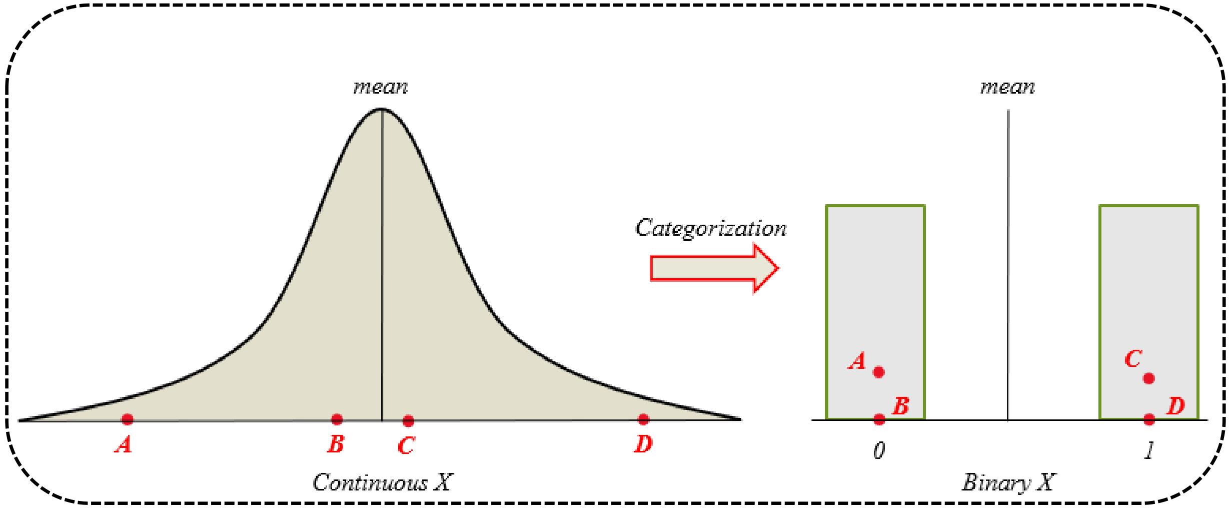
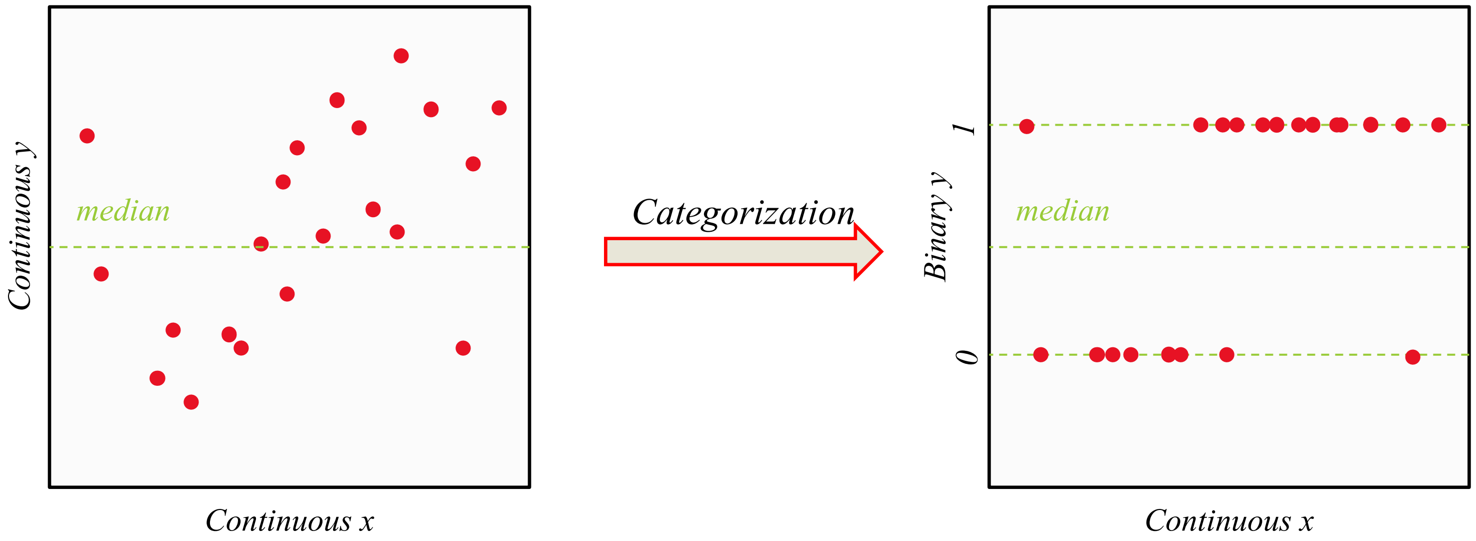
Categorizing Continuous Data in QSAR



Scott Kolmar
U.S. Environmental Protection Agency
Center for Computational Toxicology and Exposure
March 21st, 2022

This presentation does not reflect EPA policy.

