Machine Learning for econometrics

Flexible models for tabular data

Matthieu Doutreligne

February 18th, 2025

Reminder from previous session

- Statistical learning 101: bias-variance trade-off
- Regularization for linear models: Lasso, Ridge, Elastic Net
- Transformation of variables: polynomial regression
-

Reminder from previous session

- Statistical learning 101: bias-variance trade-off
- Regularization for linear models: Lasso, Ridge, Elastic Net
- Transformation of variables: polynomial regression
- 🤔 But… How to select the best model? the best hyper-parameters?

Table of contents

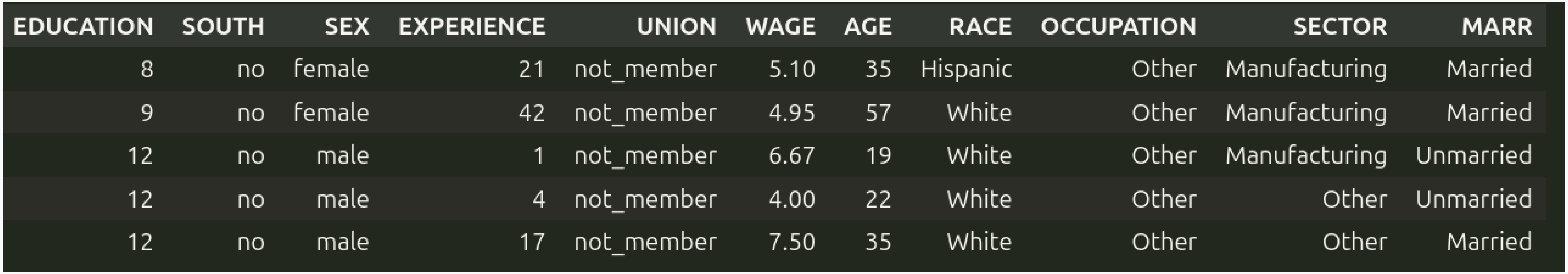
1. Model evaluation and selection with cross-validation
2. Flexible models: Tree, random forests and boosting
3. A word on other families of models

Model evaluation and selection with cross-validation

A closer look at model evaluation: Wage example

Example with the Wage dataset

- Raw dataset: (N=534, p=11)



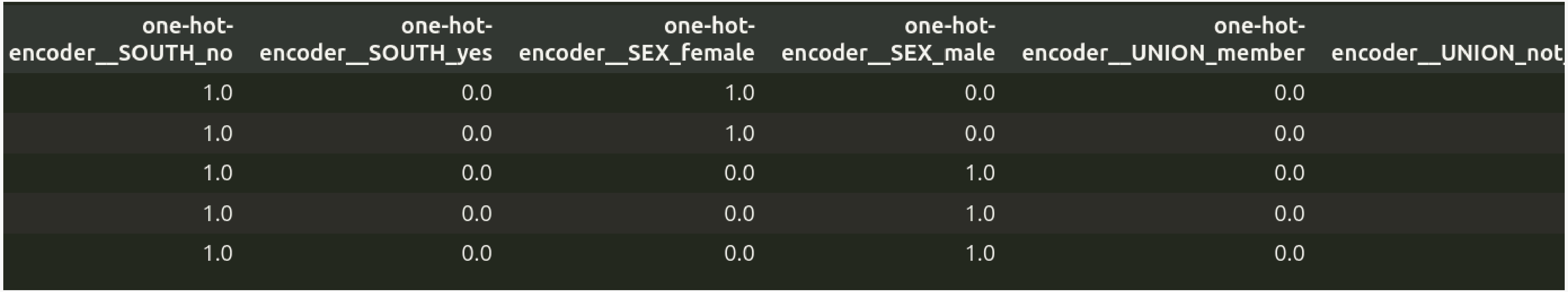
EDUCATION	SOUTH	SEX	EXPERIENCE	UNION	WAGE	AGE	RACE	OCCUPATION	SECTOR	MARR
8	no	female	21	not_member	5.10	35	Hispanic	Other	Manufacturing	Married
9	no	female	42	not_member	4.95	57	White	Other	Manufacturing	Married
12	no	male	1	not_member	6.67	19	White	Other	Manufacturing	Unmarried
12	no	male	4	not_member	4.00	22	White	Other	Other	Unmarried
12	no	male	17	not_member	7.50	35	White	Other	Other	Married

-
-

A closer look at model evaluation: Wage example

Example with the Wage dataset

- Raw dataset: (N=534, p=11)
- Transformation: encoding categorical data, scaling numerical data: (N=534, p=23)

