Applied Machine Learning #2

Manoel Horta Ribeiro
manoel@cs.princeton.edu

COS 424 / Fall 2026



MENU

APPETIZER

Quiz time!

MAIN COURSE

Unsupervised learning

DESSERT

A bag of tricks



Quizz Time!





MENU

APPETIZER
Quiz time!

MAIN COURSE
Unsupervised learning

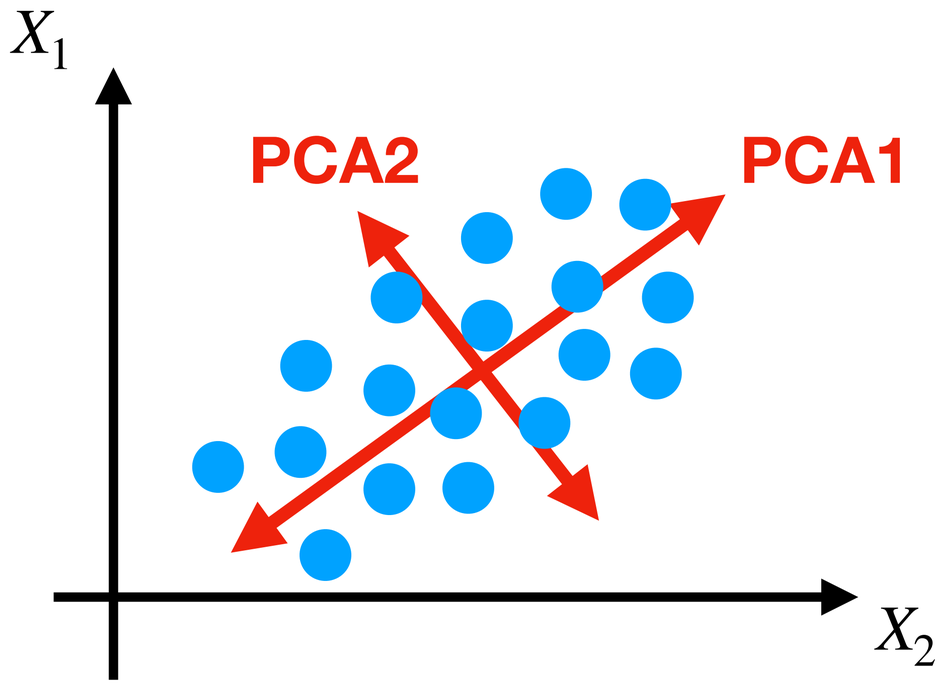
DESSERT
A bag of tricks

Unsupervised learning

- Consider *features* $\{X_1, \ldots, X_n\}$ ~~and an *outcome* Y~~
- Given a sample $\{x_1^i, x_2^i, \ldots, x_n^i, y_i\}$, learn $f(\mathbf{X}) \simeq Y$ such that Y is a "simpler" representation of the data
- We want $f(\mathbf{X})$ to make our life easier; e.g., give us insight
  - If Y is continuous $\rightarrow$ **"dimensionality reduction,"**
  - If Y is categorical $\rightarrow$ **"clustering."**

Dimensionality Reduction: Principal Component Analysis

- PCA finds new variables (principal components) that
  - …are linear combinations of variables!
  - …capture most of the variance!
  - … mutually orthogonal (uncorrelated)!
- Based on elegant linear algebra!
  - The principal components are the eigenvectors of the covariance matrix!



Why do dimensionality reduction?

- **Back in the day…** Dimensionality reduction improved ML performance (e.g., for k-NN or SVMs).
- For modern methods, e.g., XGBoost, dimensionality reduction does not matter that much!
- So what is it used for?
  - … visualize high-dimensional data.
  - … compress features

Example: World Value Survey

- Large survey administered across 120+ countries!
- Items ranging from demographics and ethical values to well-being and stereotypes.
- How to make sense of all this data?

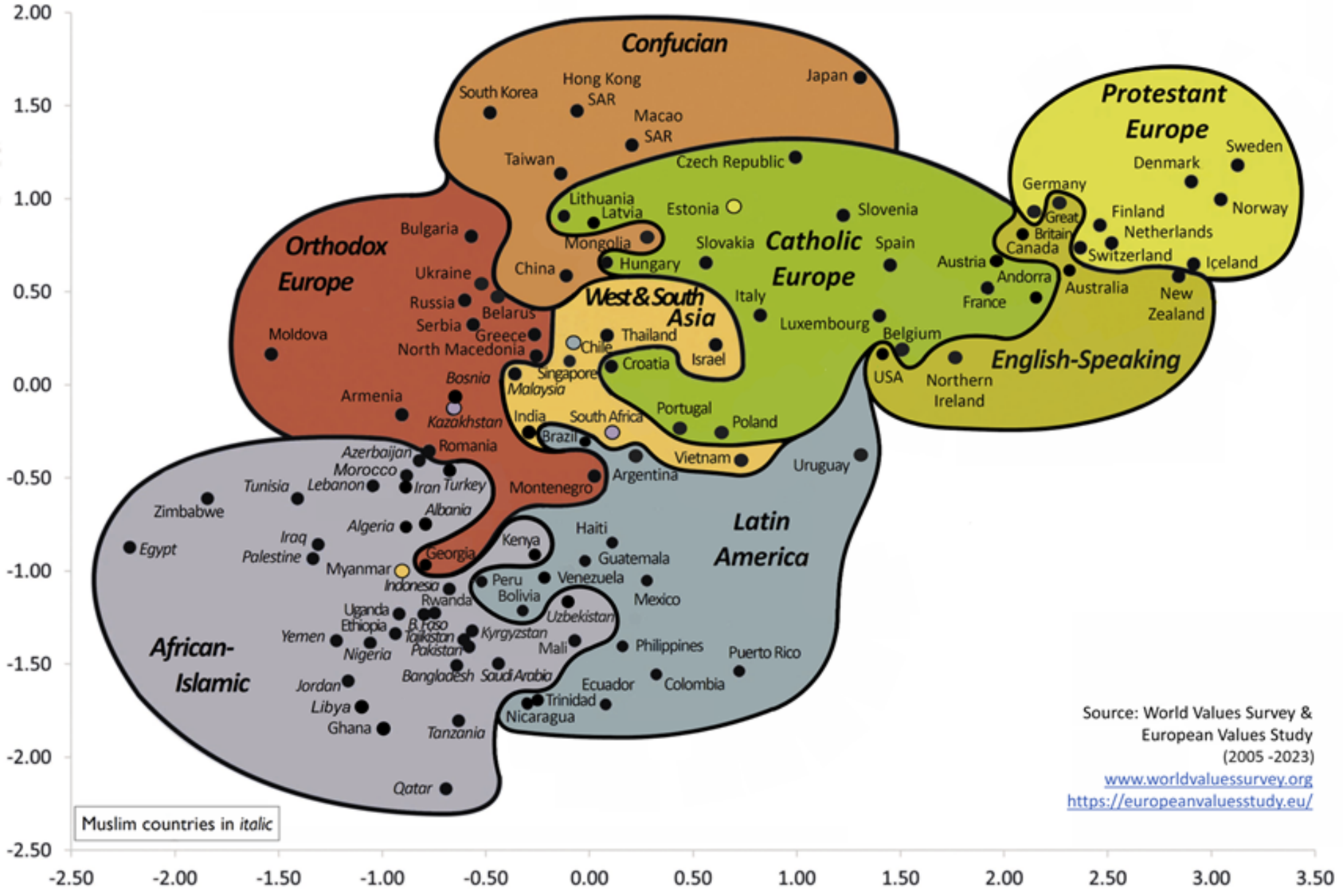
ETHICAL VALUES AND NORMS (16)

