EPA United States Environmental Protection Agency

Machine Learning in Predictive Toxicology: An Overview and Case Study

Caroline L. Ring[1]; Julia E. Rager[2]
[1]United States Environmental Protection Agency, Center for Computational Toxicology and Exposure, Research Triangle Park, North Carolina, USA
[2]University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, USA

The views expressed in this presentation are those of the author(s) and do not necessarily reflect the views or policies of the U.S. EPA

A frequent problem in toxicology….

- We have a large number of chemicals to screen for potential risk
- We want to know about hazard or toxicity that is *hard* to measure for all these chemicals (expensive, slow, unethical…)
  - We have some previously-measured examples of this information for some chemicals
- We have information that is *easier* to measure (rapid, inexpensive)
  - Molecular structure
  - *In vitro* bioactivity in high-throughput screening assays
- We don't have a clear idea of how the "hard" info relates to the "easy" info
  - no mechanistic model
- How can we use the available "easy" data to *predict* the "hard-to-measure" data?

Idakwo et al. 2018; Wang et al. 2020

Machine learning: Computational algorithms that can infer patterns from data

Data			
Target/ response	**Features**		
y1	**x1**	**x2**	**…**
[value1]	[value1]	[value1]	…
[value2]	[value2]	[value2]	…
…	…	…	…

Target/response (y): what we want to predict (e.g. toxicity or hazard)
Features (x1, x2…): Information available to predict response (e.g. structure, *in vitro* HTS bioactivity, etc.)

EPA United States Environmental Protection Agency

Two categories of patterns to be inferred

Supervised: Infer relationship between target and features. Goal: predict target from features.

Data			
Target/ response	**Features**		
y1	**x1**	**x2**	**…**
[value1]	[value1]	[value1]	…
[value2]	[value2]	[value2]	…
…	…	…	…

Unsupervised: No target to predict. Infer descriptive patterns in features (e.g., clustering).

Data		
Features		
x1	**x2**	**…**
[value1]	[value1]	…
[value2]	[value2]	…
…	…	…

This presentation will focus on supervised machine learning

Supervised: Infer relationship between target and features. Goal: predict target from features.

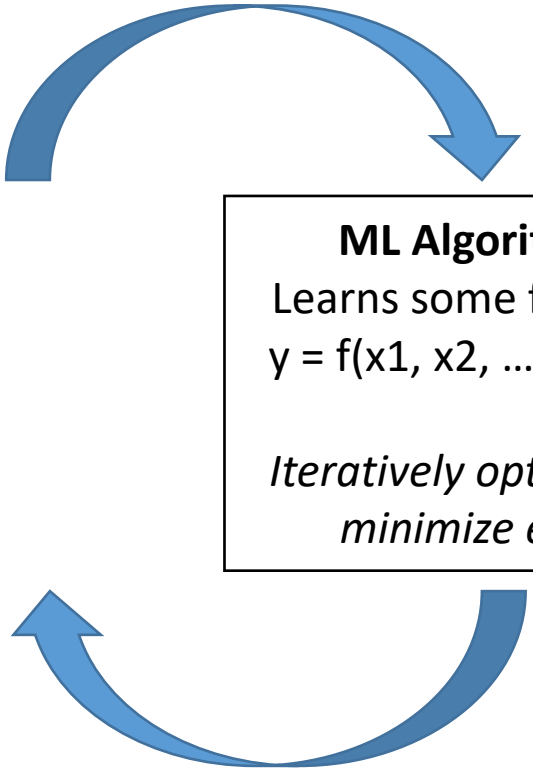
Data			
Target/ response	**Features**		
y1	**x1**	**x2**	**…**
[value1]	[value1]	[value1]	…
[value2]	[value2]	[value2]	…
…	…	…	…

Unsupervised: No target to predict. Infer descriptive patterns in features (e.g., clustering).

Data		
Features		
x1	**x2**	**…**
[value1]	[value1]	…
[value2]	[value2]	…
…	…	…

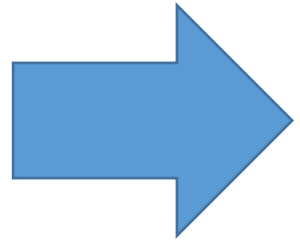
Training a supervised ML model

Training data			
Target/ response	**Features (predictors)**		
y1	**x1**	**x2**	**…**
[value1]	[value1]	[value1]	…
[value2]	[value2]	[value2]	…
…	…	…	…



ML Algorithm:
Learns some function
y = f(x1, x2, …) + error

Iteratively optimize to minimize error



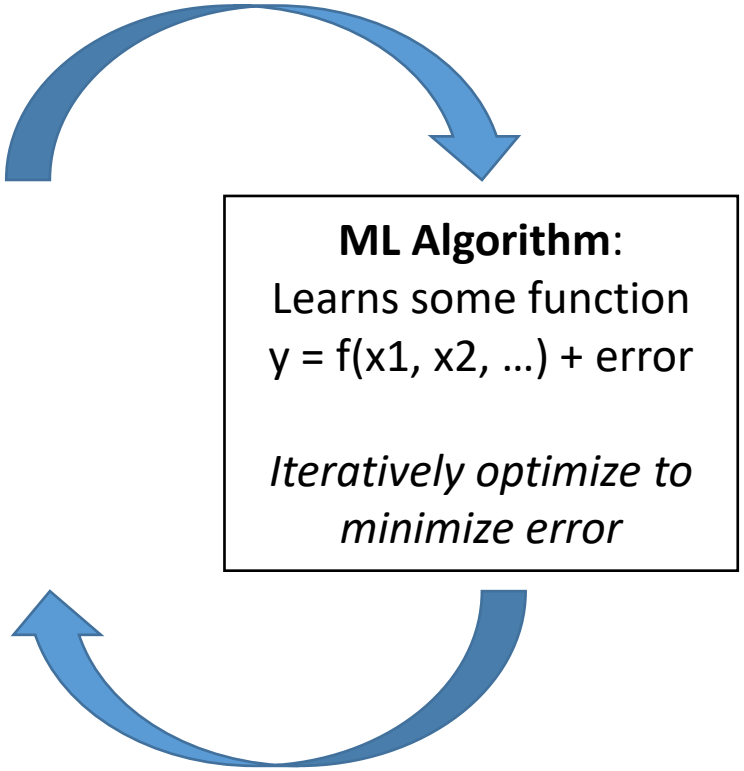
MODEL
y = f (x1, x2, …)

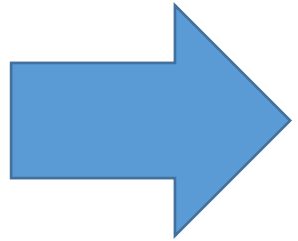
Algorithms can include:

- k nearest neighbors
- decision trees
- support vector machine
- naïve Bayes
- random forest
- artificial neural networks
- etc.

Two types of model: regression and classification

Training data			
Target/ response	**Features (predictors)**		
y1	**x1**	**x2**	**…**
[value1]	[value1]	[value1]	…
[value2]	[value2]	[value2]	…
…	…	…	…



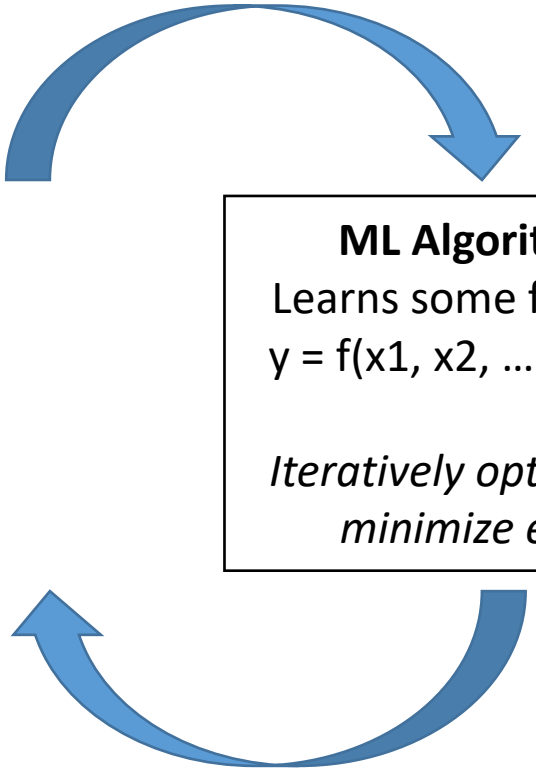


MODEL
y = f (x1, x2, …)

Regression model: y is numeric & continuous (e.g. LD50)
Classification model: y is categorical (e.g. hepatic toxicity yes/no)

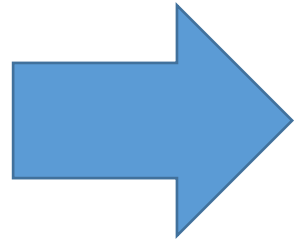
This presentation focuses on classification models