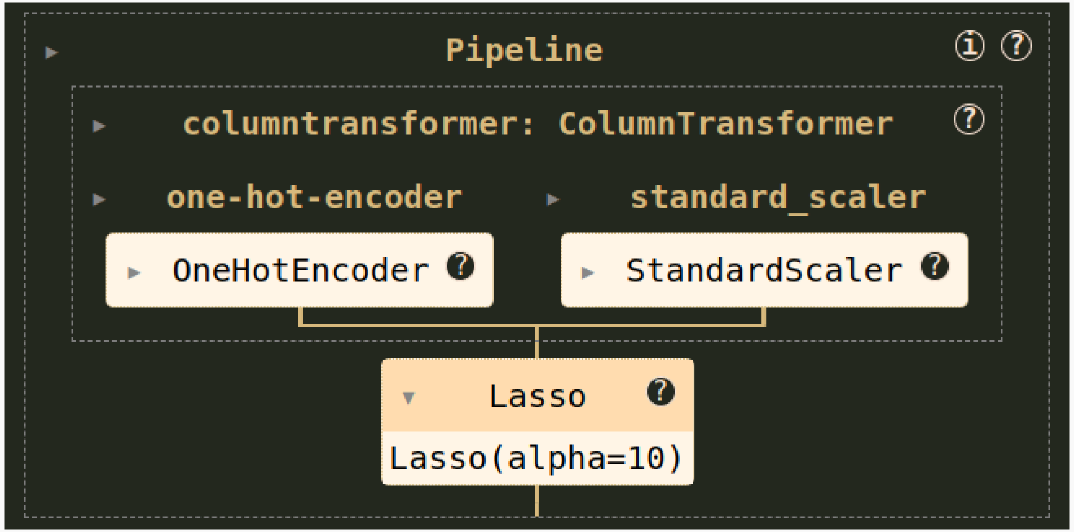


one-hot-encoder__SOUTH_no	one-hot-encoder__SOUTH_yes	one-hot-encoder__SEX_female	one-hot-encoder__SEX_male	one-hot-encoder__UNION_member	encoder__UNION_not
1.0	0.0	1.0	0.0	0.0	
1.0	0.0	1.0	0.0	0.0	
1.0	0.0	0.0	1.0	0.0	
1.0	0.0	0.0	1.0	0.0	
1.0	0.0	0.0	1.0	0.0	

