Responsible Data Science

Taming technical bias

Prof. Julia Stoyanovich

Center for Data Science &
Computer Science and Engineering
New York University

data RESPONSIBLY

NYU

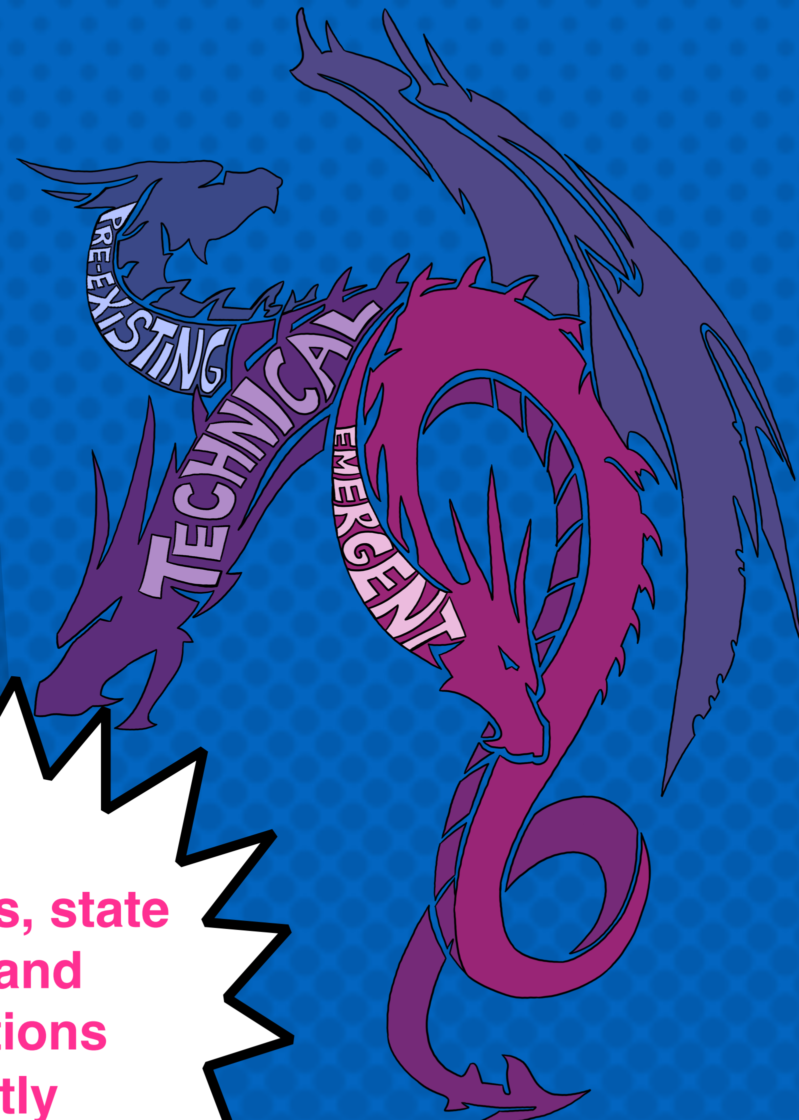
Bias in ADS, revisited

Pre-existing: exists independently of algorithm, has origins in society

Technical: introduced or exacerbated by the technical properties of an ADS

Emergent: arises due to context of use

to fight bias, state beliefs and assumptions explicitly



@FalaahArifKhan

Model development lifecycle

Goal

design a model to predict an appropriate level of compensation for job applicants

Problem

accuracy is lower for middle-aged women - **a fairness concern!**

now what?

