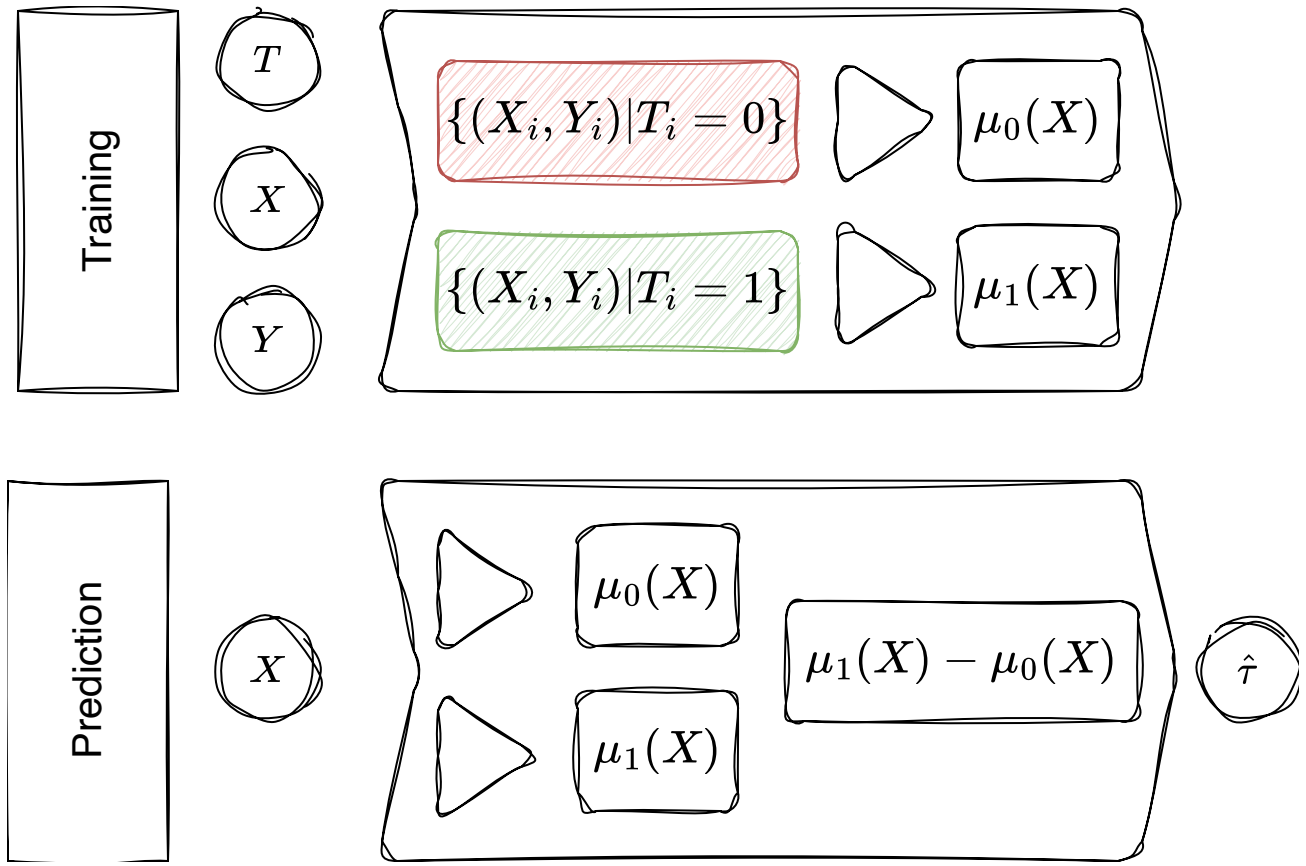
MetaLearners

- T: Treatment assignments
- X: Covariates/features
- Y: Observed outcomes
- $\hat{\tau}(X)$: Estimate of the heterogeneous treatment effect/CATE

