

A tutorial on interpretable machine learning algorithms for understanding factors related to childhood autism

Toby Dylan Hocking
toby.hocking@nau.edu
toby.hocking@r-project.org

March 26, 2024


Motivation and data for predicting childhood autism

- ▶ We have National Survey of Children's Health data (NSCH).
- ▶ Each year a number of people fill out the survey (rows), and we have data for their responses (columns).
- ▶ One column, k2q35a "Autism ASD" (Yes or No) represents if the child has Autism.
- ▶ **Data pre-processing**: operations prior to machine learning.
- ▶ **Prediction accuracy in a given year**: can we predict Autism variable (output/label/dependent), given the others? (inputs/features/independent)
- ▶ **Model interpretation / feature selection**: which inputs are most useful for prediction?
- ▶ **Similarity/difference between years**: Can we train on one survey year, and accurately predict on another?


Machine learning overview and citation

- ▶ In supervised machine learning, train data are paired inputs x (images below; survey questions for NSCH) and outputs y (integer class); goal is accurate prediction on test data.
- ▶ Hocking TD. Chapter *Introduction to machine learning and neural networks* for book *Land Carbon Cycle Modeling: Matrix Approach, Data Assimilation, and Ecological Forecasting* edited by Luo Y, published by Taylor and Francis (2022).

Learning Algorithm Train data Learned function Predictions on test data

Learn() \rightarrow g

$g(\text{0}) = 0$
 $g(\text{1}) = 1$
 $g(\text{7}) = 1$

Learn() \rightarrow h

$h(\text{shirt}) = 0$
 $h(\text{shirt}) = 0$
 $h(\text{pants}) = 1$

Data pre-processing

Prediction accuracy in a given year

Model interpretation / feature selection

Similarity/difference between years

Discussion and Conclusions

Data pre-processing overview

- ▶ Goal: for machine learning, need a CSV table with rows for people, columns for survey questions.
- ▶ Download NSCH data from public Census web site.
- ▶ For each year, keep all columns with less than 10% missing values, then remove all rows with at least one missing value.
- ▶ One-hot recoding of categorical variables (create 0/1 dummy/indicator variable for each value).
- ▶ Then keep only columns in common between both years:
result is 46,010 rows and 366 columns. Details:

year	data.type	nrow	ncol	questions	%Autism	%rowsNA	%colsNA
2019	raw	29433	443	443	2.9615	100.0000	90.0677
2019	processed	18202	377	187	2.9997	0.0000	0.0000
2020	raw	42777	443	443	2.9758	100.0000	90.0677
2020	processed	27808	373	185	3.0818	0.0000	0.0000

Data source:

http://www2.census.gov/programs-surveys/nsch/datasets/2019/nsch_2019_topical_Stata.zip nsch_2019_topical.dta, nsch_2019_topical.do

http://www2.census.gov/programs-surveys/nsch/datasets/2020/nsch_2020_topical_Stata.zip nsch_2020_topical.dta, nsch_2020_topical.do

One-hot encoding of categorical variables

Sometimes called dummy/indicator variables in statistics.
For each value, we create a new column with 0/1 values.
For example, from `nsch_2020_topical.do`

```
label var k4q24_r "Specialist Visit"  
label define k4q24_r_lab 1 "Yes"  
label define k4q24_r_lab 2 "No, but this child needed  
to see a specialist", add  
label define k4q24_r_lab 3 "No, this child did not  
need to see a specialist", add
```

code above means there is a column named `k4q24_r` in
`nsch_2020_topical.dta`, with values 1, 2, 3.

In our analysis we use a one-hot encoding, which means deleting
that column, and creating two 0/1 columns:

Specialist Visit=Yes and

Specialist Visit=No, but this child needed to see a specialist

R software citations

We used the following free/open-source software:

Base R system. R Core Team (2023). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Reading Stata dta files in R. Wickham H, Miller E, Smith D (2023). haven: Import and Export 'SPSS', 'Stata' and 'SAS' Files. R package version 2.5.4.

Data manipulation, reshaping, summarization. Barrett T, Dowle M, Srinivasan A, Gorecki J, Chirico M, Hocking T (2024). data.table: Extension of data.frame. R package version 1.15.0.

Regular expressions for parsing Stata do files in R. Hocking TD (2023). nc: Named Capture to Data Tables. R package version 2023.8.24.

