Super Learning (SL) and `sl3`

Rachael Phillips
rachaelvphillips@berkeley.edu

JSM 2023

Berkeley
UNIVERSITY OF CALIFORNIA

Overview of Super Learner

Super Learner (SL)

LIBRARY

Linear model

BART

Random Forest

Neural Network

Lasso

HAL

Regression splines

Super Learner (SL)

LIBRARY	COMPETITION

Cross-validated
performance of
learners + ensembles

Linear model

BART

Random Forest

Neural
Network

Lasso

HAL

Regression splines

Learning
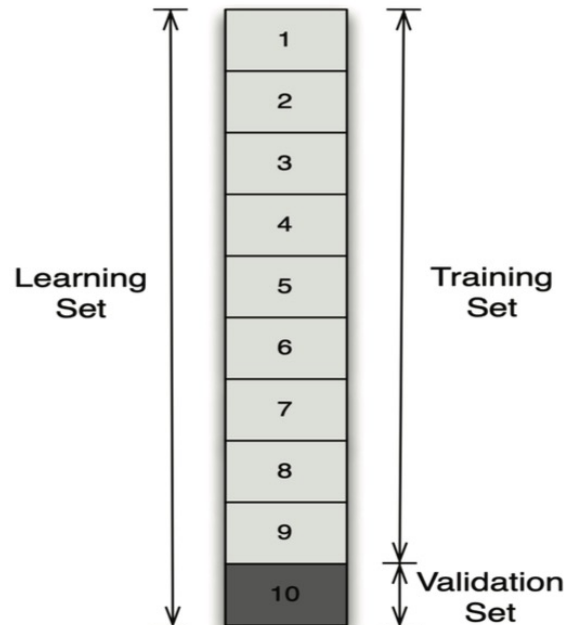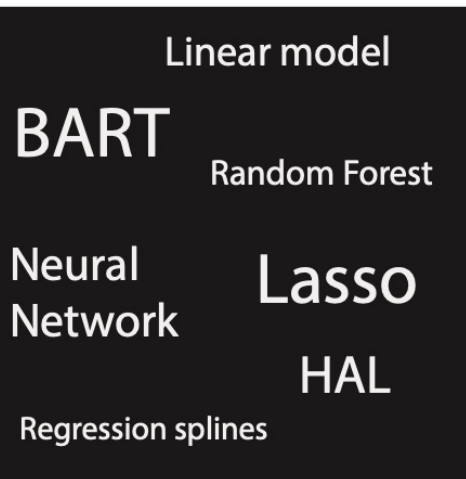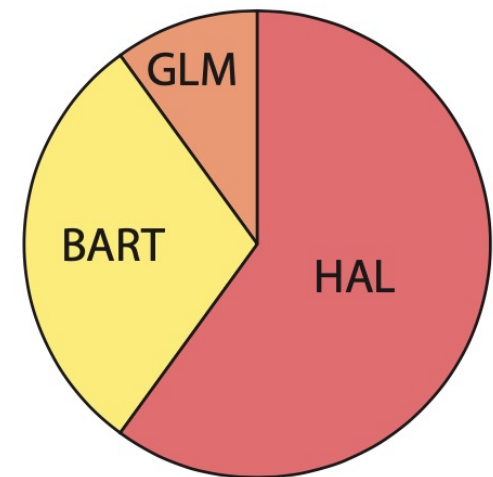Set

Training
Set

1
2
3
4
5
6
7
8
9
10

Validation
Set

Super Learner (SL)