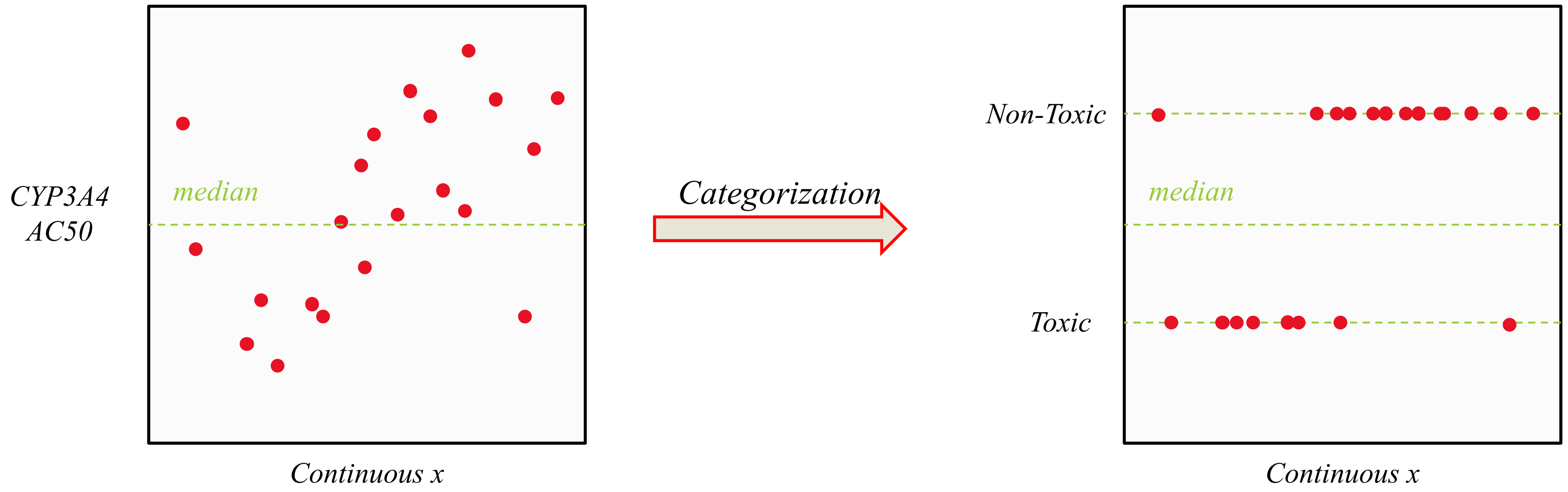
Splitting Data



Modelers often split (categorize) *continuous data* into *categorical data*

This leads to a *loss of information*, *loss of effect size*, and *loss of statistical significance* between variables

Splitting Data

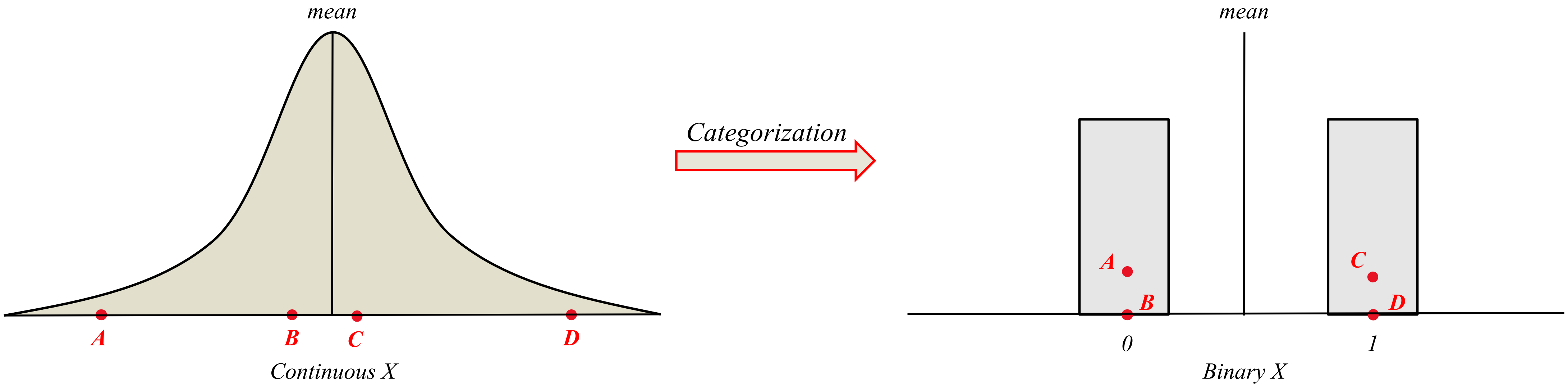


For ADME modeling:

Separate lead compounds into "Non-Toxic" and "Toxic" bins by splitting them on some enzyme activity threshold

- It is far more informative to predict HOW TOXIC a compound is
- Thresholds for splitting are subjective
- Predictions of toxicity can always be categorized AFTER prediction

Loss of Information



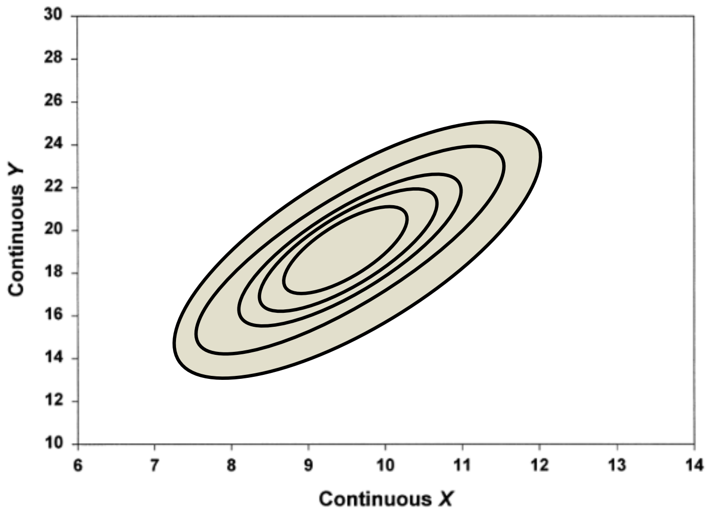
Scenario:

