Validation

How to validate models?

Luis Moneda, Data Scientist at Nubank

Outline

1. Supervised Learning summarized
2. ML 101 validation
3. Real world supervised learning
4. Examples, cases...

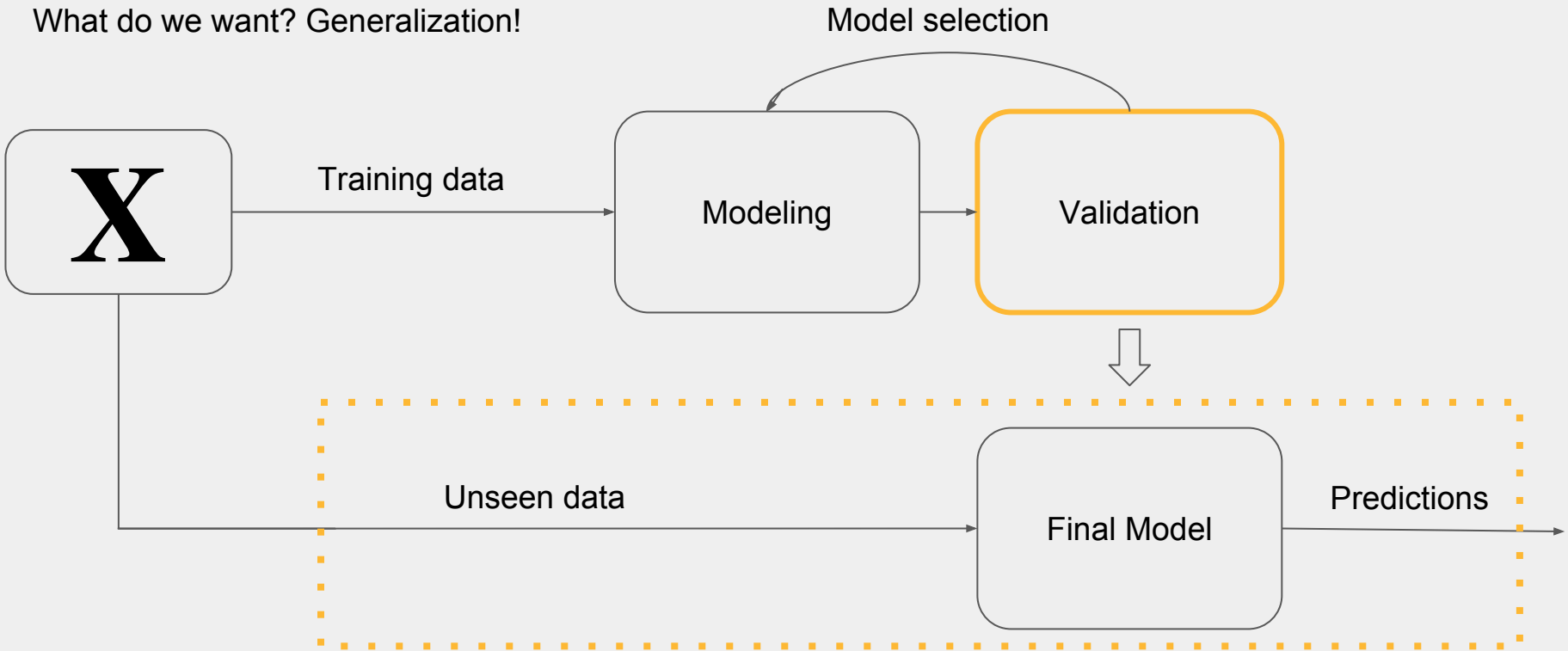
Supervised Learning summarized

$$\mathbf{X} \xrightarrow{f} \mathrm{y}$$

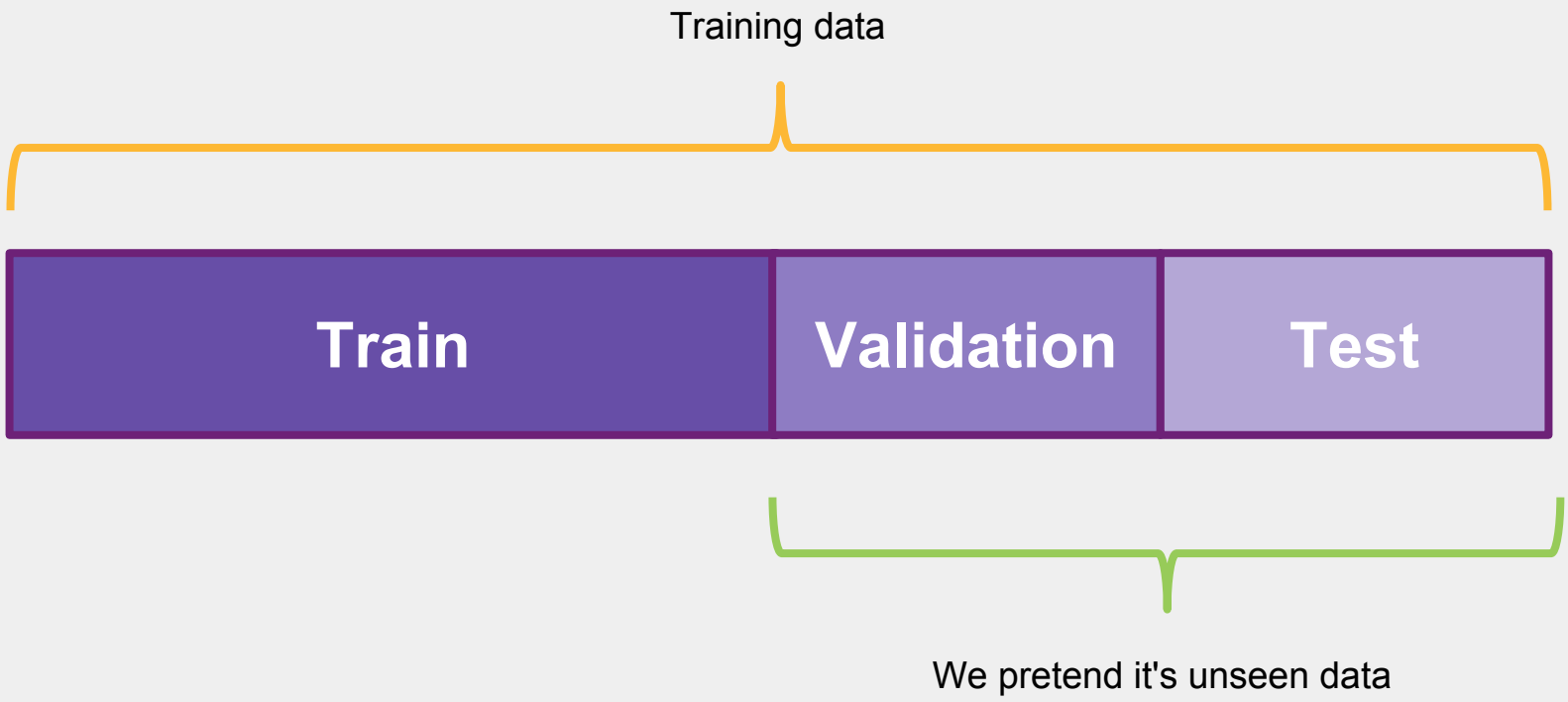
- Empirical Risk Minimization
- Statistical Learning
- Independently identically distributed (iid)
- We want to predict things nicely, we don't care about what is the f