Data visualization. H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Data pre-processing

Prediction accuracy in a given year

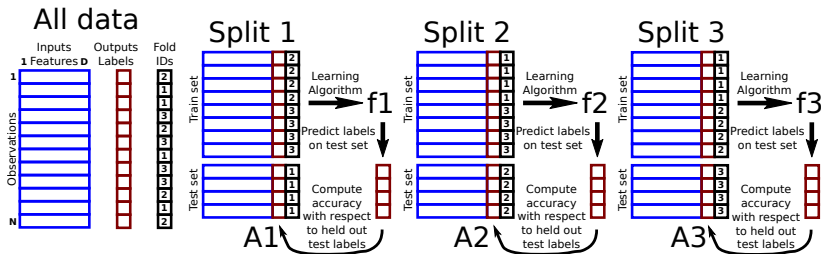
Model interpretation / feature selection

Similarity/difference between years

Discussion and Conclusions

K -fold cross-validation: a standard algorithm used to estimate the prediction accuracy in machine learning

- ▶ $K = 3$ folds shown in figure below, meaning three different models trained, and three different prediction/test accuracy rates computed.
- ▶ It is important to use several train/test splits, so we can see if there are statistically significant differences between algorithms.
- ▶ Rows/observations are people, inputs/features are survey questions, and output/label is Autism response (Yes or No).



Learning algorithms we consider

We use R packages that implement the following learning algorithms, in the mlr3 R package framework:

`cv_glmnet` L1-regularized linear model (feature selection).
Friedman, *et al.* (2010).

`xgboost` Extreme gradient boosting (non-linear). Chen and Guestrin (2016).

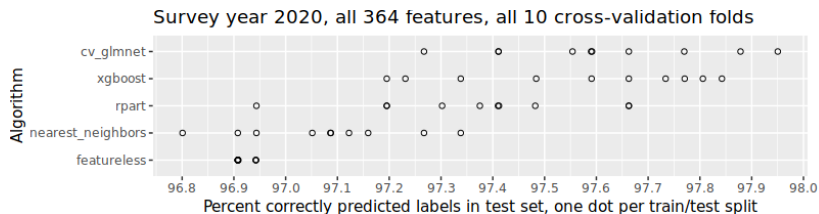
`rpart` Recursive partitioning, decision tree (non-linear, feature selection). Therneau and Atkinson (2023).

`nearest_neighbors` classic non-linear algorithm, as implemented in `knn` R package. Schliep and Hechenbichler (2016).

`featureless` un-informed baseline, ignores all inputs/features, and always predicts the most frequent label in train data (Autism=No in our case). Nomenclature from `mlr3` R package, Lang, *et al.*, (2019).

Each learning algorithm has different properties (non-linear, feature selection, etc). For details see Hastie, *et al.* (2009) textbook.

10-fold cross-validation for comparing learning algorithms



Each dot can be computed in parallel:

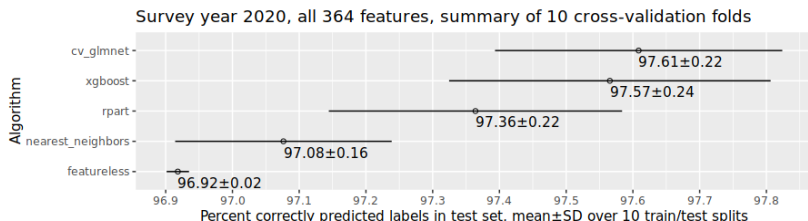
50x speedups for this figure, 5 algorithms \times 10 cross-validation folds.

NAU Monsoon super-computer cluster with 4000 CPUs, managed with SLURM scheduler software. Typically 50-500x speedups relative to sequential computation (1 CPU).

batchtools R package interface to SLURM system.

mlr3batchmark R package for running machine learning computations in parallel using batchtools.

Summarize 10 folds with mean and standard deviation



Learning algorithms we consider:

cv_glmnet L1-regularized linear model (feature selection).

xgboost Extreme gradient boosting (non-linear).

rpart Recursive partitioning, decision tree (non-linear, feature selection).

nearest_neighbors classic non-linear algorithm.

featureless un-informed baseline, ignores all inputs/features, and always predicts the most frequent label in train data (Autism=No in our case).