A closer look at model evaluation: Wage example

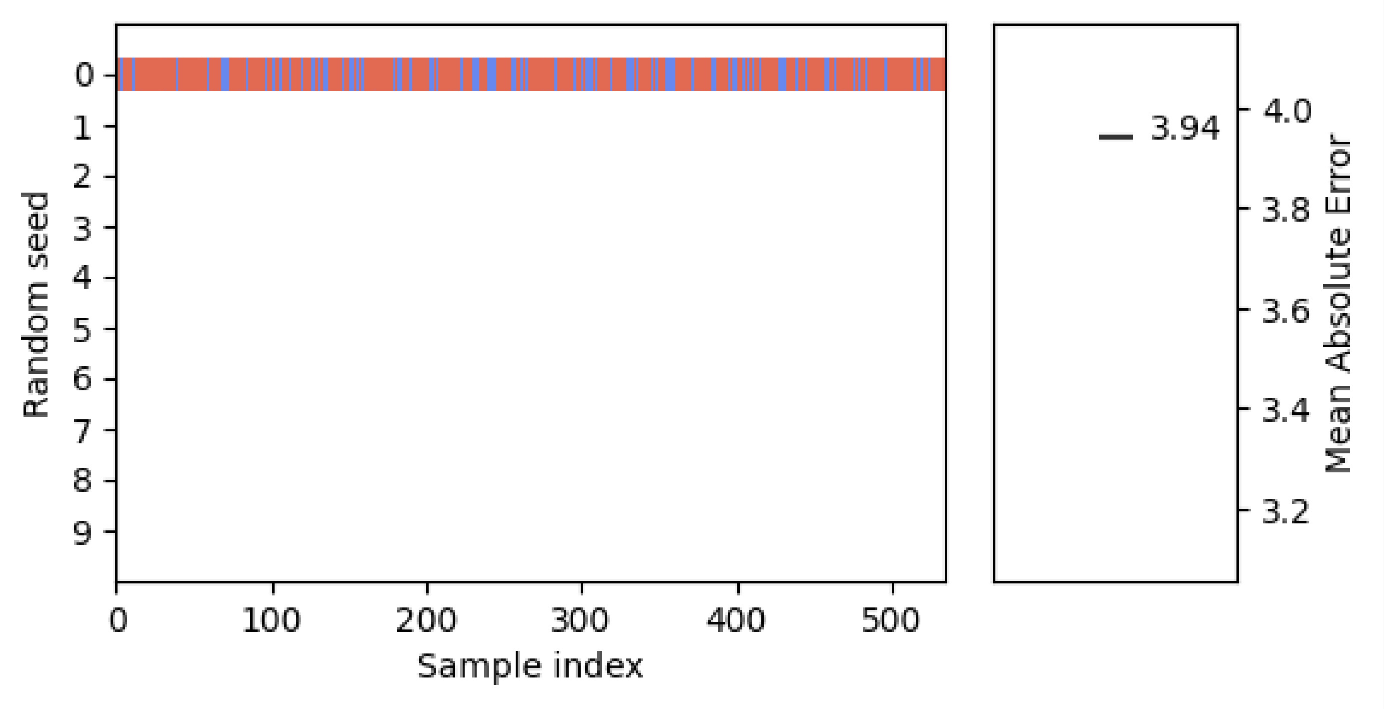
Example with the Wage dataset

- Raw dataset: (N=534, p=11)
- Transformation: encoding categorical data, scaling numerical data: (N=534, p=23)
- Regressor: Lasso with regularization parameter ($\alpha = 10$)