LIBRARY   COMPETITION   WINNER

Cross-validated
performance of
learners + ensembles

Linear model
BART
Random Forest
Neural
Network   Lasso
HAL
Regression splines

Learning
Set

Training
Set

1
2
3
4
5
6
7
8
9
10

Validation
Set

GLM
BART
HAL
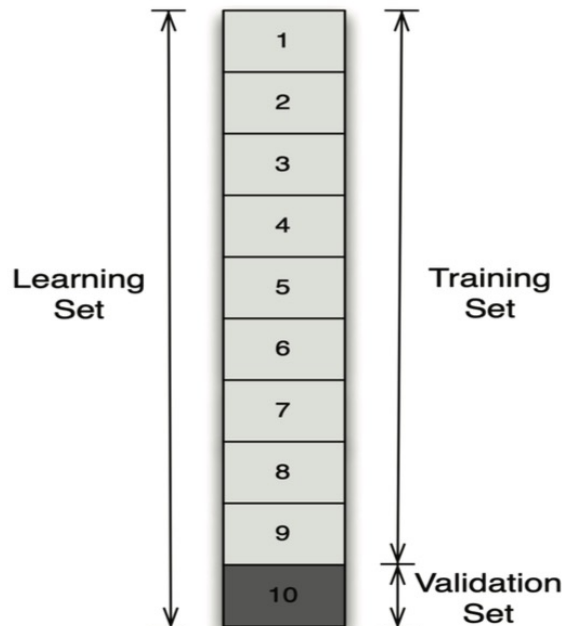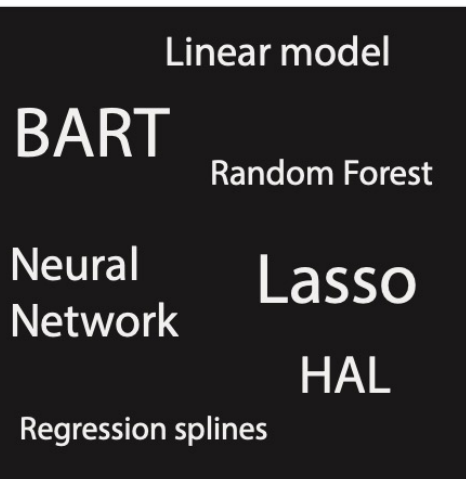
Super Learner (SL)

- Uses a library of algorithms for estimating a prediction function

  - Analyst specifies Alg_1, … Alg_K

  - Create an optimal combination

    - Optimal with respect to V-fold cross-validated (CV) risk

    - Example risk functions: Negative log likelihood, mean squared error, 1-AUC

- SL predicted values, $\hat{Y}_{SL}$, are a combination of $\hat{Y}_{Alg_1}$, …, $\hat{Y}_{Alg_K}$

  - Discrete SL: "winner-take-all", predictions from algorithm with best CV risk

  - Ensemble SL: predictions from multiple algorithms are combined

    - weighted combination

    - some other, possibly complex function of the algorithms' predictions

Berkeley
UNIVERSITY OF CALIFORNIA

Why super learner (SL)?

- No need to select the one "right" strategy
- Can consider diverse set
- Grounded in statistical optimality theory
- Pre–specified also flexible
- Mitigate statistical model misspecification
- Good initial estimation for estimators common for causal inference (e.g., AIPW, TMLE)

What does user specify for SL?

a) Measure of performance
b) Cross-validation scheme
c) Diverse library of candidate learners

Super learner

1. Specify
   a) Measure of performance
   b) Cross-validation scheme
   c) Diverse library of candidate learners

What are you learning from the data? What do you want to optimize for?

Performance measure should be valid (i.e., optimized by underlying target), bounded, corresponds to desired goal

Super learner

1. Specify
   a) Measure of performance
   b) Cross-validation scheme
   c) Diverse library of candidate learners

e.g. V-fold cross-validation

Super learner

1. Specify
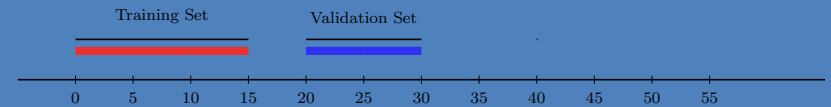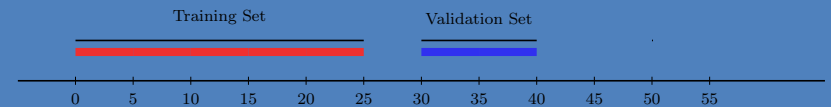   a) Measure of performance
   b) Cross-validation scheme
   c) Diverse library of candidate learners