ML101 Validation

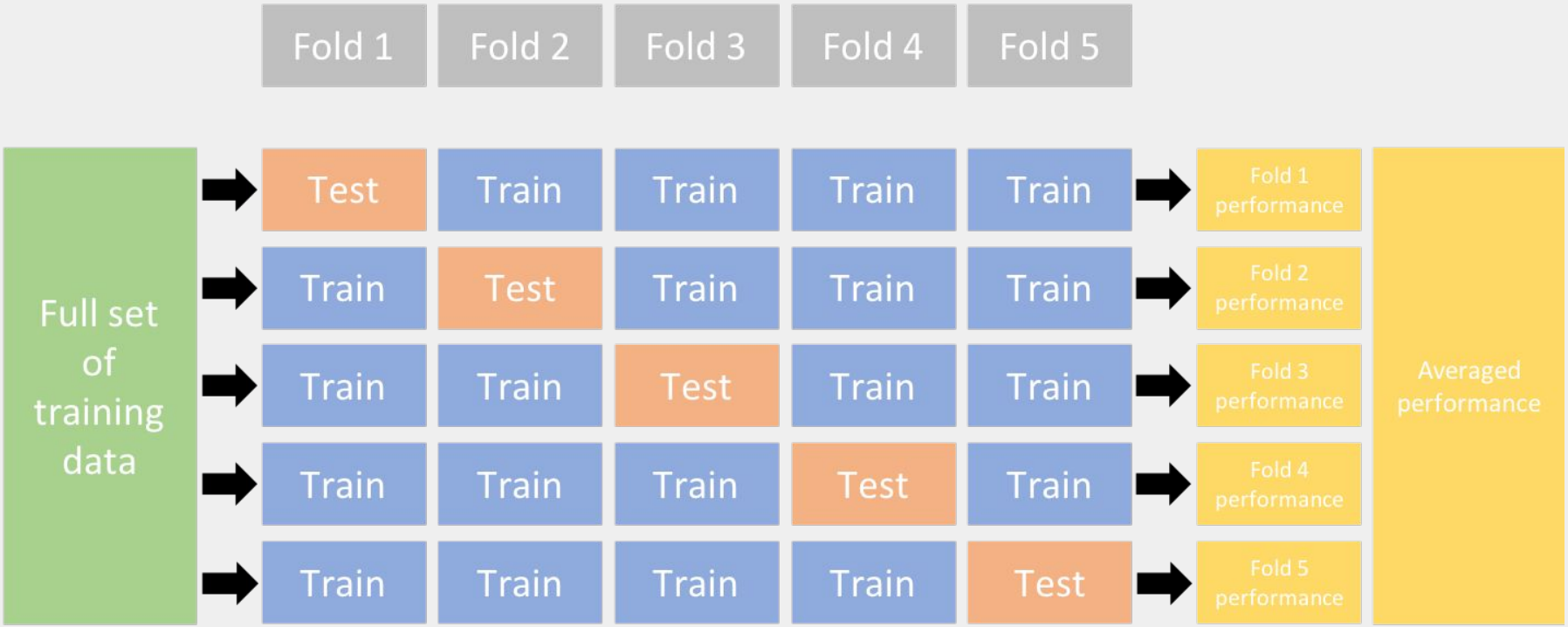
What do we want? Generalization!



ML101 Validation: Simple split



ML101 Validation: K-Fold



Source: **UC Business Analytics R Programming Guide**

ML101 Validation

So after your ML101 classes it may look very clear:

We want generalization, i.e. performing well on unseen data, so:

1) Leave some data out of the training process and pretend it's unseen;
2) Check if the learned model performs well on this unseen data;
3) If it performs reasonably, pick it!
4) Put in production!

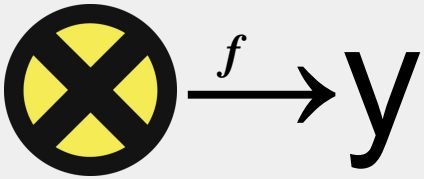
What could possibly go wrong?

Then you go to the real world and...

Real World Supervised Learning

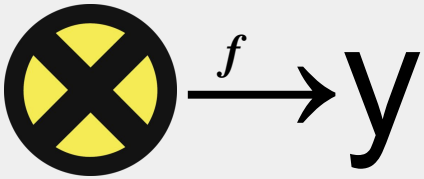
$$\mathbf{X} \xrightarrow{f} \mathrm{y}$$

Real World Supervised Learning



Well, it turns out that in **most of the cases** the $\mathbf{X}$ is **mutant**!