demographics

employment

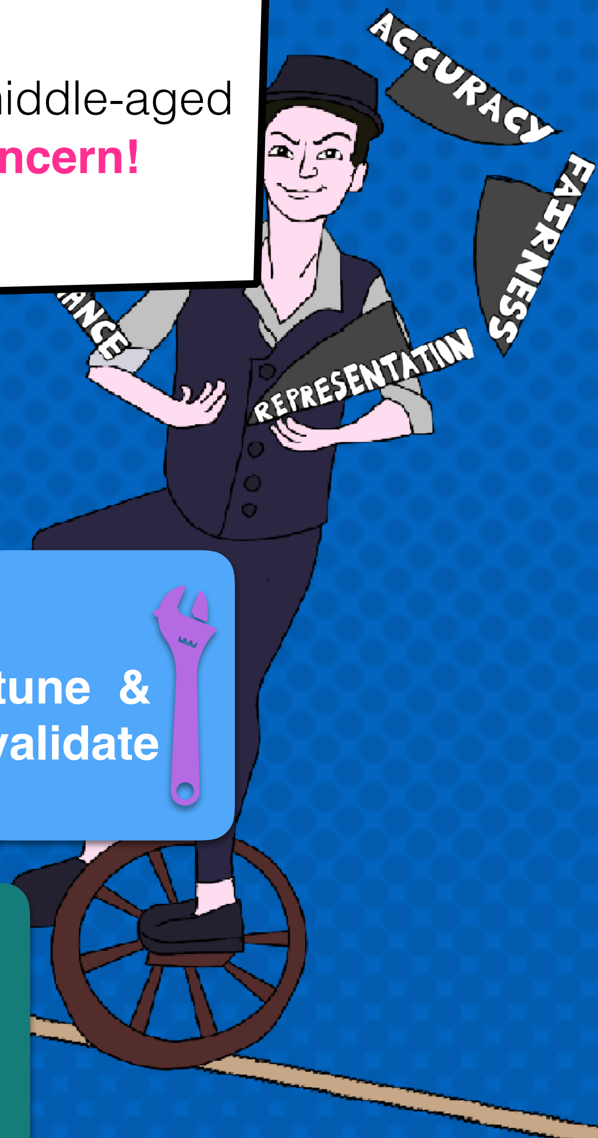
split

preprocess

interpolate missing

select model

tune & validate



[Schelter, He, Khilnani, Stoyanovich (2020)]

dimensions of technical bias

50 shades of null

- **Unknown** - some value definitely belongs here, but I don't know what it is (e.g., unknown birthdate)
- **Inapplicable** - no value makes sense here (e.g., if marital status = single then spouse name should not have a value)
- **Unintentionally omitted** - values is left unspecified unintentionally, by mistake
- **Optional** - a value may legitimately be left unspecified (e.g., middle name)
- **Intentionally withheld** (e.g., an unlisted phone number)
- …..

should we be filling these in? if so, how?

https://www.vertabelo.com/blog/technical-articles/50-shades-of-null-or-how-a-billion-dollar-mistake-has-been-stalking-a-whole-industry-for-decades

Missing value imputation

are values **missing at random** (e.g., gender, age, disability on job applications)?

are we ever interpolating **rare categories** (e.g., Native American)

are **all categories** represented (e.g., non-binary gender)?

how are we evaluating performance of missing value imputation? what's the **performance baseline**?

Data filtering

recall: selection and join in relational algebra; both are “filtering” operations, **can arbitrarily change promotions of protected groups**

select by zip code, country, years of C++ experience, others?

another example: using **pre-trained word embeddings**

age_group	county
60	CountyA
60	CountyA
20	CountyA
60	CountyB
20	CountyB
20	CountyB