e.g. rolling origin cross-validation
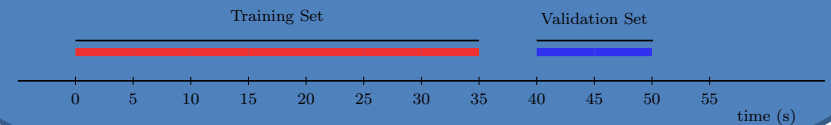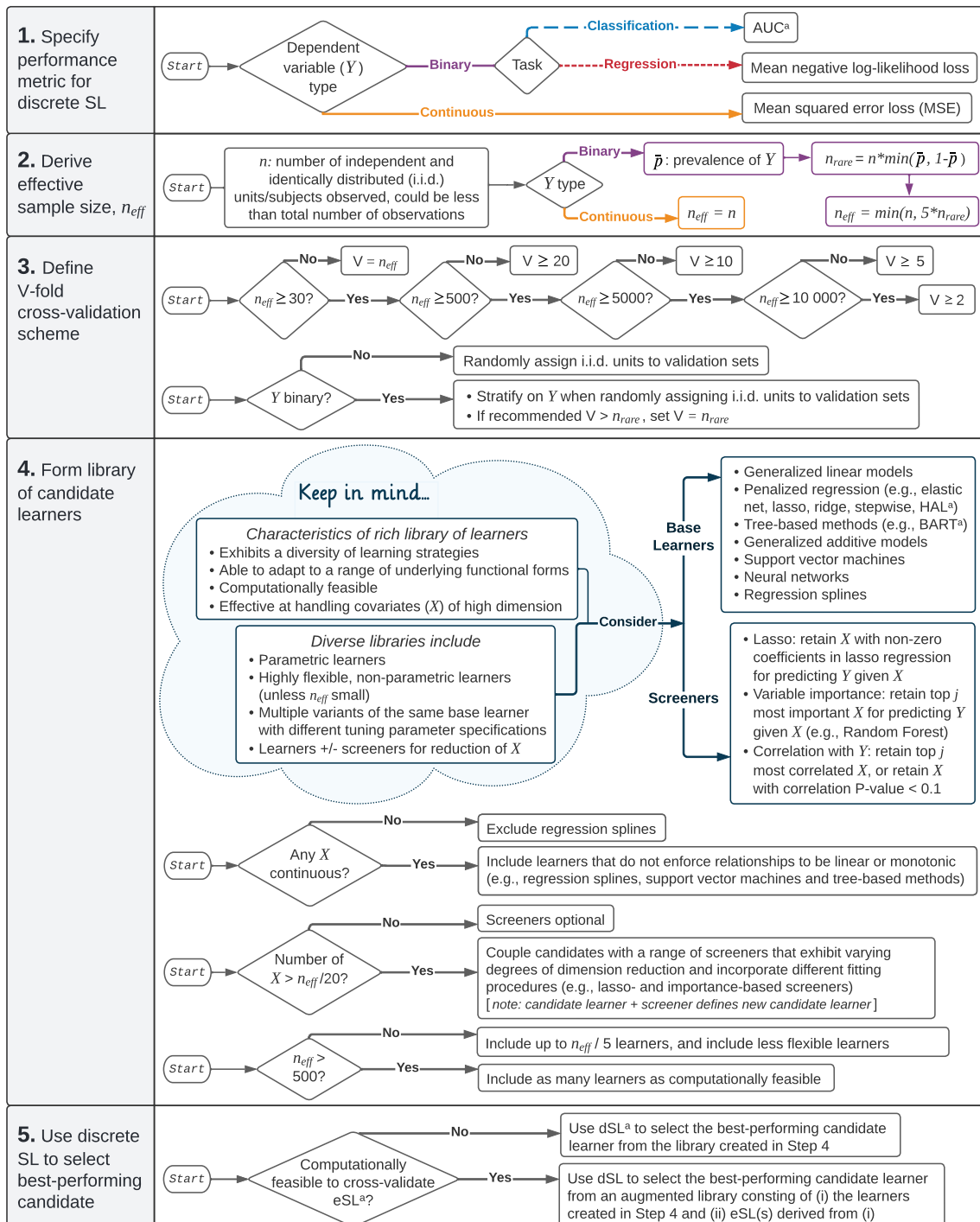(cross-validation for time series data)

Overview of the algorithm

1. Make metalevel dataset with cross-validated candidate predictions and validation set outcomes
2. Fit meta-learner to the metalevel dataset
3. Full-fit candidates
4. Define the SL

Practical considerations for specifying SL

Rachael V Phillips, Mark J van der Laan, Hana Lee, Susan Gruber, Practical considerations for specifying a super learner, *International Journal of Epidemiology*, 2023.
https://doi.org/10.1093/ije/dyad023

1. Specify performance metric for discrete SL

Start → Dependent variable (Y) type —Binary— Task
- —Classification— → AUC[a]
- —Regression— → Mean negative log-likelihood loss
- —Continuous— → Mean squared error loss (MSE)

2. Derive effective sample size, n_{eff}

Start → n: number of independent and identically distributed (i.i.d.) units/subjects observed, could be less than total number of observations → Y type
- —Binary— → $\bar{p}$: prevalence of Y → $n_{rare} = n*min(\bar{p}, 1-\bar{p})$ → $n_{eff} = min(n, 5*n_{rare})$
- —Continuous— → $n_{eff} = n$

3. Define V-fold cross-validation scheme

Start → $n_{eff} \geq 30$?
- —No— → V = n_{eff}
- —Yes— → $n_{eff} \geq 500$?
  - —No— → V ≥ 20
  - —Yes— → $n_{eff} \geq 5000$?
    - —No— → V ≥ 10
    - —Yes— → $n_{eff} \geq 10\ 000$?
      - —No— → V ≥ 5
      - —Yes— → V ≥ 2

Start → Y binary?
- —No— → Randomly assign i.i.d. units to validation sets
- —Yes— →
  - Stratify on Y when randomly assigning i.i.d. units to validation sets
  - If recommended V > n_{rare}, set V = n_{rare}

4. Form library of candidate learners

Keep in mind...