Real World Supervised Learning



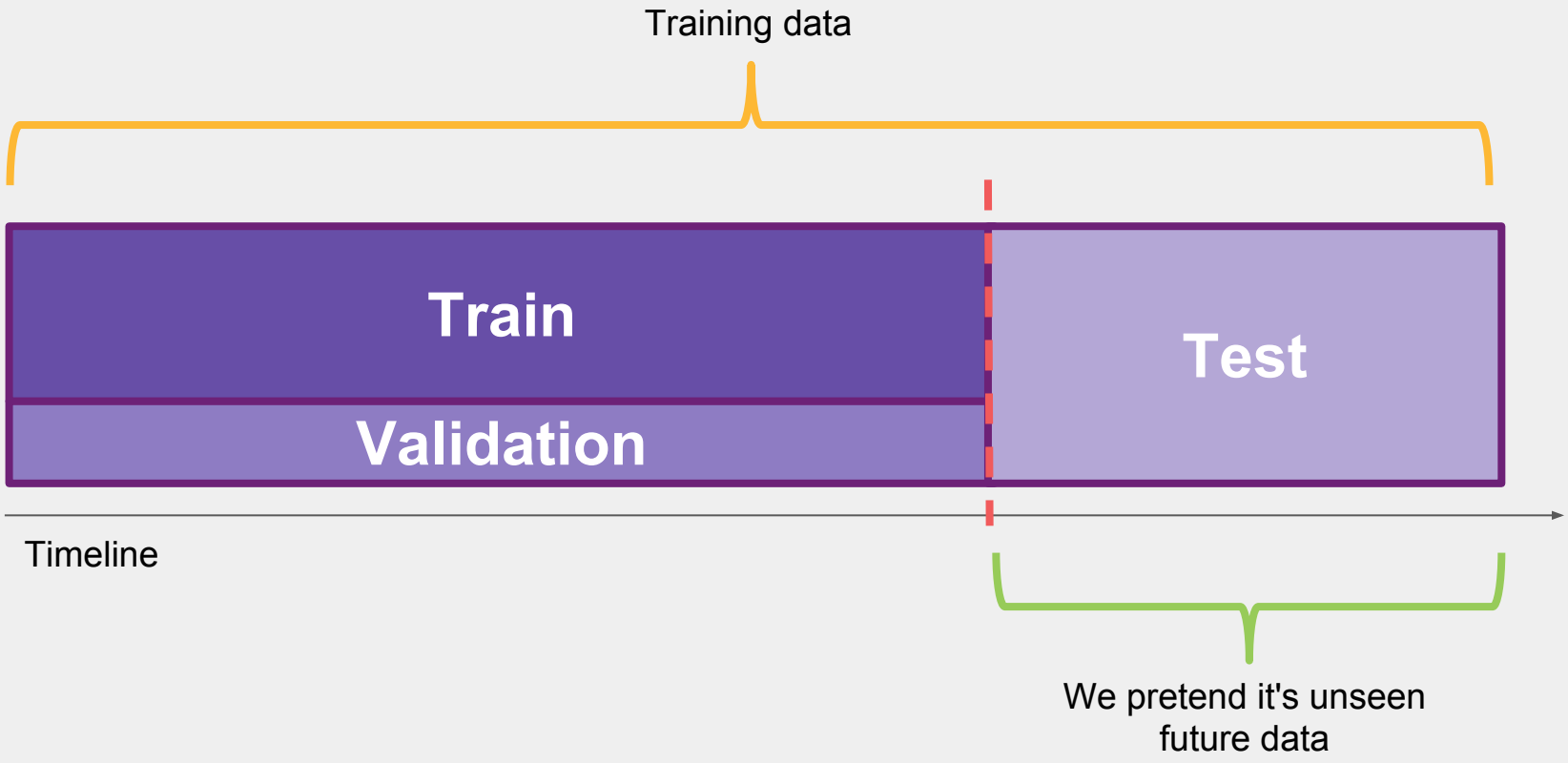
Well, it turns out that in **most of the cases** the $\mathbf{X}$ is **mutant**!

- Temporally
- Spatially

Also:

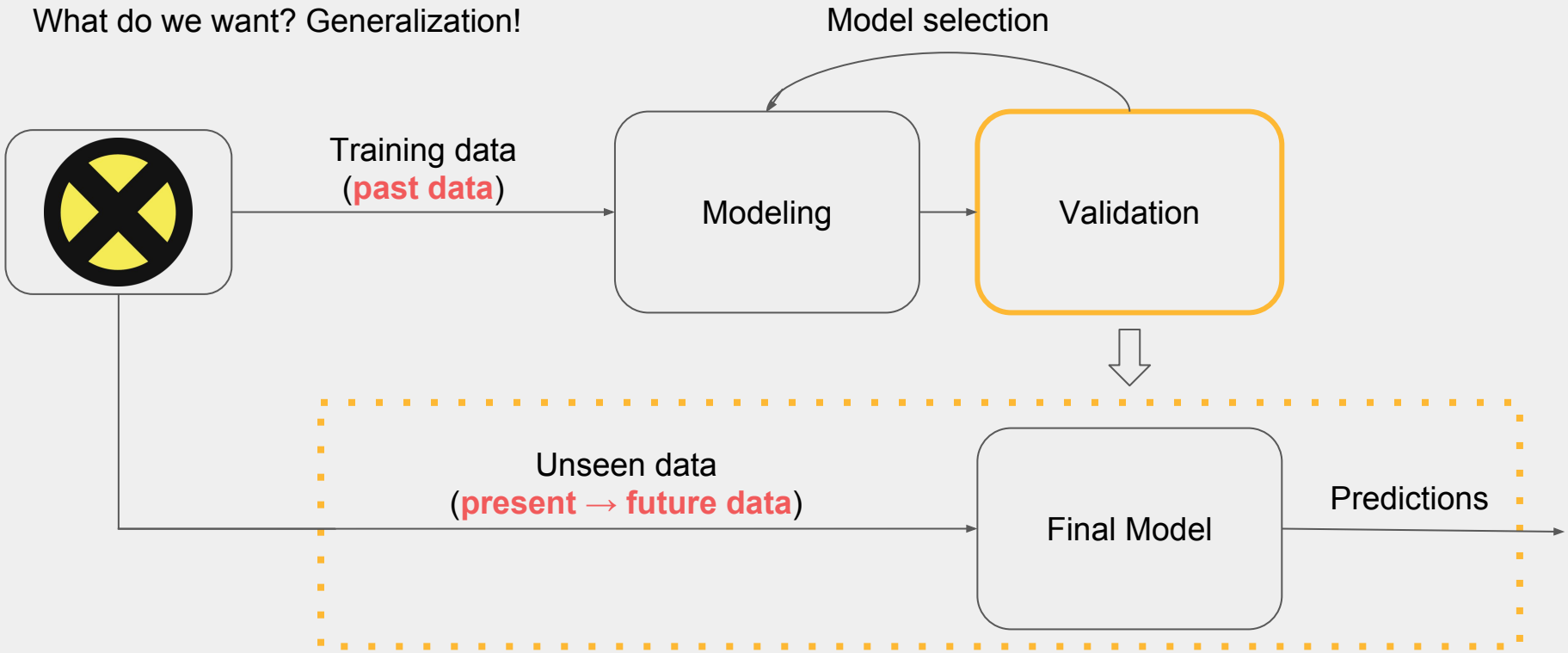
- The training data may be just a subset
- It's not perfectly distributed along its features

Real World Validation: Temporal split



Real World Validation

What do we want? Generalization!