50% vs 50%

→

age_group	county
60	CountyA
60	CountyA
20	CountyA

66% vs 33%

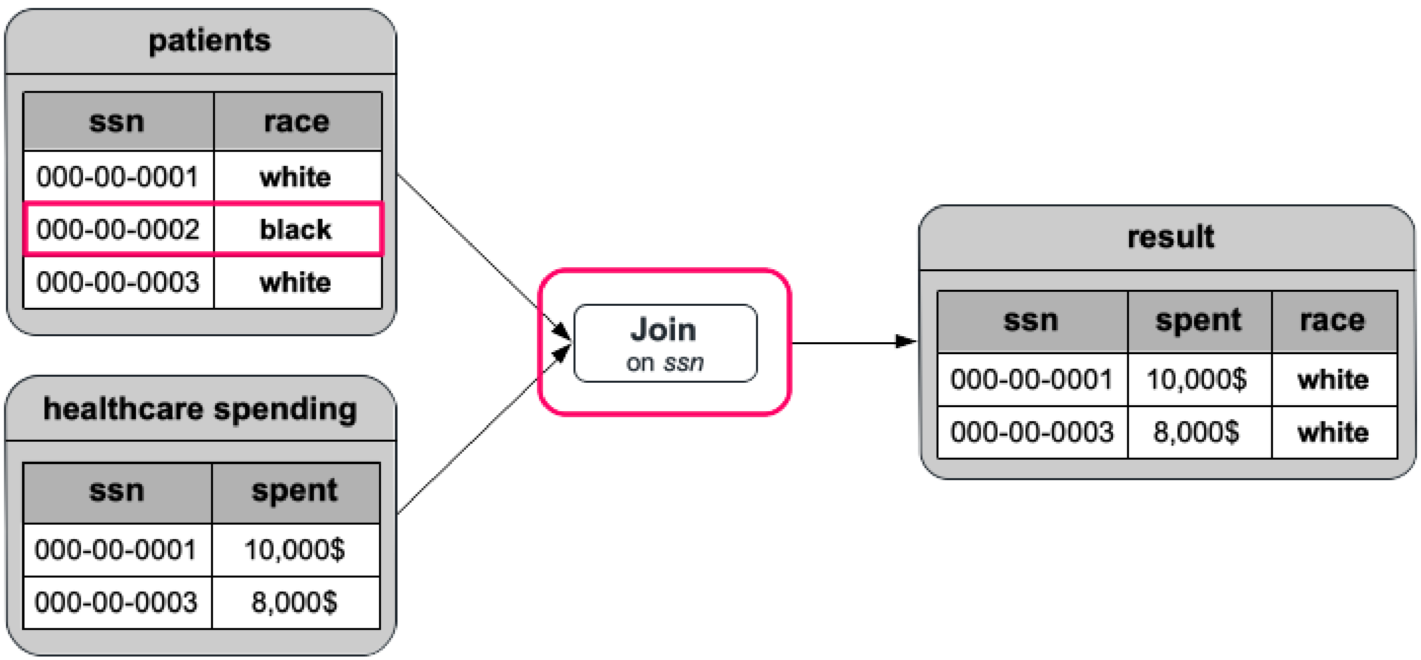
[Grafberger, Stoyanovich, Schelter (2021)]

Data filtering

recall: selection and join in relational algebra; both are “filtering” operations, **can arbitrarily change promotions of protected groups**

select by zip code, country, years of C++ experience, others?

another example: using **pre-trained word embeddings**



[Grafberger, Stoyanovich, Schelter (2021)]

Data debugging: mlinspect

Potential issues in preprocessing pipeline:

1. Join might change proportions of groups in data
2. Column 'age_group' projected out, but required for fairness
3. Selection might change proportions of groups in data
4. Imputation might change proportions of groups in data
5. 'race' as a feature might be illegal!
6. Embedding vectors may not be available for rare names!

Python script for preprocessing, written exclusively with native pandas and sklearn constructs