Characteristics of rich library of learners
- Exhibits a diversity of learning strategies
- Able to adapt to a range of underlying functional forms
- Computationally feasible
- Effective at handling covariates (X) of high dimension

Diverse libraries include
- Parametric learners
- Highly flexible, non-parametric learners (unless n_{eff} small)
- Multiple variants of the same base learner with different tuning parameter specifications
- Learners +/- screeners for reduction of X

Consider →

Base Learners
- Generalized linear models
- Penalized regression (e.g., elastic net, lasso, ridge, stepwise, HAL[a])
- Tree-based methods (e.g., BART[a])
- Generalized additive models
- Support vector machines
- Neural networks
- Regression splines

Screeners
- Lasso: retain X with non-zero coefficients in lasso regression for predicting Y given X
- Variable importance: retain top j most important X for predicting Y given X (e.g., Random Forest)
- Correlation with Y: retain top j most correlated X, or retain X with correlation P-value < 0.1

Start → Any X continuous?
- —No— → Exclude regression splines
- —Yes— → Include learners that do not enforce relationships to be linear or monotonic (e.g., regression splines, support vector machines and tree-based methods)

Start → Number of $X > n_{eff}/20$?
- —No— → Screeners optional
- —Yes— → Couple candidates with a range of screeners that exhibit varying degrees of dimension reduction and incorporate different fitting procedures (e.g., lasso- and importance-based screeners) [*note: candidate learner + screener defines new candidate learner*]

Start → $n_{eff} > 500$?
- —No— → Include up to n_{eff} / 5 learners, and include less flexible learners
- —Yes— → Include as many learners as computationally feasible

5. Use discrete SL to select best-performing candidate

Start → Computationally feasible to cross-validate eSL[a]?
- —No— → Use dSL[a] to select the best-performing candidate learner from the library created in Step 4
- —Yes— → Use dSL to select the best-performing candidate learner from an augmented library consting of (i) the learners created in Step 4 and (ii) eSL(s) derived from (i)

Additional helpful, introductory articles for applying TL

1. Gruber S, Lee H, Phillips RV, Ho M, van der Laan M. **Developing a Targeted Learning-Based** Statistical Analysis Plan. *Statistics in Biopharmaceutical Research*. 2022. https://doi.org/10.1080/19466315.2022.2116104

2. Gruber S, Phillips RV, Lee H, Ho M, Concato J, van der Laan MJ. **Targeted Learning: Toward a Future Informed by** Real-World Evidence. *Statistics in Biopharmaceutical Research*. 2023. https://doi.org/10.1080/19466315.2023.2182356

3. Gruber S, Phillips RV, Lee H, Concato J, van der Laan M. **Evaluating and** Improving Real-World Evidence **with Targeted Learning**. *BMC Medical Research Methodology*. 2023. https://doi.org/10.1186/s12874-023-01998-2

Berkeley
UNIVERSITY OF CALIFORNIA

sl3
SL software package in tlverse

Introductory overview of `sl3`

- Task
- Learners
- Functions

Introductory overview of `sl3`

- **Task**
- Learners
- Functions

Tasks in sl3

What is the prediction task?

data, covariates, outcome,

weights, id, outcome_type,offset, drop_missing_outcome, folds

https://tlverse.org/sl3/reference/sl3_Task.html

Introductory overview of `sl3`

- Task
- **Learners**
- Functions

Berkeley
UNIVERSITY OF CALIFORNIA

Exercise: Training learner with `sl3` wrapper versus package's function

- Prediction tasks in `sl3`
- Comparison of R6 methods and S3 methods
  - `sl3` R package function, `Lrnr_earth`
  - `earth` R package function, `earth`
- Looking up learners, the `sl3` wrapper, and arguments
- Modifying learner parameters

- **Data example**: Collaborative Perinatal Project (CPP) was a multisite prospective cohort study designed to identify the effects of complications during either pregnancy or the perinatal period on birth and child outcomes.

Learners in sl3

How to estimate prediction function?

Learners in sl3

How to estimate prediction function?