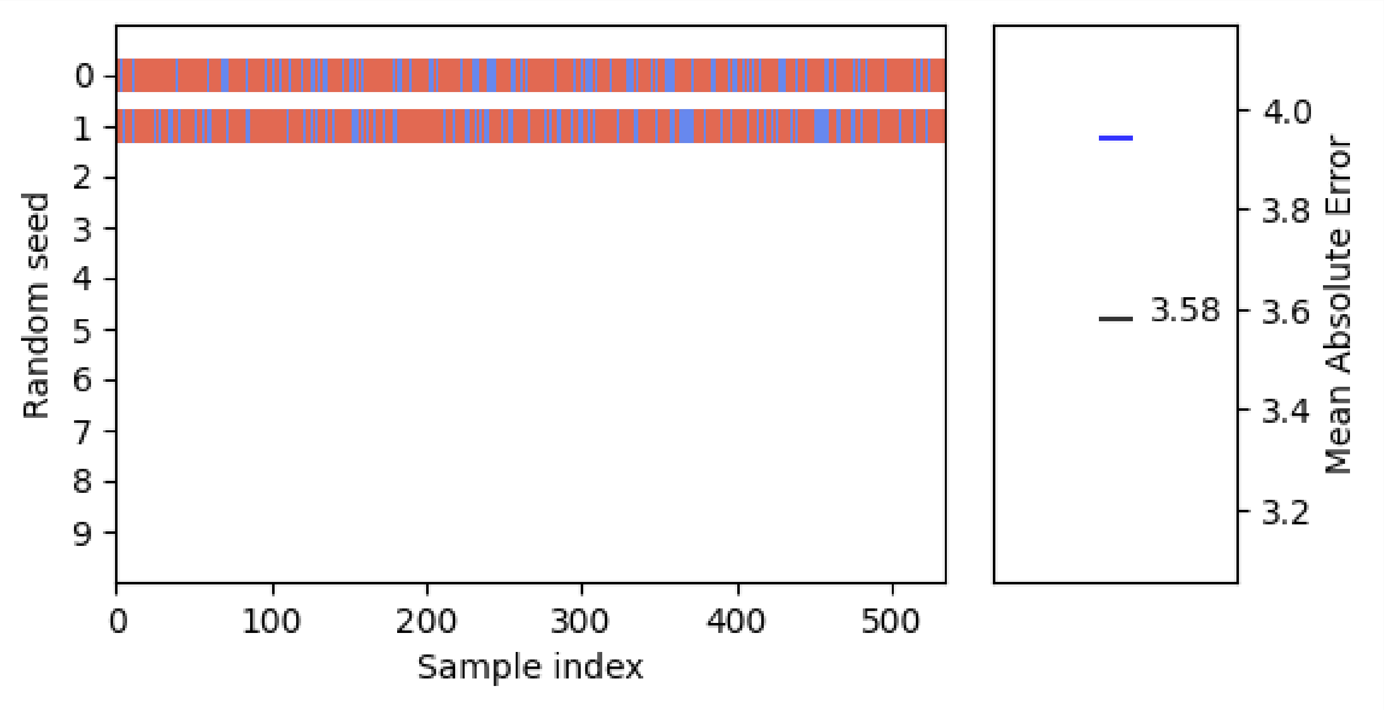
Repeated train/test splits

Splitting once: In red, the training set, in blue, the test set



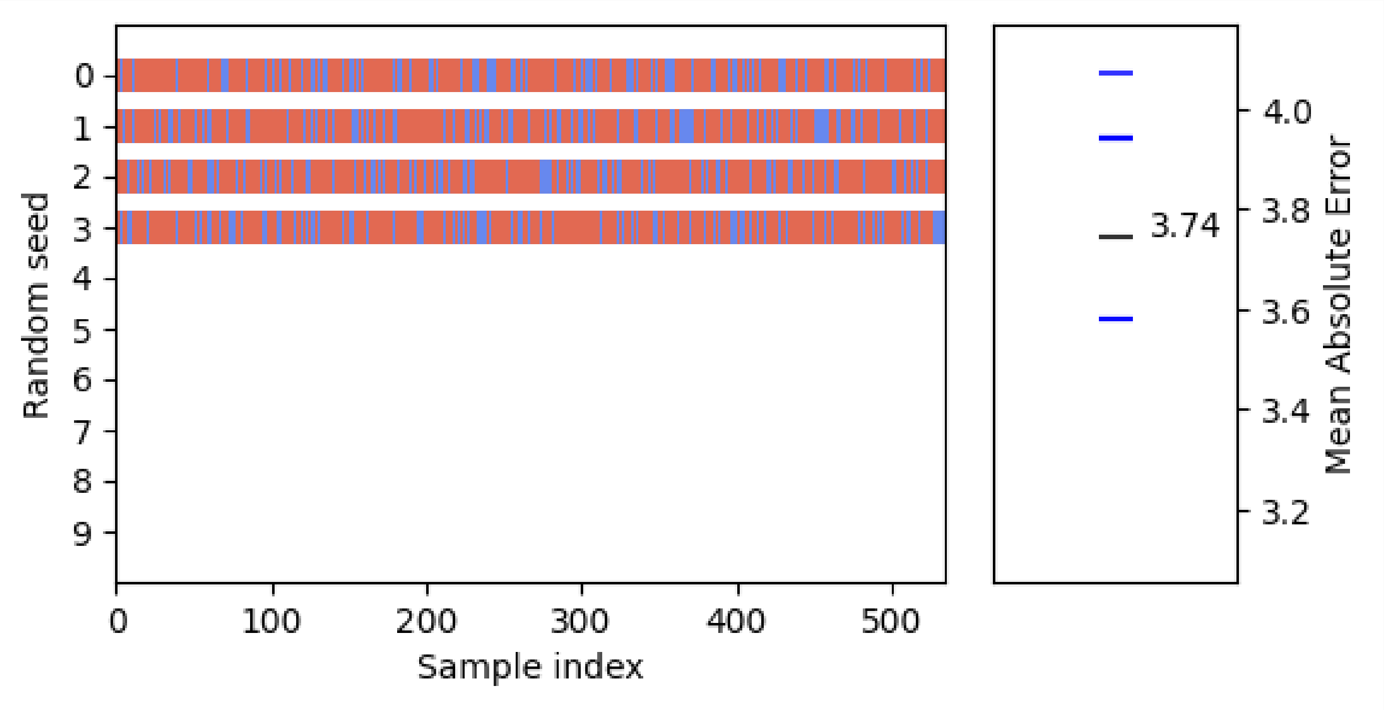
Repeated train/test splits

But we could have chosen another split ! Yielding a different MAE



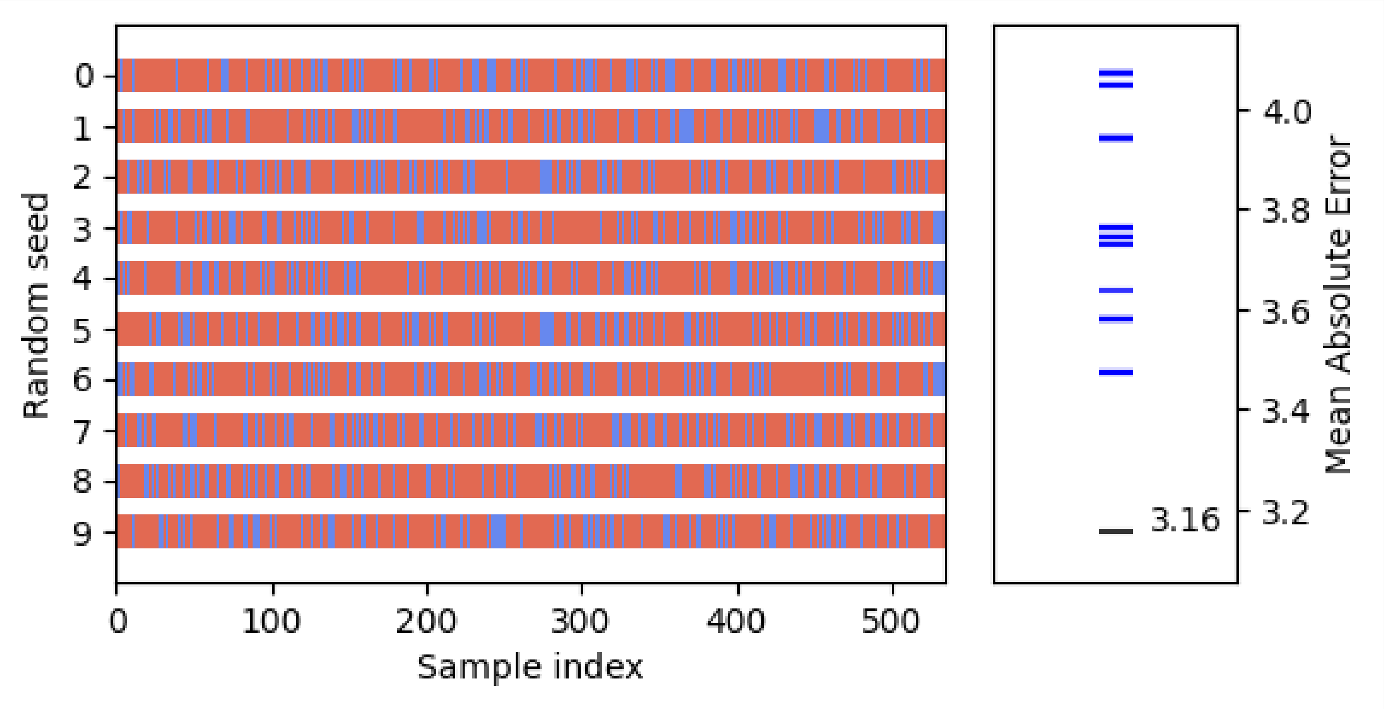
Repeated train/test splits

And another split...



Repeated train/test splits

Splitting ten times



🎉 **Distribution of MAE:** 3.71 ± 0.26

Repeated train/test splits = Cross-validation

Cross-validation

- In sklearn, it can be instantiated with `cross_validate`.

```python
from sklearn.model_selection import cross_validate
from sklearn.model_selection import ShuffleSplit

cv = ShuffleSplit(n_splits=40, test_size=0.3, random_state=0)
cv_results = cross_validate(
    regressor, data, target, cv=cv, scoring="neg_mean_absolute_error"
)
```