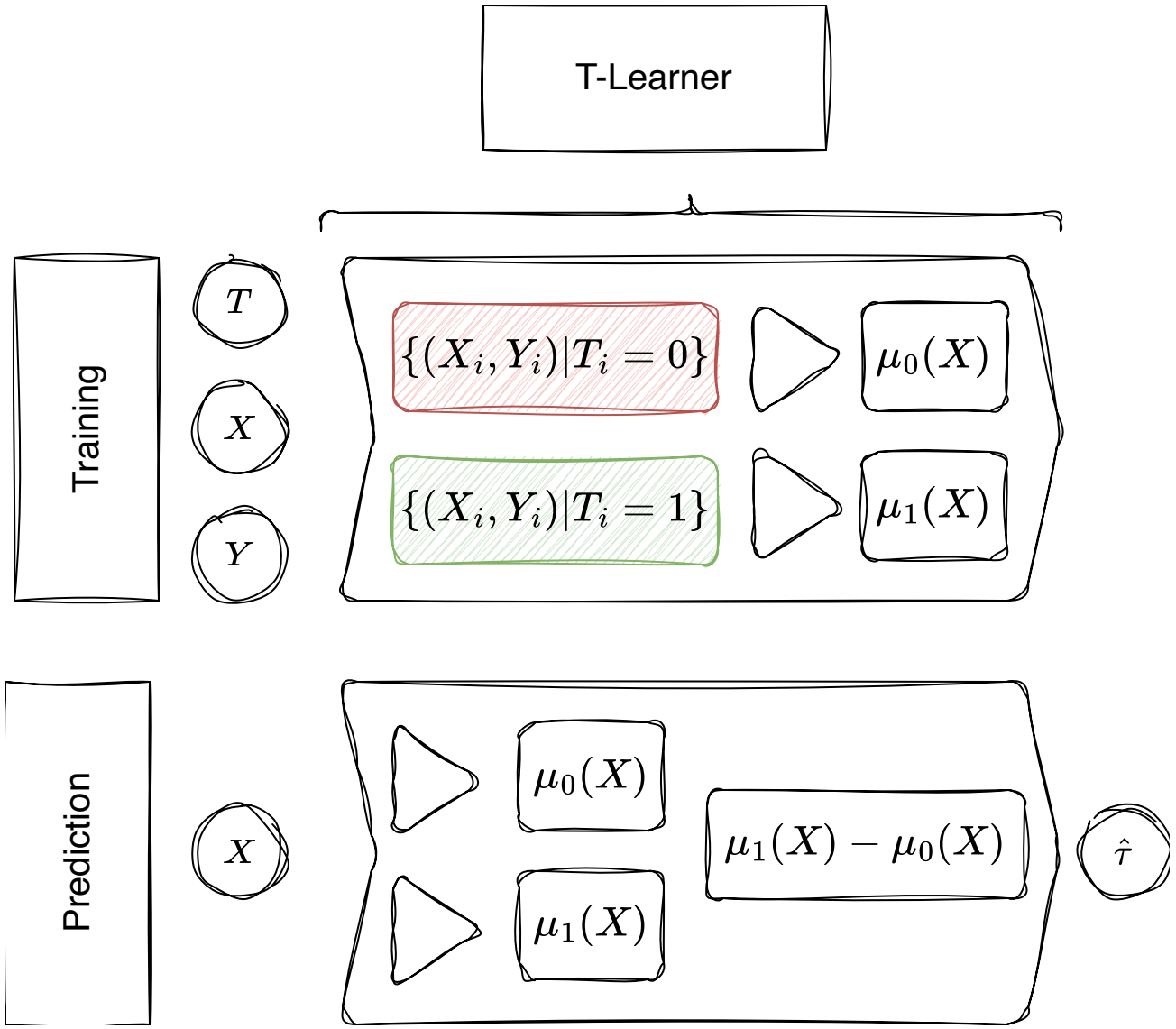
The T-Learner



The T-Learner



The T-Learner



4. How can we estimate heterogeneous treatment effects in practice?

The open-source libraries for CATE estimation

	`EconML`	`CausalML`
Developed by	MSR/py-why	Uber
License	MIT	Apache 2.0
#releases in past 2 years	4	7
Features	CATE estimation	CATE estimation
	direct policy learnig	
	inference (e.g. p-values)	
MetaLearner API	`sklearn`	`sklearn`

Risotto consumption: a simulation

age	nationality	chef_rating	gas_stove	$\mu(X)$	T	$\tau(X)$	Y
50.77	Indonesia	0.53	1	20.73	1	0.34	21.08
59.48	Iraq	0.46	0	20.46	0	0.76	20.46
22.21	Italy	0.58	0	15.90	1	0.88	16.79

$\mu(X) \equiv$ the 'base outcome', i.e. outcome/payment without stirring

$T \equiv$ the treatment, whether the risotto has been stirred (1) or not (0)