Training data			
Target/ response	Features (predictors)		
y1	x1	x2	…
[value1]	[value1]	[value1]	…
[value2]	[value2]	[value2]	…
…	…	…	…



ML Algorithm:
Learns some function
y = f(x1, x2, …) + error

Iteratively optimize to minimize error



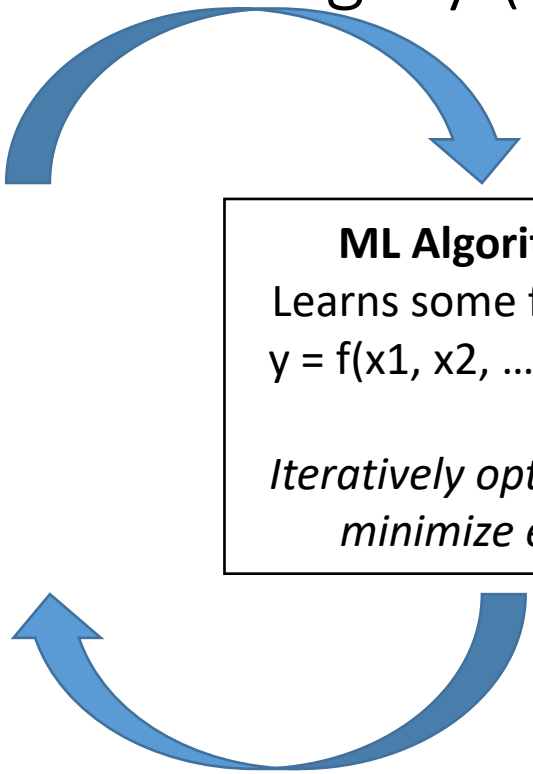
MODEL
y = f (x1, x2, …)

Regression model: y is numeric & continuous (e.g. LD50)

Classification model: y is categorical (e.g. hepatic toxicity yes/no)

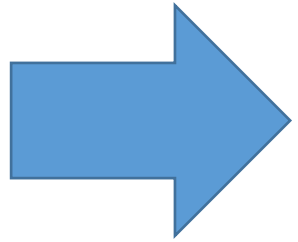
Classification model: Usually predicts *probability* of positive/negative for a category (e.g., hepatotoxicity)

Training data			
Target/ response	**Features (predictors)**		
y1: Positive?	**x1**	**x2**	**…**
0	[value1]	[value1]	…
1	[value2]	[value2]	…
…	…	…	…



ML Algorithm:
Learns some function
y = f(x1, x2, …) + error

Iteratively optimize to minimize error



MODEL
y = f (x1, x2, …)

Predicted prob. positive
0.4
0.5
0.7
…

Make predictions categorical by applying a threshold on predicted probabilities — typically 0.5, but doesn't have to be

Training data			
Target/ response	**Features (predictors)**		
y1: Positive?	**x1**	**x2**	**…**
0	[value1]	[value1]	…
1	[value2]	[value2]	…
…	…	…	…

ML Algorithm:
Learns some function
y = f(x1, x2, …) + error

Iteratively optimize to minimize error

MODEL
y = f (x1, x2, …)

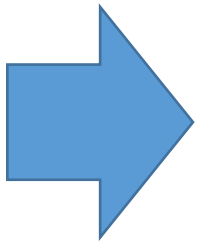
Predicted y1: Positive?
0
1
….

Threshold = 0.5

Predicted prob. positive
0.4
0.7
…

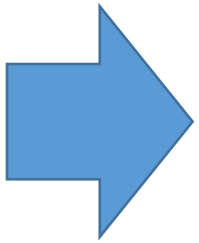
Evaluating performance of a ML model

Test data			
Target/ response	Features (predictors)		
y	x1	x2	…
[value1]	[value1]	[value1]	…
[value2]	[value2]	[value2]	…
…	…	…	…

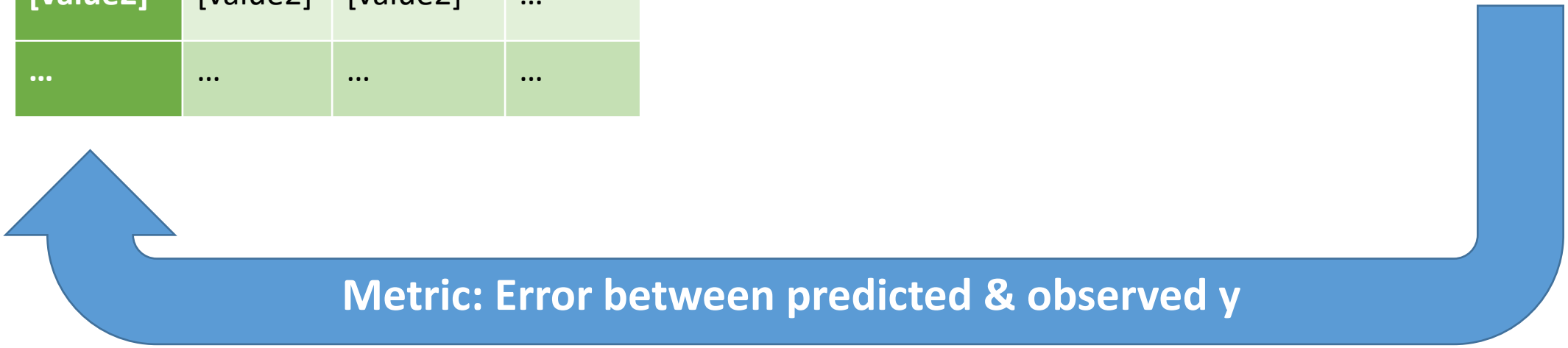


MODEL

y = f (x1, x2, …)



Model-predicted response
y_pred
[value1]
[value2]
…



For regression, this could be sum of squared errors

$$\sum(y - y\_pred)^2$$

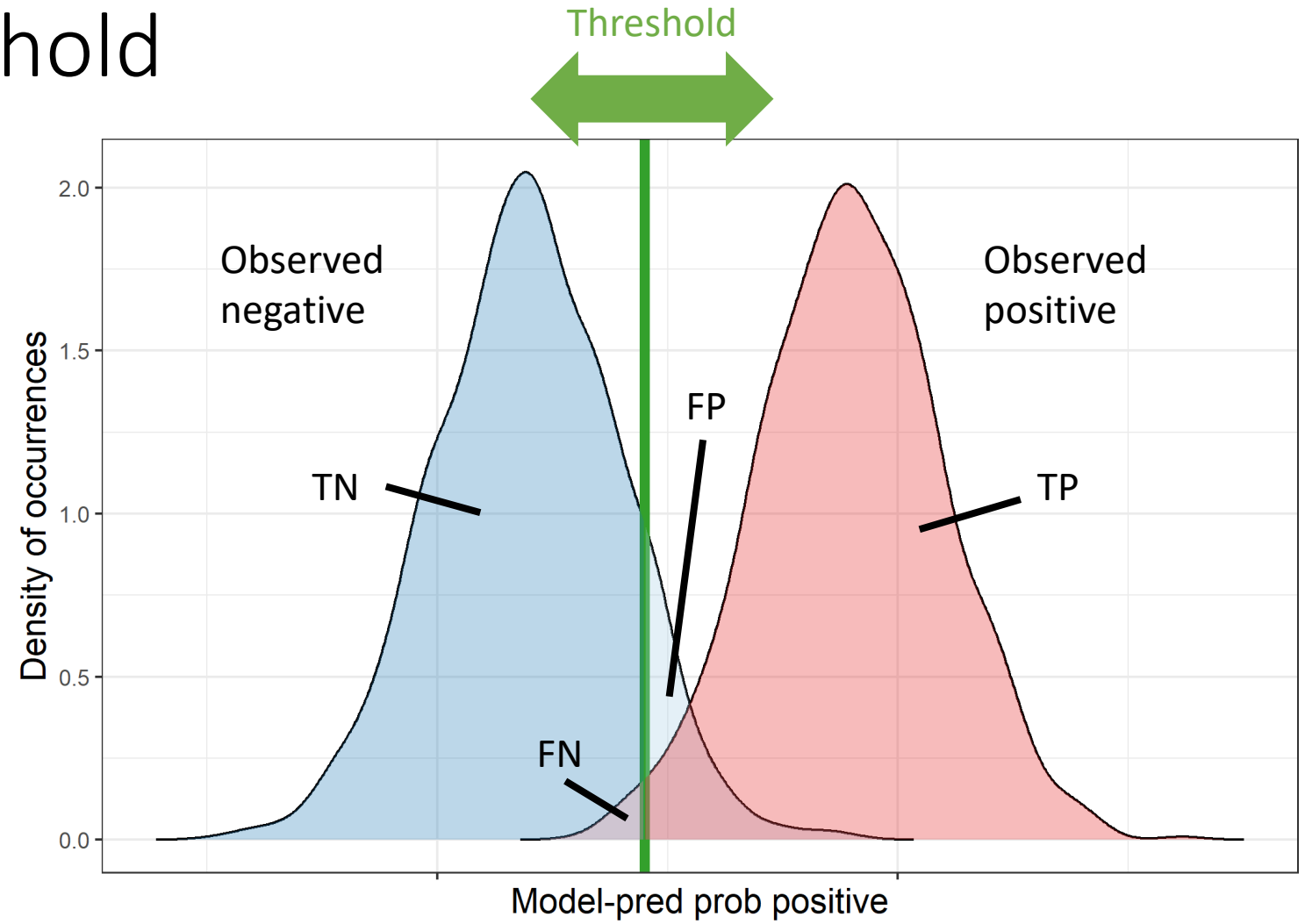
EPA United States Environmental Protection Agency

Error metrics for (binary) classification models: *confusion matrix*

	Predicted negative	Predicted positive
Observed negative	True negatives (TN)	False positives (FP)
Observed positive	False negatives (FN)	True positives (TP)

Accuracy: (TN + TP) / (TN + TP + FN + FP)
Sensitivity (true positive rate, TPR): TP/(TP + FN)
Specificity (true negative rate, TNR): TN/(TN + FP)
Balanced Accuracy: (Sensitivity + Specificity)/2
Positive Predictive Value (PPV): TP / (TP + FP)
False Discovery Rate: 1 – PPV
[…lots more!]