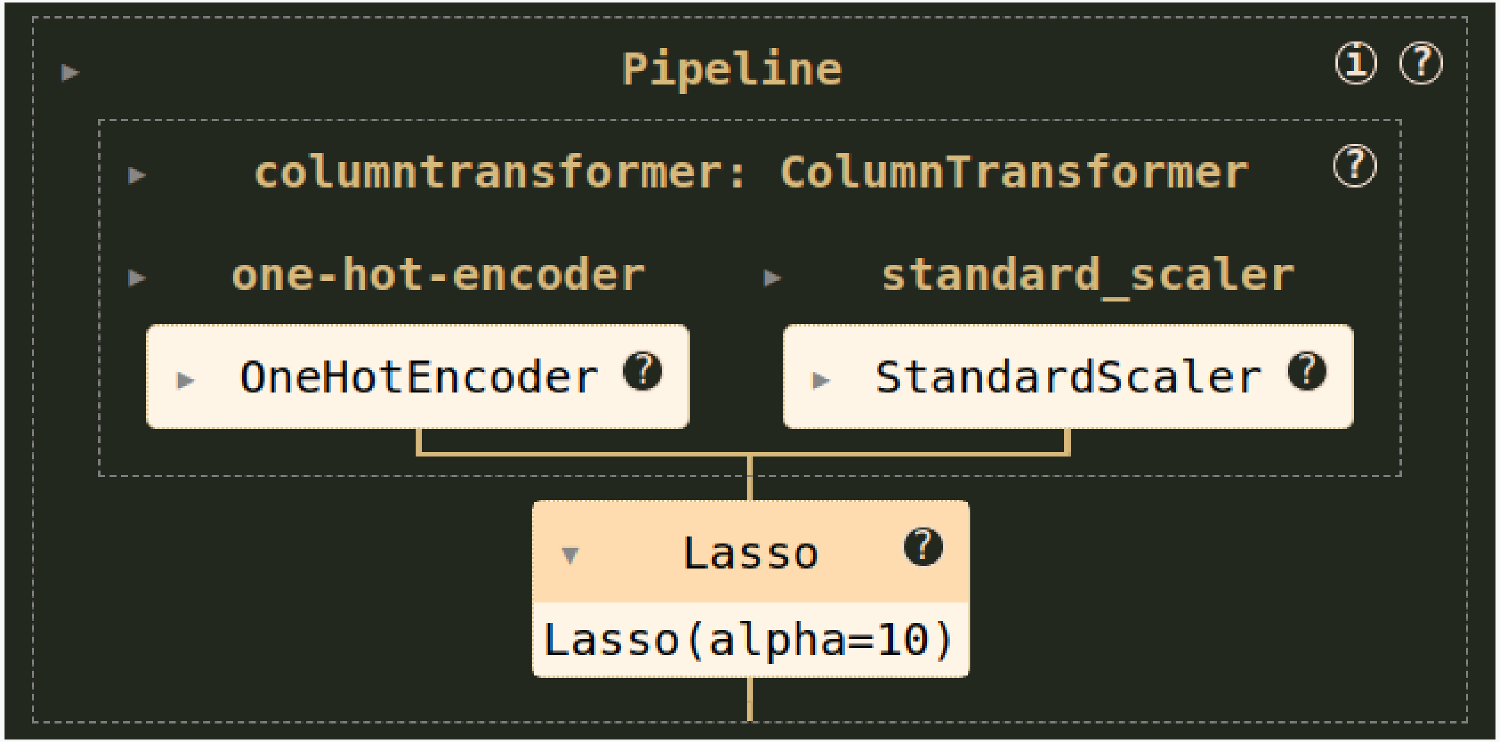
Repeated train/test splits = Cross-validation

Cross-validation

- In sklearn, it can be instantiated with `cross_validate`.
- 🙂 Robustly estimate generalization performance
- 🤩 Estimate variability of the performance: similar to bootstrapping (but different).
- 🚀 Let's use it to select the best models among several canditates!
- Proof that it selects the best model (averaging on the folds): (Lecué & Mitchell, 2012)

Cross-validation for model selection: choose best α for lasso

- Wage pipeline



```
Lasso(alpha=10)
```

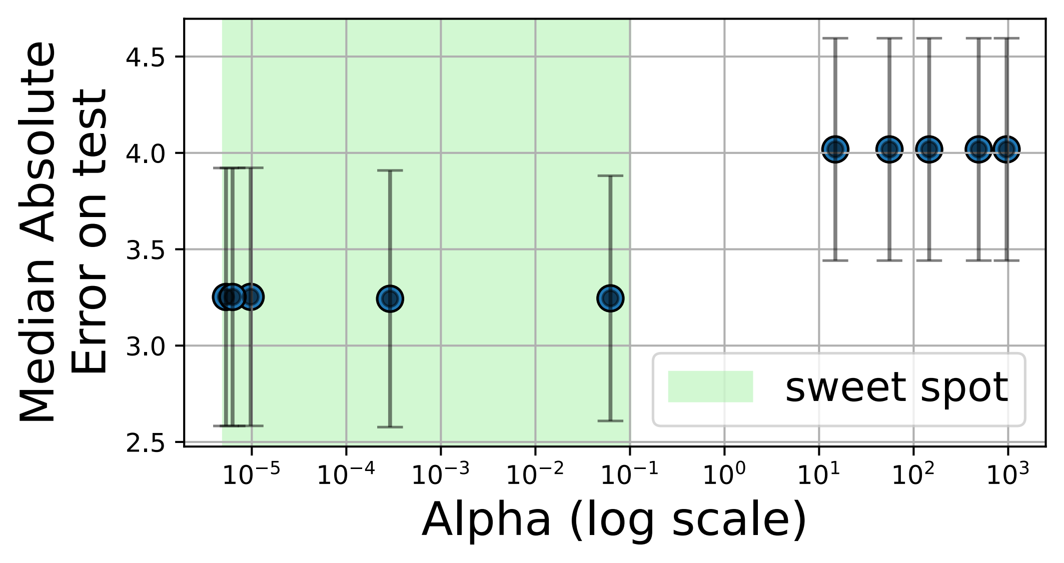
Cross-validation for model selection: choose best α for lasso