$\tau(X) \equiv$ the heterogeneous treatment effect

$Y \equiv$ the outcome, the final payment

$$Y = \mu(X) + T \cdot \tau(X)$$

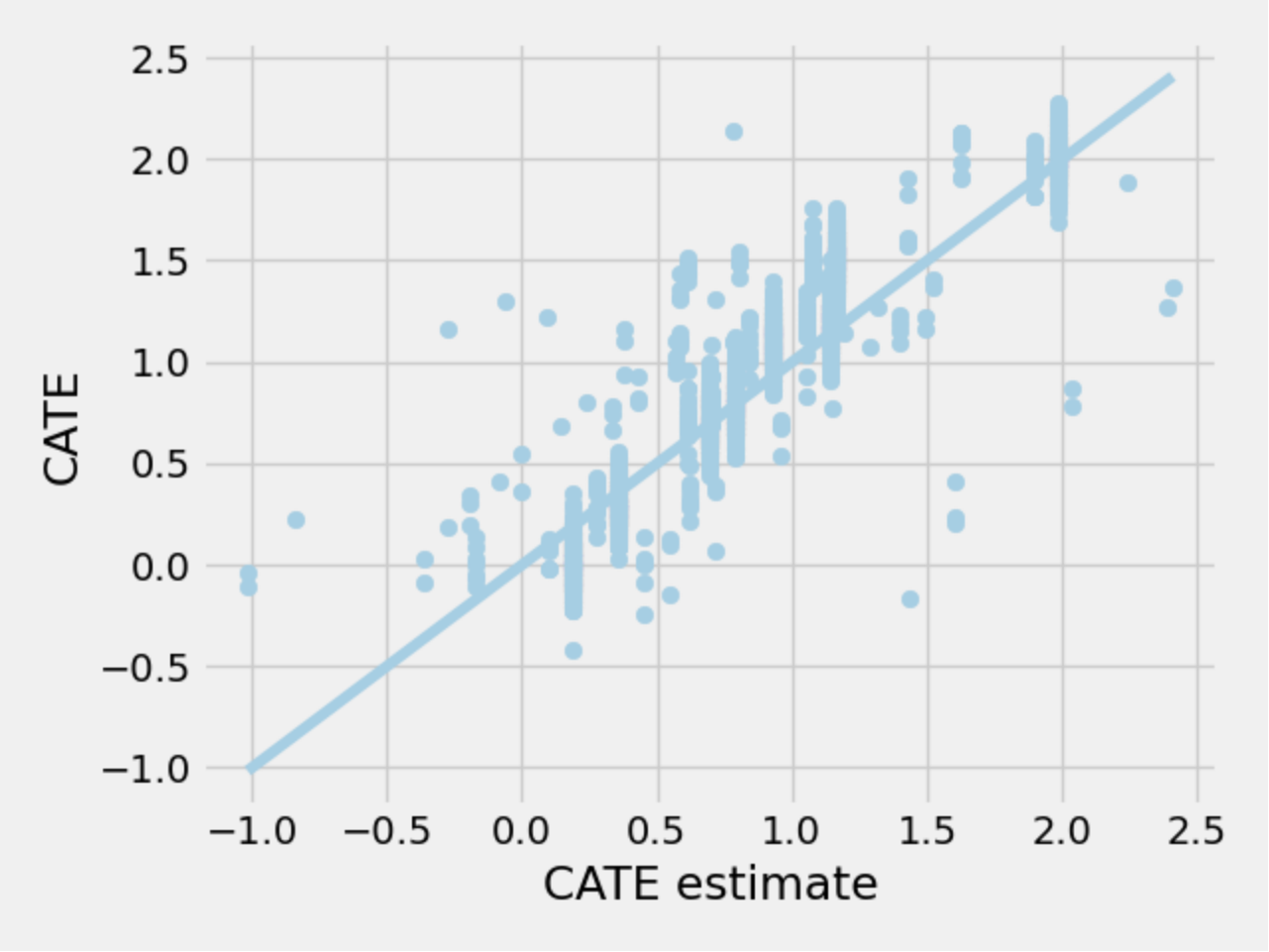
Training a CATE model with `CausalML`

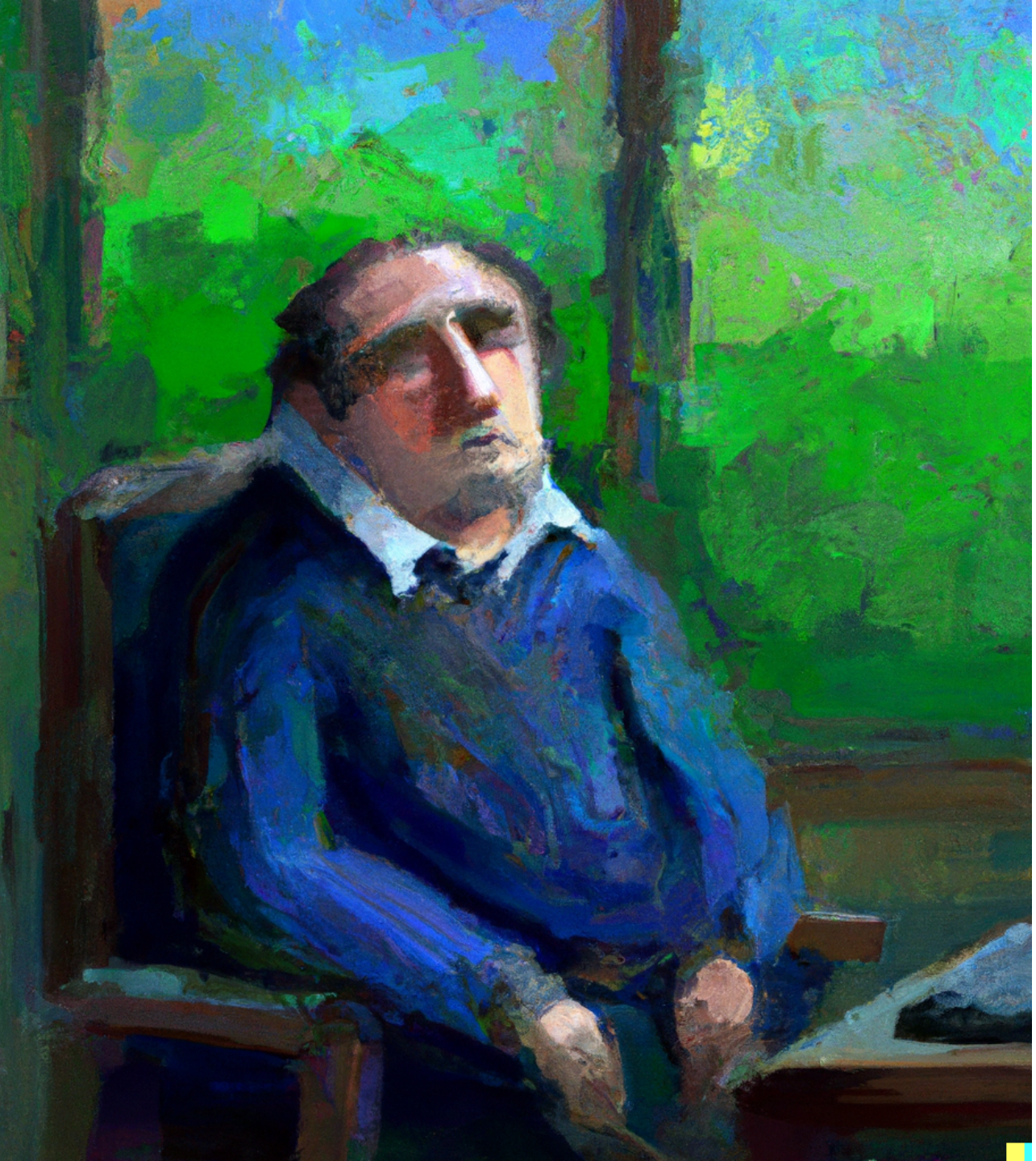
```
# One-hot encoding
X = pd.concat([
    df[numerical_covariates],
    pd.get_dummies(df["nationality"])
], axis=1)

# Model definition
reg = lgbm.LGBMRegressor()
clf = lgbm.LGBMClassifier()
model = causalml.BaseRRegressor(
    outcome_learner=reg,
    effect_learner=reg,
    propensity_learner=clf,
)

# Model training and prediction
model.fit(X=X, treatment=df[treatment], y=df[outcome])
cate_estimates = model.predict(X)
```