When temporal validation can help us?

Basically, **always**!

All datasets have a temporal aspect because it is generated as the time passes by, but its effect depend on the problem.

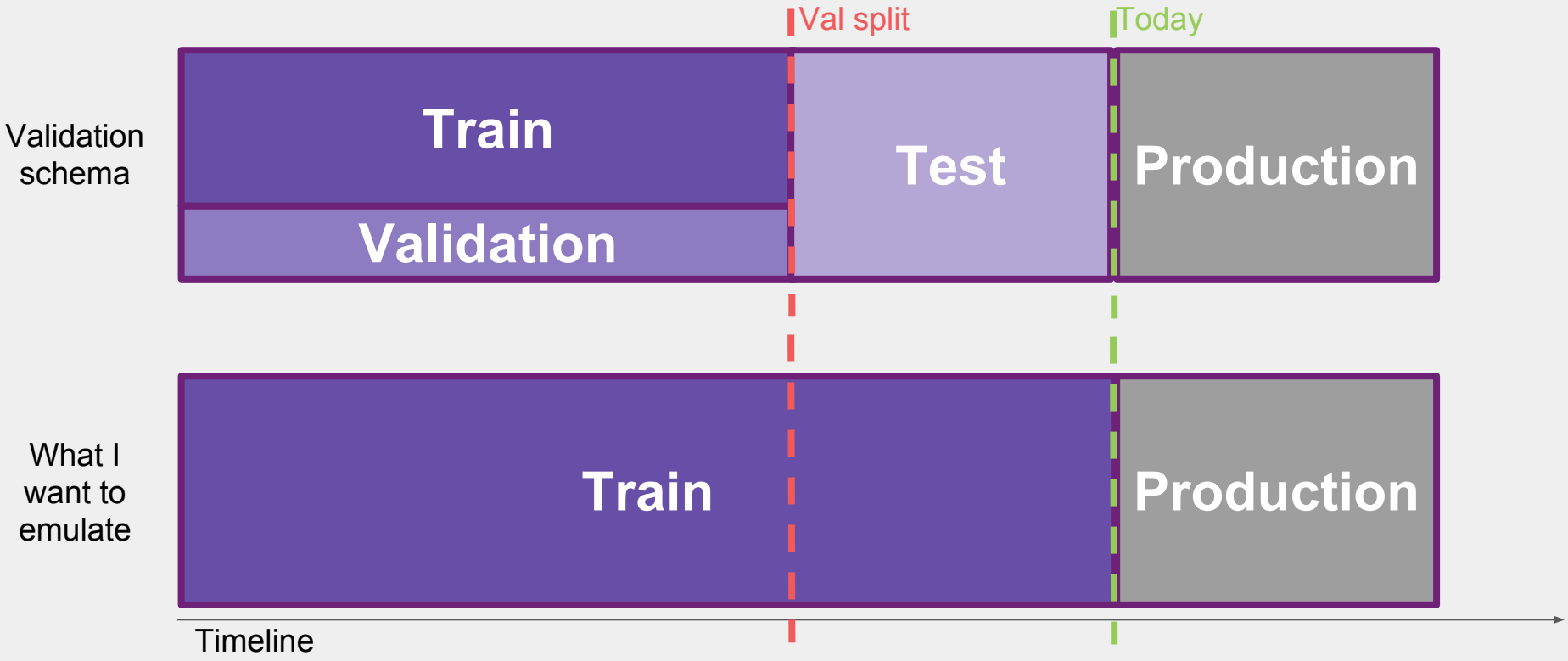
Weak

- Images
- Text

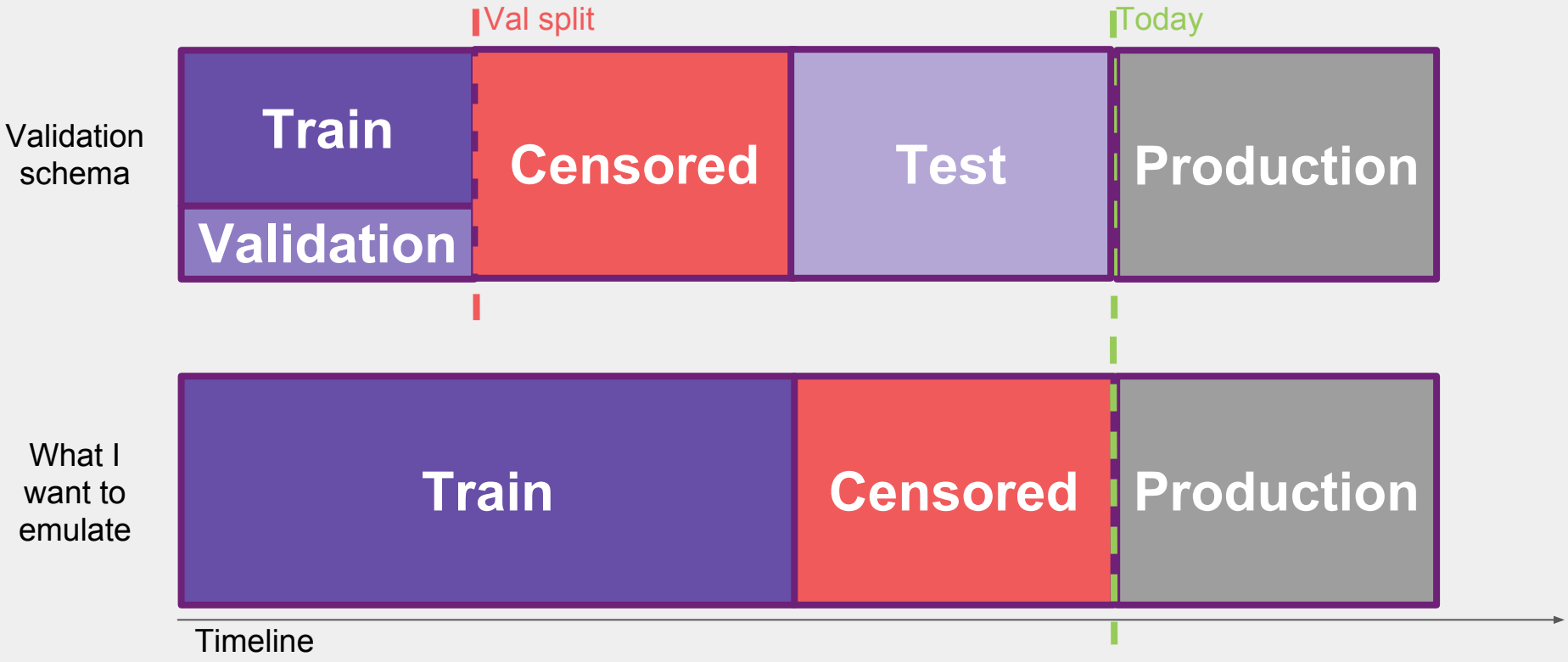
Strong

- Time series
- Tabular data

Real World Problem: target observation



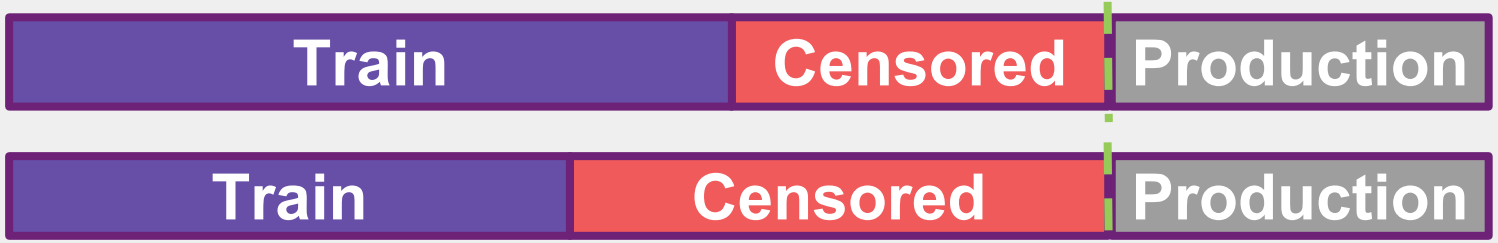
Real World Problem: target observation



Prediction gap

When is it relevant?