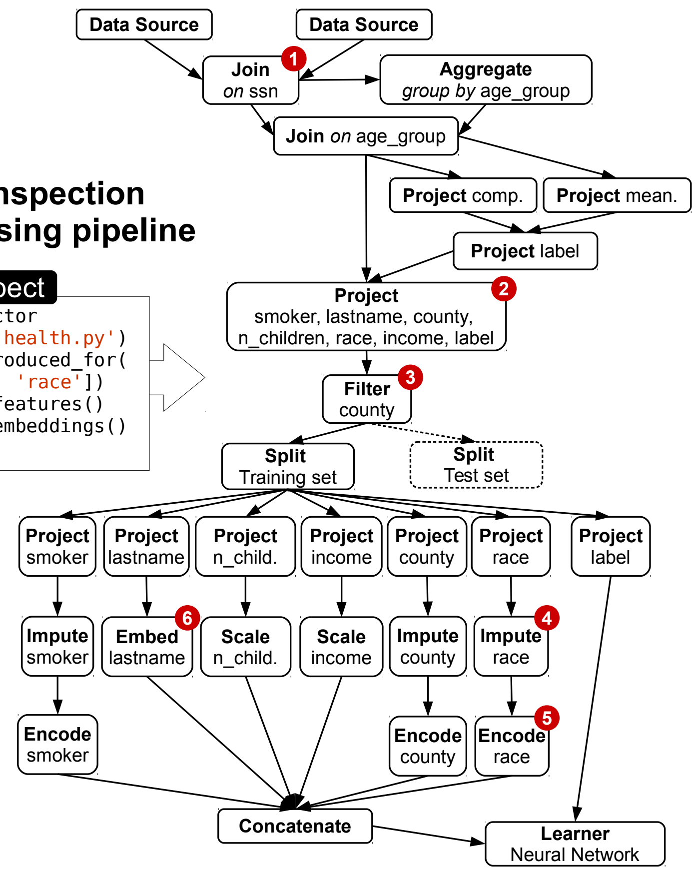
```
# load input data sources, join to single table
patients = pandas.read_csv(…)
histories = pandas.read_csv(…)
data = pandas.merge([patients, histories], on=['ssn'])

# compute mean complications per age group, append as column
complications = data.groupby('age_group')
 .agg(mean_complications=('complications','mean'))
data = data.merge(complications, on=['age_group'])

# Target variable: people with frequent complications
data['label'] = data['complications'] >
  1.2 * data['mean_complications']

# Project data to subset of attributes, filter by counties
data = data[['smoker', 'last_name', 'county',
             'num_children', 'race', 'income', 'label']]
data = data[data['county'].isin(counties_of_interest)]

# Define a nested feature encoding pipeline for the data
impute_and_encode = sklearn.Pipeline([
  (sklearn.SimpleImputer(strategy='most_frequent')),
  (sklearn.OneHotEncoder())])
featurisation = sklearn.ColumnTransformer(transformers=[
  (impute_and_encode, ['smoker', 'county', 'race']),
  (Word2VecTransformer(), 'last_name')
  (sklearn.StandardScaler(), ['num_children', 'income']])

# Define the training pipeline for the model
neural_net = sklearn.KerasClassifier(build_fn=create_model())
pipeline = sklearn.Pipeline([
  ('features', featurisation),
  ('learning_algorithm', neural_net)])

# Train-test split, model training and evaluation
train_data, test_data = train_test_split(data)
model = pipeline.fit(train_data, train_data.label)
print(model.score(test_data, test_data.label))
```

Declarative inspection of preprocessing pipeline

mlinspect

```
PipelineInspector
.on_pipeline('health.py')
 .no_bias_introduced_for(
 ['age_group', 'race'])
 .no_illegal_features()
 .no_missing_embeddings()
.verify()
```

Corresponding dataflow DAG for instrumentation, extracted by *mlinspect*



[Grafberger, Stoyanovich, Schelter (2021)]

Data debugging: mlinspect

- similar to code inspection in modern IDEs, but specifically for data
- works on existing pipeline code using libraries like pandas and scikit-learn
- negligible performance overhead

ACM SIGMOD 2021 demo (4 min)

https://surfdrive.surf.nl/files/index.php/s/ybriyzsdc6vcd2w

CIDR 2021 talk (10 min)