Confusion matrix varies with threshold

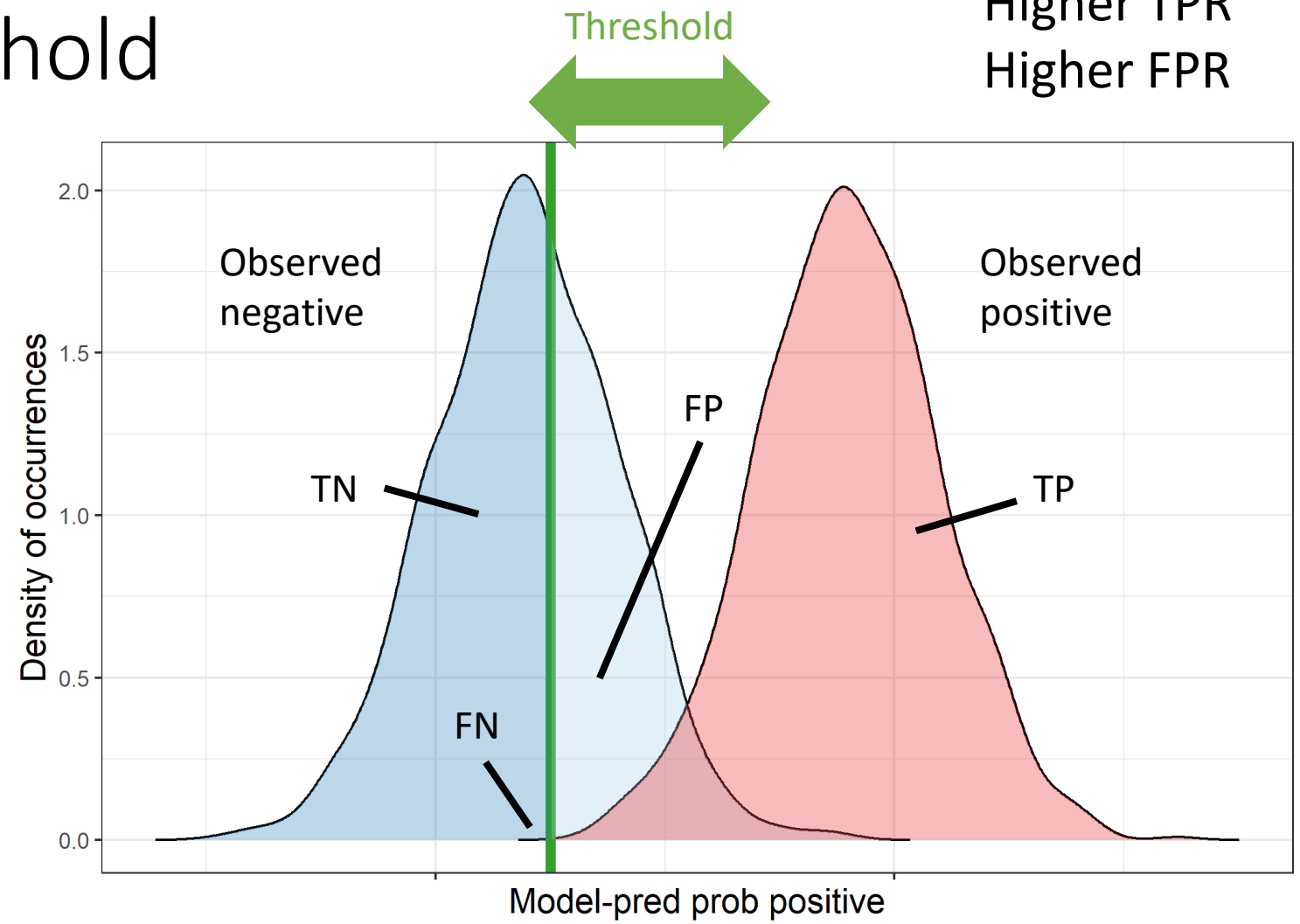


More separation between peaks = more informative model

(synthetic example data)

Confusion matrix varies with threshold

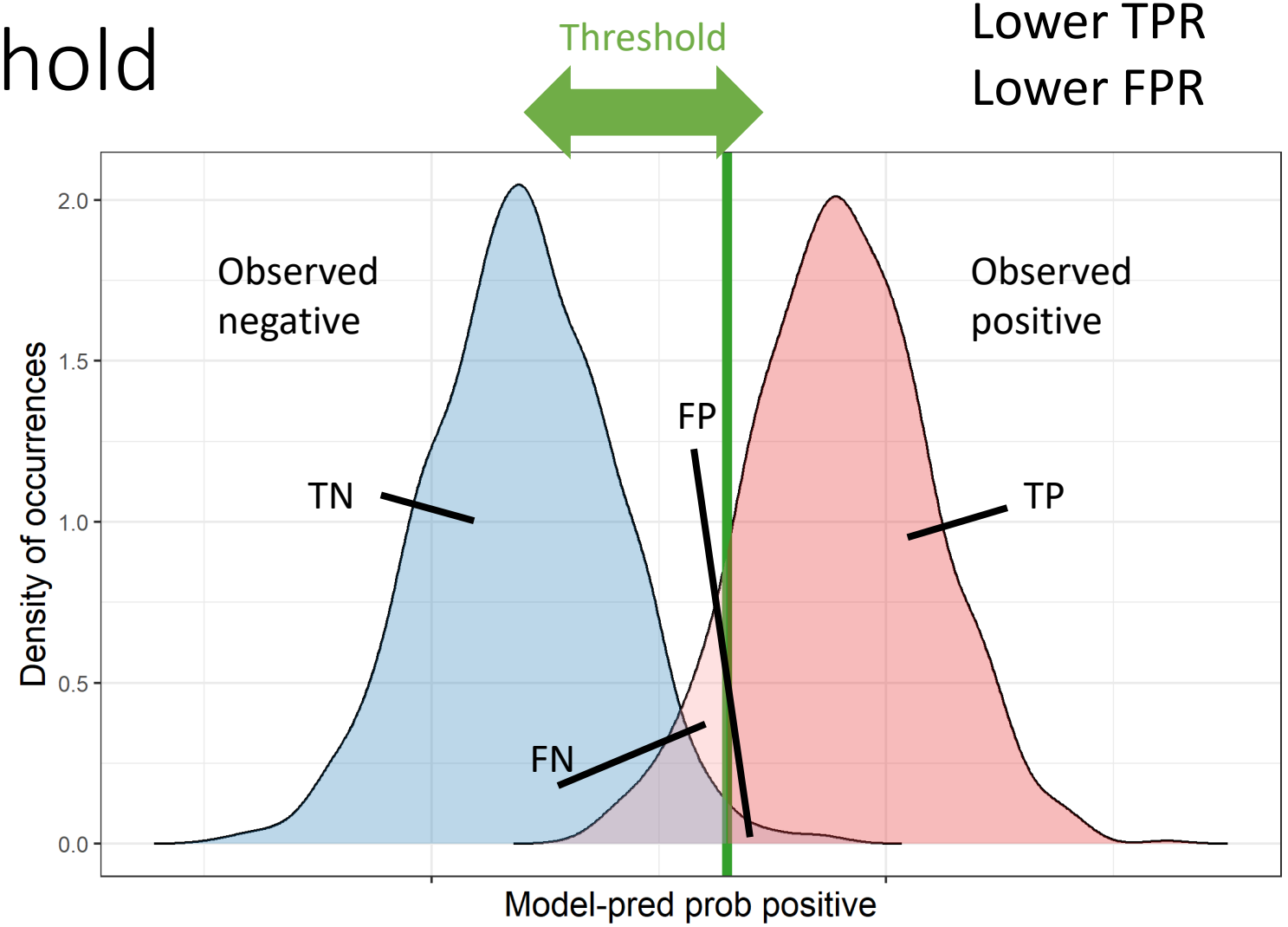
Lower threshold =
Higher TPR
Higher FPR



(synthetic example data)