CATE estimation results

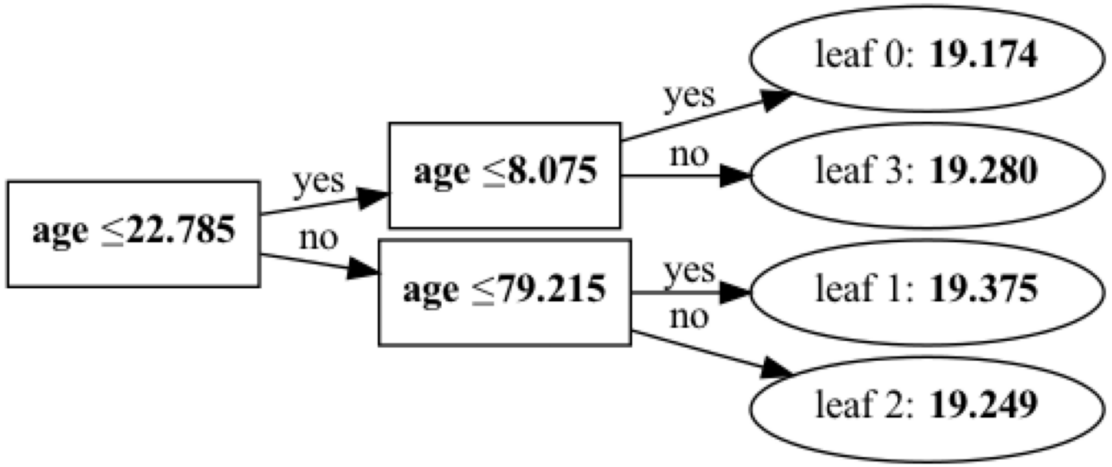


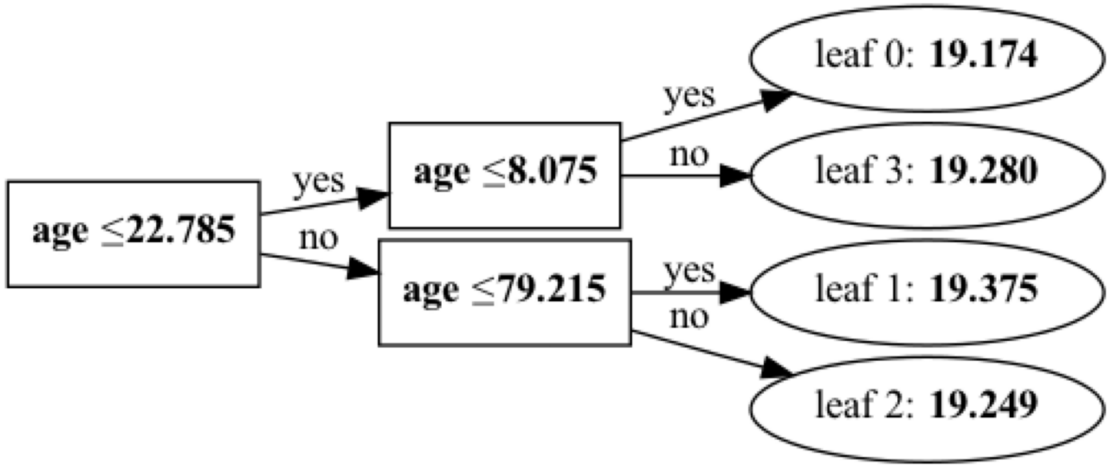


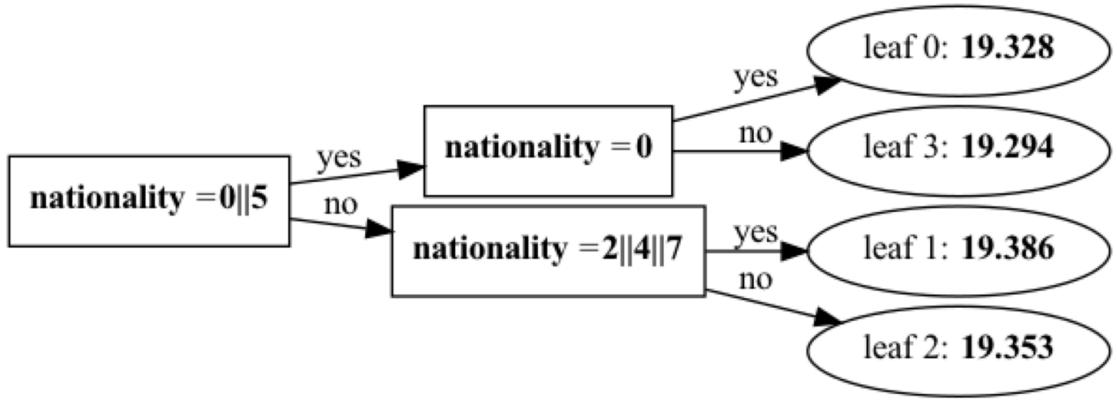
5. Pains and problems in practice (P^3)

P^3#1: Categorical features

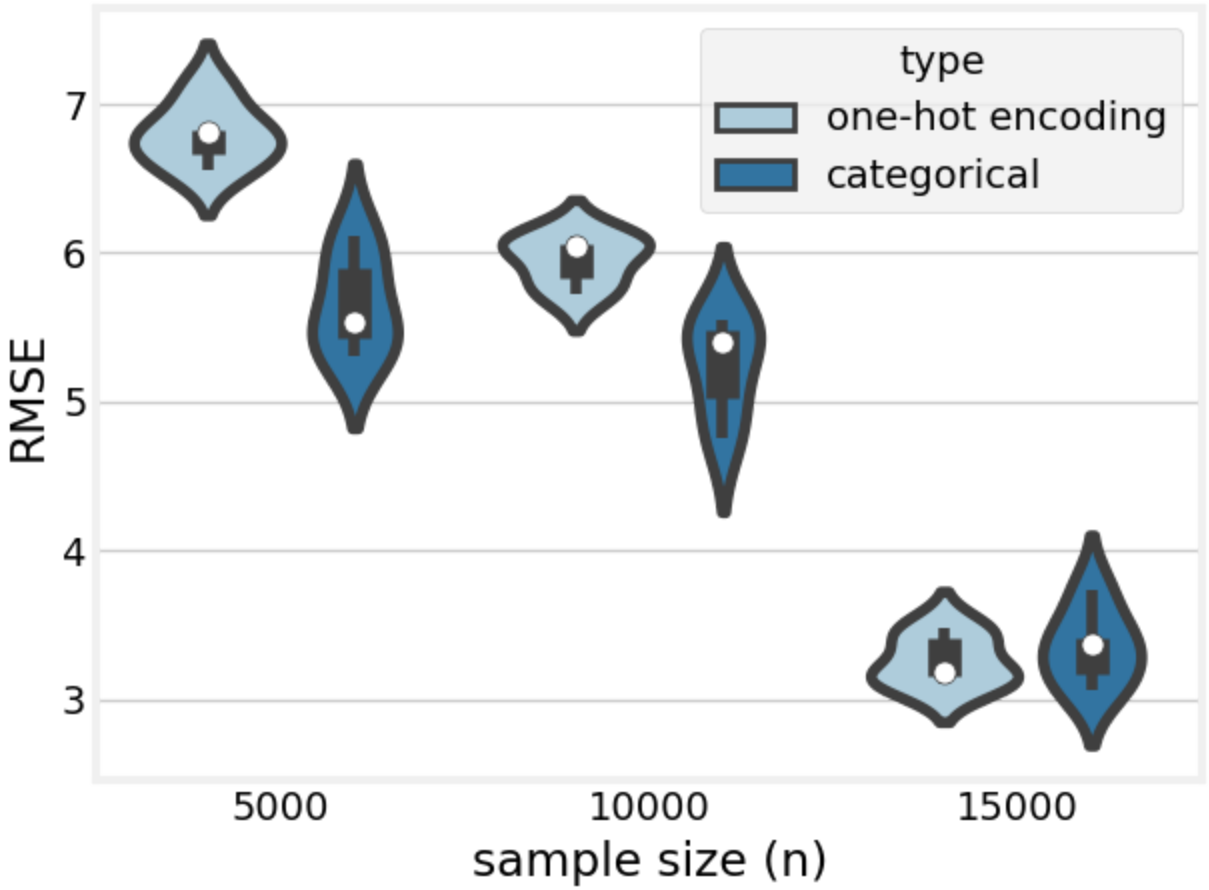
- `lightgbm` is a very popular choice for prediction on tabular datasets.
- In particular, it has native support for working with categorical features.
  Instead of having to **one-hot encode** categoricals, one can indicate that a column is to be treated as a categorical.