Testing different target definition: churn as an inactive user for 5, 10… 60 days. It will change the censored length.



So the validation can mimic the production environment and address the trade-off between **target stability** and **less and older training data**.

Examples: churn, default

Ready to rock!

Ok, let's recapitulate:

- Now you know the **inherent role of time** in every dataset
- You can design a validation schema that considers the **prediction gap**

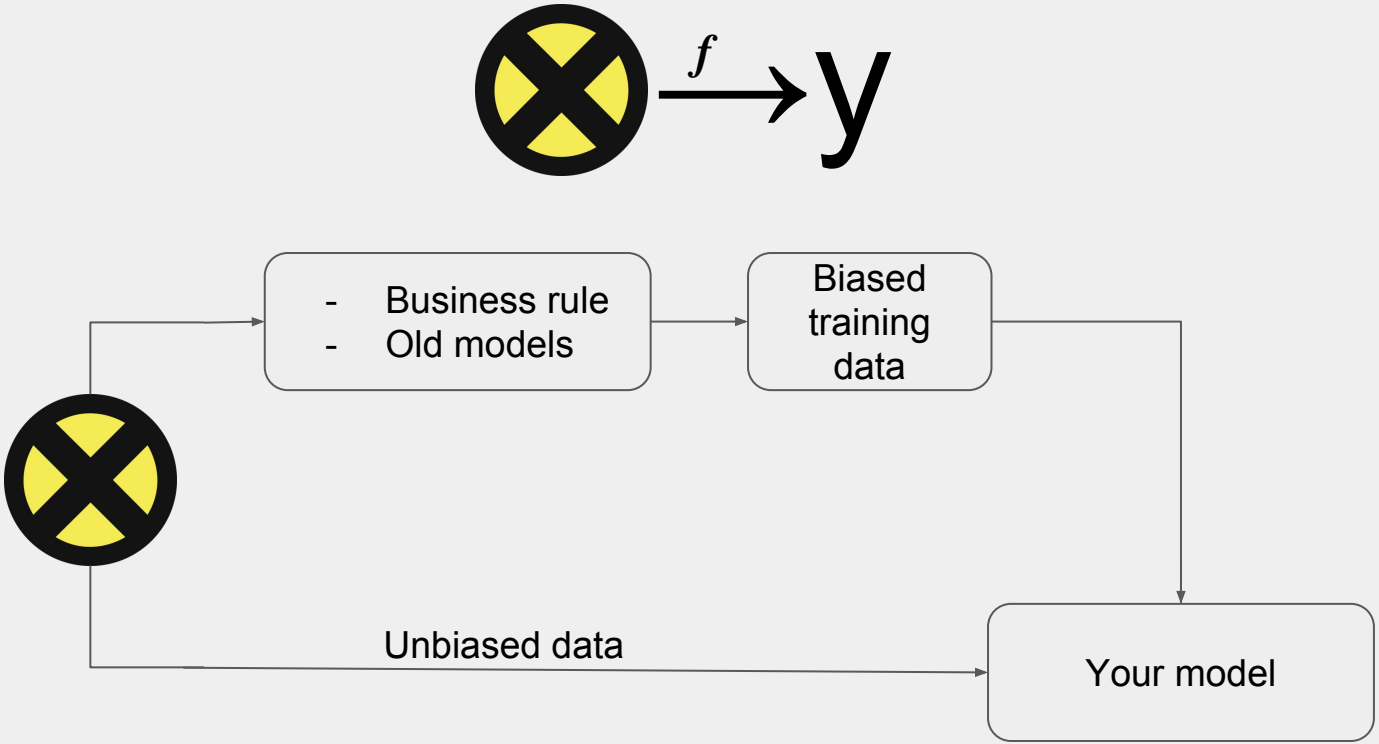
Now you pick a company's problem and ask how they solve it currently.