(***SHOW CARD 15***)

Please tell me for each of the following actions whether you think it can always be justified, never be justified, or something in between, using this card. (*Read out and code one answer for each statement*):

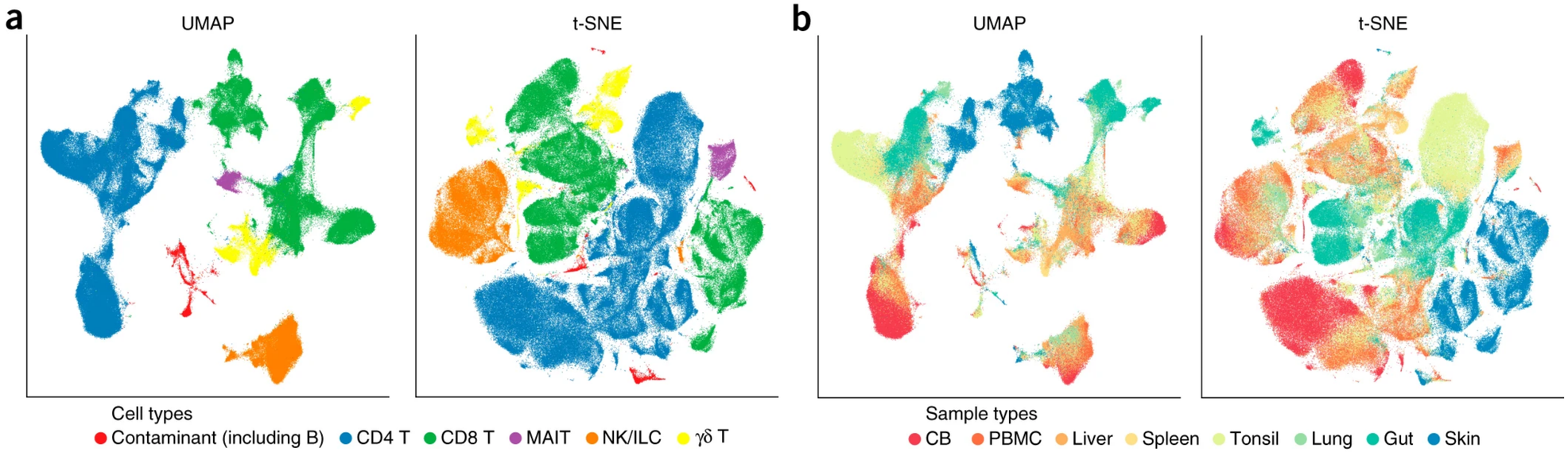
		Never justifiable									Always justifiable
Q130	Claiming government benefits to which you are not entitled	1	2	3	4	5	6	7	8	9	10
Q131	Avoiding a fare on public transport	1	2	3	4	5	6	7	8	9	10
Q132	Stealing property	1	2	3	4	5	6	7	8	9	10
Q133	Cheating on taxes if you have a chance	1	2	3	4	5	6	7	8	9	10
Q134	Someone accepting a bribe in the course of their duties	1	2	3	4	5	6	7	8	9	10
Q135	Homosexuality	1	2	3	4	5	6	7	8	9	10
Q136	Abortion	1	2	3	4	5	6	7	8	9	10
Q137	Divorce	1	2	3	4	5	6	7	8	9	10
Q138	Sex before marriage	1	2	3	4	5	6	7	8	9	10
Q139	Suicide	1	2	3	4	5	6	7	8	9	10
Q140	Euthanasia	1	2	3	4	5	6	7	8	9	10
Q141	For a man to beat his wife	1	2	3	4	5	6	7	8	9	10
Q142	Parents beating children	1	2	3	4	5	6	7	8	9	10
Q143	Violence against other people	1	2	3	4	5	6	7	8	9	10
Q144	Political violence	1	2	3	4	5	6	7	8	9	10
Q145	Death penalty	1	2	3	4	5	6	7	8	9	10

The Inglehart-Welzel World Cultural Map 2023



Other popular methods: UMAP and t-SNE

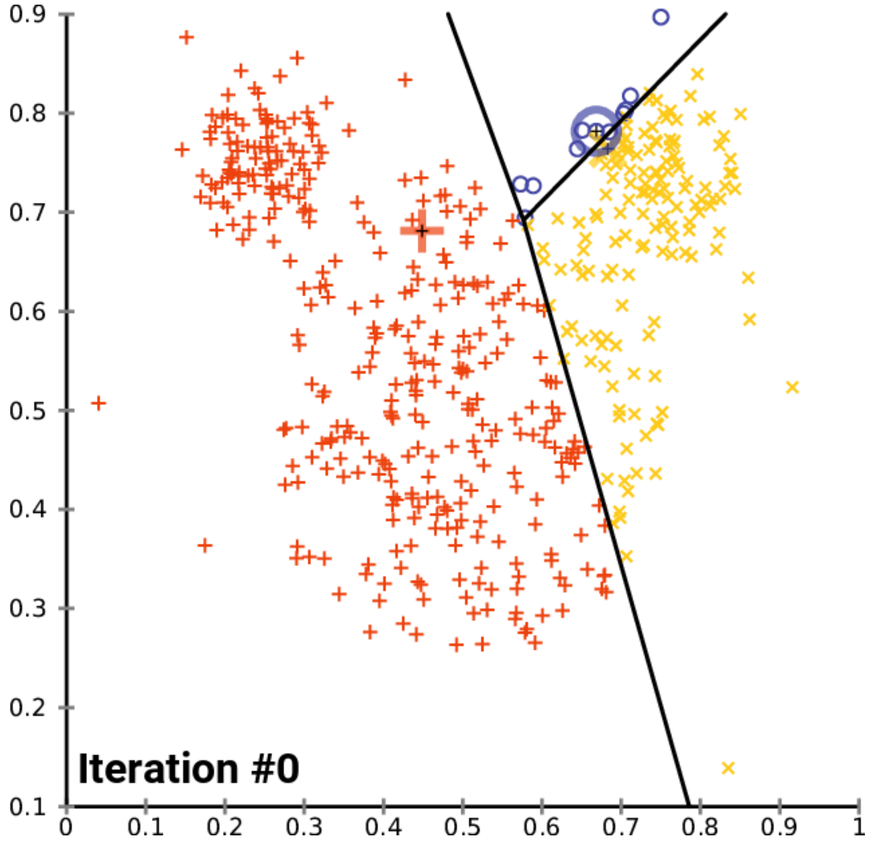
UMAP and t-SNE are nonlinear dimensionality reduction methods that preserve “local neighborhood structure”



