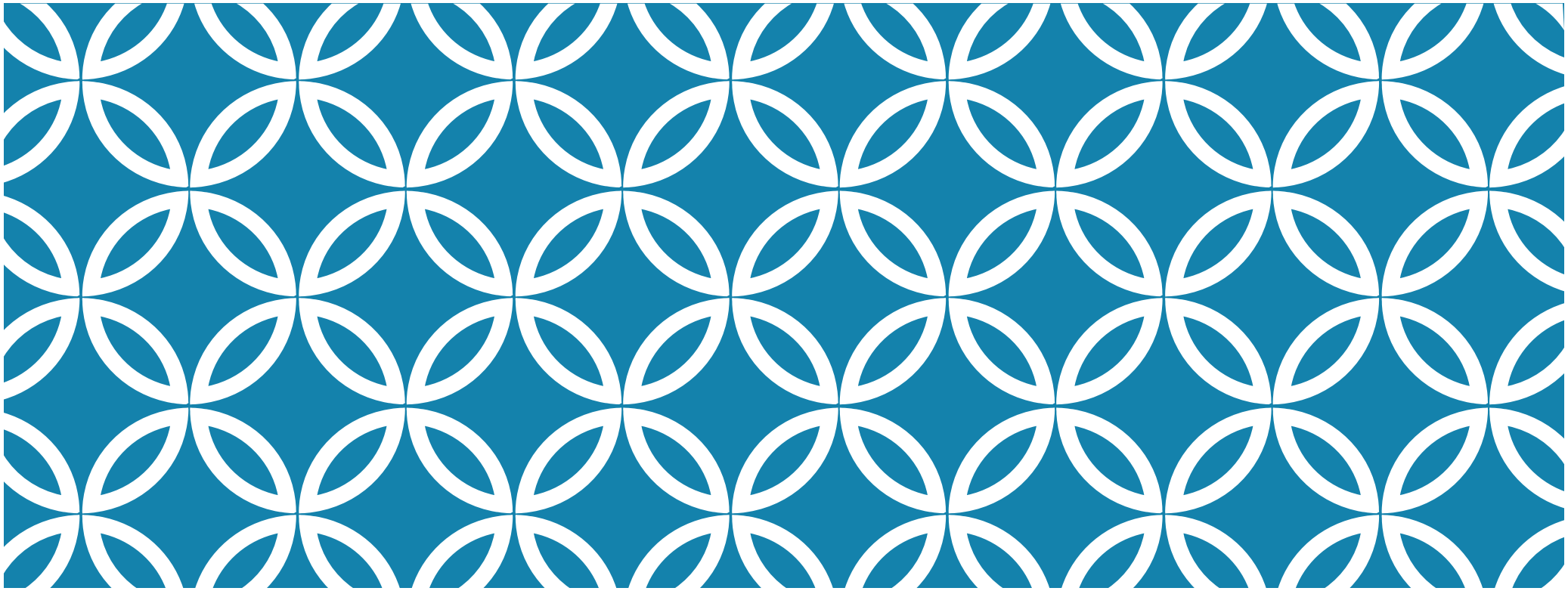
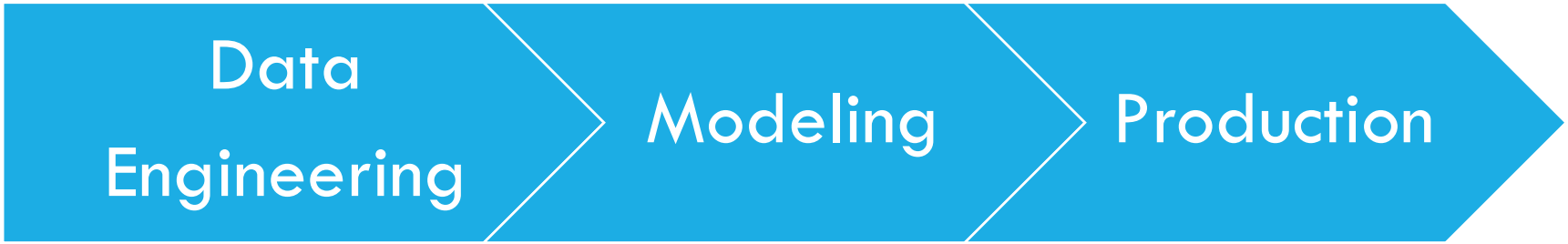