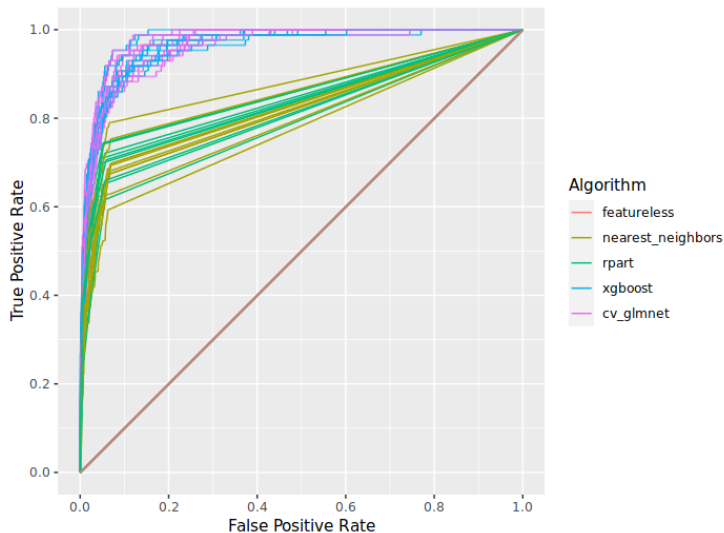
Confusion matrix and error rates

	Label 0/No Autism	Label 1/Yes Autism
Predict 0	True Negative (TN)	False Negative (FN)
Predict 1	False Positive (FP)	True Positive (TP)

- ▶ Each has a corresponding rate which is a proportion between zero and one, for example $FPR = \text{False Positive Rate}$.
- ▶ Rates are related, $TPR = 1 - FNR$ quantifies accuracy for positive labels, and $TNR = 1 - FPR$ is for negative labels.
- ▶ TN/TP are good (want to maximize), whereas FP/FN are bad (want to minimize).
- ▶ Ideal rates are $FPR = 0$ and $TPR = 1$ but that is not possible to achieve in most real data.
- ▶ Receiver Operating Characteristic (ROC) curves trace TPR as a function of FPR, for every threshold of the predicted scores $f(x) \in \mathbb{R}$ (default threshold is typically 0, smaller thresholds result in more positive predictions, etc).

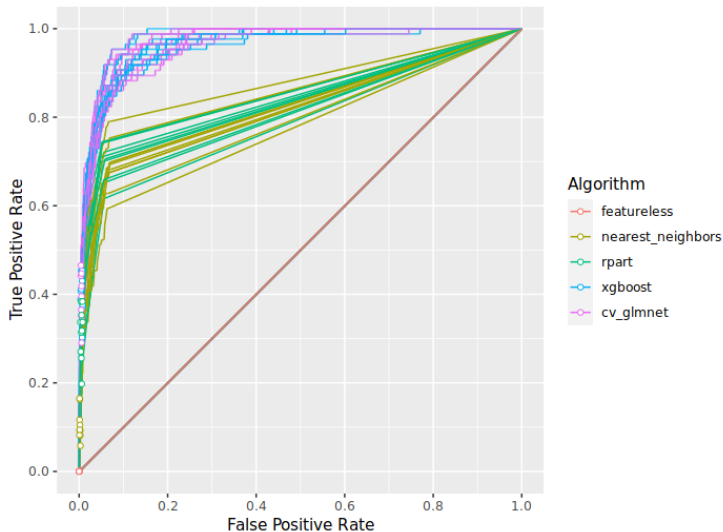
ROC curves show all tradeoffs between TPR and FPR

Survey year 2020, all 364 features,
One ROC curve per cross-validation fold



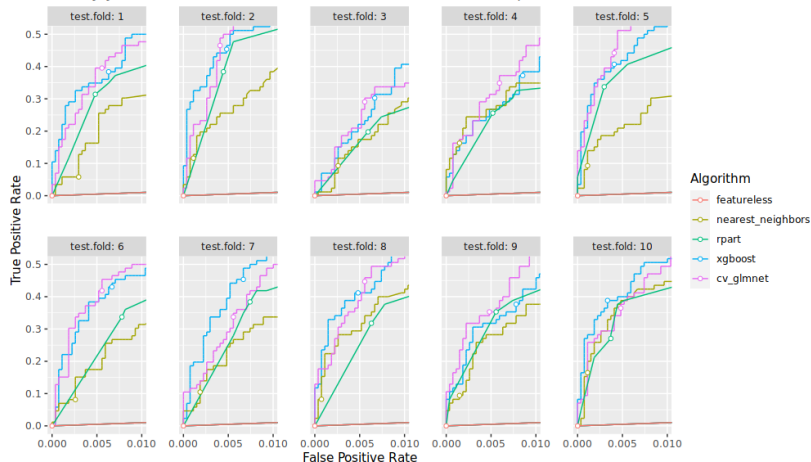
Default prediction threshold can be viewed as a dot