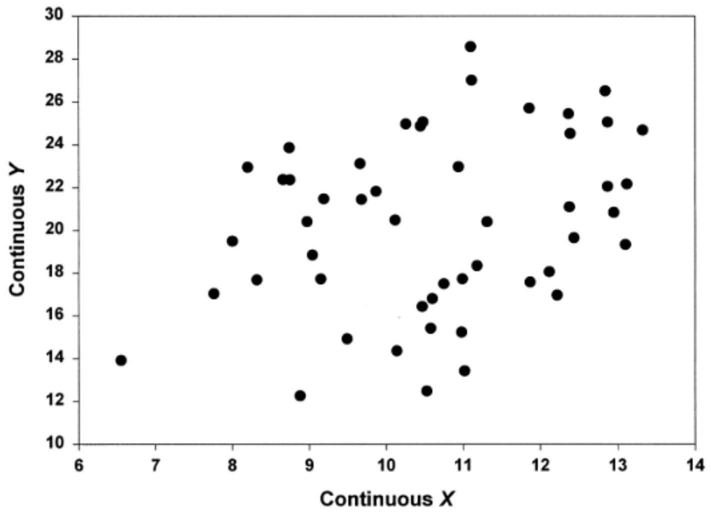
- *C* is closer to *B* than to *D*

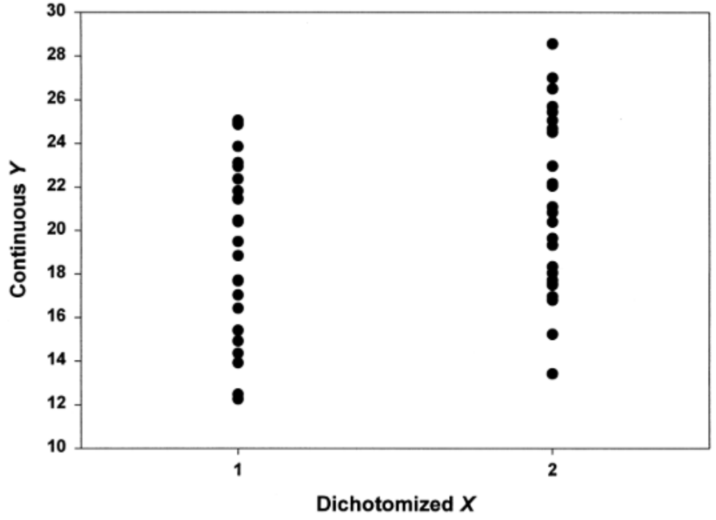
Result:

- Loss of individual differences between observations
- *C* and *D* are judged to be more similar than *C* and *B*

Loss of Effect Size and Statistical Significance







Population:

- $n = > 1x10^{6}$
- $\rho_{xy} = 0.40$

Continuous Sample:

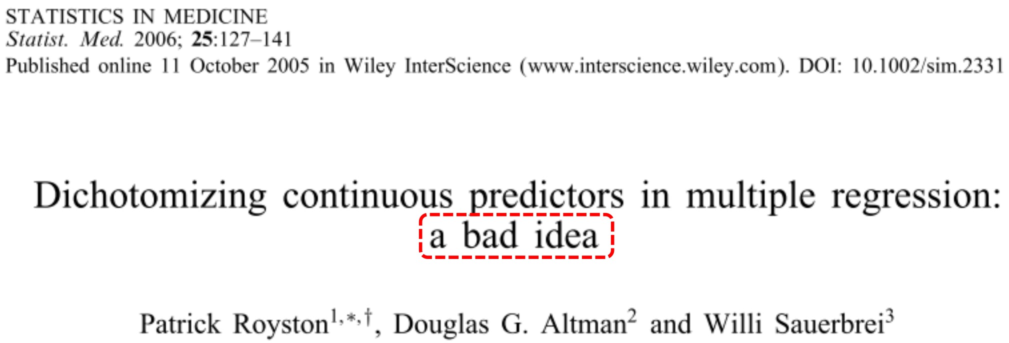
- $n = 50$
- $r_{xy} = 0.30$
- *95% CI* = [0.02, 0.53]
- *Null Hypothesis*: $\rho_{xy} = 0.0$
- $t(48) = 2.19, p = 0.03$

Dichotomized Sample:

- $n = 50$
- $r_{xy} = 0.21$
- *95% CI* = [-0.07, 0.46]
- *Null Hypothesis*: $\mu_1 = \mu_2$
- $t(48) = 1.47, p = 0.15$

MacCallum et al. *Psychological Methods*, **2002**, *7*, 19.

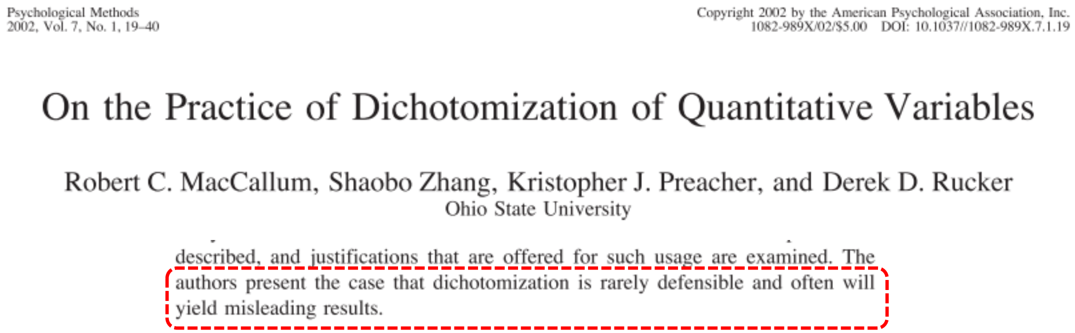
Splitting Up: It's a Bad Idea..



STATISTICS IN MEDICINE
Statist. Med. 2006; **25**:127–141
Published online 11 October 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/sim.2331

Dichotomizing continuous predictors in multiple regression: a bad idea

Patrick Royston[1,*,†], Douglas G. Altman[2] and Willi Sauerbrei[3]



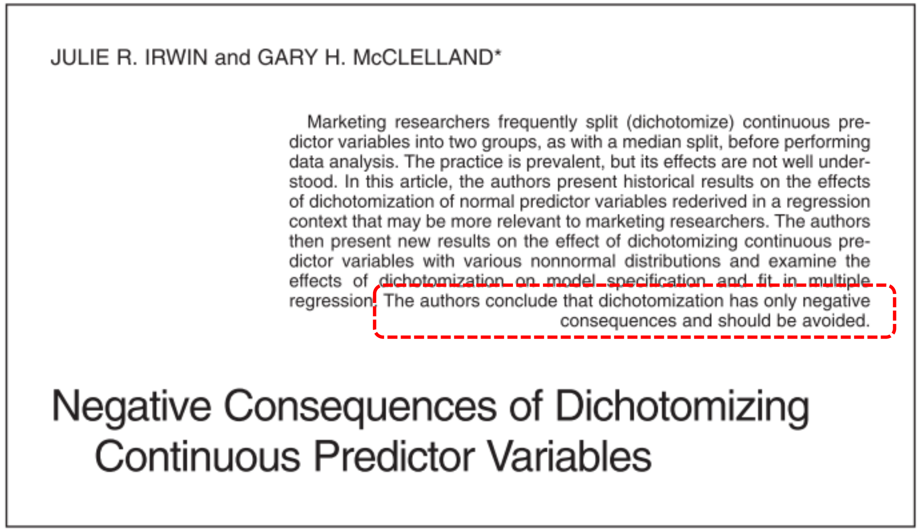
Psychological Methods
2002, Vol. 7, No. 1, 19–40