Other Learners: Lrnr_HarmonicReg, Lrnr_arima, Lrnr_bartMachine, Lrnr_base, Lrnr_bayesglm, Lrnr_bilstm, Lrnr_caret, Lrnr_cv_selector, Lrnr_cv, Lrnr_dbarts, Lrnr_define_interactions, Lrnr_density_discretize, Lrnr_density_hse, Lrnr_density_semiparametric, Lrnr_earth, Lrnr_expSmooth, Lrnr_gam, Lrnr_ga, Lrnr_gbm, Lrnr_glm_fast, Lrnr_glmnet, Lrnr_glm, Lrnr_grf, Lrnr_gru_keras, Lrnr_gts, Lrnr_h2o_grid, Lrnr_hal9001, Lrnr_haldensify, Lrnr_hts, Lrnr_independent_binomial, Lrnr_lightgbm, Lrnr_lstm_keras, Lrnr_mean, Lrnr_multiple_ts, Lrnr_multivariate, Lrnr_nnet, Lrnr_nnls, Lrnr_optim, Lrnr_pca, Lrnr_pkg_SuperLearner, Lrnr_polspline, Lrnr_pooled_hazards, Lrnr_randomForest, Lrnr_ranger, Lrnr_revere_task, Lrnr_rpart, Lrnr_rugarch, Lrnr_screener_augment, Lrnr_screener_coefs, Lrnr_screener_correlation, Lrnr_screener_importance, Lrnr_sl, Lrnr_solnp_density, Lrnr_solnp, Lrnr_stratified, Lrnr_subset_covariates, Lrnr_svm, Lrnr_tsDyn, Lrnr_ts_weights, Lrnr_xgboost, Pipeline, Stack, define_h2o_X(),

https://tlverse.org/sl3/reference/index.html#sl-learners

Learners in sl3

How to estimate prediction function?

Other Learners: `Lrnr_HarmonicReg`, `Lrnr_arima`, `Lrnr_bartMachine`, `Lrnr_base`, `Lrnr_bayesglm`, `Lrnr_bilstm`, `Lrnr_caret`, `Lrnr_cv_selector`, `Lrnr_cv`, `Lrnr_dbarts`, `Lrnr_define_interactions`, `Lrnr_density_discretize`, `Lrnr_density_hse`, `Lrnr_density_semiparametric`, `Lrnr_earth`, `Lrnr_expSmooth`, `Lrnr_gam`, `Lrnr_ga`, `Lrnr_gbm`, `Lrnr_glm_fast`, `Lrnr_glmnet`, `Lrnr_glm`, `Lrnr_grf`, `Lrnr_gru_keras`, `Lrnr_gts`, `Lrnr_h2o_grid`, `Lrnr_hal9001`, `Lrnr_haldensify`, `Lrnr_hts`, `Lrnr_independent_binomial`, `Lrnr_lightgbm`, `Lrnr_lstm_keras`, `Lrnr_mean`, `Lrnr_multiple_ts`, `Lrnr_multivariate`, `Lrnr_nnet`, `Lrnr_nnls`, `Lrnr_optim`, `Lrnr_pca`, `Lrnr_pkg_SuperLearner`, `Lrnr_polspline`, `Lrnr_pooled_hazards`, `Lrnr_randomForest`, `Lrnr_ranger`, `Lrnr_revere_task`, `Lrnr_rpart`, `Lrnr_rugarch`, `Lrnr_screener_augment`, `Lrnr_screener_coefs`, `Lrnr_screener_correlation`, `Lrnr_screener_importance`, `Lrnr_sl`, `Lrnr_solnp_density`, `Lrnr_solnp`, `Lrnr_stratified`, `Lrnr_subset_covariates`, `Lrnr_svm`, `Lrnr_tsDyn`, `Lrnr_ts_weights`, `Lrnr_xgboost`, `Pipeline`, `Stack`, `define_h2o_X()`,

Berkeley
UNIVERSITY OF CALIFORNIA

https://tlverse.org/sl3/reference/index.html#sl-learners

Learners in sl3

How to estimate prediction function?

