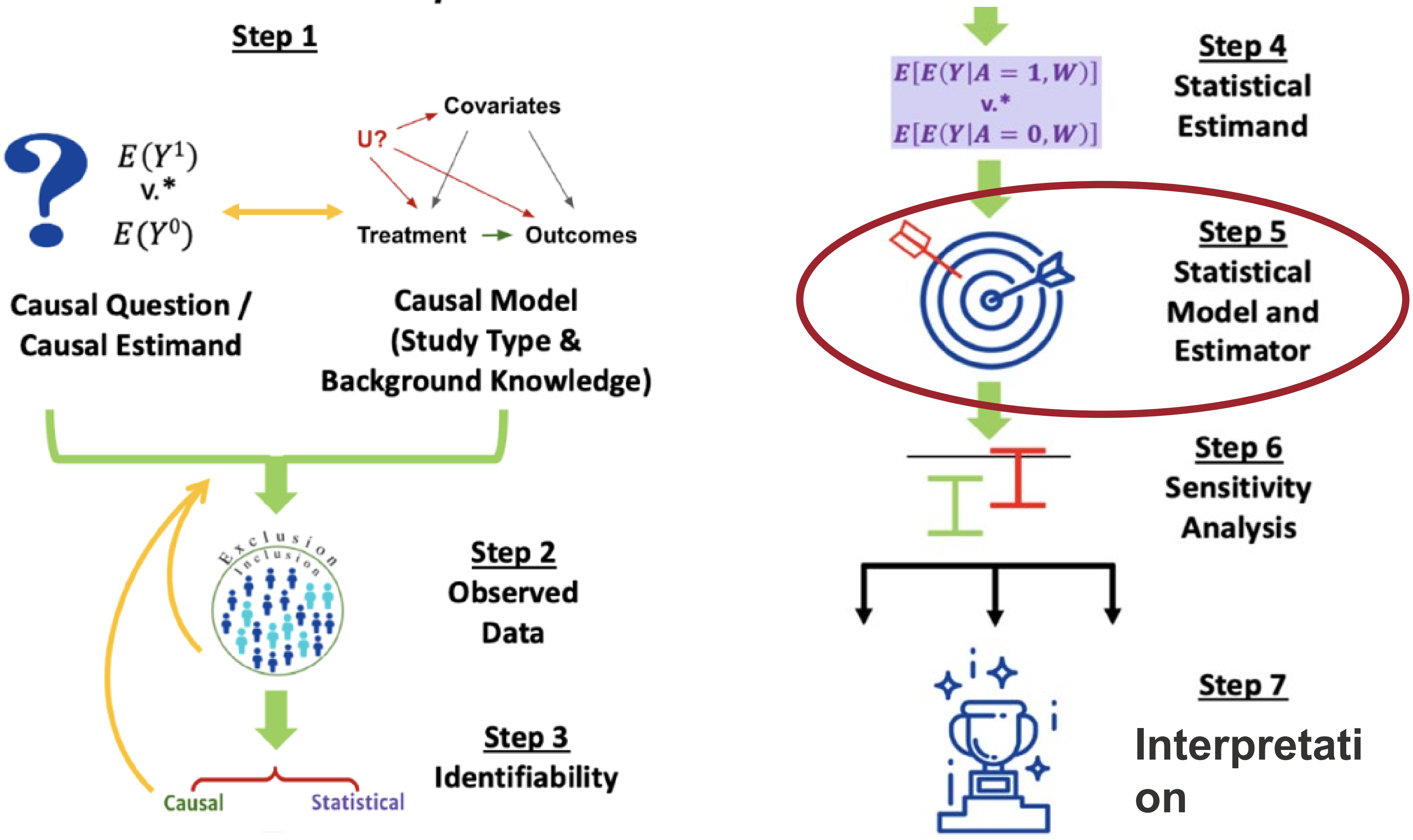
UNIVERSITY OF COPENHAGEN

Targeted register analyses
PhD short course

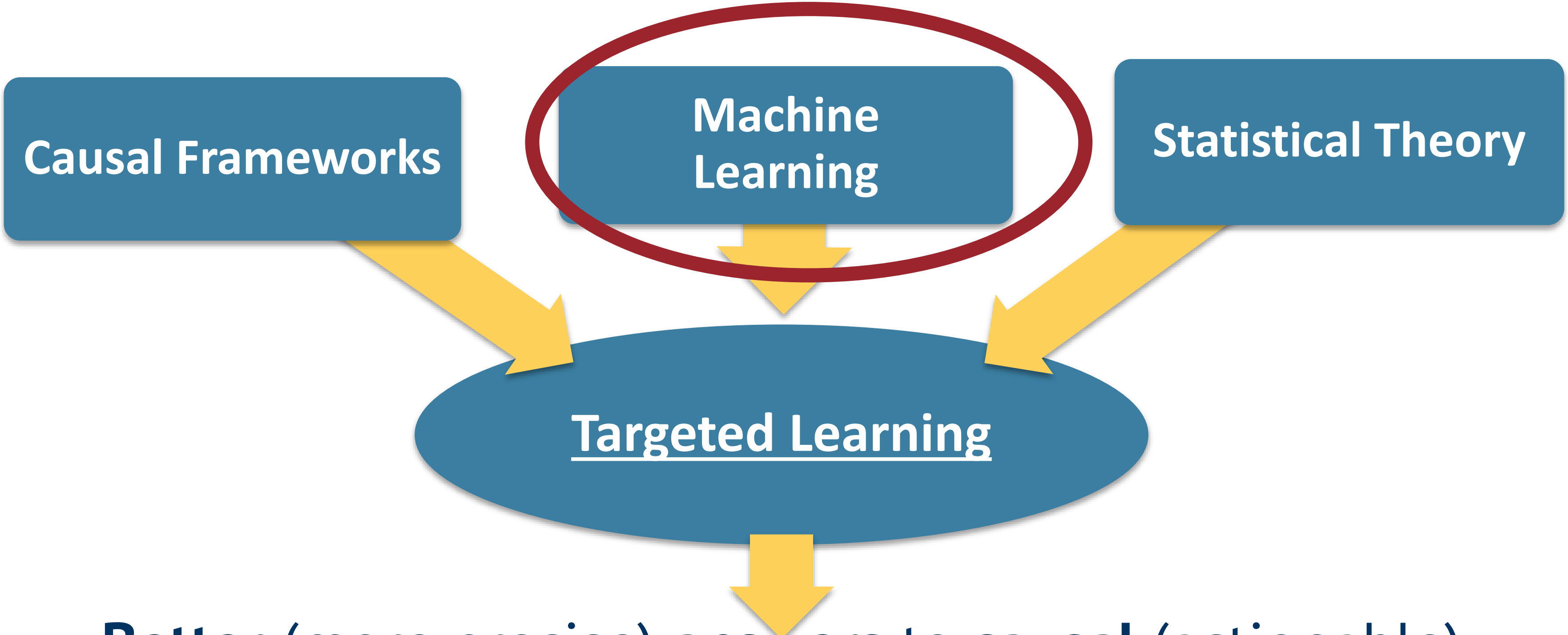
An Introduction to SuperLearner

Andrew Mertens
University of California, Berkeley, Division of Biostatistics

Figure 1: The *Causal Roadmap*



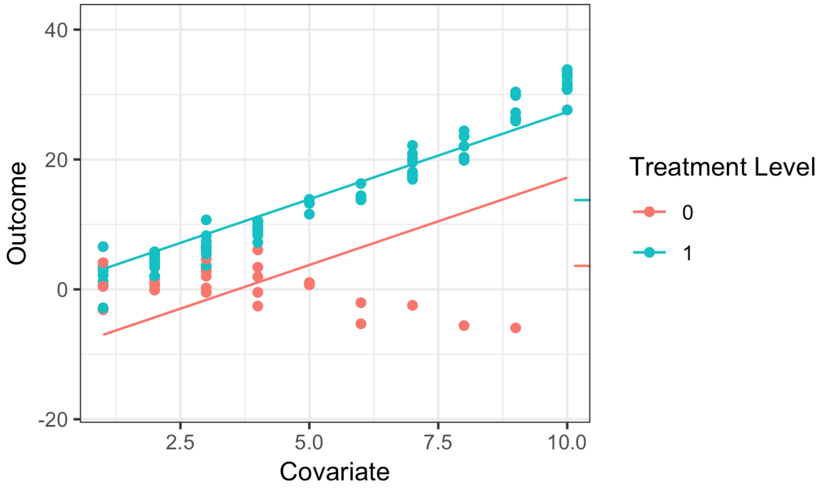
Dang et al. 2023



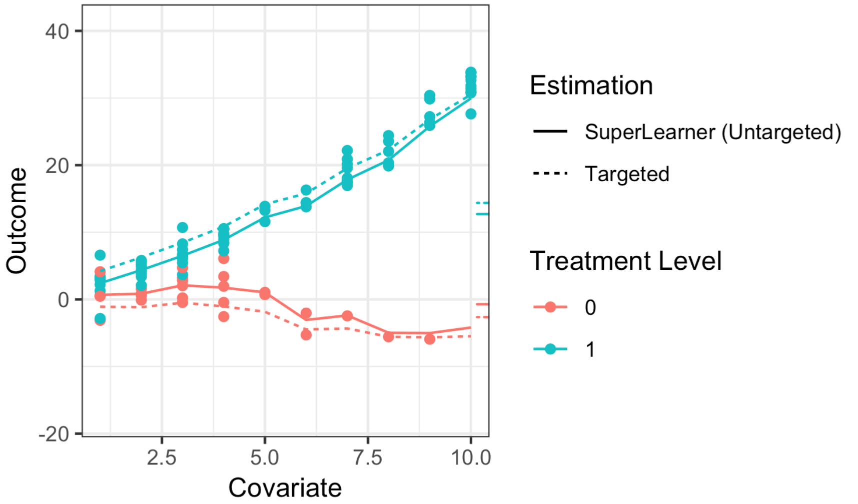
Better (more precise) **answers** to **causal** (actionable) **questions** with **accurate quantification of uncertainty** (signal from noise)