Confusion matrix varies with threshold

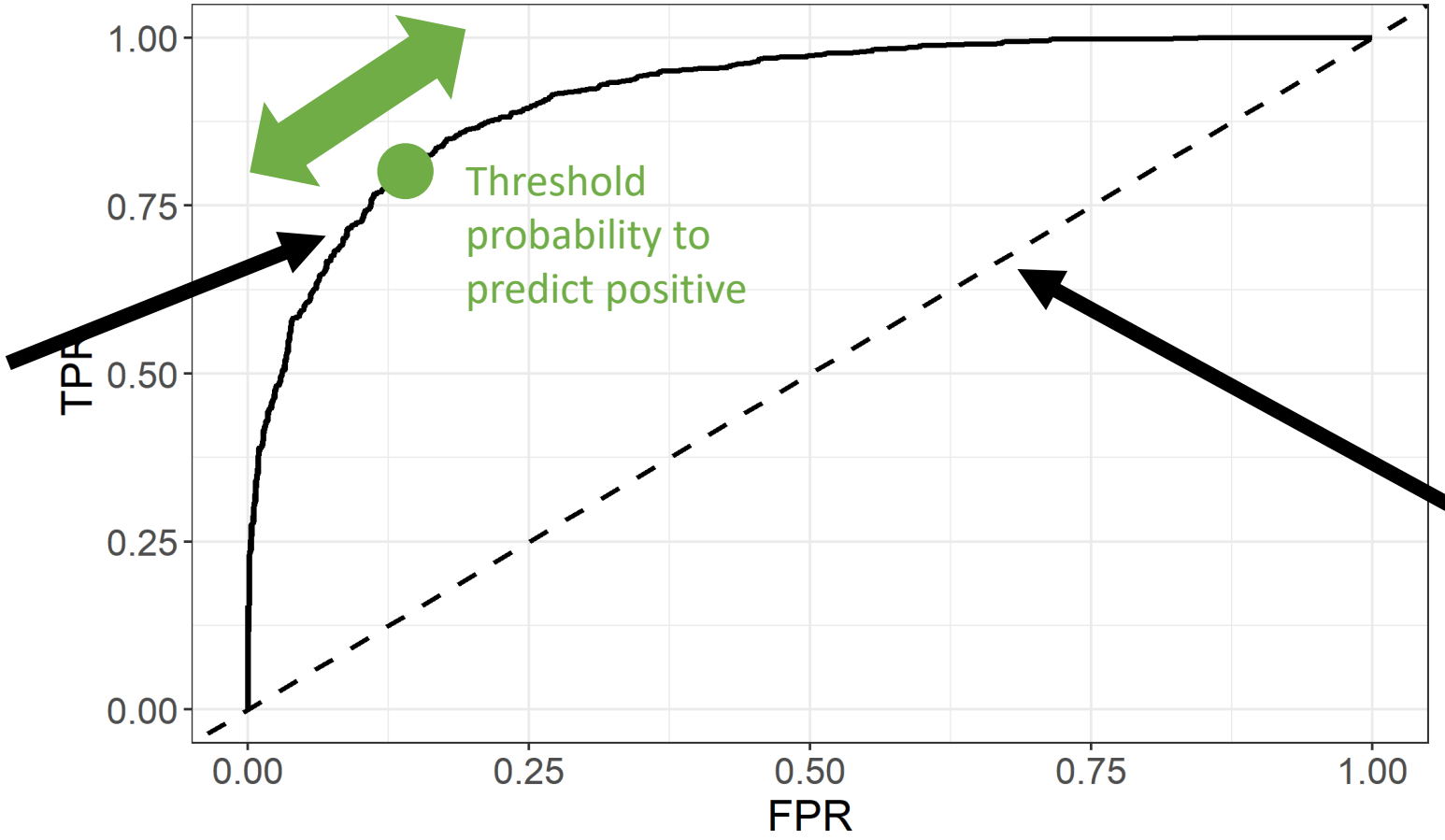
Higher threshold =
Lower TPR
Lower FPR



(synthetic example data)

Area under receiver-operator characteristic (ROC) curve (AUROC) tells us about separation between peaks & model performance over all thresholds

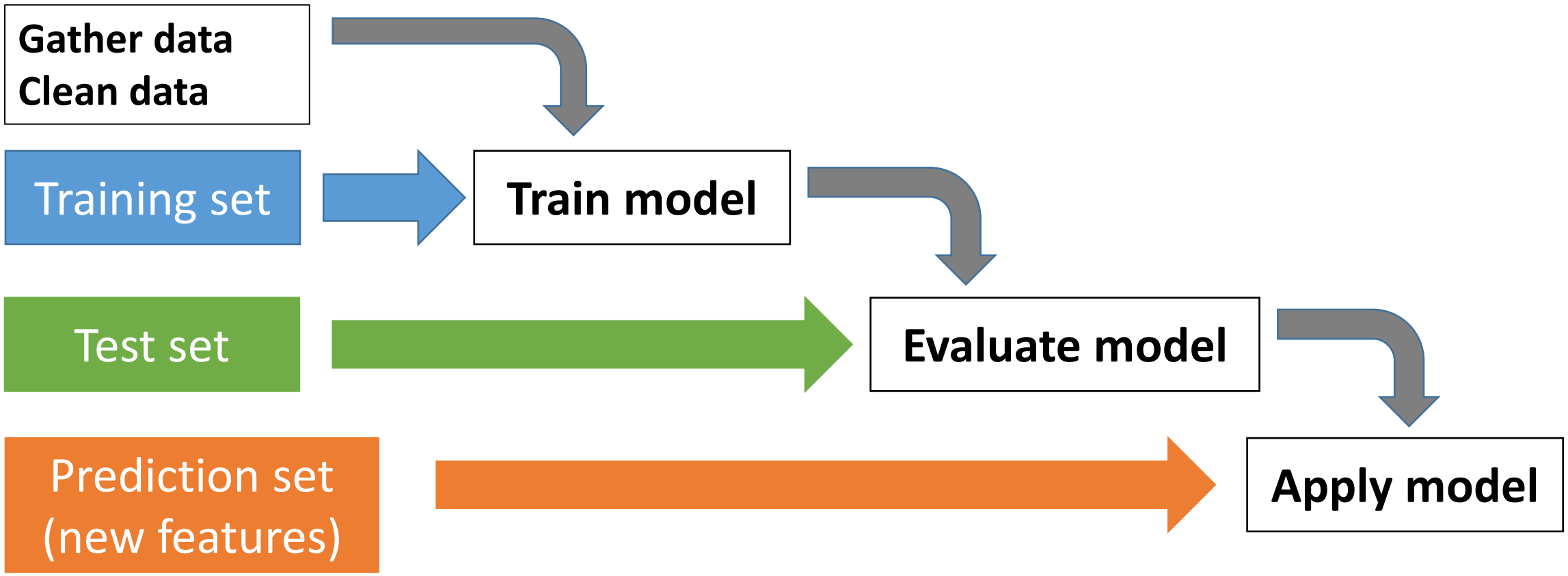
Trained model AUROC = 0.91 (for this fake example)



Perfectly uninformative model (no separation between peaks) AUROC = 0.5

EPA United States Environmental Protection Agency

Summary of machine-learning model process



Challenge in classification models for predictive toxicology: *imbalanced data*

From Mansouri et al. 2020: ComPARA training set
(response = *in vitro* androgen activity in ToxCast, yes/no)

Table 1. Training set chemicals for binding, agonist and antagonist data sets.

Number of	Binding	Agonist	Antagonist
Actives	198	43	159
Inactives	1,464	1,616	1,366
Total	1,662	1,659	1,525

88% inactive for binding

97% inactive for agonism

90% inactive for antagonism

Problem:

A ML model that simply predicted "inactive" for *everything* would have a 97% accuracy rate for agonism!

Many toxicology-related data sets are imbalanced like this (Idakwo et al. 2018; Wang et al. 2020)

How can we build a ML model that properly predicts the minority class?