Other Learners: `Lrnr_HarmonicReg`, `Lrnr_arima`, `Lrnr_bartMachine`, `Lrnr_base`, `Lrnr_bayesglm`, `Lrnr_bilstm`, `Lrnr_caret`, `Lrnr_cv_selector`, `Lrnr_cv`, `Lrnr_dbarts`, `Lrnr_define_interactions`, `Lrnr_density_discretize`, `Lrnr_density_hse`, `Lrnr_density_semiparametric`, `Lrnr_earth`, `Lrnr_expSmooth`, `Lrnr_gam`, `Lrnr_ga`, `Lrnr_gbm`, `Lrnr_glm_fast`, `Lrnr_glmnet`, `Lrnr_glm`, `Lrnr_grf`, `Lrnr_gru_keras`, `Lrnr_gts`, `Lrnr_h2o_grid`, `Lrnr_hal9001`, `Lrnr_haldensify`, `Lrnr_hts`, `Lrnr_independent_binomial`, `Lrnr_lightgbm`, `Lrnr_lstm_keras`, `Lrnr_mean`, `Lrnr_multiple_ts`, `Lrnr_multivariate`, `Lrnr_nnet`, `Lrnr_nnls`, `Lrnr_optim`, `Lrnr_pca`, `Lrnr_pkg_SuperLearner`, `Lrnr_polspline`, `Lrnr_pooled_hazards`, `Lrnr_randomForest`, `Lrnr_ranger`, `Lrnr_revere_task`, `Lrnr_rpart`, `Lrnr_rugarch`, `Lrnr_screener_augment`, `Lrnr_screener_coefs`, `Lrnr_screener_correlation`, `Lrnr_screener_importance`, `Lrnr_sl`, `Lrnr_solnp_density`, `Lrnr_solnp`, `Lrnr_stratified`, `Lrnr_subset_covariates`, `Lrnr_svm`, `Lrnr_tsDyn`, `Lrnr_ts_weights`, `Lrnr_xgboost`, `Pipeline`, `Stack`, `define_h2o_X()`,

https://tlverse.org/sl3/reference/index.html#sl-learners

Learners in sl3

How to estimate prediction function?

Other Learners: Lrnr_HarmonicReg, Lrnr_arima, Lrnr_bartMachine, Lrnr_base, Lrnr_bayesglm, Lrnr_bilstm, Lrnr_caret, Lrnr_cv_selector, Lrnr_cv, Lrnr_dbarts, Lrnr_define_interactions, Lrnr_density_discretize, Lrnr_density_hse, Lrnr_density_semiparametric, Lrnr_earth, Lrnr_expSmooth, Lrnr_gam, Lrnr_ga, Lrnr_gbm, Lrnr_glm_fast, Lrnr_glmnet, Lrnr_glm, Lrnr_grf, Lrnr_gru_keras, Lrnr_gts, Lrnr_h2o_grid, Lrnr_hal9001, Lrnr_haldensify, Lrnr_hts, Lrnr_independent_binomial, Lrnr_lightgbm, Lrnr_lstm_keras, Lrnr_mean, Lrnr_multiple_ts, Lrnr_multivariate, Lrnr_nnet, Lrnr_nnls, Lrnr_optim, Lrnr_pca, Lrnr_pkg_SuperLearner, Lrnr_polspline, Lrnr_pooled_hazards, Lrnr_randomForest, Lrnr_ranger, Lrnr_revere_task, Lrnr_rpart, Lrnr_rugarch, Lrnr_screener_augment, Lrnr_screener_coefs, Lrnr_screener_correlation, Lrnr_screener_importance, Lrnr_sl, Lrnr_solnp_density, Lrnr_solnp, Lrnr_stratified, Lrnr_subset_covariates, Lrnr_svm, Lrnr_tsDyn, Lrnr_ts_weights, Lrnr_xgboost, Pipeline, Stack, define_h2o_X(),

https://tlverse.org/sl3/reference/index.html#sl-learners

Learners in sl3

How to estimate prediction function?

Other Learners: Lrnr_HarmonicReg, Lrnr_arima, Lrnr_bartMachine, Lrnr_base, Lrnr_bayesglm, Lrnr_bilstm, Lrnr_caret, Lrnr_cv_selector, Lrnr_cv, Lrnr_dbarts, Lrnr_define_interactions, Lrnr_density_discretize, Lrnr_density_hse, Lrnr_density_semiparametric, Lrnr_earth, Lrnr_expSmooth, Lrnr_gam, Lrnr_ga, Lrnr_gbm, Lrnr_glm_fast, Lrnr_glmnet, Lrnr_glm, Lrnr_grf, Lrnr_gru_keras, Lrnr_gts, Lrnr_h2o_grid, Lrnr_hal9001, Lrnr_haldensify, Lrnr_hts, Lrnr_independent_binomial, Lrnr_lightgbm, Lrnr_lstm_keras, Lrnr_mean, Lrnr_multiple_ts, Lrnr_multivariate, Lrnr_nnet, Lrnr_nnls, Lrnr_optim, Lrnr_pca, Lrnr_pkg_SuperLearner, Lrnr_polspline, Lrnr_pooled_hazards, Lrnr_randomForest, Lrnr_ranger, Lrnr_revere_task, Lrnr_rpart, Lrnr_rugarch, Lrnr_screener_augment, Lrnr_screener_coefs, Lrnr_screener_correlation, Lrnr_screener_importance, Lrnr_sl, Lrnr_solnp_density, Lrnr_solnp, Lrnr_stratified, Lrnr_subset_covariates, Lrnr_svm, Lrnr_tsDyn, Lrnr_ts_weights, Lrnr_xgboost, Pipeline, Stack, define_h2o_X(),

Berkeley
UNIVERSITY OF CALIFORNIA

Learners in sl3

How to estimate prediction function?