"We have some rules to decide what to do: we apply some IFs and..."

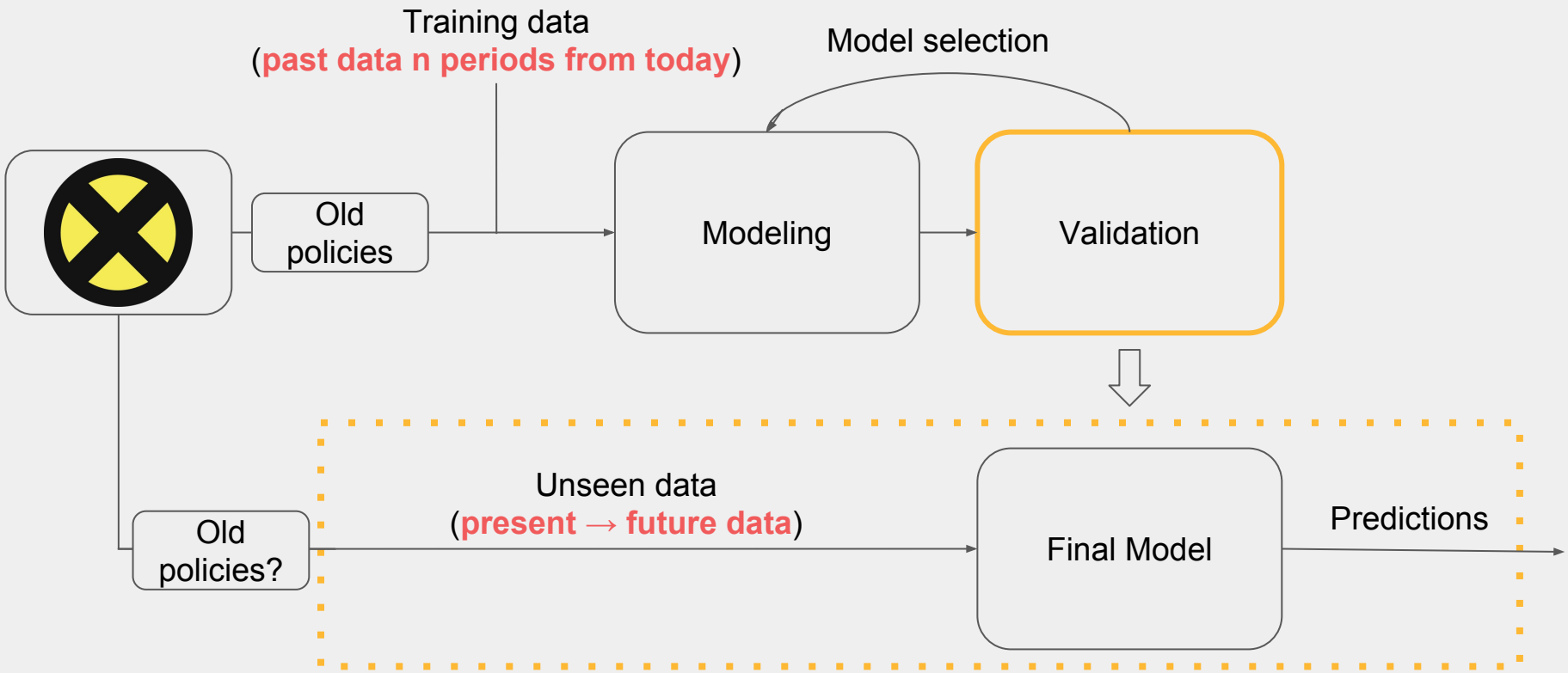
"Oh, do you think you can improve it?"

But your model fails miserably and you don't get what you're missing!

Old policies and models bias



Real World Validation



Engineering and Business

Engineering

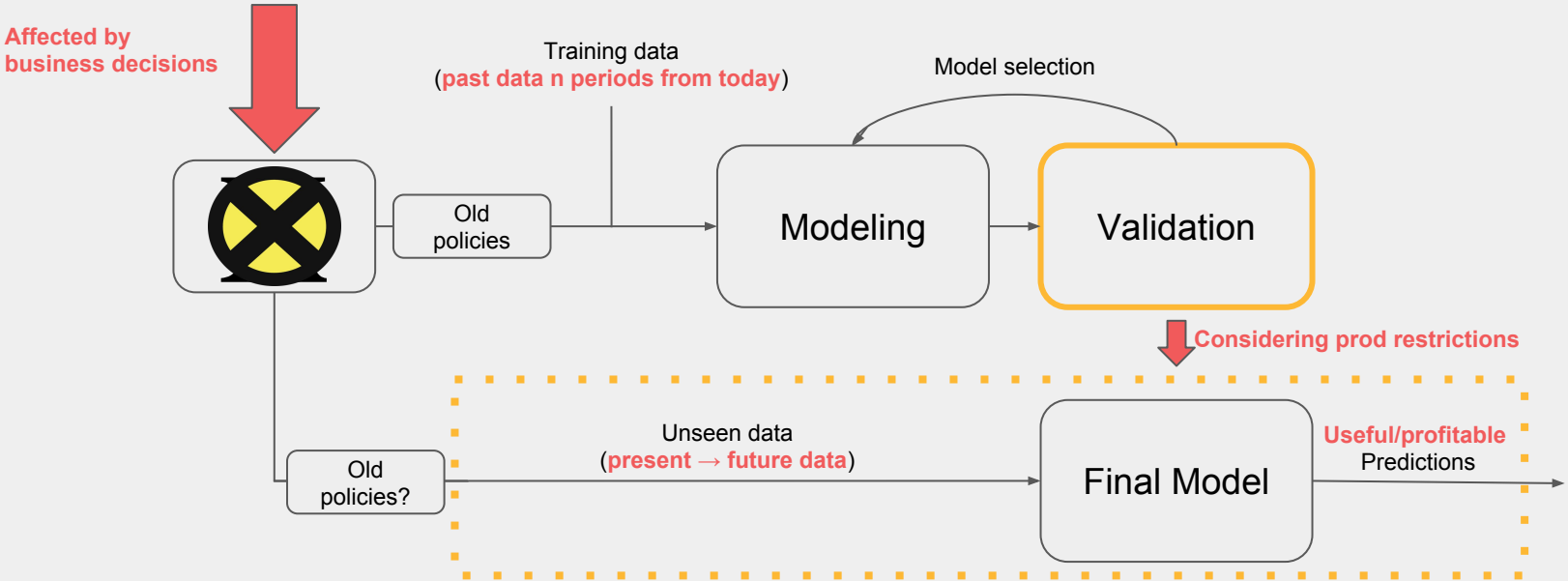
- How often can I update my model?
- Is there any time constraint?