Survey year 2020, all 364 features,
One ROC curve per cross-validation fold



Default prediction threshold can be viewed as a dot

Survey year 2020, all 364 features, zoom to show FPR/TPR of predictions

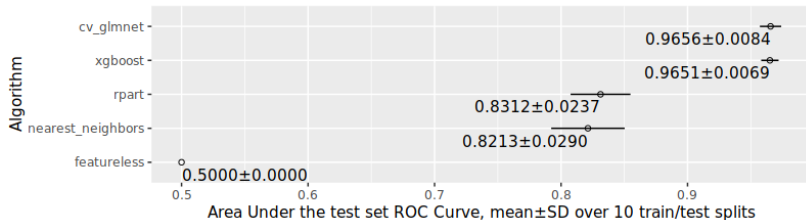


Relatively small FPR because there are so few positive labels (Autism=Yes only 3% of 27808 rows in 2020).

Area Under ROC Curve (AUC) quantifies accuracy over all thresholds

- ▶ Different learning algorithms result in different FPR/TPR at default prediction threshold, which can make it difficult to fairly compare.
- ▶ For example, nearest neighbors always had lower FPR/TPR than other algorithms.
- ▶ Is there an algorithm which has a larger TPR, for a given FPR? If so, then it is objectively better.
- ▶ An algorithm with larger AUC means more often larger TPR, for a given FPR (averaged over all prediction thresholds).

Survey year 2020, all 364 features, AUC over 10 cross-validation folds



Data pre-processing

Prediction accuracy in a given year

Model interpretation / feature selection

Similarity/difference between years

Discussion and Conclusions

Column categorization

Each input/feature column was assigned a category.

column_name,category

survey_year,

Autism,

State FIPS Code=Alabama,state

...

Number of Children in Household=1,home

...

Sex of Selected Child=Male,birth

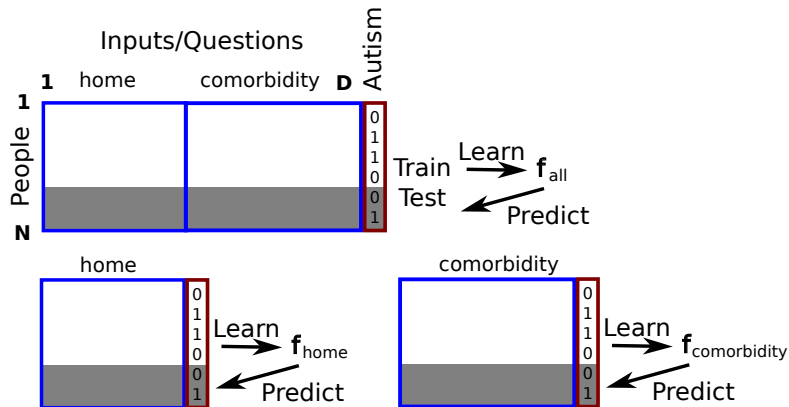
...

Deafness=Yes,comorbidity

...

	behavior	birth	comorbidity
2	15	24	30
culture	healthcare	home	state
14	88	130	50
wealth			
13			

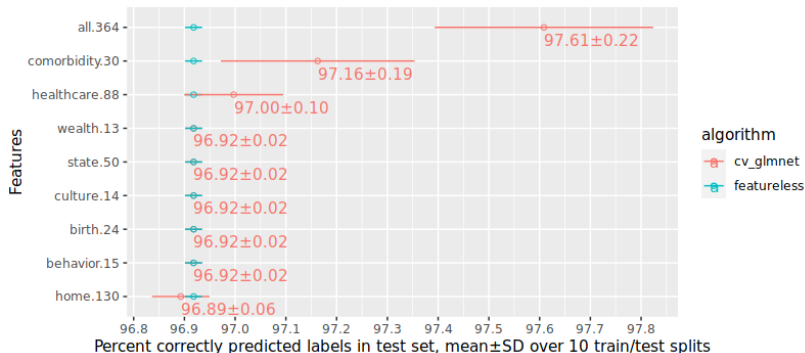
Cross-validation for category importance



- Do inputs/questions from home category, or comorbidity category, result in more accurate predictions on the test set?