Copyright 2002 by the American Psychological Association, Inc.
1082-989X/02/$5.00 DOI: 10.1037//1082-989X.7.1.19

On the Practice of Dichotomization of Quantitative Variables

Robert C. MacCallum, Shaobo Zhang, Kristopher J. Preacher, and Derek D. Rucker
Ohio State University

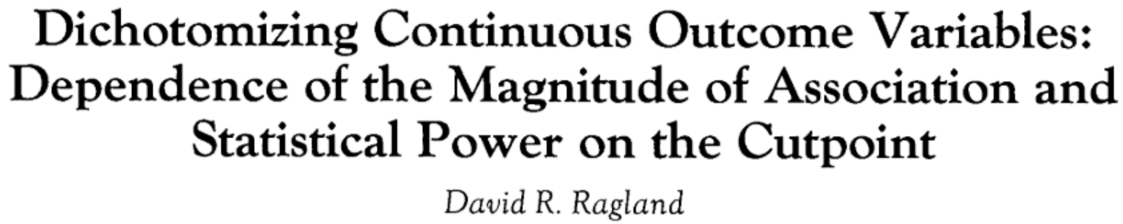
described, and justifications that are offered for such usage are examined. The authors present the case that dichotomization is rarely defensible and often will yield misleading results.



JULIE R. IRWIN and GARY H. McCLELLAND*

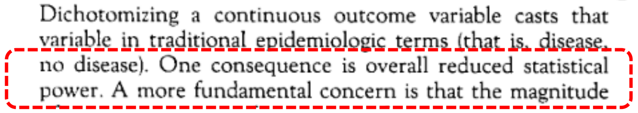
Marketing researchers frequently split (dichotomize) continuous predictor variables into two groups, as with a median split, before performing data analysis. The practice is prevalent, but its effects are not well understood. In this article, the authors present historical results on the effects of dichotomization of normal predictor variables rederived in a regression context that may be more relevant to marketing researchers. The authors then present new results on the effect of dichotomizing continuous predictor variables with various nonnormal distributions and examine the effects of dichotomization on model specification and fit in multiple regression. The authors conclude that dichotomization has only negative consequences and should be avoided.

Negative Consequences of Dichotomizing Continuous Predictor Variables

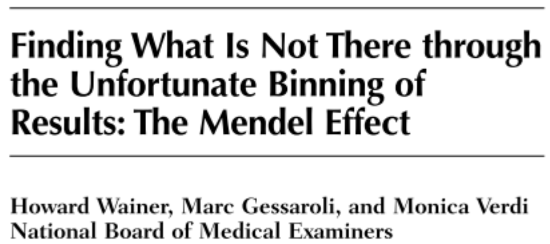


Dichotomizing Continuous Outcome Variables: Dependence of the Magnitude of Association and Statistical Power on the Cutpoint

David R. Ragland

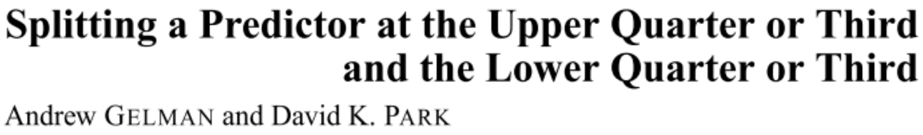


Dichotomizing a continuous outcome variable casts that variable in traditional epidemiologic terms (that is, disease, no disease). One consequence is overall reduced statistical power. A more fundamental concern is that the magnitude



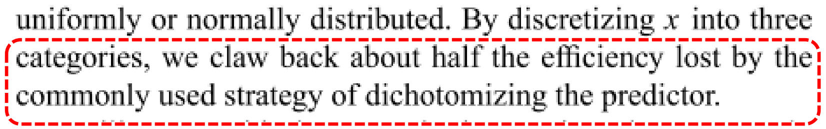
Finding What Is Not There through the Unfortunate Binning of Results: The Mendel Effect

Howard Wainer, Marc Gessaroli, and Monica Verdi
National Board of Medical Examiners



Splitting a Predictor at the Upper Quarter or Third and the Lower Quarter or Third

Andrew GELMAN and David K. PARK



uniformly or normally distributed. By discretizing x into three categories, we claw back about half the efficiency lost by the commonly used strategy of dichotomizing the predictor.

Strategy

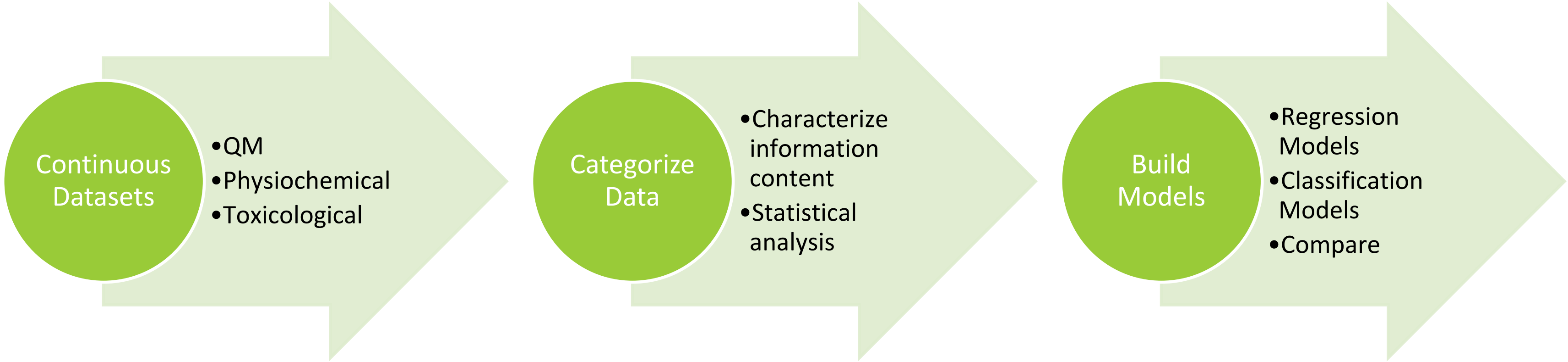
Hypothesis

Categorization of continuous data is bad statistical practice and distorts the relationship between variables.