https://www.nature.com/articles/nbt.4314

Clustering: k-means

- **What?** Assign each data point to one of the clusters!
- **How?** Greedily minimize the distance of points to their *centroids*
  1. Draw random centroids from the data.
  2. Repeat until convergence:
     A. Assign each point to its nearest centroid.
     B. Recompute centroids.



What are “good” cluster?

- E.g., k is arbitrary; how to choose the number of clusters?
- Principles:
  - **High internal cohesion:** points within the same cluster are close together
  - **High external separation:** different clusters are far apart.
- Silhouette Score of a cluster i

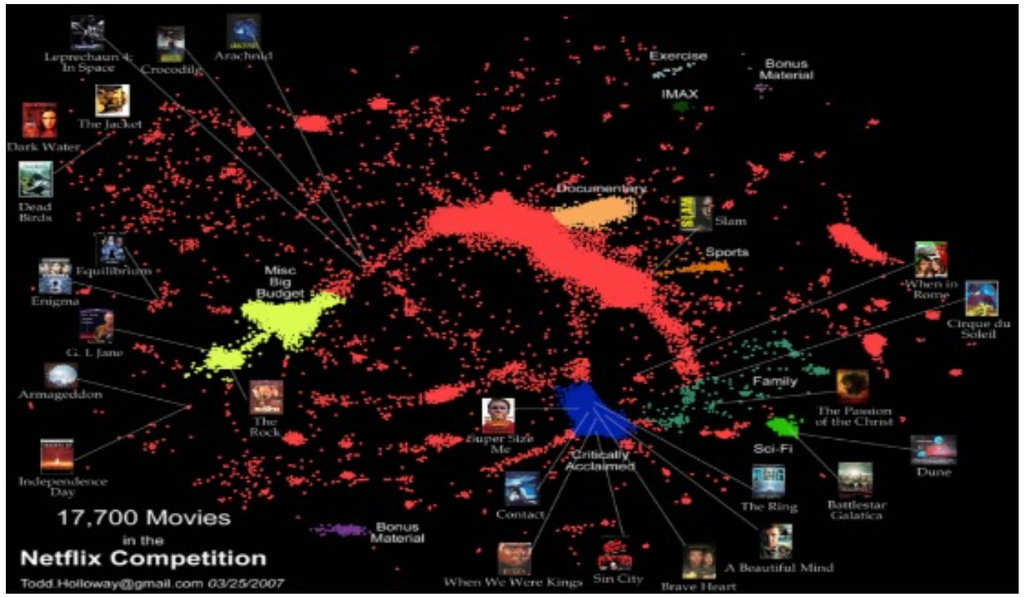
$$s(i) = b(i) - a(i)/max(b(i), a(i))$$

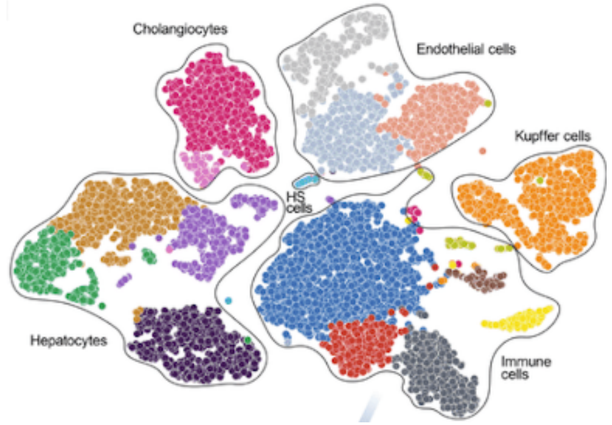


$a(i)$: average distance between datapoints within i (cohesion)

$b(i)$: average distance between data points across i and other clusters (separation)

Example of Clustering Problems

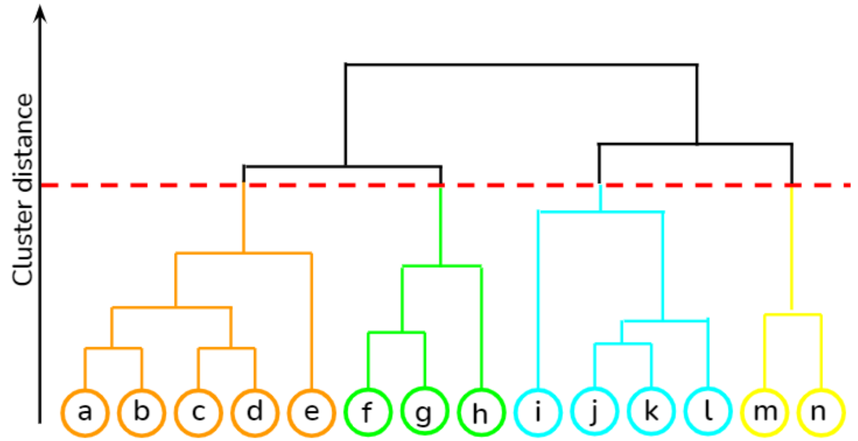




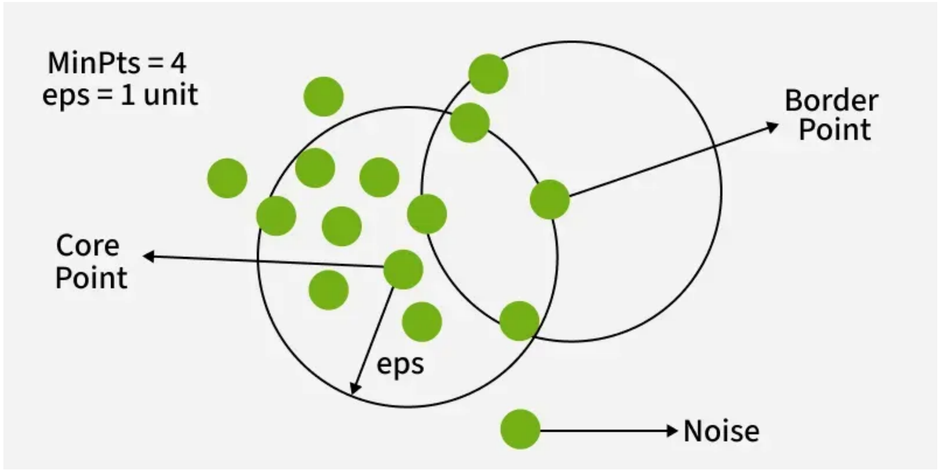


Honorary mentions

Hierarchical clustering



DB Scan



usually not
Clustering is ~~not always~~ a good idea!