age ≤22.785
yes
no
age ≤8.075
age ≤79.215
yes
no
yes
no
leaf 0: 19.174
leaf 3: 19.280
leaf 1: 19.375
leaf 2: 19.249

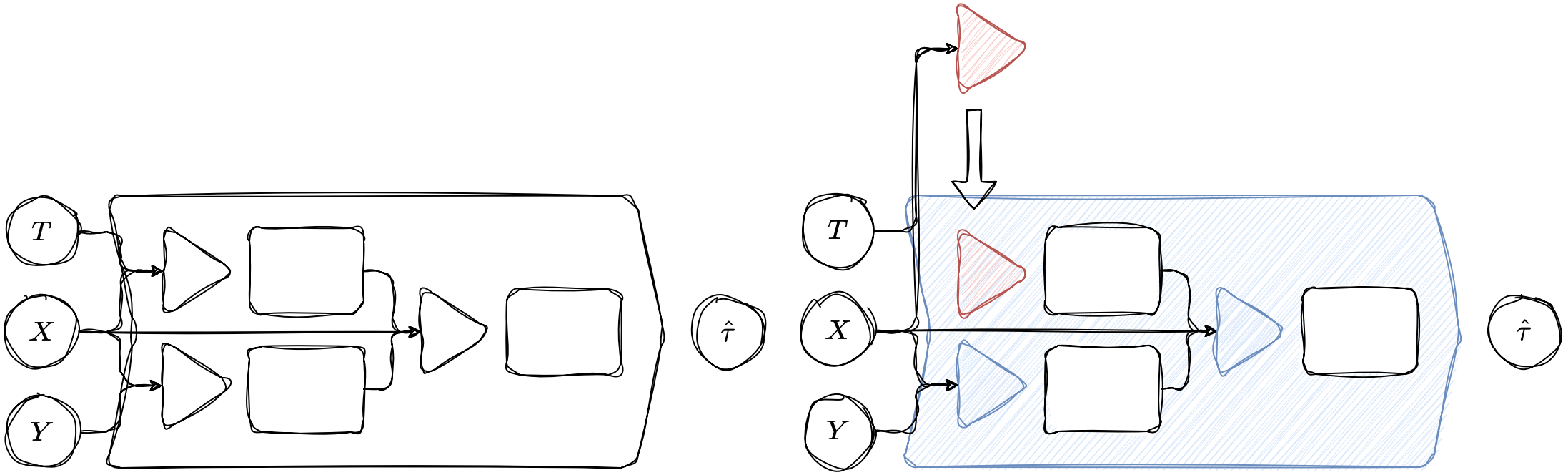

age ≤22.785
yes
no
age ≤8.075
age ≤79.215
yes
no
yes
no
leaf 0: 19.174
leaf 3: 19.280
leaf 1: 19.375
leaf 2: 19.249


nationality =0||5
yes
no
nationality =0
nationality =2||4||7
yes
no
yes
no
leaf 0: 19.328
leaf 3: 19.294
leaf 1: 19.386
leaf 2: 19.353

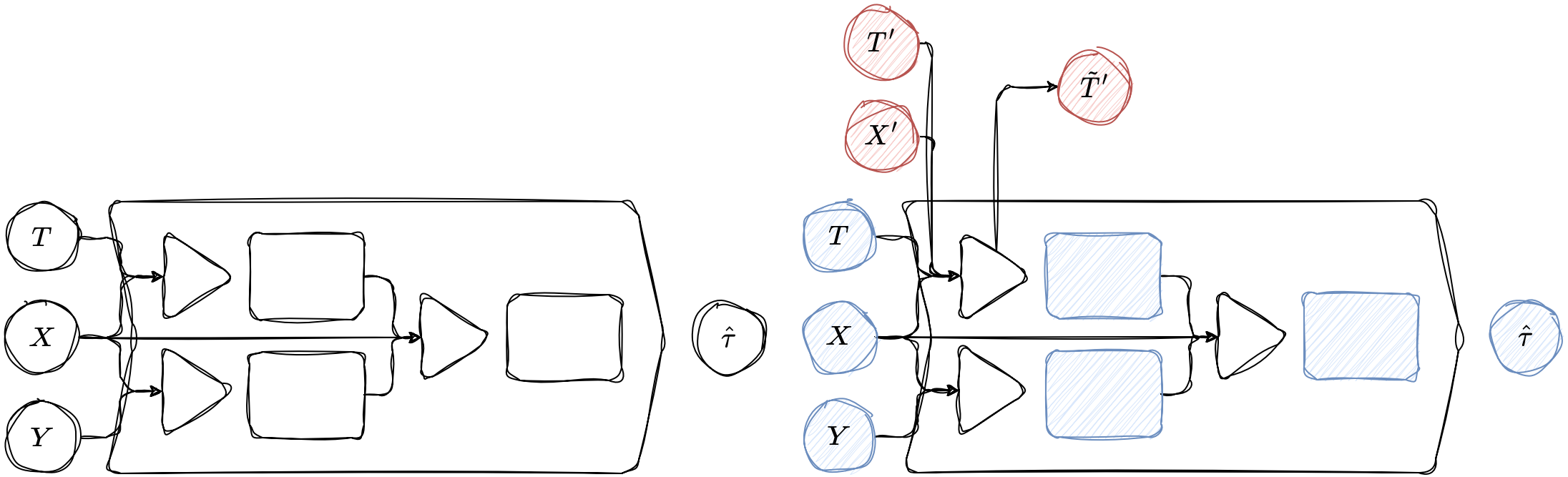
Tying back to our example: what's the difference?