Other Learners: `Lrnr_HarmonicReg`, `Lrnr_arima`, `Lrnr_bartMachine`, `Lrnr_base`, `Lrnr_bayesglm`, `Lrnr_bilstm`, `Lrnr_caret`, `Lrnr_cv_selector`, `Lrnr_cv`, `Lrnr_dbarts`, `Lrnr_define_interactions`, `Lrnr_density_discretize`, `Lrnr_density_hse`, `Lrnr_density_semiparametric`, `Lrnr_earth`, `Lrnr_expSmooth`, `Lrnr_gam`, `Lrnr_ga`, `Lrnr_gbm`, `Lrnr_glm_fast`, `Lrnr_glmnet`, `Lrnr_glm`, `Lrnr_grf`, `Lrnr_gru_keras`, `Lrnr_gts`, `Lrnr_h2o_grid`, `Lrnr_hal9001`, `Lrnr_haldensify`, `Lrnr_hts`, `Lrnr_independent_binomial`, `Lrnr_lightgbm`, `Lrnr_lstm_keras`, `Lrnr_mean`, `Lrnr_multiple_ts`, `Lrnr_multivariate`, `Lrnr_nnet`, `Lrnr_nnls`, `Lrnr_optim`, `Lrnr_pca`, `Lrnr_pkg_SuperLearner`, `Lrnr_polspline`, `Lrnr_pooled_hazards`, `Lrnr_randomForest`, `Lrnr_ranger`, `Lrnr_revere_task`, `Lrnr_rpart`, `Lrnr_rugarch`, `Lrnr_screener_augment`, `Lrnr_screener_coefs`, `Lrnr_screener_correlation`, `Lrnr_screener_importance`, `Lrnr_sl`, `Lrnr_solnp_density`, `Lrnr_solnp`, `Lrnr_stratified`, `Lrnr_subset_covariates`, `Lrnr_svm`, `Lrnr_tsDyn`, `Lrnr_ts_weights`, `Lrnr_xgboost`, `Pipeline`, `Stack`, `define_h2o_X()`,

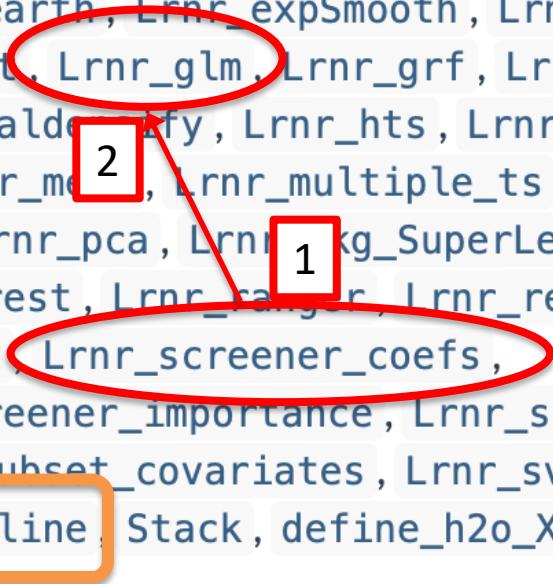https://tlverse.org/sl3/reference/index.html#sl-learners

Learners in sl3

How to estimate prediction function?