- May “categorize” variables that are better understood on a continuous scale!
- Challenging to “generalize”
- Very helpful to get a sense of the data





MENU

APPETIZER
Quiz time!

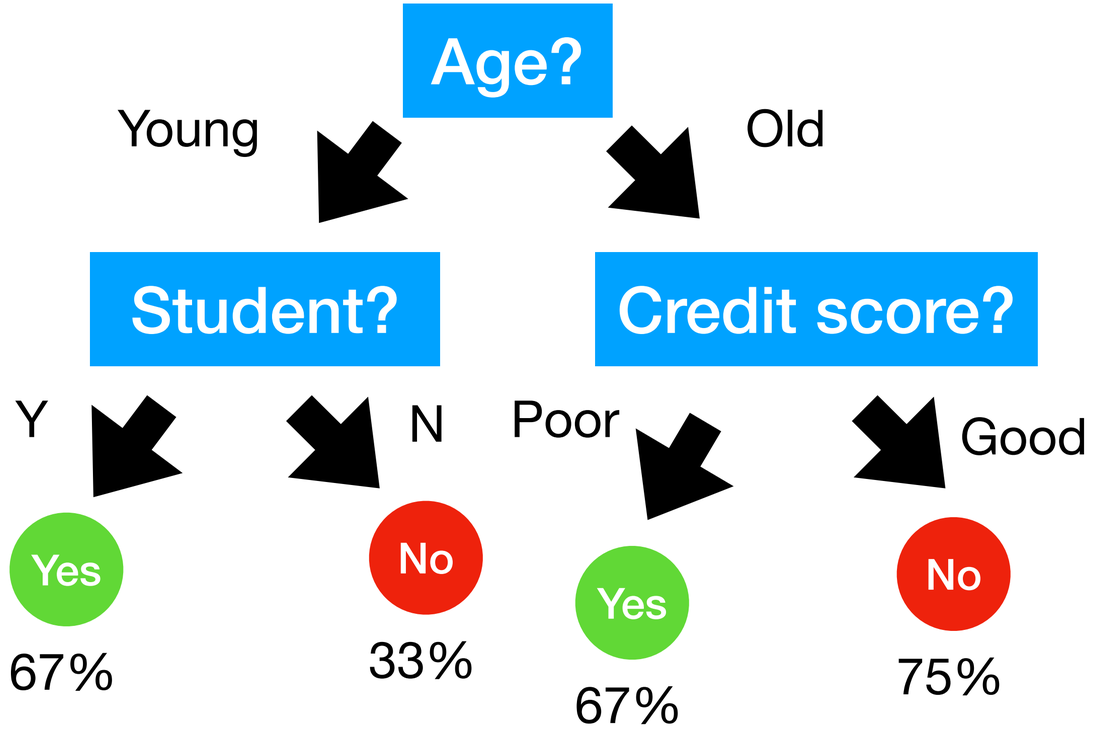
MAIN COURSE
Unsupervised learning

DESSERT
A bag of tricks



Model Calibration

- Consider 100 data points, all ⟨Young, Student, Poor⟩
  A. All 100 bought a computer
  B. Only 67 bought a computer
- Which model is better calibrated?



A classifier is calibrated if its predicted probabilities match the true frequencies of the events they represent.