- Wage pipeline
- Random search over a distribution of α values

```python
param_distributions = {"lasso__alpha": loguniform(1e-6, 1e3)}
model_random_search = RandomizedSearchCV(
    pipeline,
    param_distributions=param_distributions,
    n_iter=10, # number of hyper-parameters sampled
    cv=5, # number of folds for the cross-validation
    scoring="neg_mean_absolute_error", # score to optimize
)
model_random_search.fit(X, y)
```

Cross-validation for model selection: choose best α for lasso

- Wage pipeline
- Random search over a distribution of α values
- Identify the best α value(s)



What final model to use for new prediction?

- Either refit on full data the model with the best hyper-parameters on the full data
- Or use the aggregation of outputs from the cross-validation of the best model:

$\hat{y} = \frac{1}{K} \sum_{k=1}^{K} \hat{y}_k$ where $\hat{y}_k$ is the prediction of the model trained on the k-th fold

Naive cross-validation to select AND estimate the best performances

Nested cross-validation to select the best model

Flexible models: Tree, random forests and boosting

Tree for predictive inference

Random Forests for predictive inference

Boosting

Ensemble models

A word on other families of models

Why not use deep learning everywhere?

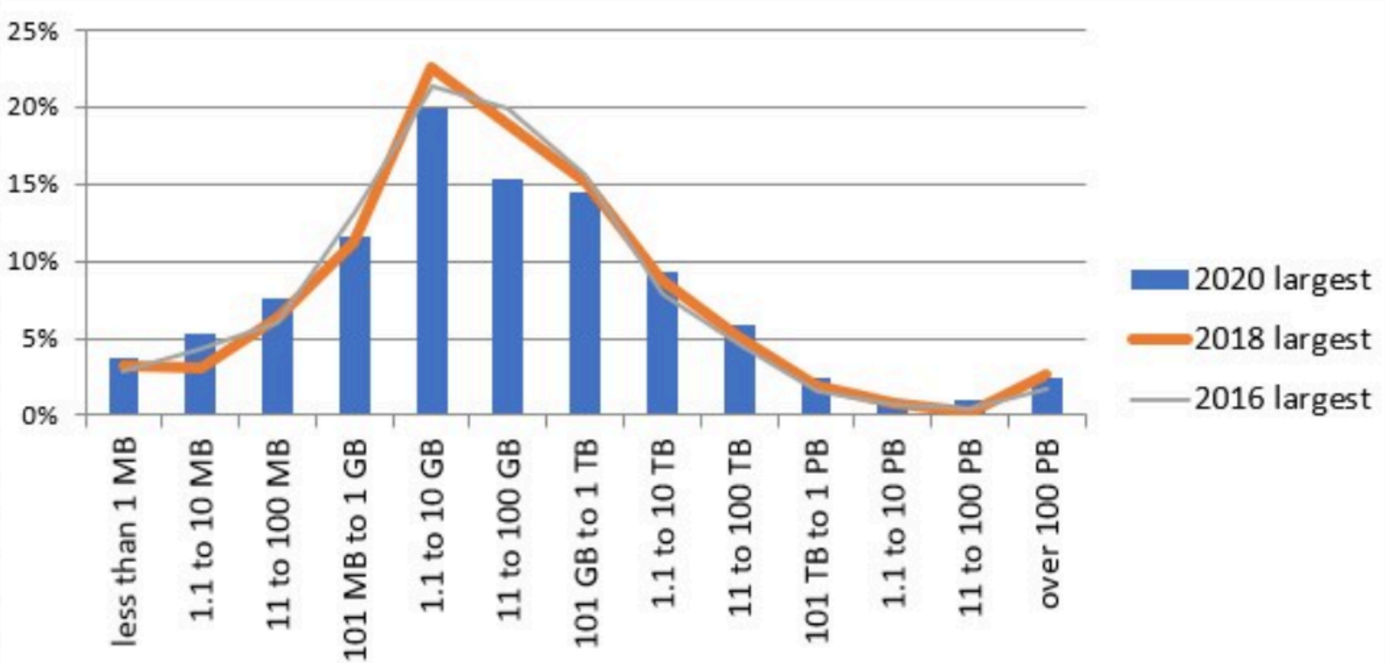
- Success of deep learning (aka deep neural networks) in image, speech recognition and text
- Why not so used in econometrics?