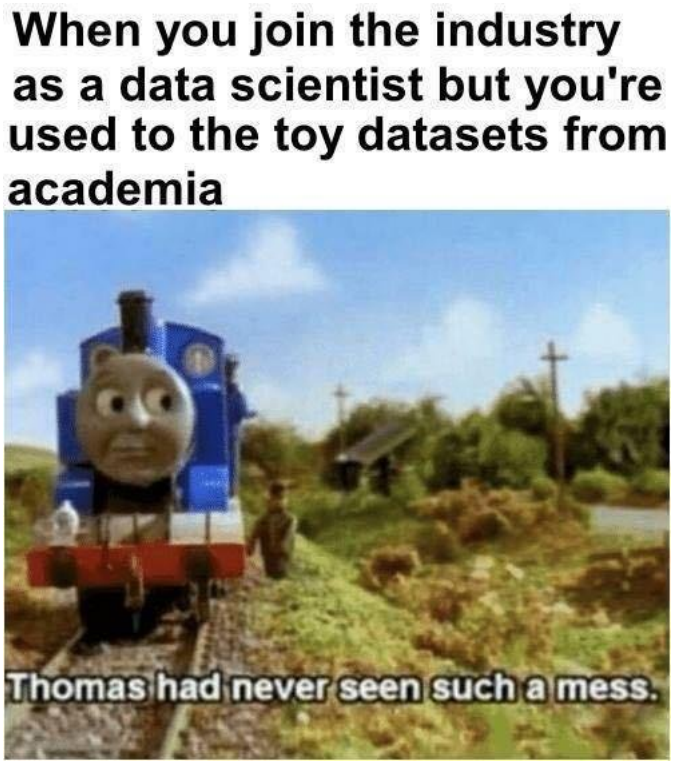
CONSIDERATIONS FOR REAL WORLD PROJECTS

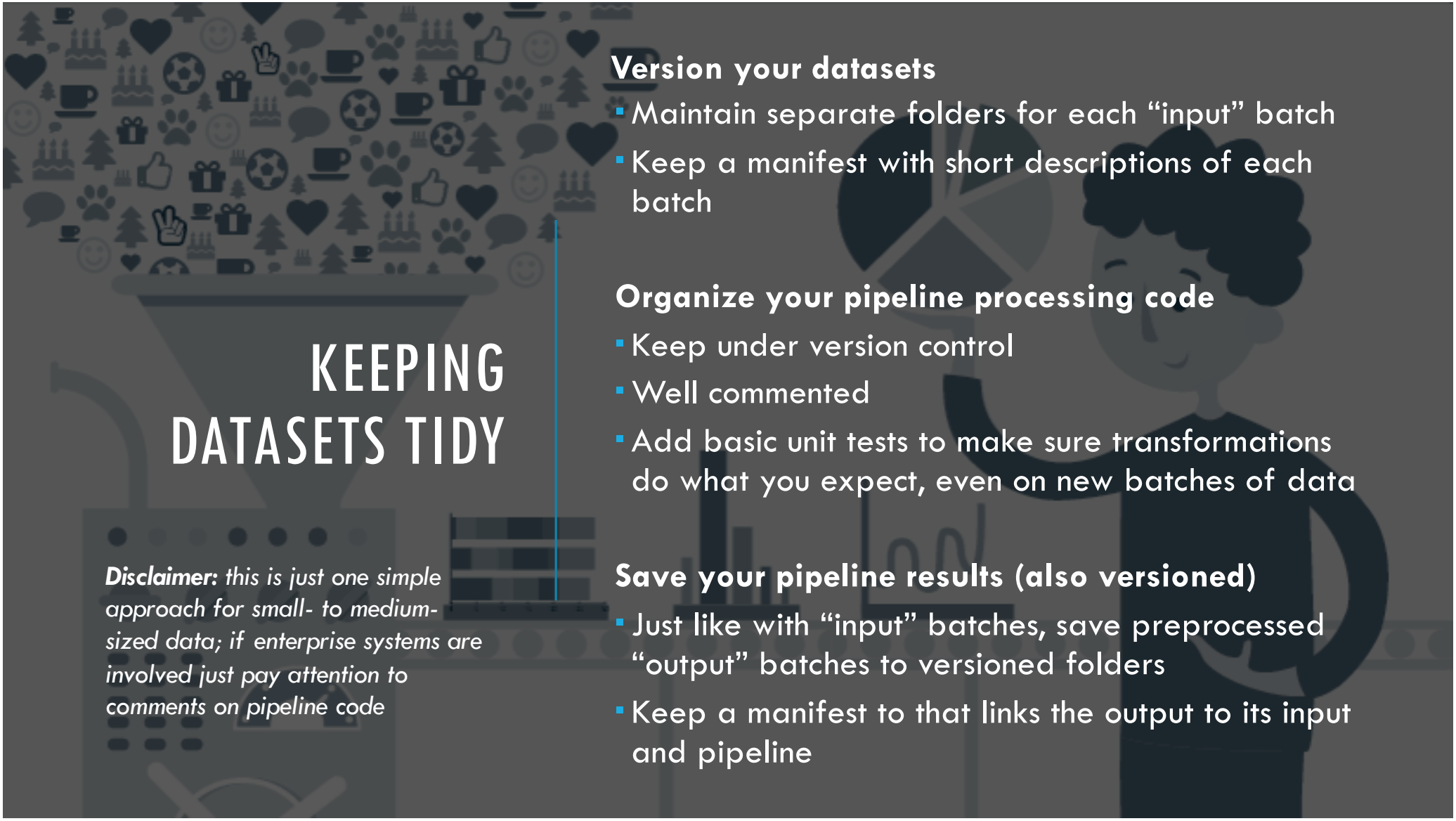
STAGES



REAL DATASETS ARE A MESS

- Real data does not come prepackaged like tutorial data sets
  - Each sample or batch of samples may not follow the same schema, file formats, file name patterns, etc.
- Data often needs to be preprocessed for modeling
  - Each sample or batch may need a different pipeline
  - Different modeling approaches may require different input formats





KEEPING DATASETS TIDY

Disclaimer: *this is just one simple approach for small- to medium-sized data; if enterprise systems are involved just pay attention to comments on pipeline code*