$$P(Y = 1 \mid \hat{p} = p) = p$$

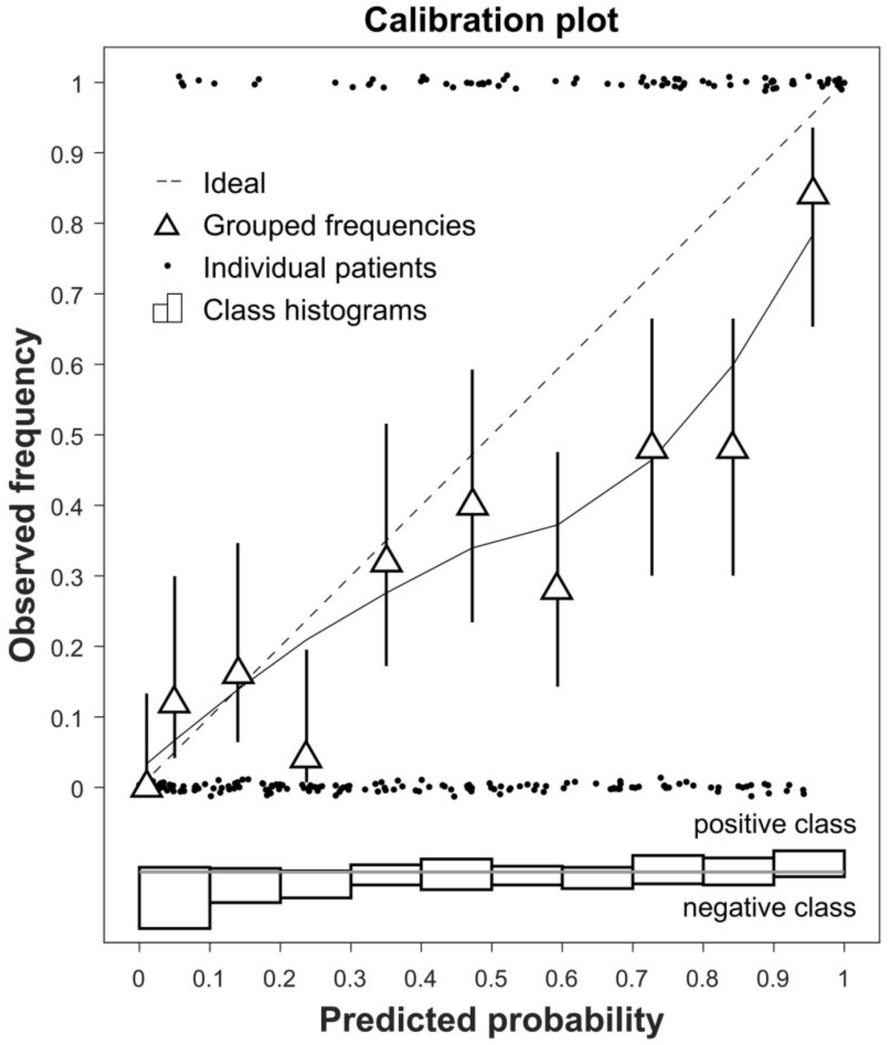
Intuition: Medical Test

- Models' predictions incur a "cost" in a medical test…
  - True Negatives: 0 (**Good! You don't do anything**)
  - False Positives: 10 (Unnecessary treatment)
  - False Negatives: 100 (Missed disease!)
  - True Positive: 20 (Treatment provided)
- If a model is well-calibrated, we can consider these costs when treating a patient!
  - E.g., if a model predicts a 15% chance of disease, how should we treat?

Assessing Calibration

- Calibration Plot
  1. Bin the data according to the outcome's predicted probability
  2. For each bin, estimate the fraction of positives.
  3. A calibrated classifier should yield something close to a horizontal line
- Expected Calibration Error

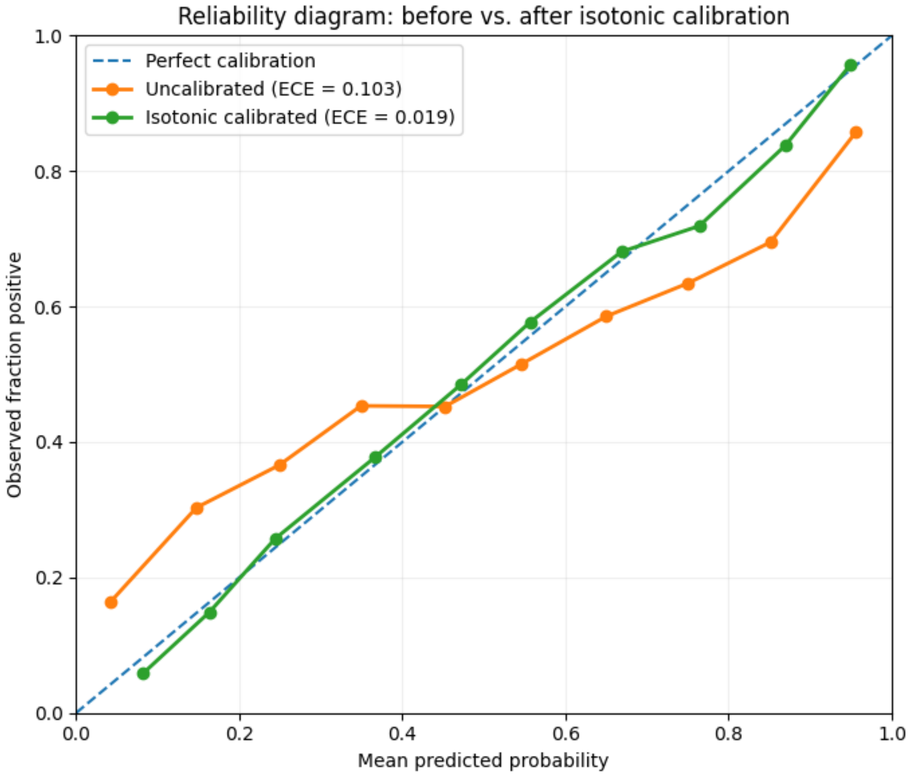
$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \right|,$$



Doing Calibration

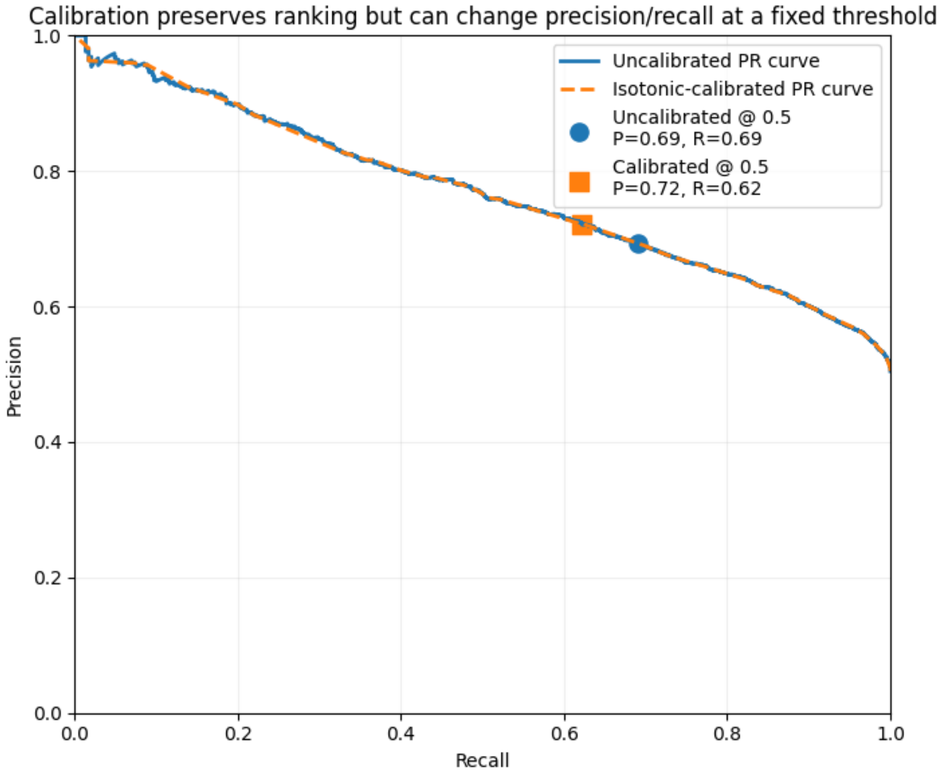
- **Idea**: Train a model to map the predicted probabilities into *corrected* probabilities.
- Isotonic Regression:

$$\min_f \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2, \hat{y}_i \leq \hat{y}_j \implies f(\hat{y}_i) \leq f(\hat{y}_j)$$



What calibration does not change!

- Calibration does not change the precision-recall curve!
- But it does change the decision thresholds associated with each precision-recall trade-off!