Deep learning needs a lot of data (typically $N \approx 1$ million)

  - Do we have this much data in econometrics?

Limited data settings

- Typically in economics (but also everywhere), we have a limited number of observations



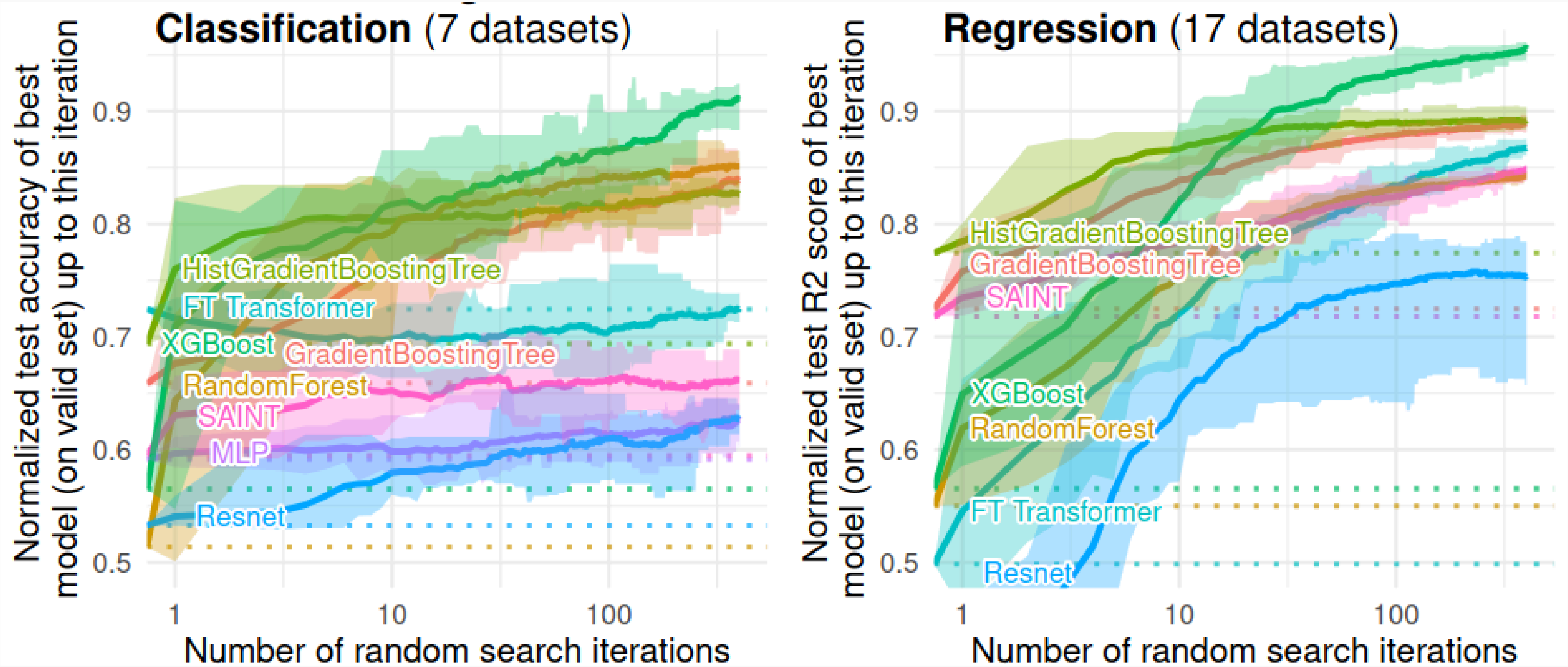
Typical dataset are mid-sized. This does not change with time.[1]

[1] https://www.kdnuggets.com/2020/07/poll-largest-dataset-analyzed-results.html

Deep learning underperforms on data tables

Tree-based methods outperform tailored deep learning architectures (Grinsztajn et al., 2022)

Deep learning underperforms on data tables



DAG for a RCT: the treatment is independent of the confounders

Other well known families of models

Generalized linear models

Support vector machines

Gaussian processes

Bibliography

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data?. Advances in Neural Information Processing Systems, 35, 507–520.

Lecué, G., & Mitchell, C. (2012). Oracle inequalities for cross-validation type procedures.