EPA United States Environmental Protection Agency

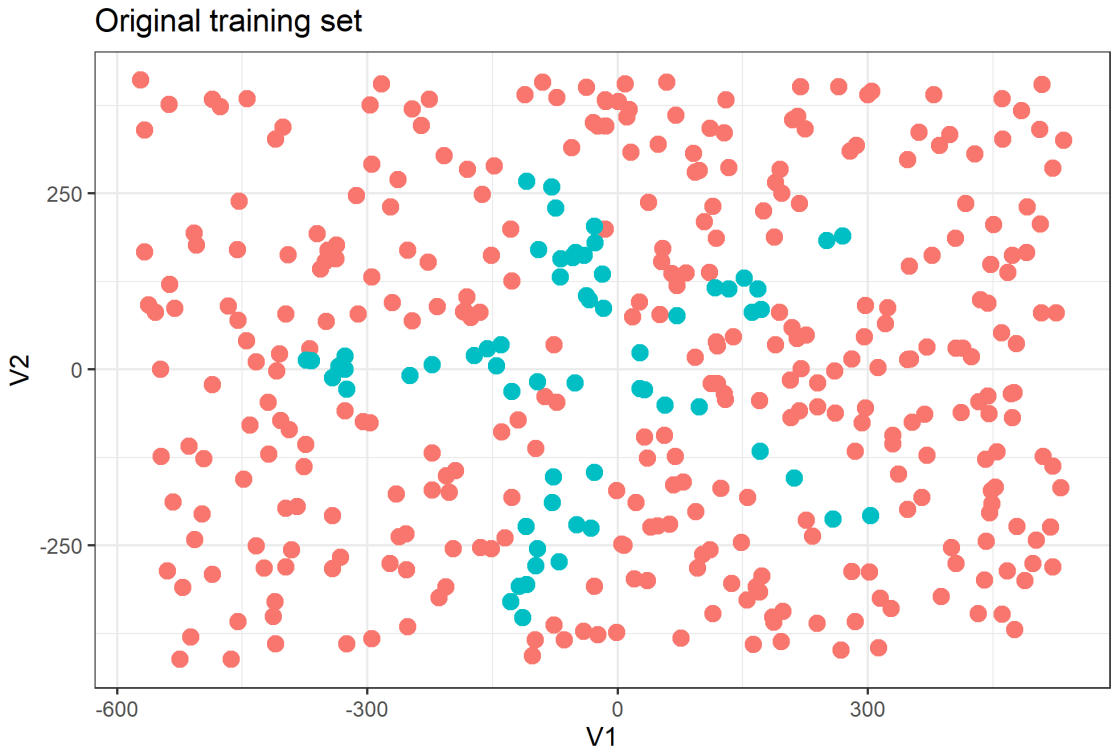
Strategies to address imbalanced training data

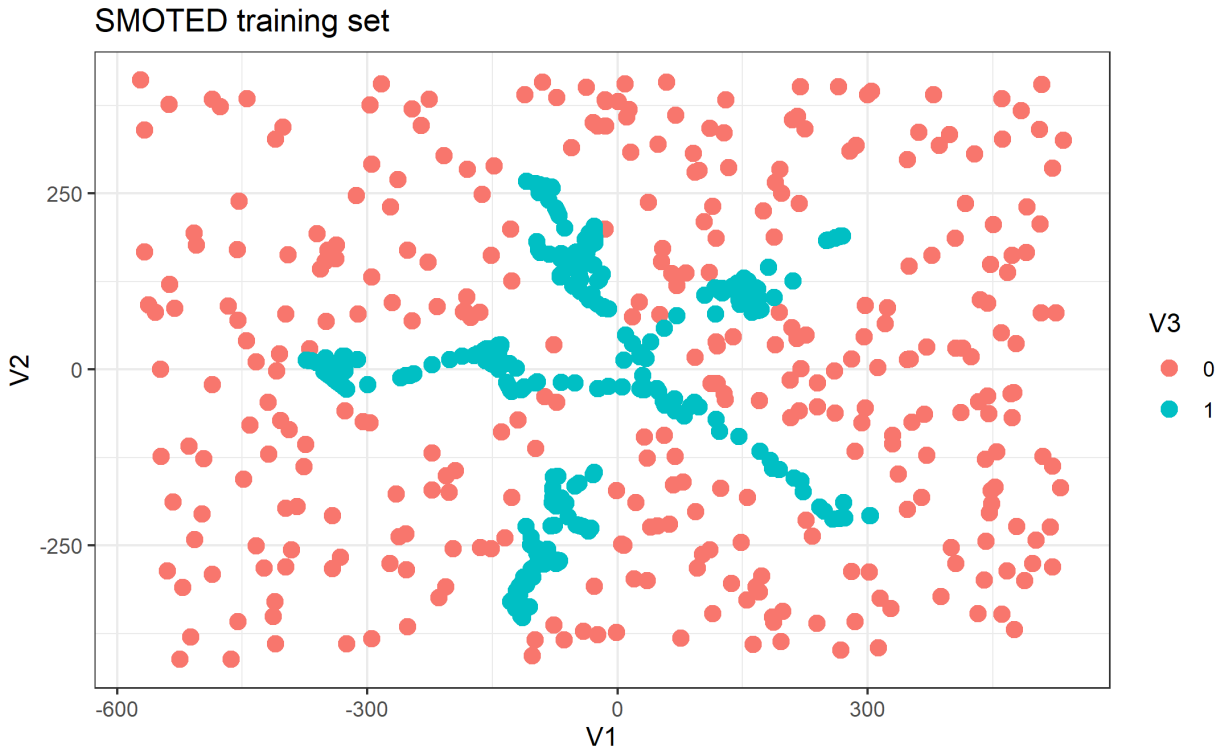
(Branco et al., 2016)

- Algorithm-based: Make the model less sensitive to imbalance
  - *Boosting*: iteratively correct misidentified instances in the training class
  - *Bagging:* trains multiple versions of the model on subsets or bootstrap-resampled versions of the data set
  - *Cost function*: During model training, weight errors more heavily for minority-class examples
- Sampling-based: Pre-process training data to balance out the classes
  - *Undersampling*: Remove some majority-class examples
  - *Oversampling*: Repeat some minority-class examples – or create synthetic new ones
- Synthesizing new minority-class examples
  - Generative Adversarial Networks (GAN): train a second ML model to synthesize plausible minority examples (Douzas & Bacao, 2018; Green et al. 2021)
  - Interpolate between minority-class examples & nearest neighbors: e.g. SMOTE (Synthetic Minority Over-sampling TEchnique) (Chawla et al., 2002)

Example of SMOTE

"clover" data from https://sci2s.ugr.es/keel/datasets.php (Alcalá-Fdez et al. 2011)





Drawback: SMOTE can blur boundaries by interpolating to majority-class near neighbors

It turns out that ML algorithms and imbalanced data strategies perform very differently for different data sets