Feature Engineering

Transform raw data into representations that make the signal easier for the model to learn.

purchase_ts
1749643833

11th June 2025 12:10:33, Wednesday

purchase_year	purchase_hour
2025	12
purchase_dow	purchase_month
02	06

Common tricks

Transform

- $\log(x)$ $\arctan(x)$ $\log(x+1)$
- Standardization: $\frac{x - E[x]}{Var[x]}$
- Binning continuous variables

Aggregate

- Suppose customer transactions are logged as (id, item, cost, timestamp)
- How would you represent them?
  - purchases_last_7_days
  - average_spend_last_30_days
  - days_since_last_purchase

Common tricks

Interactions

- Sometimes the interactions between two variables has meaning that transcends each individual variable
  - humidity · temperature
  - clicks/impressions

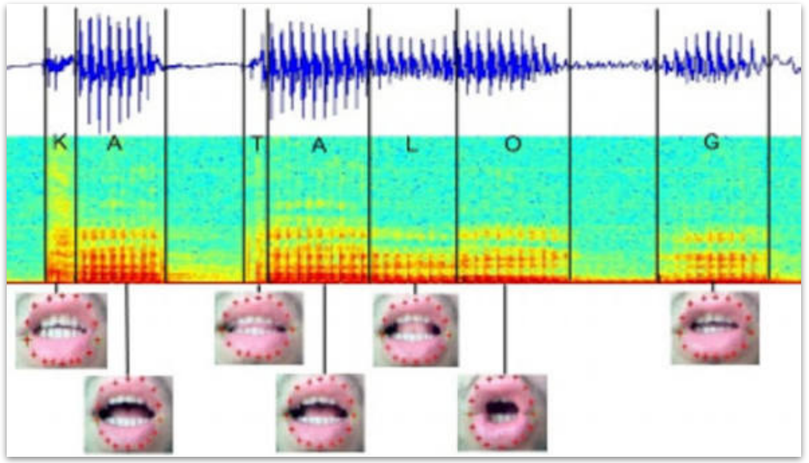
Categorical

- City $\in$ {Princeton, NYC, Boston, …}
- One-hot encoding:
  - NYC $\rightarrow [0,1,0,\ldots]$
- Problematic with high cardinality
  - Target encoding (in a second)

The Bitter Lesson

- **Feature engineering**: we should incorporate our own prior knowledge!
- This view has been disrupted by the deep learning revolution!
- The Bitter Lesson: more “raw” data and generic algorithms triumph over carefully constructed features or methods.
- Still, at low n, it may matter!





Target Encoding

- Instead of one-hot encoding a category, use the **category's average target**.
- Very powerful for high-cardinality categorical features…
- But there is a trap! An observation's own y_i value helps construct its feature!
  - **Solution**: split training data into k folds, create encodings with other folds

City	$\bar{y}$
NYC	0.34
Princeton	0.21
Boston	0.17
…	…



Target Encoding (continued)

- Suppose you calculated target encoding for the whole dataset!
- “Information” would leak from the test set to the training set, e.g., if there are two datapoints from NYC:
  - $y_{\text{train}} = 0.20 \rightarrow y_{\text{test}} = 0.48$
- Performance in the test set would not generalize to new data

City	$\bar{y}$
NYC	0.34
Princeton	0.21
Boston	0.17
…	…

Preventing data leakage

- **A very common issue:** information sneaks from the test set into the training set!
- Kapoor and Narayanan found this to affect more than 329 papers across 17 fields!



Leakage and the Reproducibility Crisis in ML-based Science

Sayash Kapoor [1] **Arvind Narayanan** [1]

Abstract

The use of machine learning (ML) methods for prediction and forecasting has become widespread across the quantitative sciences. However, there are many known methodological pitfalls, including data leakage, in ML-based science. In this paper, we systematically investigate reproducibility issues in ML-based science. We show that data leakage is indeed a widespread problem and has led to severe reproducibility failures. Specifically, through a survey of literature in research communities that adopted ML methods, we find 17 fields where errors have been found, collectively affecting 329 papers and in some cases leading to wildly overoptimistic conclusions. Based on our survey, we present a fine-grained taxonomy of 8 types of leakage that range from textbook errors to open research problems.

We argue for fundamental methodological changes to ML-based science so that cases of leakage can be caught before publication. To that end, we propose model info sheets for reporting scientific claims based on ML models that would address all types of leakage identified in our survey. To investigate the impact of reproducibility errors and the efficacy of model info sheets, we undertake a reproducibility study in a field where complex ML models are believed to vastly outperform older statistical models such as Logistic Regression (LR): civil war prediction. We find that all papers claiming the superior performance of complex ML models compared to LR models fail to reproduce due to data leakage, and complex ML models don't perform substantively better than decades-old LR models. While none of these errors could have been caught by reading the papers, model info sheets would enable the detection of leakage in each case.

[1]Department of Computer Science and Center for Information Technology Policy, Princeton University. Correspondence to: Sayash Kapoor <sayashk@princeton.edu>.

1. Overview

There has been a marked shift towards the paradigm of predictive modeling across quantitative science fields. This shift has been facilitated by the widespread use of machine learning (ML) methods. However, pitfalls in using ML methods have led to exaggerated claims about their performance. Such errors can lead to a feedback loop of overoptimism about the paradigm of prediction—especially as non-replicable publications tend to be cited more often than replicable ones (Serra-Garcia & Gneezy, 2021). It is therefore important to examine the reproducibility of findings in communities adopting ML methods.

Scope. We focus on reproducibility issues in ML-based science, which involves making a scientific claim using the performance of the ML model as evidence. There is a much better known reproducibility crisis in research that uses traditional statistical methods (Open Science Collaboration, 2015). We also situate our work in contrast to other ML domains, such as methods research (creating and improving widely-applicable ML methods), ethics research (studying the ethical implications of ML methods), engineering applications (building or improving a product or service), and modeling contests (improving predictive performance on a fixed dataset created by an independent third party). Investigating the validity of claims in all of these areas is important, and there is ongoing work to address reproducibility issues in these domains (Hullman et al., 2022; Pineau et al., 2020; Erik Gundersen, 2021; Bell & Kampman, 2021).