Version your datasets

- Maintain separate folders for each "input" batch
- Keep a manifest with short descriptions of each batch

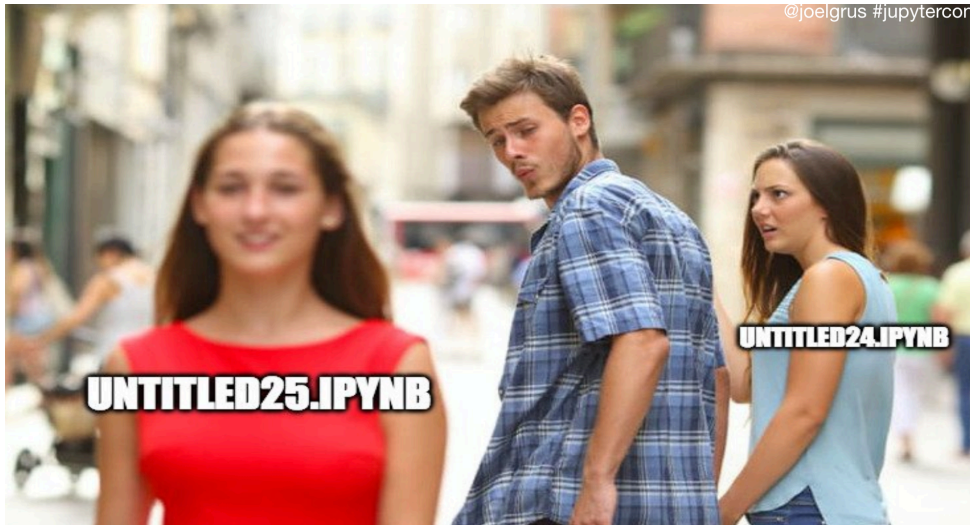
Organize your pipeline processing code

- Keep under version control
- Well commented
- Add basic unit tests to make sure transformations do what you expect, even on new batches of data

Save your pipeline results (also versioned)

- Just like with "input" batches, save preprocessed "output" batches to versioned folders
- Keep a manifest to that links the output to its input and pipeline

MODELING:
CREATING MODELS IS CREATING SOFTWARE



Designing good machine learning systems should follow many good software engineering principles:

- Modularity
- Reusability
- Good Documentation
- Version Control

Notebooks are good for exploration but can hinder good development practices

- See Joel Grus' presentation "I don't like notebooks" for more detail

Source: https://docs.google.com/presentation/d/1n2RlMdmv1p25Xy5thJUhkKGvjtV-dkAIsUXP-AL4ffI/edit?usp=sharing

MODELING:
APPROACH DATA SCIENCE LIKE SCIENCE

Sacred

> *Every experiment is sacred*
> *Every experiment is great*
> *If an experiment is wasted*
> *God gets quite irate*

Make sure your experiments are repeatable

- Documentation for running/installing
- Seed random values
- **Specify versions of libraries used**
- Use modular code under test
- Containers!

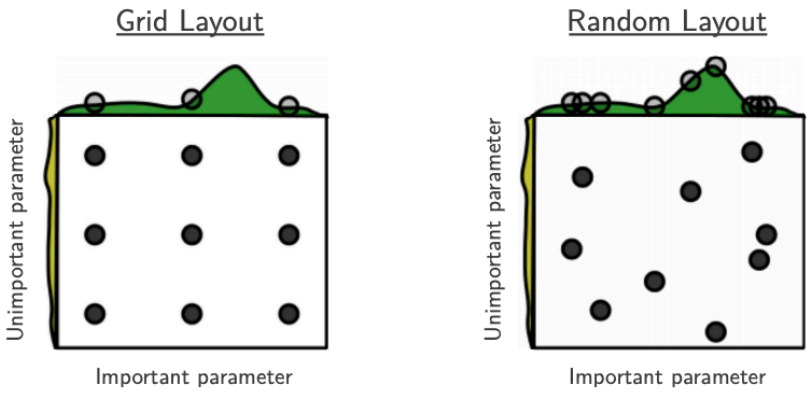
Critically important for Python!

Don't let experiments go to waste

- Save your hyperparameters/learned parameters
- Document your experiments (possibly with semantic names)
- Consider using an experiment management library (i.e., Sacred, Beaker, MLflow

Figure: https://github.com/IDSIA/sacred

MODELING:
HYPERPARAMETER TUNING IS A MUST

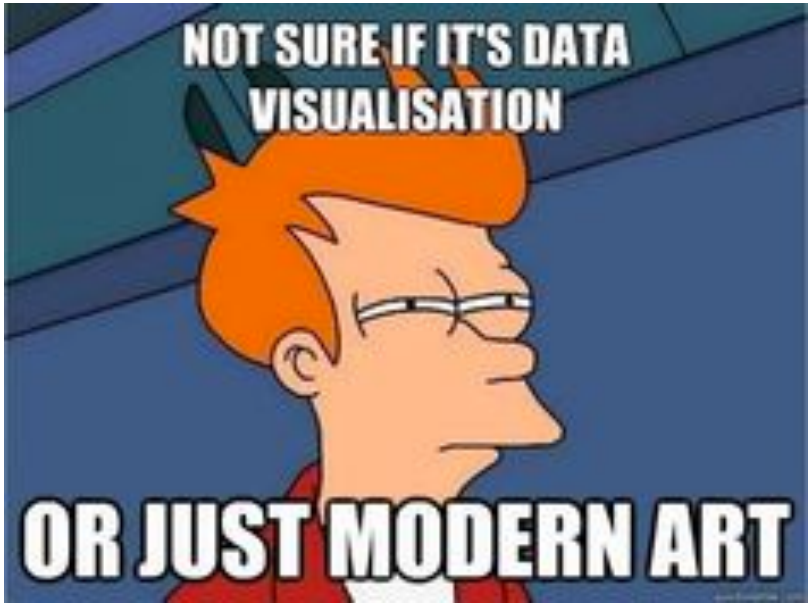


Make sure you tune your hyperparameters

- No one guesses best hyperparameters the first time
- Use a dev set (separate from your test set) for turning
- Using random search is as good as grid search and easy to implement

Figure: http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

PRODUCTION:
GIVE YOUR MODEL A GOOD INTERFACE



Several good libraries for adding a quick interface to show off your model

- Streamlit
- Dash by Plotly
- https://github.com/obazoud/awesome-dashboard

Figure: https://www.pinterest.de/exasolag/fun-about-data/?autologin=true