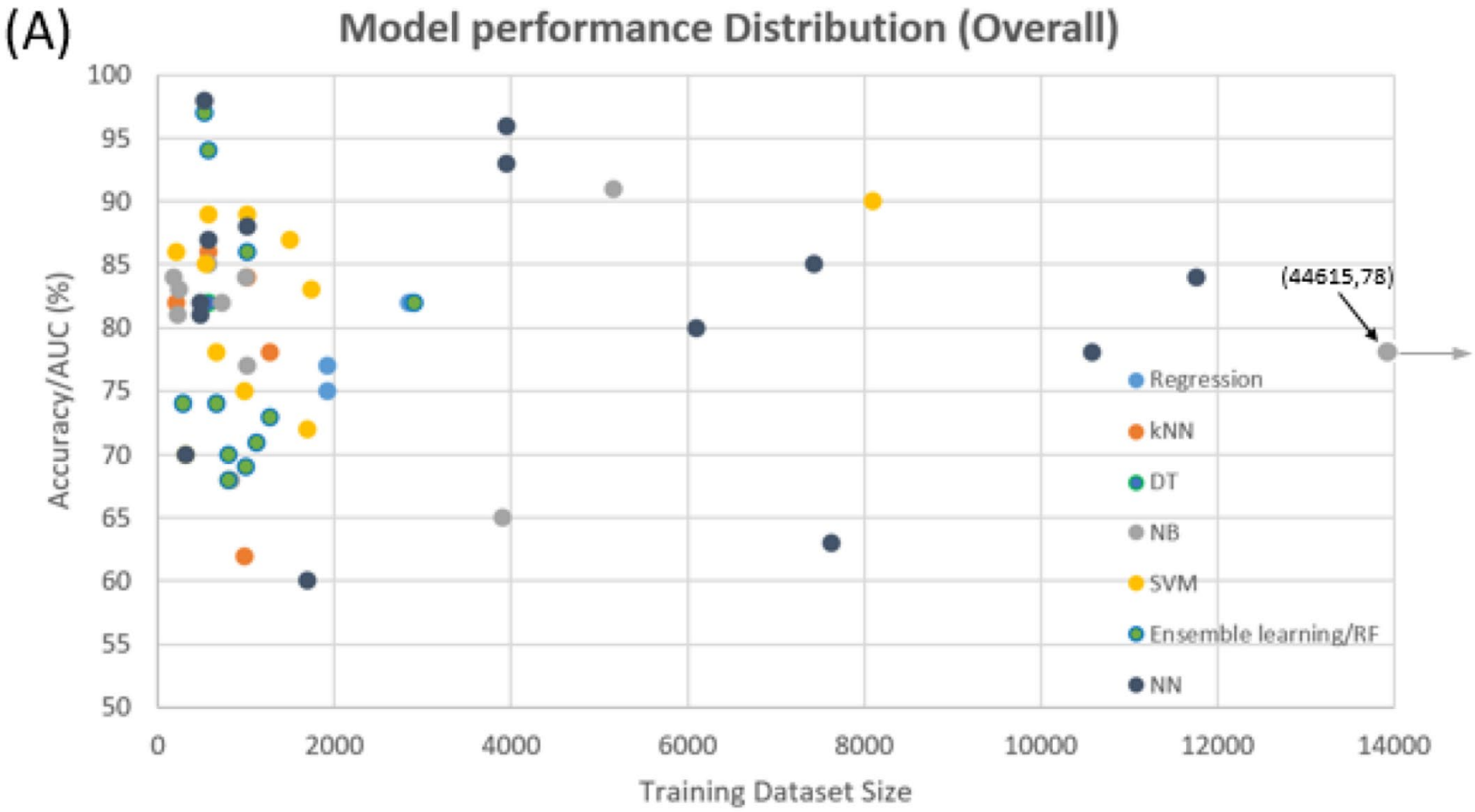


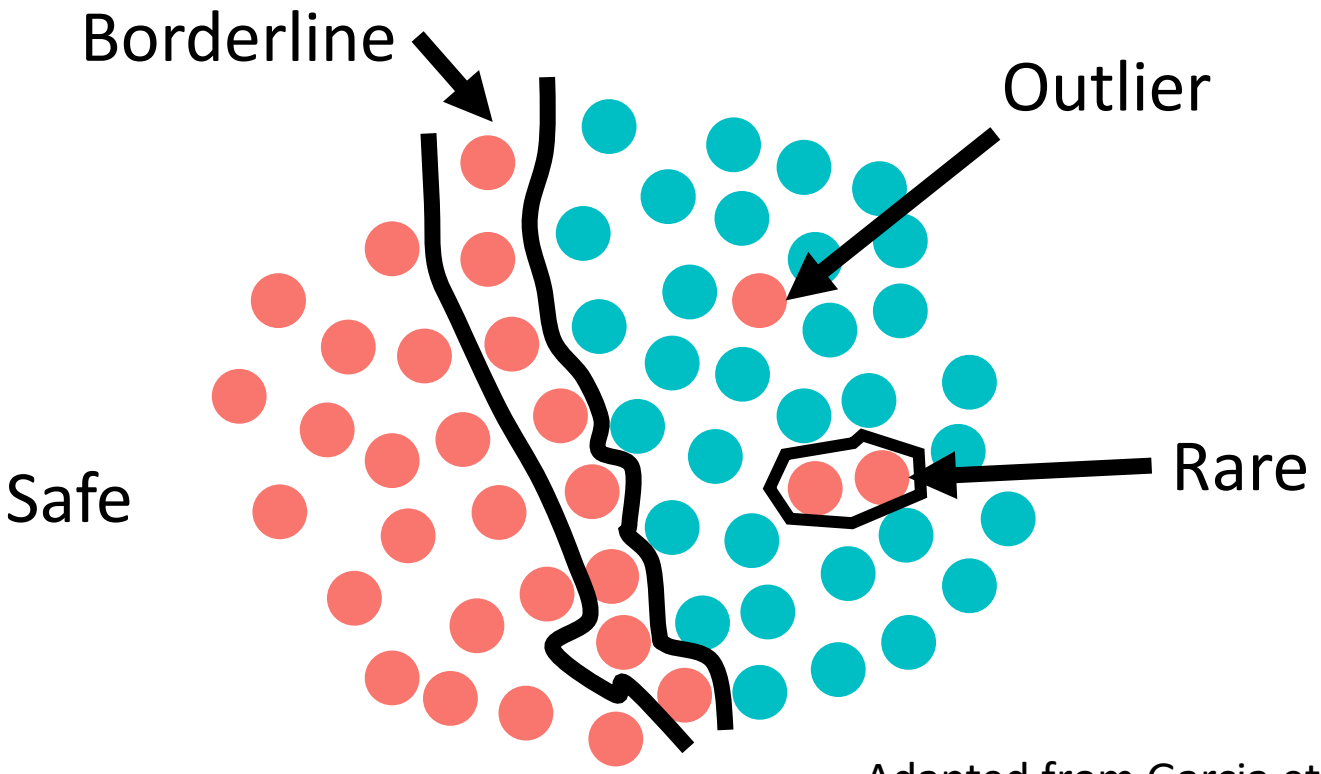
Figure 9A from Wang et al. 2020 (literature review of ML methods in predictive toxicology)

Variability in model performance *not* explained by training dataset size or by ML algorithm used

Authors suspect dataset-specific difficulties

“Data difficulty factors”: ML performance depends on frequency of 4 different types of data points

(Napierla & Stefanowski 2015; Garcia et al. 2020; Stefanowski 2016)



Napierla & Stefanowski 2015:

Undersampling seems to work better for borderline examples

SMOTE seems to work better for outlier & rare examples

Adapted from Garcia et al. (2020)

Suggestion: Develop a more systematic approach to characterize these “data difficulty factors” in predictive toxicology datasets

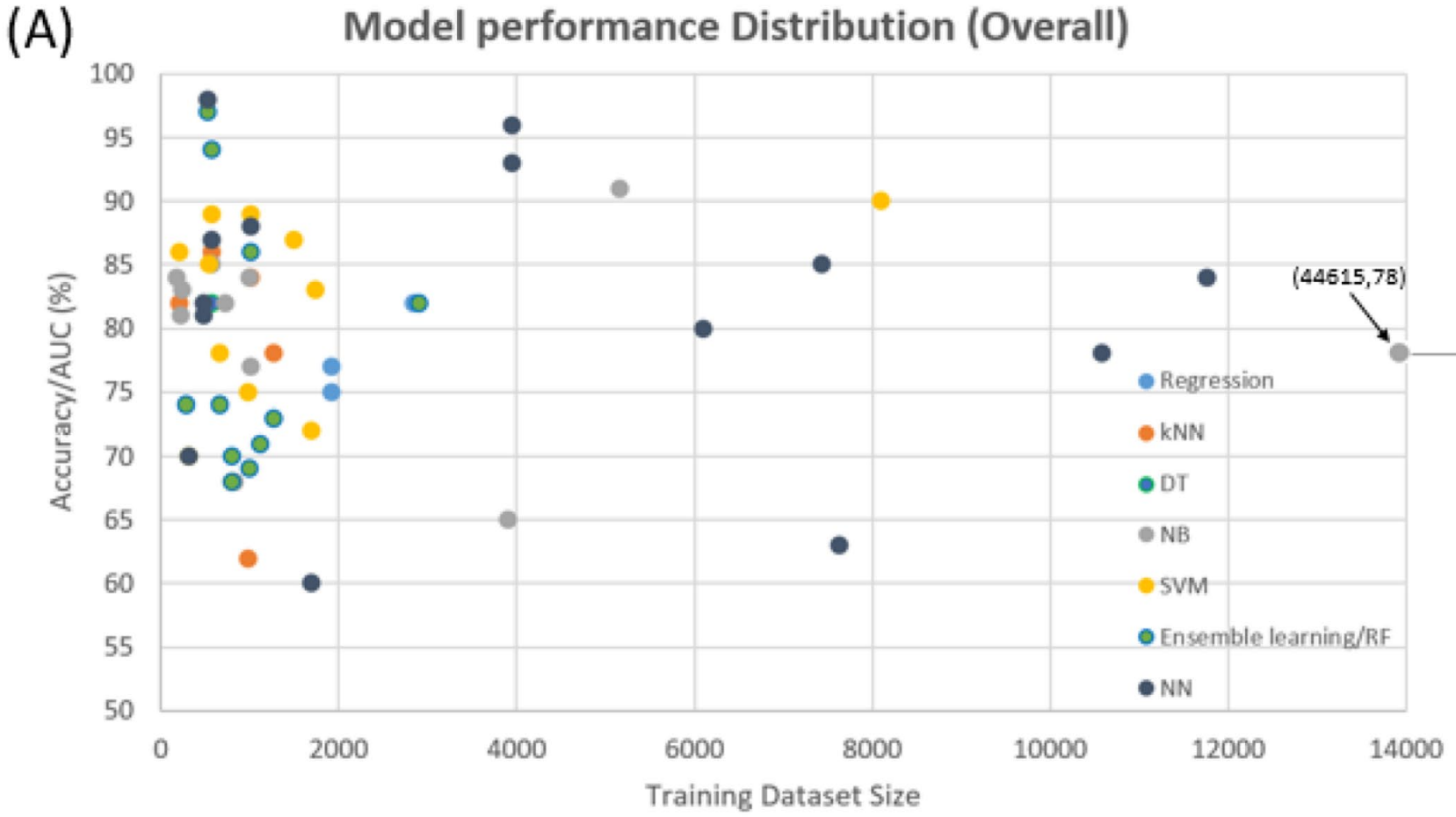


Figure 9A from Wang et al. 2020 (literature review of ML methods in predictive toxicology)