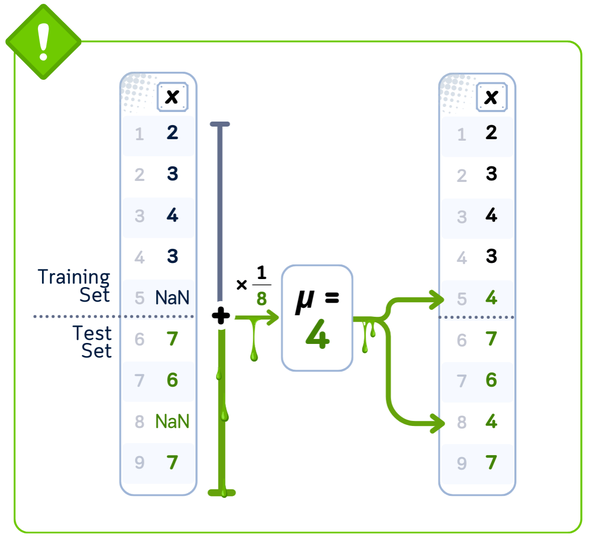
We define a research finding as reproducible if the code and data used to obtain the finding are available and the data is correctly analyzed (Hofman et al., 2021a; Leek & Peng, 2015; Pineau et al., 2020). This is a broader definition than computational reproducibility — when the results in a paper can be replicated using the exact code and dataset provided by the authors (see Appendix A).

Leakage. Data leakage has long been recognized as a leading cause of errors in ML applications (Nisbet et al., 2009). In formative work on leakage, Kaufman et al. (2012) provide an overview of different types of errors and give several recommendations for mitigating these errors. Since this paper was published, the ML community has investigated leakage in several engineering applications and modeling competitions (Fraser, 2016; Ghani et al., 2020; Becker, 2018; Brownlee, 2016; Collins-Thompson). However, leakage occurring in ML-based science has not been comprehensively

https://arxiv.org/abs/2207.07048

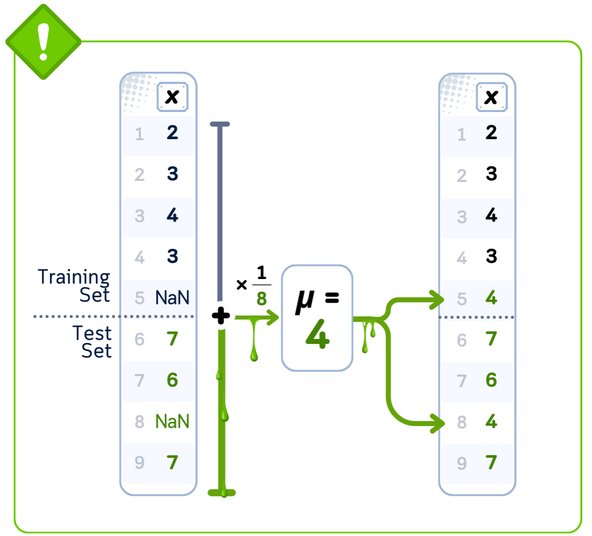
Usual suspect #1: Preprocessing

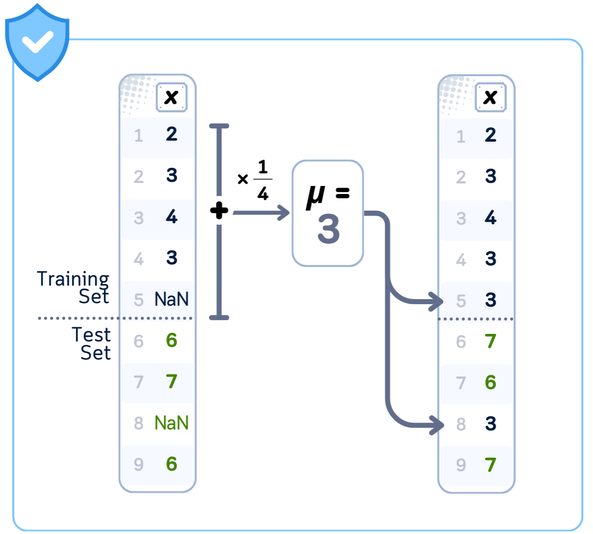
- **Example**: data imputation, i.e., substituting the NaN values with the average value.
- Leaks information from the test set into the training set!
- Could also happen with scaling features, target encoding, etc!



Usual suspect #1: Preprocessing

- **Solution:** Any kind of pre-processing cannot “breach” the train-test split.
- For example, if you have a heuristic to fill missing values, calculate it on the training data and apply it to the test data.





Usual suspect #2: Correlated Variables

- Leakage may occur if you have correlated variables spread between train and test
- E.g., in temporal data, days close to each other are correlated!
- Similar problem: multiple data points corresponding to the same person!

Day	…	Split	Price
168	…	Train	$100
169	…	Test	$98
170	…	Train	$105
171	…	Test	$102
…	…	…	…

Usual suspect #2: Correlated Variables

- **Solution**: ensure there is a clean split of correlated units between train and test sets!
- In the temporal prediction case, prediction will naturally become harder as you depart from the training set.