Will this fundamental principle result in less predictive machine learning models?
How does categorization affect the prediction accuracy?

Approach

Using continuous datasets, make predictions *before* (Regression) and *after* (Classification) categorization.



Continuous Datasets
- QM
- Physiochemical
- Toxicological

Categorize Data
- Characterize information content
- Statistical analysis

Build Models
- Regression Models
- Classification Models
- Compare

Datasets

Dataset	Category	Number of Molecules[a]	Endpoint	Range
G298_atom[1]	Quantum Mechanical	131,082	ΔG^{o}_{at} (kcal mol^{-1})	-2,417 − -288
Solv	Physiochemical	642	ΔG^{o}_{hyd} (kcal mol^{-1})	-25.5 − 3.4
Tox_102[b,2]	Toxicological *in vitro*	971	$\log AC_{50}$	-2.1 − 4.7
Tox_134[c,2]	Toxicological *in vitro*	1,347	$\log AC_{50}$	-4.0 – 2.8

[a] Original size of the dataset. If datasets have more than 1,000 molecules, they were randomly sampled down to a size of 1,000 before modeling.

[b] Includes data exclusively from the ATG-PPre-cis assay

[c] Inclues data exclusively from the ATG-PPARg-trans assay

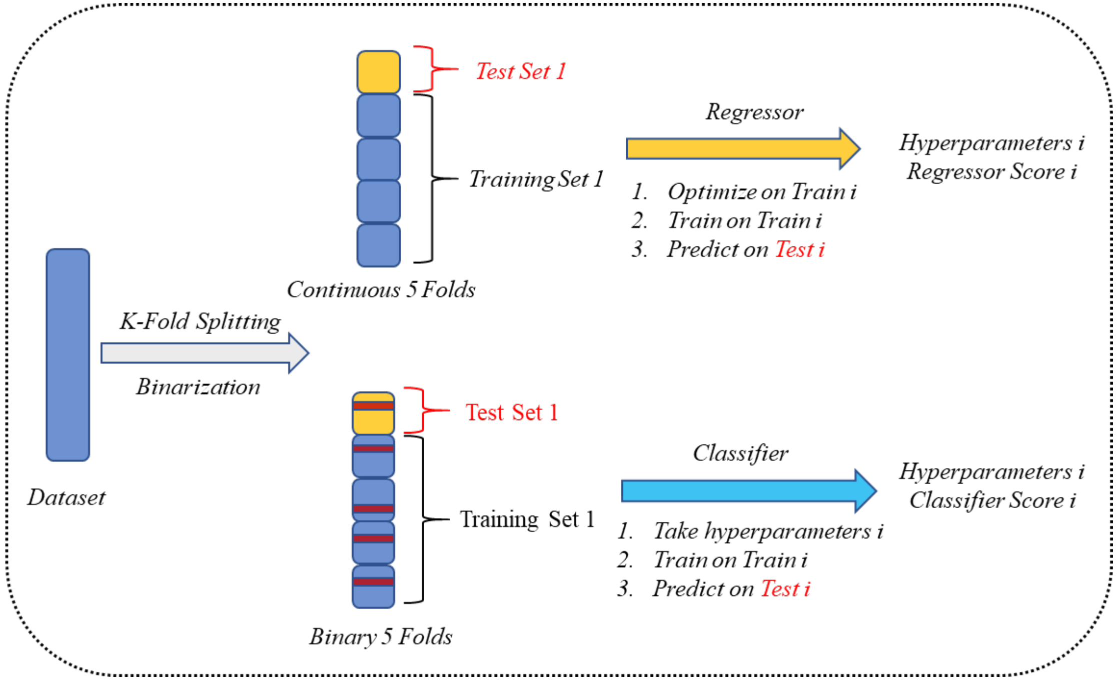
[1] Blum and Reymond, *J. Amer. Chem. Soc.*, **2009**, *131*, 8732.

[2] Richard et al., *Chem. Res. in Toxicol.*, **2016**, *29*, 1225.