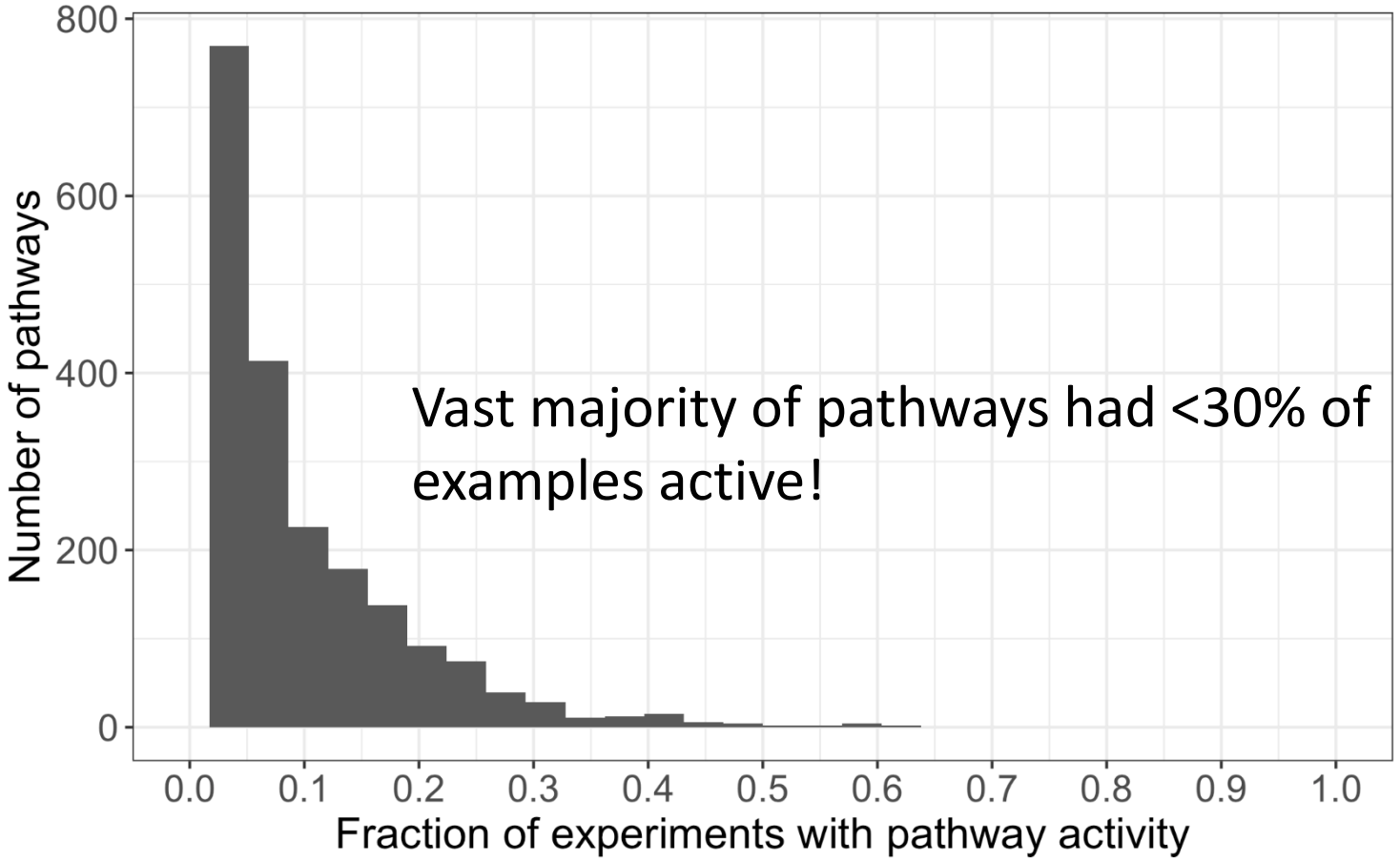
Could it be color-coded by proportion of safe, borderline, rare, and outlier data?

Could we identify “best practices” based on these dataset characteristics?

Case study: Machine learning for *in vitro-in vivo* extrapolation (Ring, Rager, et al. 2021)

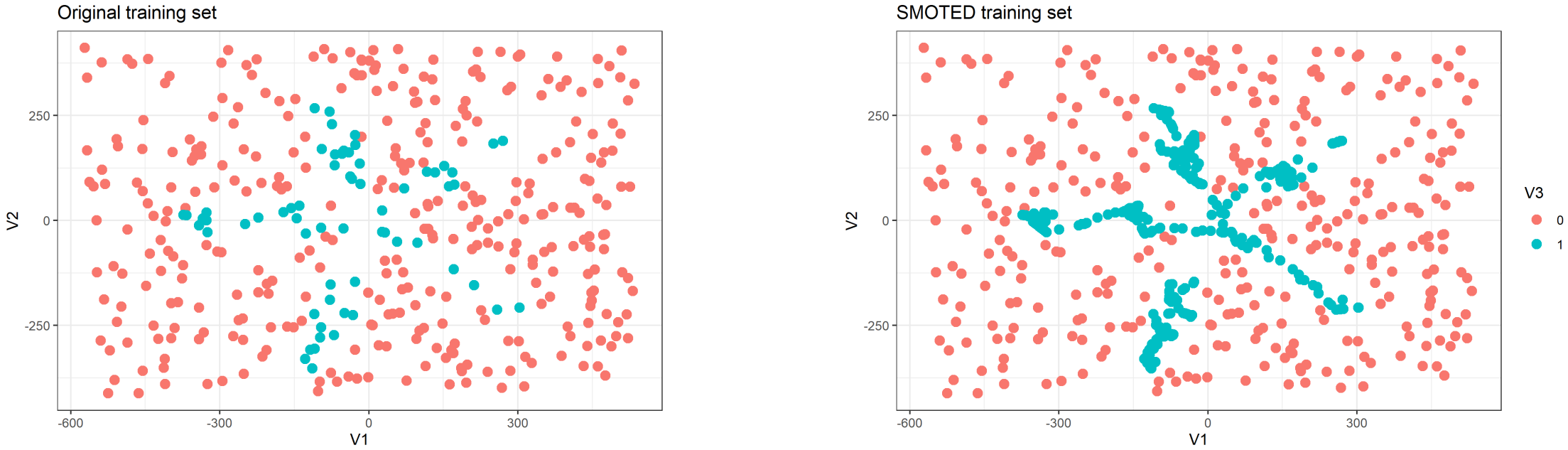
- Target: *in vivo* pathway-level transcriptomic activity (active/inactive) in rat liver, for a given chemical & dose (DrugMatrix and TG-Gates datasets)
- Features:
  - *in vitro* Tox21 bioactive concentrations (AC50) for 144 assays
  - phys-chem properties
  - *in vivo* dose
  - toxicokinetic model predictions of plasma & liver concentration at *in vivo* dose

Challenge: Highly imbalanced data



Approach to imbalanced data: SMOTE (Ring, Rager, et al. 2021)

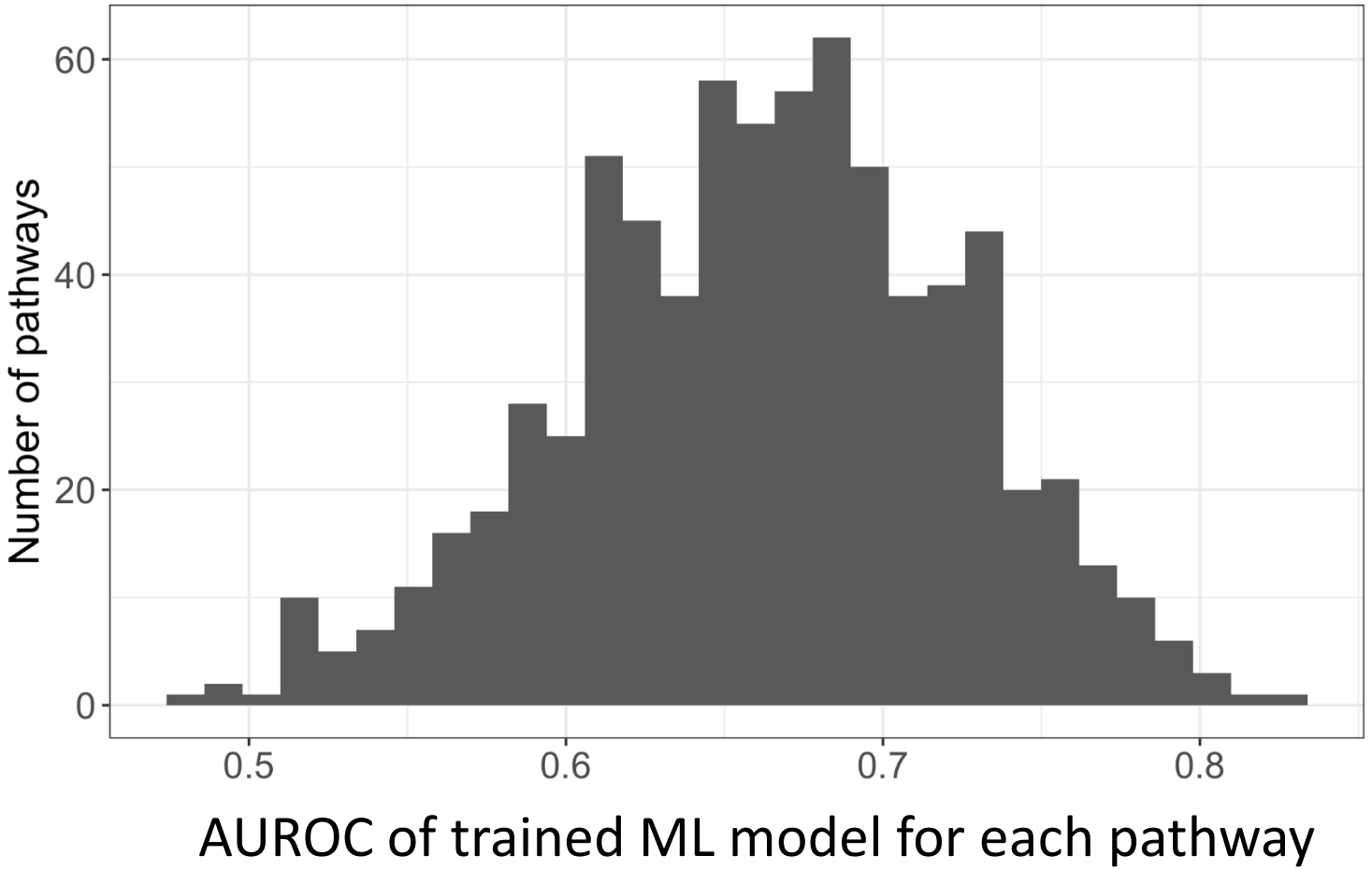
(Illustration of SMOTE on example data from earlier, **not** our actual data)