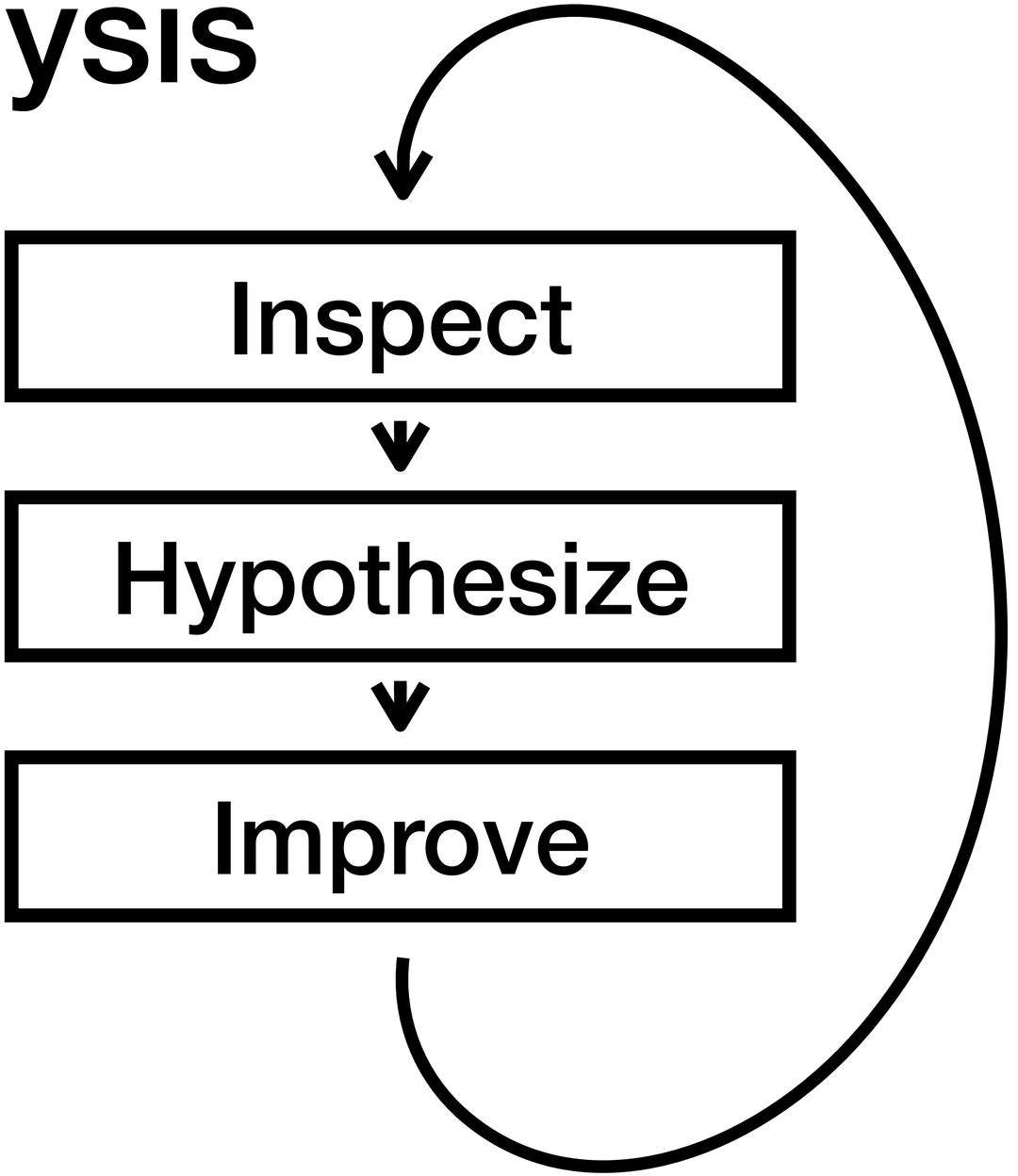


Patient	…	Split	Y
3	…	Train	45
3	…	Train	66
4	…	Test	34
4	…	Test	29
…	…	…	…



Error analysis

- Aggregate metrics shed little light on *why* a model is failing!
- **Error analysis**: examining mistakes to identify systematic patterns.









True Positives







False Negatives



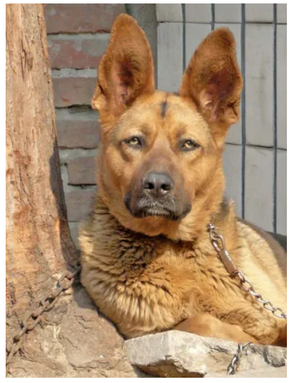




False Positives







True Negatives

Check the model confidence!



13% Cat

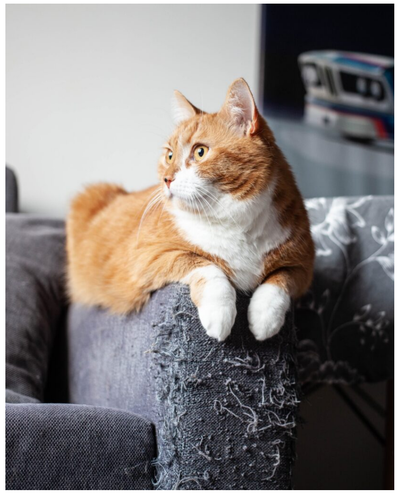


31% Cat



89% Cat

Is the problem the model or the label?



- **Labeled as “not a cat”**
- **Model:** 76% chance of cat

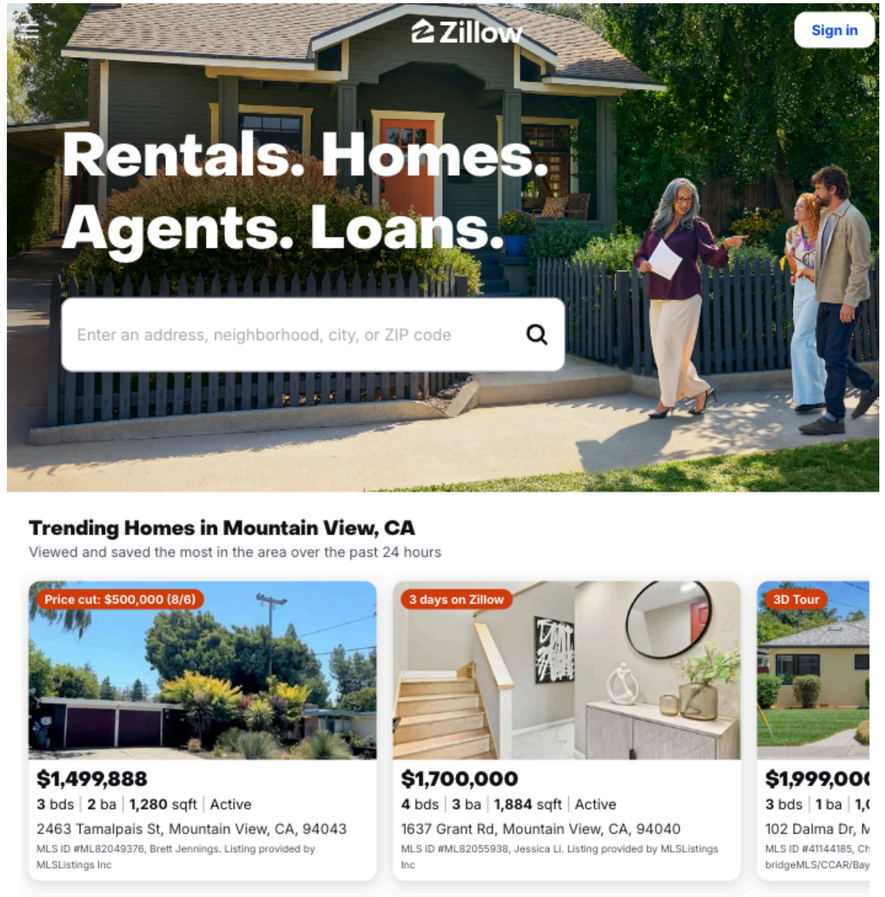


- **Should this be a cat?**

Predicting House Prices

- A regression model predicts house prices based on
  - Year built
  - Number of rooms
  - Square footage

- You look at the top 50 predictions with the highest residuals!
  - Big houses next to downtown (lowball)
  - Big houses far from downtown (highball)



Poor subgroup performance	→	Collect targeted data!	→	*Find more data on people dressed as cats*
Label errors	→	Improve labeling	→	*Use crowdsourcing to find mistakes in my dataset*
Missing signal	→	Obtain new data	→	*Let me consider the distance to downtown*


Give us some feedback!
https://tinyurl.com/reasoningwithdata