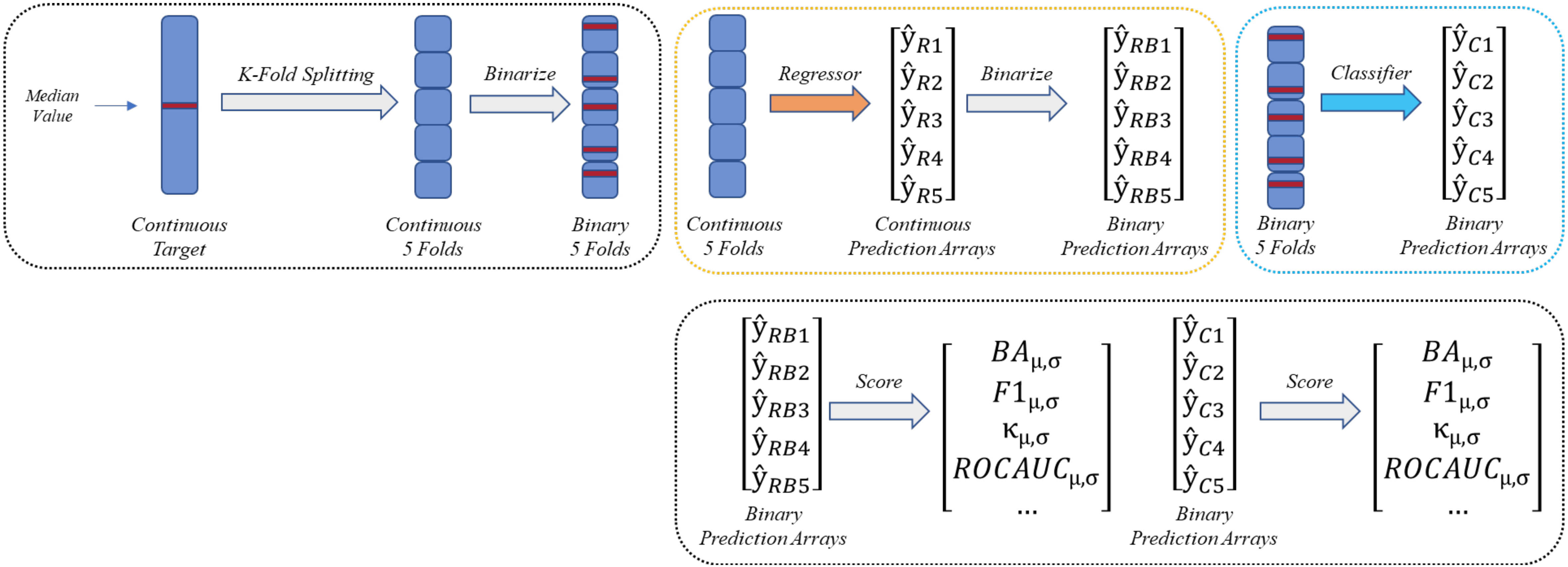
Algorithms and Hyperparameters

Algorithm	Hyperparameters Searched in Optimization[a,b]
Decision Tree (DT)	*max depth* $\in (50, 100, 200, 500)$
	min samples split $\in (2, 5, 10, 20, 40)$
	min samples leaf $\in (1, 5, 10, 20)$
k- Nearest Neighbors (kNN)	$k \in (2,\ 3, \dots, 22)$
Random Forest (RF)	*n estimators* $\in (10, 25, 50, 100, 150, 200)$
	max depth $\in (50, 100, 200, 500)$
	min samples split $\in (2, 5, 10, 20, 40)$
	min samples leaf $\in (1, 5, 10, 20)$
Support Vector Machines (SVM)	*kernel*: RBF, Sigmoid
	$C \in (0.001, 0.01,\ 0.1,\ 1, 10)$
Deep Neural Network (DNN)	*N hidden layers* $\in (2,3,4,5,6,7,8)$
	N hidden units per layer $\in (32, 64, 128)$
	Regularizer: L1, L2, No regularizer
	Output layer bias: True or False
	Class weighting: True or False

Hyperparameter optimization



Comparing Classification and Regression



Median Value

Continuous Target

K-Fold Splitting

Continuous 5 Folds

Binarize

Binary 5 Folds

Continuous 5 Folds

Regressor

$\begin{bmatrix} \hat{y}_{R1} \\ \hat{y}_{R2} \\ \hat{y}_{R3} \\ \hat{y}_{R4} \\ \hat{y}_{R5} \end{bmatrix}$

Continuous Prediction Arrays

Binarize

$\begin{bmatrix} \hat{y}_{RB1} \\ \hat{y}_{RB2} \\ \hat{y}_{RB3} \\ \hat{y}_{RB4} \\ \hat{y}_{RB5} \end{bmatrix}$

Binary Prediction Arrays

Binary 5 Folds

Classifier

$\begin{bmatrix} \hat{y}_{C1} \\ \hat{y}_{C2} \\ \hat{y}_{C3} \\ \hat{y}_{C4} \\ \hat{y}_{C5} \end{bmatrix}$

Binary Prediction Arrays

$\begin{bmatrix} \hat{y}_{RB1} \\ \hat{y}_{RB2} \\ \hat{y}_{RB3} \\ \hat{y}_{RB4} \\ \hat{y}_{RB5} \end{bmatrix}$

Binary Prediction Arrays

Score

$\begin{bmatrix} BA_{\mu,\sigma} \\ F1_{\mu,\sigma} \\ \kappa_{\mu,\sigma} \\ ROCAUC_{\mu,\sigma} \\ \dots \end{bmatrix}$

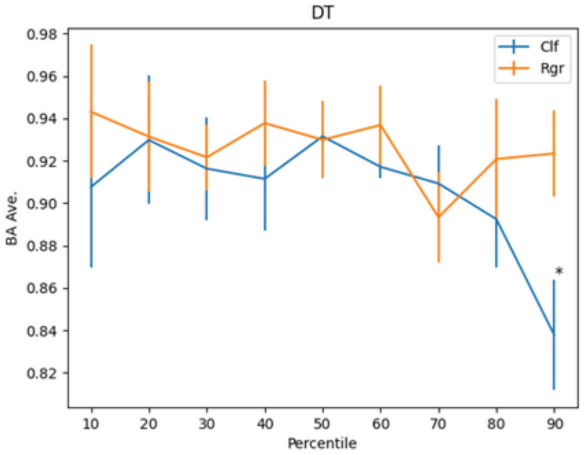
$\begin{bmatrix} \hat{y}_{C1} \\ \hat{y}_{C2} \\ \hat{y}_{C3} \\ \hat{y}_{C4} \\ \hat{y}_{C5} \end{bmatrix}$