Other Learners: `Lrnr_HarmonicReg`, `Lrnr_arima`, `Lrnr_bartMachine`, `Lrnr_base`, `Lrnr_bayesglm`, `Lrnr_bilstm`, `Lrnr_caret`, `Lrnr_cv_selector`, `Lrnr_cv`, `Lrnr_dbarts`, `Lrnr_define_interactions`, `Lrnr_density_discretize`, `Lrnr_density_hse`, `Lrnr_density_semiparametric`, `Lrnr_earth`, `Lrnr_expSmooth`, `Lrnr_gam`, `Lrnr_ga`, `Lrnr_gbm`, `Lrnr_glm_fast`, `Lrnr_glmnet`, `Lrnr_glm`, `Lrnr_grf`, `Lrnr_gru_keras`, `Lrnr_gts`, `Lrnr_h2o_grid`, `Lrnr_hal9001`, `Lrnr_haldensify`, `Lrnr_hts`, `Lrnr_independent_binomial`, `Lrnr_lightgbm`, `Lrnr_lstm_keras`, `Lrnr_mean`, `Lrnr_multiple_ts`, `Lrnr_multivariate`, `Lrnr_nnet`, `Lrnr_nnls`, `Lrnr_optim`, `Lrnr_pca`, `Lrnr_pkg_SuperLearner`, `Lrnr_polspline`, `Lrnr_pooled_hazards`, `Lrnr_randomForest`, `Lrnr_ranger`, `Lrnr_revere_task`, `Lrnr_rpart`, `Lrnr_rugarch`, `Lrnr_screener_augment`, `Lrnr_screener_coefs`, `Lrnr_screener_correlation`, `Lrnr_screener_importance`, `Lrnr_sl`, `Lrnr_solnp_density`, `Lrnr_solnp`, `Lrnr_stratified`, `Lrnr_subset_covariates`, `Lrnr_svm`, `Lrnr_tsDyn`, `Lrnr_ts_weights`, `Lrnr_xgboost`, `Pipeline`, `Stack`, `define_h2o_X()`,

https://tlverse.org/sl3/reference/index.html#sl-learners

Learners in sl3

How to estimate prediction function?

Other Learners: Lrnr_HarmonicReg, Lrnr_arima, Lrnr_bartMachine, Lrnr_base, Lrnr_bayesglm, Lrnr_bilstm, Lrnr_caret, Lrnr_cv_selector, Lrnr_cv, Lrnr_dbarts, Lrnr_define_interactions, Lrnr_density_discretize, Lrnr_density_hse, Lrnr_density_semiparametric, Lrnr_earth, Lrnr_expSmooth, Lrnr_gam, Lrnr_ga, Lrnr_gbm, Lrnr_glm_fast, Lrnr_glmnet, Lrnr_glm, Lrnr_grf, Lrnr_gru_keras, Lrnr_gts, Lrnr_h2o_grid, Lrnr_hal9001, Lrnr_haldensify, Lrnr_hts, Lrnr_independent_binomial, Lrnr_lightgbm, Lrnr_lstm_keras, Lrnr_mean, Lrnr_multiple_ts, Lrnr_multivariate, Lrnr_nnet, Lrnr_nnls, Lrnr_optim, Lrnr_pca, Lrnr_pkg_SuperLearner, Lrnr_polspline, Lrnr_pooled_hazards, Lrnr_randomForest, Lrnr_ranger, Lrnr_revere_task, Lrnr_rpart, Lrnr_rugarch, Lrnr_screener_augment, Lrnr_screener_coefs, Lrnr_screener_correlation, Lrnr_screener_importance, Lrnr_sl, Lrnr_solnp_density, Lrnr_solnp, Lrnr_stratified, Lrnr_subset_covariates, Lrnr_svm, Lrnr_tsDyn, Lrnr_ts_weights, Lrnr_xgboost, Pipeline, Stack, define_h2o_X(),

2

1

Berkeley
UNIVERSITY OF CALIFORNIA

Introductory overview of `sl3`

- Task
- Learners
- **Other functions**

Other sl3 Functions

- Performance measures:
  - loss functions (e.g., squared error, negative log-likelihood, multinomial log-likelihood)
  - metrics based on `ROCR` software package, like AUC, AUCPR, accuracy, sensitivity, with `custom_ROCR_risk()`
- Variable importance with `importance`
- Table with each candidate learner's cross-validated predictive performance with `cv_risk`
- Cross-validated SL (to see CV risk of SL itself) with `cv_sl`

Berkeley
UNIVERSITY OF CALIFORNIA

Live coding exercise with sl3 using WASH Benefits data

WASH Benefits Bangladesh

- Study aiming to understand the **impact of water quality, sanitation, hand washing, and nutritional (WASH) interventions on child development in rural Bangladesh** (WASH Benefits Bangladesh): a cluster randomized controlled trial (Tofail et al. 2018)

- Enrolled pregnant women in first or second trimester from the rural villages of Bangladesh

WASH Benefits Bangladesh: Example Dataset for sl3 Prediction

- We concentrate on child growth (size for age) as the outcome of interest.

- We will use SL (sl3 software) to estimate the **prediction function** for predicting a child's weight-for-height z-score (whz) given covariate information.

- Next section on TMLE (tmle3 software) focuses on estimation of the (causal) impact of particular intervention on an outcome (e.g., the impact of the WASH intervention on whz)

Exercise: Training a super learner with `sl3`

https://tlverse.org/tlverse-handbook/sl3.html

http://tlverse.org/jsm2023-workshop/sl3.html