P^3 #2: Using already trained components

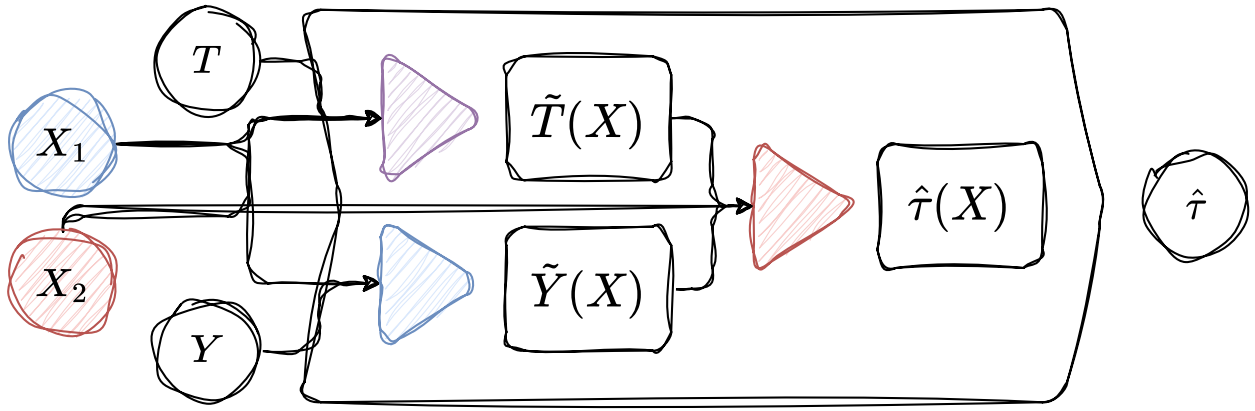


P^3 #3: Predicting with components



P^3 #4: Distinct covariate sets: Use case 1

- We might want to use different covariates for different components models inside of a MetaLearner.
    - E.g. we know that the treatment effect is only a function of `nationality` while the base outcome is a function of many more features.

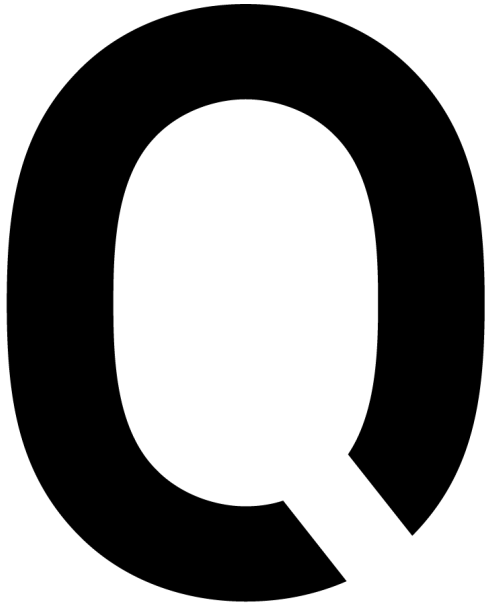


P^3 #4: Distinct covariate sets: Use case 2

- Let's assume we have 3 instead of 2 treatment variants.
treatment variant 1	treatment variant 2	covariates
No stirring	Stirring for 20'	X
No stirring	Stirring for 40'	X
Stirring for 20'	Stirring for 40'	$X \cup \{$ `spoon_type` $\}$
- Ideally, the MetaLearner implementation would always simply as many available features as possible when comparing different treatment variants.

P^3 #4: Distinct covariate sets

Whatever the motivation of using different covariate sets inside a MetaLearner, afaict `CausalML` and `EconML` don't support them.



quantco

Do you also prefer Causal Inference over aligning elements in slides?

Join us :)

https://www.quantco.com/

ENGINEERING

Cloud Site Reliability Engineer — APPLY
EUROPE ENGINEERING HYBRID

Deep Learning Engineer — APPLY
EUROPE ENGINEERING FULL-TIME HYBRID

Developer Productivity Engineer — APPLY
EUROPE ENGINEERING FULL-TIME HYBRID

Software Engineer — APPLY
EUROPE ENGINEERING FULL-TIME HYBRID

Software Engineering Intern — APPLY
EUROPE ENGINEERING INTERN ON-SITE

DATA SCIENCE

Data Science Intern (Causal Inference/Machine Learning) — APPLY
EUROPE DATA SCIENCE INTERN ON-SITE

Data Scientist (Causal Inference/Machine Learning) — APPLY
EUROPE DATA SCIENCE FULL-TIME HYBRID

Data Scientist | Commercial Insurance Pricing — APPLY
COLOGNE, NORTH RHINE-WESTPHALIA DATA SCIENCE FULL-TIME HYBRID

Medical Physicist - Virdx — APPLY
ESSEN, NORTH RHINE-WESTPHALIA DATA SCIENCE FULL-TIME HYBRID