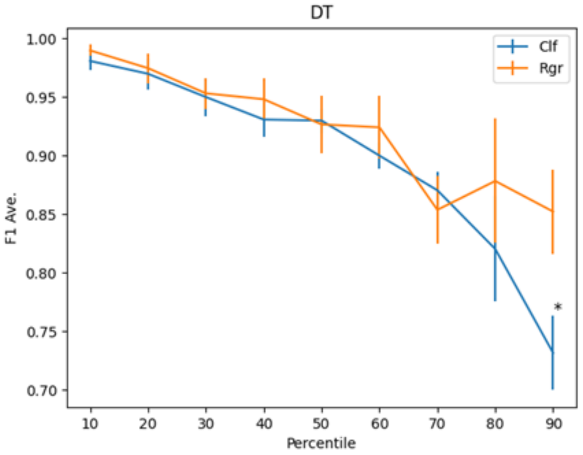
Binary Prediction Arrays

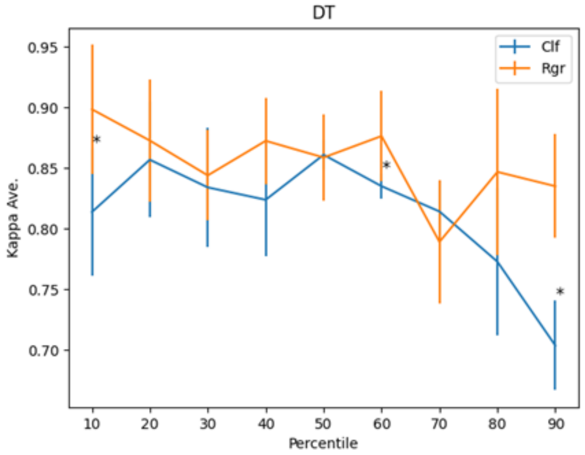
Score

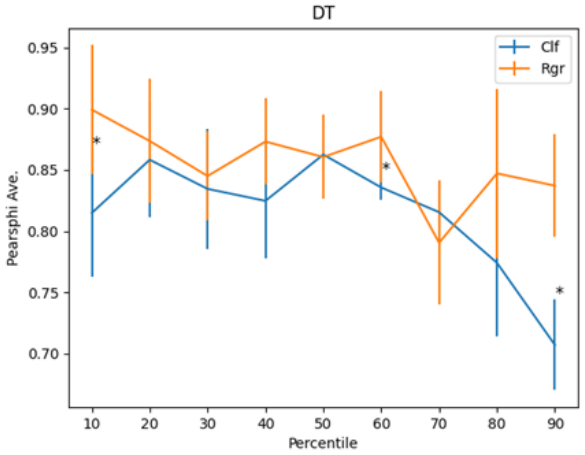
$\begin{bmatrix} BA_{\mu,\sigma} \\ F1_{\mu,\sigma} \\ \kappa_{\mu,\sigma} \\ ROCAUC_{\mu,\sigma} \\ \dots \end{bmatrix}$

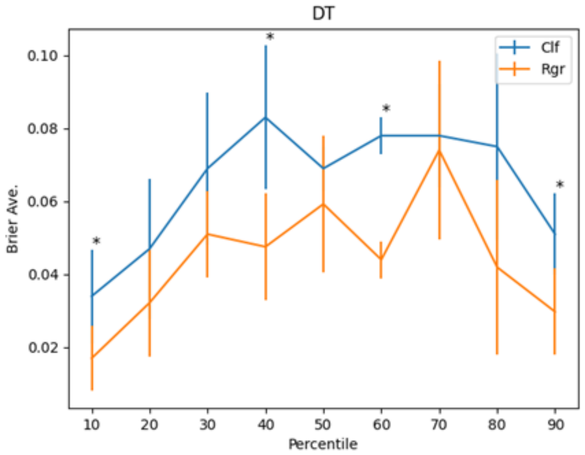
G298Atom - DT

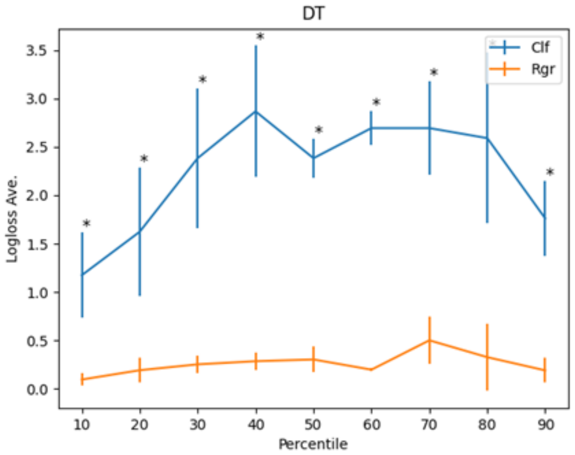


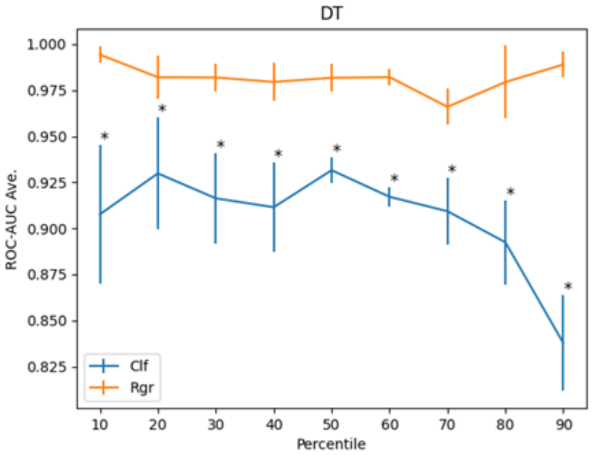












** Statistical significance of difference of means determined by independent T-test with equal variances*

G298Atom

Each cell is for splitting at the 50th percentile; ***orange****: regressor has higher score,* ***blue****: classifier has higher score,* ***white****: statistically insignificant difference*

1000

G298atom	**DT**	**kNN**	**RF**	**SVM**
BA				
F1				
Kappa				
Pearsphi			🟦	
Brier				
Logloss	🟧			
ROC-AUC	🟧		🟦	

1000, Optimized hyperparameters

G298atom	**DT**	**kNN**	**RF**	**SVM**
BA				
F1				
Kappa				
Pearsphi				
Brier				🟦
Logloss	🟧		🟦	🟦
ROC-AUC	🟧			

1000, Scaled

G298atom	**DT**	**kNN**	**RF**	**SVM**
BA				🟦
F1				🟦
Kappa				🟦
Pearsphi				🟦
Brier				🟦
Logloss	🟧		🟦	🟦
ROC-AUC	🟧		🟦	🟦

1000, Scaled, CorrFilt60

G298atom	**DT**	**kNN**	**RF**	**SVM**
BA	🟧			🟦
F1	🟧			🟦
Kappa	🟧			🟦
Pearsphi	🟧			🟦
Brier	🟧			🟦
Logloss	🟧	🟧		🟦
ROC-AUC	🟧	🟧	🟦	🟦

1000, Scaled, VarFilt25, CorrFilt60

G298atom	**DT**	**kNN**	**RF**	**SVM**
BA	🟧			🟦
F1				🟦
Kappa	🟧			🟦
Pearsphi	🟧		🟦	🟦
Brier	🟧			🟦
Logloss	🟧	🟧		🟦
ROC-AUC	🟧		🟦	🟦