https://www.youtube.com/watch?v=Ic0aD6lv5h0

https://github.com/stefan-grafberger/mlinspect

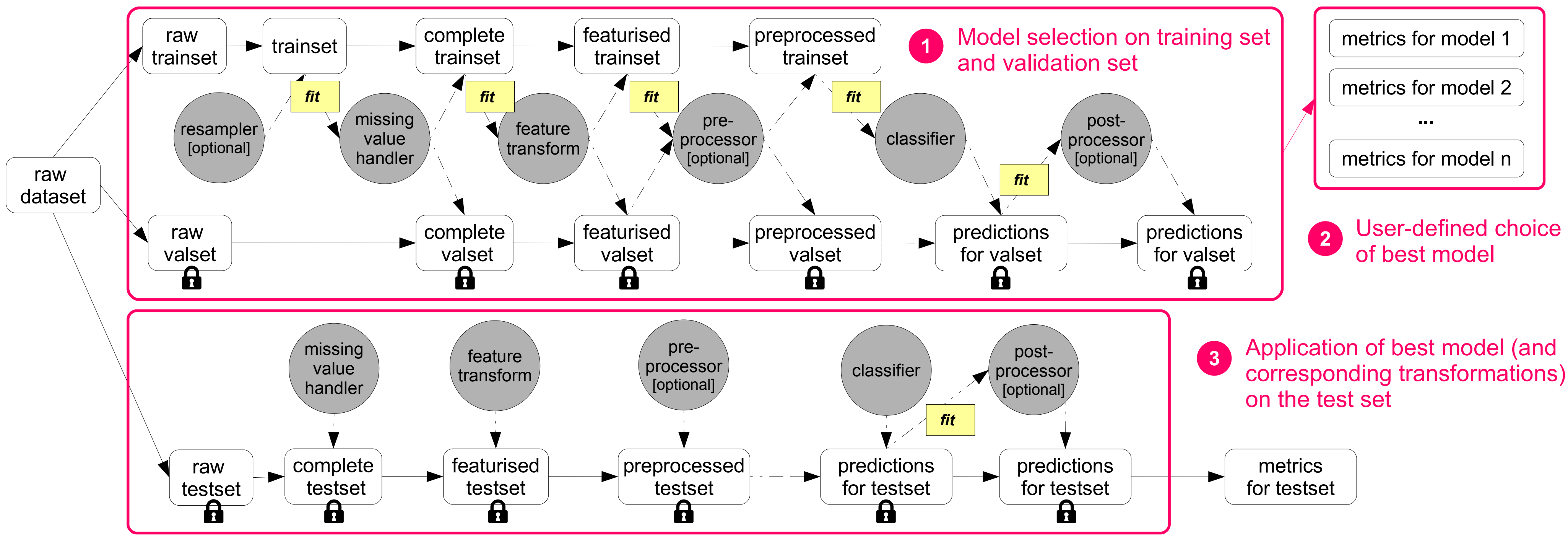
Sound experimentation

"A theory or idea shouldn't be scientific unless it could, in principle, be proven false."

Karl Popper

- software-engineering and data science best-practices
- data isolation: training / validation / test
- accounting for **variability** when observing trends
- tuning hyper-parameters: **for what objective**?

Sounds experimentation: FairPrep



[Schelter, He, Khilnani, Stoyanovich (2020)]

module take-aways

Automated Decision Systems (ADS)

Automated Decision Systems (ADS)

process data about people

help make consequential decisions

combine human & automated decision making

aim to improve **efficiency** and promote **equity**

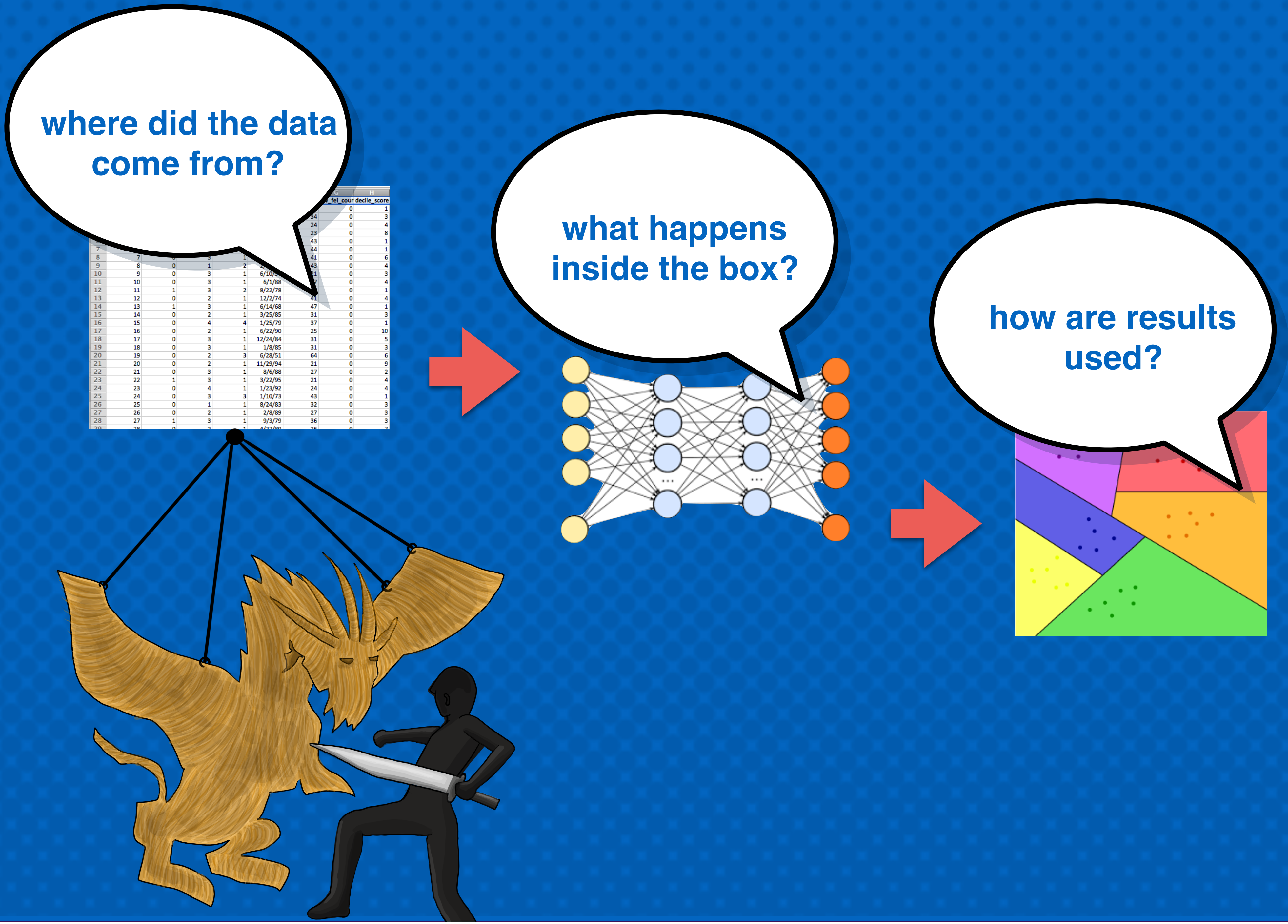
are subject to **auditing** and **public disclosure**