So the production environment we want to validate may become something like "what is the best model considering it can be updated every N periods?"

Business

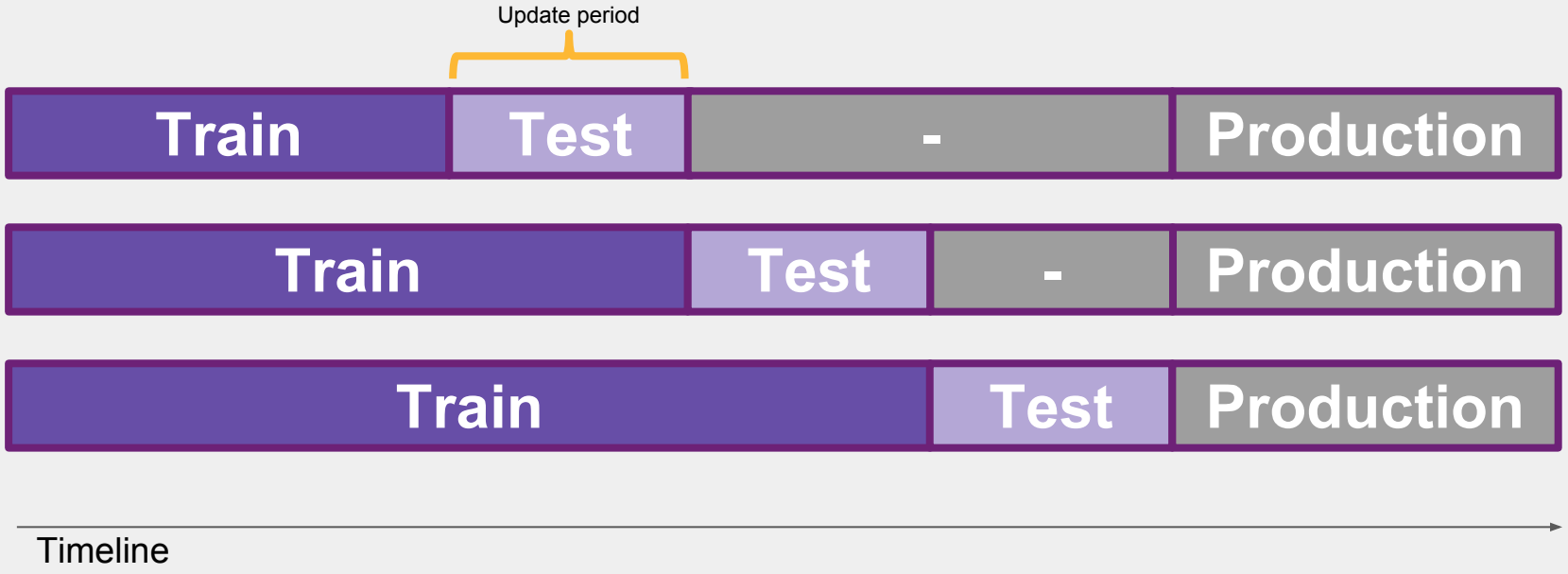
- A lot of things can change the $\mathbf{X}$ distribution:
  - Marketing
  - New products
  - Communication
  - Growth/maturity
- You want to produce meaningful/profitable/useful predictions
- Update and time constraints also
- Model objective