1000, Scaled, VarFilt25, CorrFilt95, Optimized

G298atom	**DT**	**kNN**	**RF**	**SVM**
BA				🟧
F1				🟧
Kappa				🟧
Pearsphi				🟧
Brier				
Logloss	🟧			
ROC-AUC	🟧	🟧		

DNN Results

Each cell represents the results for splitting data at the 50th percentile

No Regularization

G298atom/	**32*2**	**32*8**	**64*2**	**64*8**	**128*2**	**128*8**
BA	🟦	🟦	🟦	🟦	🟦	🟦
F1	🟦	🟦	🟦	🟦	🟦	🟦
Kappa	🟦	🟦	🟦	🟦	🟦	🟦
PearsPhi	🟦	🟦	🟦	🟦	🟦	🟦
Brier	🟦	🟦	🟦	🟦	🟦	🟦
Logloss						
ROC-AUC	🟦	🟦	🟦	🟦	🟦	🟦

L1 Regularization

G298atom/	**32*2**	**32*8**	**64*2**	**64*8**	**128*2**	**128*8**
BA	🟦	🟦	🟦	🟨		
F1	🟦	🟦	🟦	🟨		
Kappa	🟦	🟦	🟦	🟨		
PearsPhi	🟦	🟦	🟦	🟨		
Brier	🟦	🟦	🟦	🟨	🟦	
Logloss	🟦		🟦	🟨	🟦	
ROC-AUC	🟦	🟦	🟦	🟨		

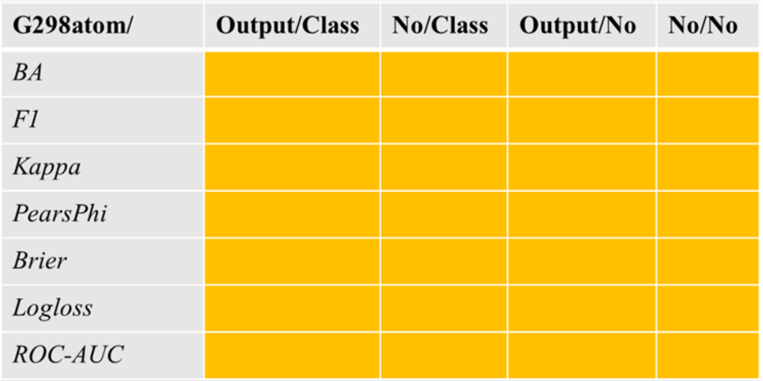
L2 Regularization

G298atom/	**32*2**	**32*8**	**64*2**	**64*8**	**128*2**	**128*8**
BA	🟦	🟦	🟦	🟦	🟦	🟦
F1	🟦	🟦	🟦	🟦	🟦	🟦
Kappa	🟦	🟦	🟦	🟦	🟦	🟦
PearsPhi	🟦	🟦	🟦	🟦	🟦	🟦
Brier	🟦	🟦	🟦	🟦	🟦	🟦
Logloss		🟦		🟦		🟦
ROC-AUC	🟦	🟦	🟦	🟦	🟦	🟦

DNN Results

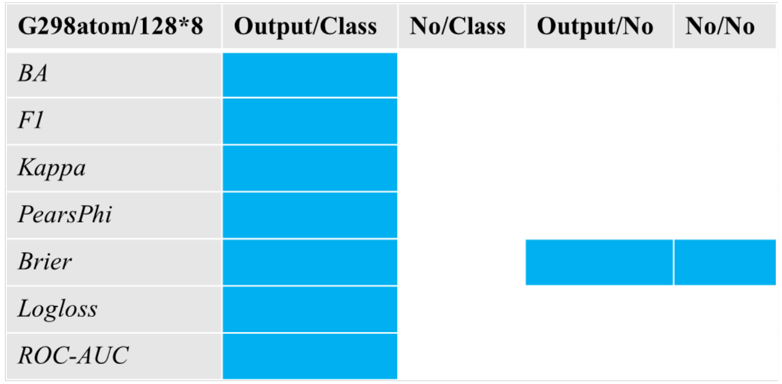
Using an architecture with 8 hidden layers and 128 nodes per layer, the class weighting and output bias of the DNN classifiers were turned on and off

L1 Regularization



G298atom/	Output/Class	No/Class	Output/No	No/No
BA	🟨	🟨	🟨	🟨
F1	🟨	🟨	🟨	🟨
Kappa	🟨	🟨	🟨	🟨
PearsPhi	🟨	🟨	🟨	🟨
Brier	🟨	🟨	🟨	🟨
Logloss	🟨	🟨	🟨	🟨
ROC-AUC	🟨	🟨	🟨	🟨

L2 Regularization



G298atom/128*8	Output/Class	No/Class	Output/No	No/No
BA	🟦			
F1	🟦			
Kappa	🟦			
PearsPhi	🟦			
Brier	🟦		🟦	🟦
Logloss	🟦			
ROC-AUC	🟦			

Conclusions

Approach

- Categorization of continuous data is bad statistical practice. But does it affect the predictivity of models?
- By making predictions *before* (*regression*) and *after* (*classification*) categorizing a continuous dataset, we can explore how categorization affects model performance

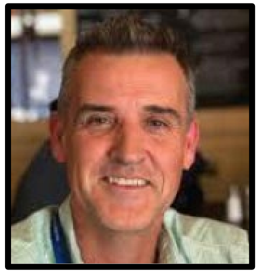
Results

- There are observable differences in model performance when continuous data is categorized
- Relative performance is dependent on cutpoint, algorithm, and dataset
- *Probabilistic metrics* are sometimes needed to distinguish performance
- *Optimization*, *variance filtering,* and *correlation filtering* change the relative performance
- The relative performance of DNN regressors and classifiers have some dependence on network architecture and regularization

Acknowledgements

Mentors

Chris Grulke



Antony Williams

Computational Chemistry and Cheminformatics Branch (CCCB)

PIs

Daniel Chang
Chris Grulke
Paul Harten
Todd Martin
Grace Patlewicz
Ann Richard
Dan Vallero
Antony Williams

Postdocs and SSCs

Matthew Boyce
Zachary Chiodini
Willysha Jenkins
Charles Lowe
Christian Ramsland
Gabriel Sinclair
Tia Tate

Tox102 and Tox134 Datasets

Katie Paul-Friedman

Thank You!

Q & A