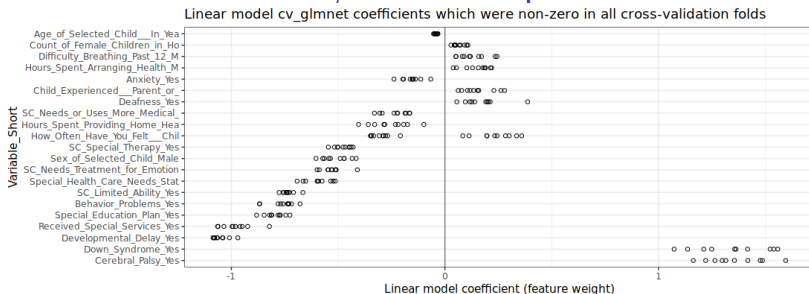
Cross-validation for category importance

Survey year 2020, train on feature subsets,
summary of 10 cross-validation folds



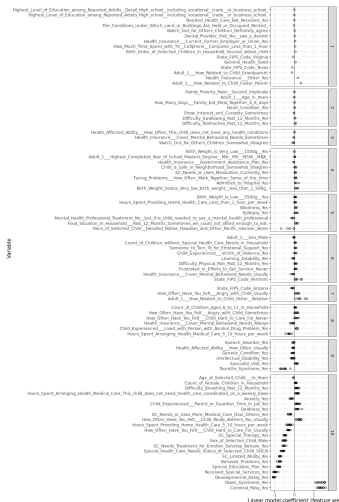
- ▶ None of the categories alone is as accurate as all of the features, which implies that some combination of categories is required for optimal prediction accuracy.
- ▶ Co-morbidity is the most accurate single category, and healthcare is second; other categories are not useful at all by themselves for prediction (same as featureless).

Linear model coefficient / feature importance



- ▶ Linear model is Likelihood of autism = $f(x) = \sum_{j=1}^D x_j \beta_j$ where x_j is input j and β_j is the learned weight/coefficient.
- ▶ For example, above likelihood is 1.4(Cerebral Palsy) + 1.3(Down Syndrome) - 1.1(Developmental Delay)+ ...
- ▶ Positive weight/coefficient β_j means that feature contributes to probability of autism=Yes, negative means autism=No.
- ▶ Above we show only most important features, with non-zero weights/coefficients in all 10 cross-validation folds (sorted by absolute mean weight/coefficient).

Full figure, variables selected in any number of CV folds



View full figure online, <https://github.com/tdhock/2024-01-ml-for-autism/blob/main/download-nsch-mlr3batchmark-registry-glmnet-coef.png>

[//github.com/tdhock/2024-01-ml-for-autism/blob/main/download-nsch-mlr3batchmark-registry-glmnet-coef.png](https://github.com/tdhock/2024-01-ml-for-autism/blob/main/download-nsch-mlr3batchmark-registry-glmnet-coef.png)

Data pre-processing

Prediction accuracy in a given year

Model interpretation / feature selection

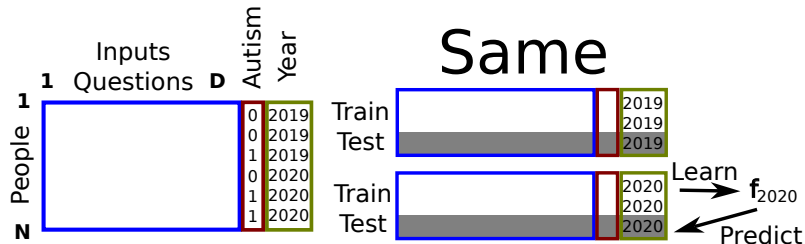
Similarity/difference between years

Discussion and Conclusions

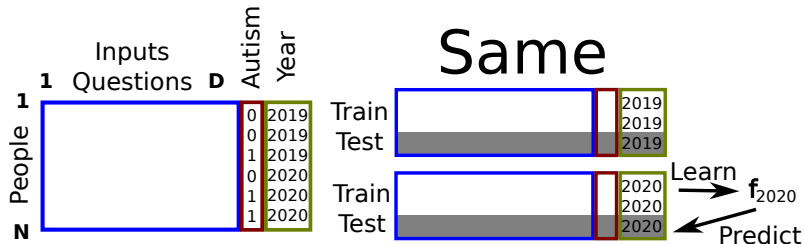
Cross-validation for determining similarity between years

		Inputs		Autism	Year
1	2	Questions	D		
People	1		0	0	2019
			0	0	2019
			1	1	2019
			0	0	2020
			1	1	2020
			1	1	2020
	2		1	1	2020

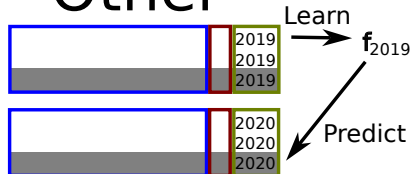
Cross-validation for determining similarity between years