- High-dimensional feature set: *in vitro* bioactivity for 144 Tox21 assays – interpolating along all of these dimensions
- We did not evaluate “data difficulty factors” in this analysis, so we don’t know about safe, borderline, rare, or outliers

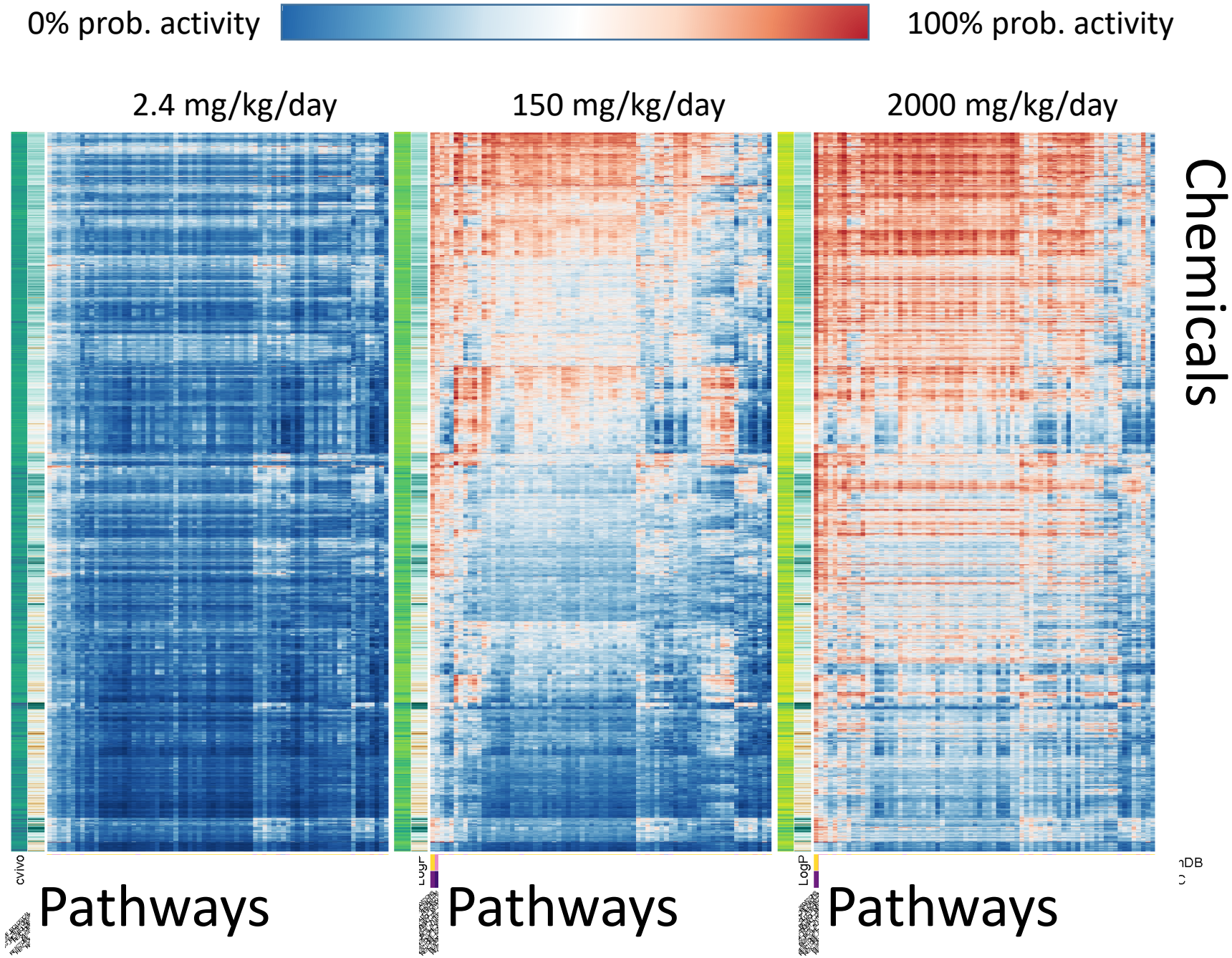
Result: Pathway models with decent AUROC

(Ring, Rager, et al. 2021)



EPA United States Environmental Protection Agency

Result: Apply models to predict pathway activity for 6617 Tox21 chemicals at three doses spanning DrugMatrix dose range (Ring, Rager, et al. 2021)



EPA United States Environmental Protection Agency

Summary

- Machine learning is a powerful tool for predictive toxicology …
- … But its performance is affected by data difficulty factors
  - Imbalanced data
  - Safe, borderline, rare, and outlier data points
- Strategies to address imbalanced data exist & are fairly successful
  - e.g. SMOTE, GAN
  - but data difficulty factors affect these strategies as well
- Suggestion: Develop a more systematic approach to characterizing data difficulty factors for predictive toxicology datasets
- Case study: Machine learning for *in vitro-in vivo* extrapolation
  - Applying SMOTE to address highly imbalanced training data

EPA United States Environmental Protection Agency

References

1. Idakwo GL, J.; Chen, M.; Hong, H.; Zhou, Z.; Gong, P.; Zhang, C. A review on machine learning methods for in silico toxicity prediction. J Environ Sci Health C Environ Carcinog Ecotoxicol Rev. 2018;36(4):169-91.
2. Wang MWH, Goodman JM, Allen TEH. Machine Learning in Predictive Toxicology: Recent Applications and Future Directions for Classification Models. Chem Res Toxicol. 2020.
3. Mansouri K, Kleinstreuer N, Abdelaziz AM, Alberga D, Alves VM, Andersson PL, et al. CoMPARA: Collaborative Modeling Project for Androgen Receptor Activity. Environ Health Perspect. 2020;128(2):27002.
4. Branco P, Torgo L, Ribeiro RP. A Survey of Predictive Modeling on Imbalanced Domains. ACM Computing Surveys. 2016;49(2):1-50.
5. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research. 2002;16:321-57.
6. Douzas G, Bacao F. Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with Applications. 2018;91:464-71.
7. Green AJ, Mohlenkamp MJ, Das J, Chaudhari M, Truong L, Tanguay RL, et al. Leveraging high-throughput screening data, deep neural networks, and conditional generative adversarial networks to advance predictive toxicology. PLoS Comput Biol. 2021;17(7):e1009135.
8. Alcalá-Fdez J, Fernandez A, Luengo J, Derrac J, García S, Sánchez L, et al. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing. 2011;17.
9. Napierala K, Stefanowski J. Types of minority class examples and their influence on learning classifiers from imbalanced data. Journal of Intelligent Information Systems. 2015;46(3):563-97.
10. García V, Sánchez JS, Marqués AI, Florencia R, Rivera G. Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. Expert Systems with Applications. 2020;158.
11. Stefanowski J. Dealing with Data Difficulty Factors While Learning from Imbalanced Data. Challenges in Computational Statistics and Data Mining. Studies in Computational Intelligence2016. p. 333-63.
12. Ring C, Sipes NS, Hsieh JH, Carberry C, Koval LE, Klaren WD, et al. Predictive modeling of biological responses in the rat liver using in vitro Tox21 bioactivity: Benefits from high-throughput toxicokinetics. Comput Toxicol. 2021;18.