may or may not use AI

may or may not have autonomy

rely heavily on data

Fair-ML view: fighting a paper dragon?



Understand your data!

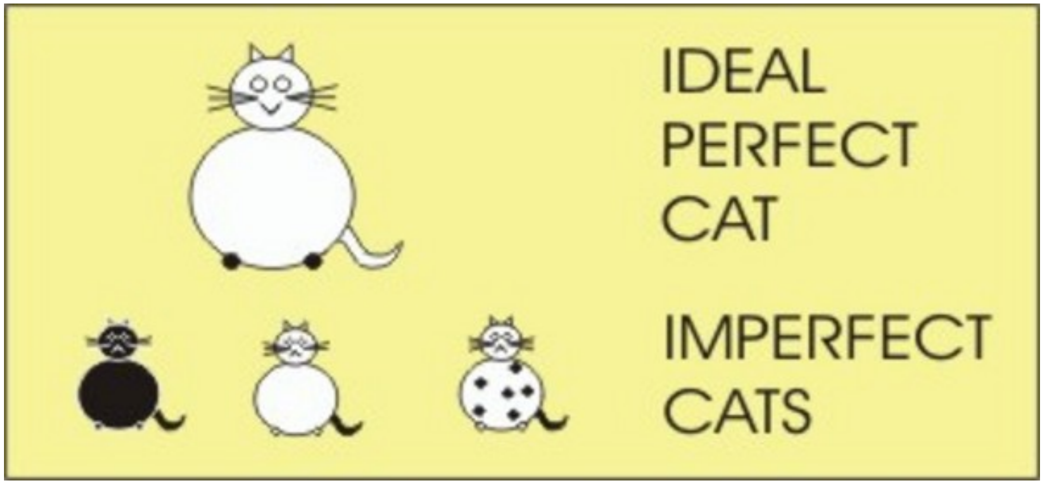
Need **metadata** to:

- enable data **re-use** (have to be able to find it!)
- determine **fitness for use** of a dataset in a task
- help establish **trust** in the data analysis process and its outcomes

Data is considered to be of high quality if it's "**fit for intended uses** in operations, decision making and planning"

[Thomas C. Redman, "Data Driven: Profiting from Your Most Important Business Asset." 2013]

DB (databases) vs. DS (data science)



https://midnightmediamusings.wordpress.com/2014/07/01/plato-and-the-theory-of-forms/

- **DB**: start with the schema, admit only data that fits; iterative refinement is possible, and common, but we are still schema-first
- **DS**: start with the data, figure out what schema it fits, or almost fits - reasons of usability, repurposing, low start-up cost

the "right" approach is somewhere between these two, **data profiling aims to bridge** between the two world views / methodologies

module 3: data protection & privacy