Real World Validation

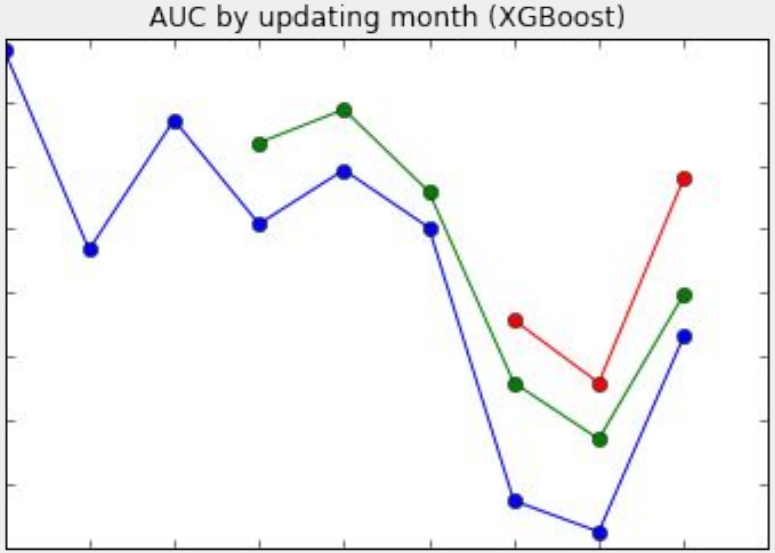


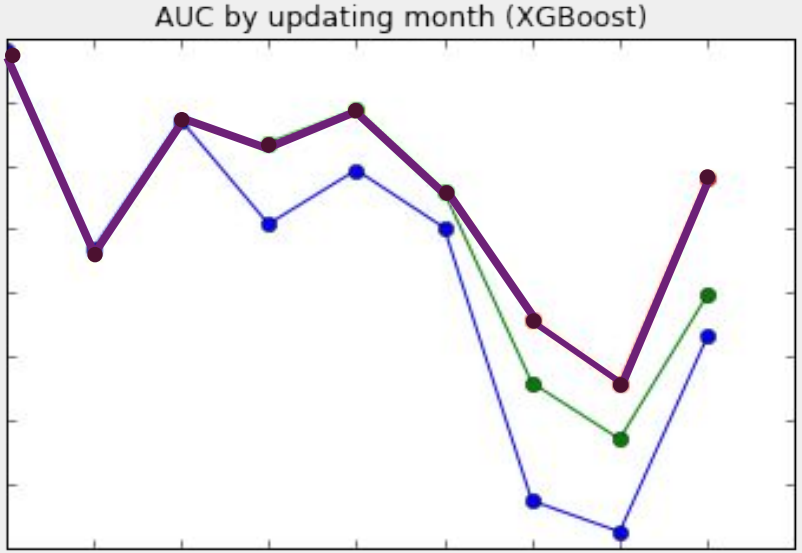
Real World Validation - Engineering

Validate considering update



RW Validation - Engineering example





Business

Business

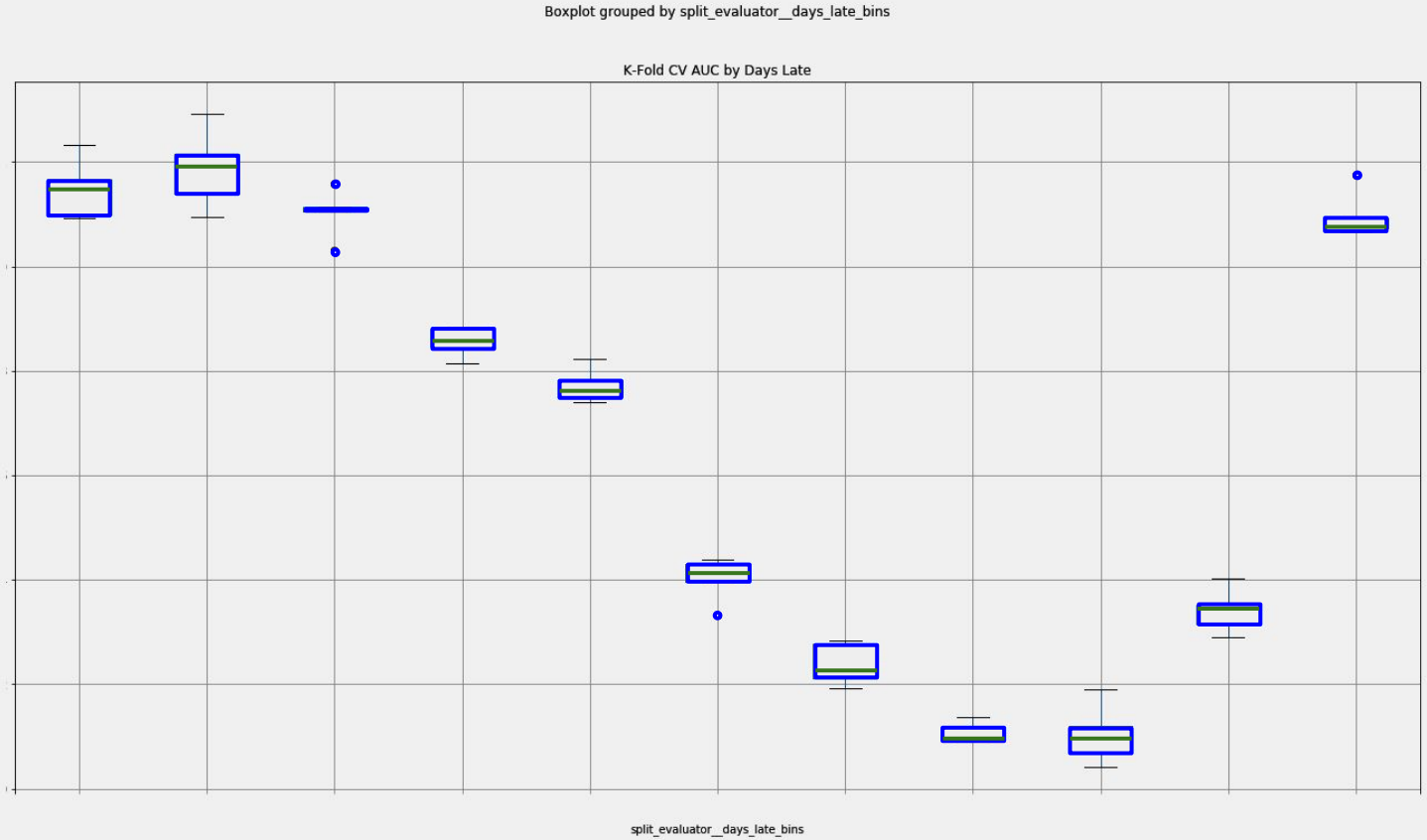
- A lot of things can change the $\mathbf{X}$ distribution
  You can't do anything at validation time, but **monitor**! You shipped something to score over X, but people won't care about, while you should.

- You want to produce meaningful/profitable/useful predictions
  Validate **considering business value**. Split by important features/groups.

- Update and time constraints also:
  Consider model performance x delay to take decisions! **Calculate the monetary trade-off** between them.

- Model objective
  If you know how your data was collected and how your model is going to be applied, it can be a **leverage** instead of a trap.

Real World Validation - Deeper look



So at the end...

X	
Train: A nice and invariant distribution I have a reasonable random sample. **Apply**: In an unseen random sample.	**Train:** Old, far from prediction time, biased by old policies and models, unequally distributed in the features you care about. **Apply:** In an unseen future data I'm not sure about how it's going to change accordingly to time and other business decisions.

Takeaways

It's hard to define a recipe for validation, but keep in mind the general idea of "**mimic the application case**":

- Use a **temporal split**
- Observe the **model degradation** in time
- Consider the **censored period** to observe the target
- Do a internal research about **how the data was collected** to be aware of all the old policies and **its bias**
- Know **how/when** your model is going to be applied
- Consider all the **engineering restrictions and possibilities**
- Think about the **important business aspects** to do a deeper validation
- Be aware in **population shifts** caused by business decisions

Twitter: @lgmoneda
E-mail: lgmoneda@gmail.com
Blog: http://lgmoneda.github.io/

Questions?