Quantitative Researcher — APPLY
GLOBAL DATA SCIENCE FULL-TIME HYBRID

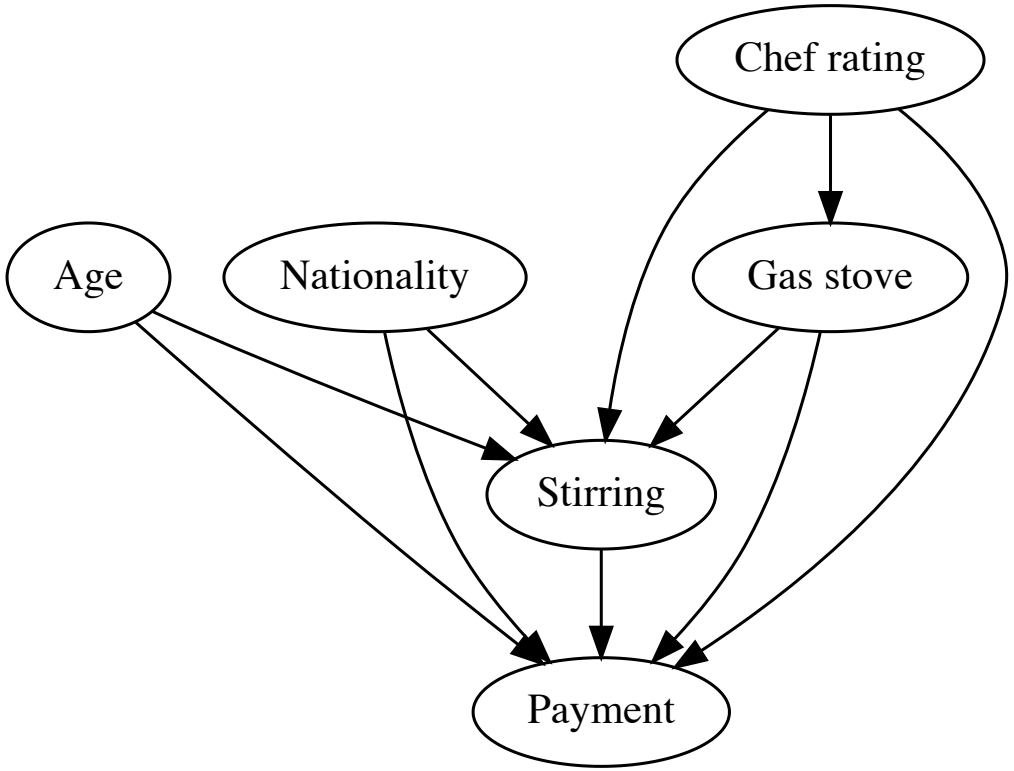
Lastly...

- Thanks for listening!
- Special thanks to Daan Nilis, Ege Karaismailoğlu, Julie Vienne, Matthias Lux, Norbert Stoop
- Shout out to Matheus Facure's Causal Inference for the Brave and True
- You can find the slides and according code at github.com/kklein/pydata_ams

Addendum

To stir or not to stir, the maths

- Assume that the cost of stirring amounts to 1$ per unit.
- Also assume that the overall revenue when never stirring is R.
- Then, the overall revenue when **always stirring** is $R - n \cdot 1 + \delta_1$
  - The plot from the previous slide tells us that $n \cdot 1 > \delta$.
- The overrall revenue of **stirring when we expect it to pay off**: $R - k \cdot 1 + \delta_\pi$
  - We can condition on certain 'covariates'/features to decide for whom it pays off.
  - When doing this 'right', we get that $\delta_\pi > k \cdot 1$.



Risotto consumption: a simulation

How can we use categoricals with `lightgbm`?

- Option 1: Use `pandas` `category` dtype

```
df["nationality"] = df["nationality"].astype("category")
model = lgbm.LGBMRegressor()
model.fit(df[["nationality"]], df["payment"])
```

- Option 2: Explicitly set `categorical_indices`

```
df["nationality"] = df["nationality"].astype("category").cat.codes
model = lgbm.LGBMRegressor(categorical_feature=[0])
model.fit(df[["nationality"]], df["payment"])
```

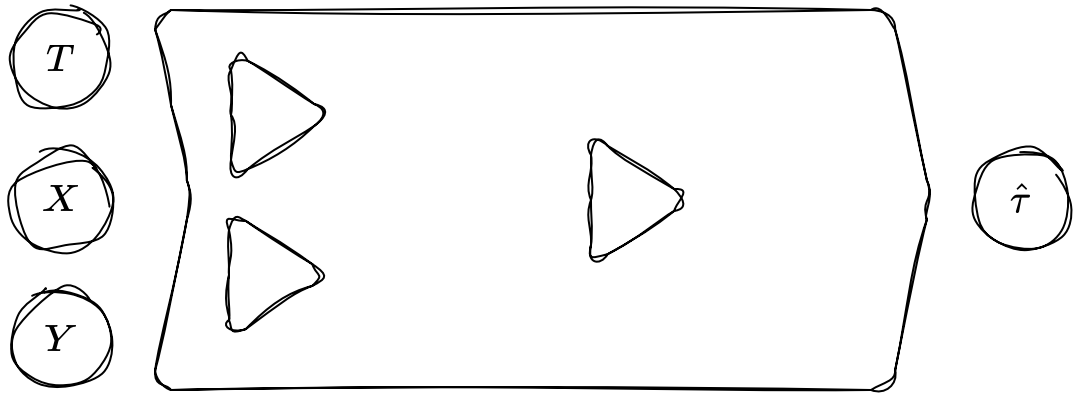
How can we use `lightgbm`'s categoricals in `EconML` and `CausalML`?

- Unfortunately, both options don't work with `CausalML` and `EconML`.
- Option 1 is not possible since both convert `pandas` input to `numpy` objects:
  - `X, treatment, y = convert_pd_to_np(X, treatment, y)`
  - https://github.com/uber/causalml/blob/3b3daaa3cd2ef1960028908c152cfd242b37712c/causalml/inference/meta/rlearner.py#L100
- Option 2 is not possible since constructor parameters can't be passed.

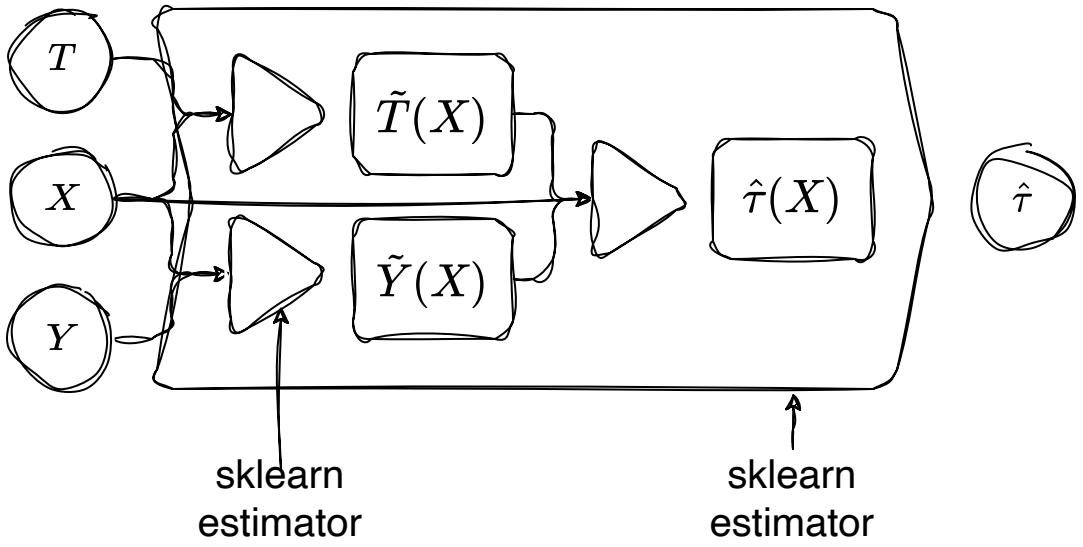
A hack is - of course - possible in order to indirectly use option 2:

```
from functools import partialmethod
from lightgbm import LGBMRegressor
LGBMRegressor.fit = partialmethod(
  LGBMRegressor.fit,
  categorical_feature=[0],
)
```