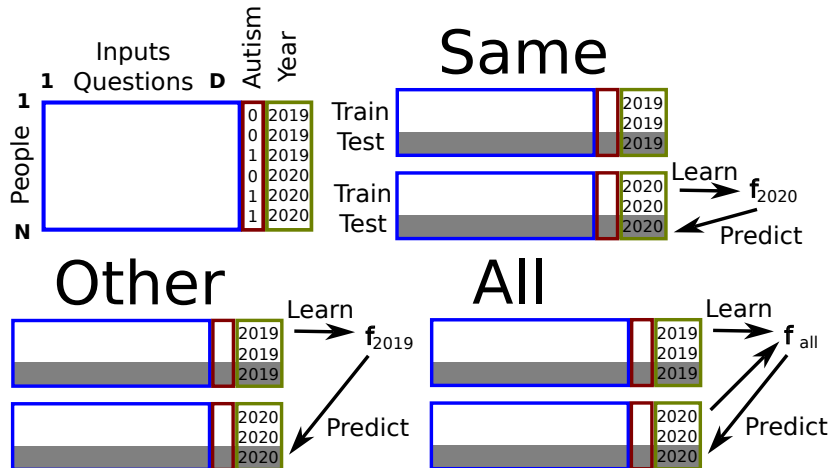
Cross-validation for determining similarity between years



Other

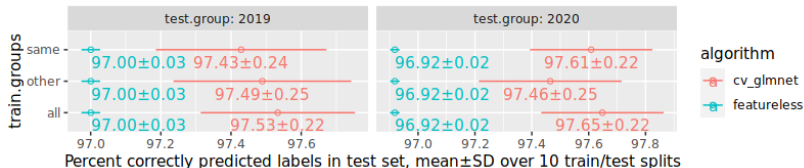


Cross-validation for determining similarity between years



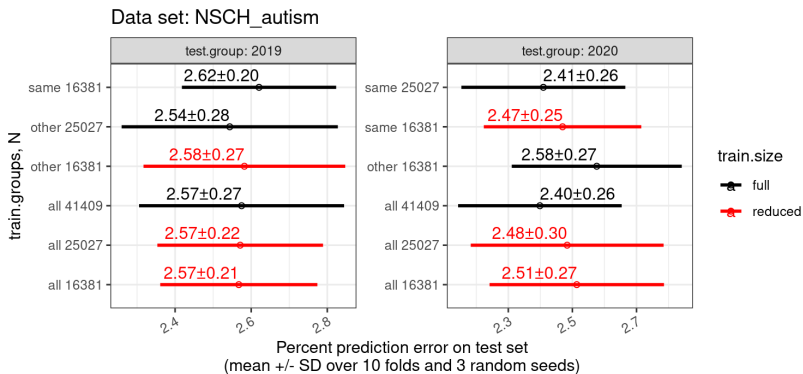
Cross-validation for determining similarity between years

Fix test group, train on same/other/all, summary of 10 cross-validation folds



- ▶ 18,202 rows in 2019, whereas 27,808 in 2020.
- ▶ For predicting in 2019 (left), training on only 2019 (same) is slightly less accurate than training on only 2020 (other), and 2019+2020 (all). This suggests 2020 data are consistent with the pattern in 2019, which is too complex to learn from the limited 2019 data alone (there is a slight advantage to combining years when training).
- ▶ For predicting in 2020 (right), training on 2019 (other) is slightly less accurate than training on 2020 (same), and 2019+2020 (all). This again suggests that 2019/2020 data are consistent, but there are not enough data in 2019 alone.

Cross-validation for determining similarity between years



► TODO

Data pre-processing

Prediction accuracy in a given year

Model interpretation / feature selection

Similarity/difference between years

Discussion and Conclusions

Discussion and Conclusions

- ▶ Cross-validation can be used to determine which learning algorithms, and features, are most accurate.
- ▶ Machine learning algorithms like L1 regularized linear models (LASSO/cv_glmnet) are additionally interpretable in terms of which features are used for prediction.
- ▶ Free/open-source software available: mlr3resampling R package on CRAN and <https://github.com/tdhock/mlr3resampling>, cross-validation for train on one year, predict on another.
- ▶ These slides are reproducible, using the code in <https://github.com/tdhock/2024-01-ml-for-autism>
- ▶ Contact: toby.hocking@nau.edu, toby.hocking@r-project.org