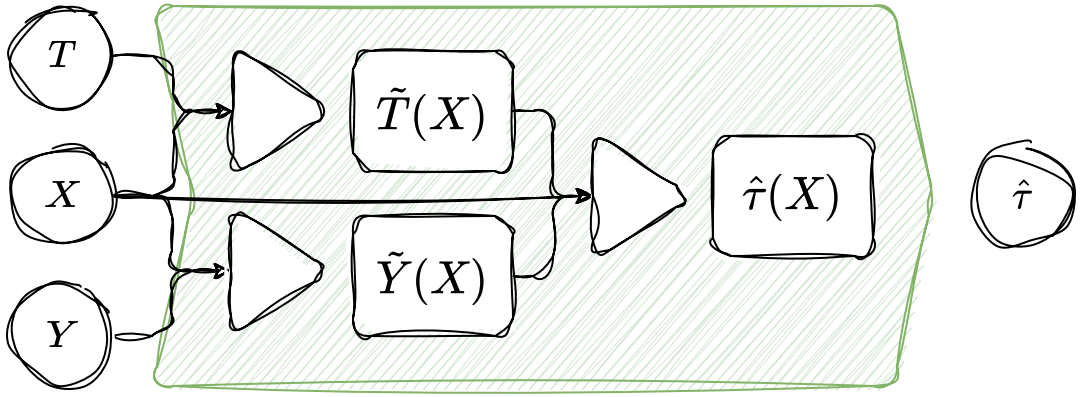
The R-Learner



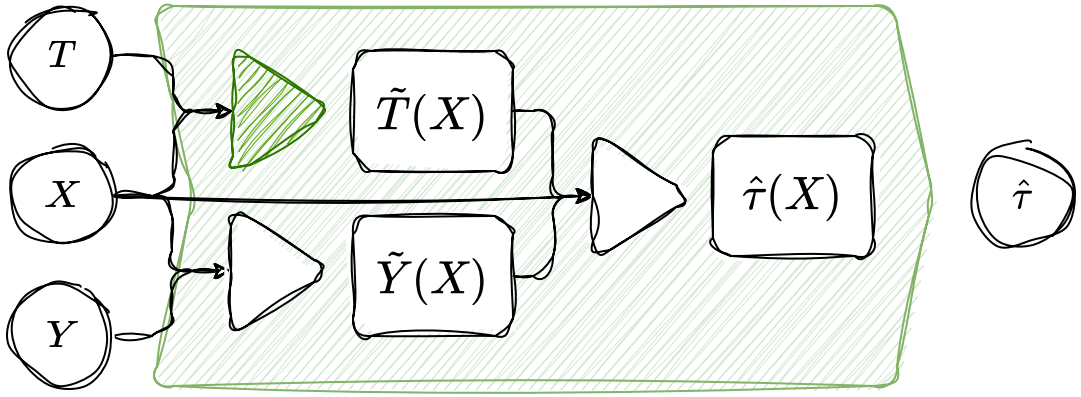
The R-Learner



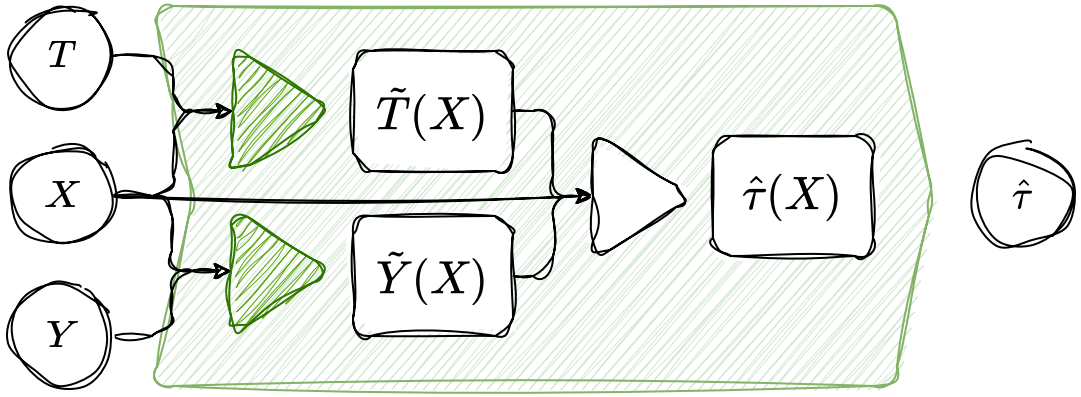
The R-Learner: Hyperparameter tuning



The R-Learner: Hyperparameter tuning



The R-Learner: Hyperparameter tuning



The R-Learner: Hyperparameter tuning

- Unfortuntaely, I haven't found a supported way of reusing already trained components with most `EconML` and `CausalML` CATE estimators.
  - See e.g. EconML issue 646.
- We can expect a ~3x increase of runtime due to not being able to train and reuse component models.
- This is even amplified when trying to use a particular component model for other MetaLearners.

And more...

- DoubleML: Biased final stage
- Tricky to combine cross-fitting with further cross-splitting
  (e.g. super learning or splits) -> also an engineering problem
  (e.g. multiprocessing)
- Read out treatment effects of categoricals when using DML

Material

- R-Learner: Nie and Wagner, 2020