Targeted Learning Schematic

Outcome Mechanism Estimated using Linear Regression



Outcome Mechanism Estimated using SuperLearner and TMLE



Superlearner: the algorithm steps

1. Pick the learners
2. Split data into cross-validation folds
3. Fit learners on cross-validation folds
4. Obtain predictions from each fitted model within folds
5. Use a metalearner to combine predictions across learners
6. Repeat for the full dataset
7. Predict on new data

Step 1: pick the learners

Superlearner is a type of "ensemble machine learning"

- The word "ensemble" from the arts. It means a collection of musicians or performers.

Step 1: pick the learners

Superlearner is a type of “ensemble machine learning”

- Ensemble strengths:

Step 1: pick the learners

Superlearner is a type of "ensemble machine learning"

- Ensemble strengths:
  - Diversity

Step 1: pick the learners

Superlearner is a type of "ensemble machine learning"

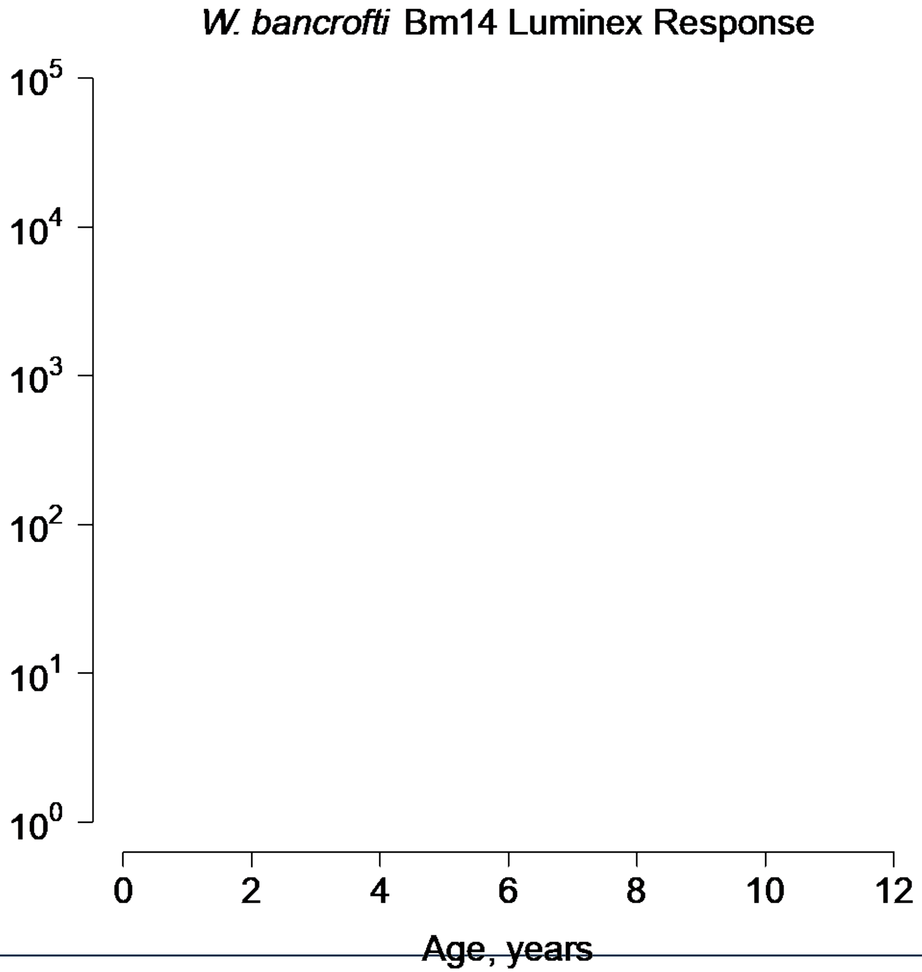
- Ensemble strengths:
  - Diversity
  - Redundancy

Step 1: pick the learners

Superlearner is a type of "ensemble machine learning"

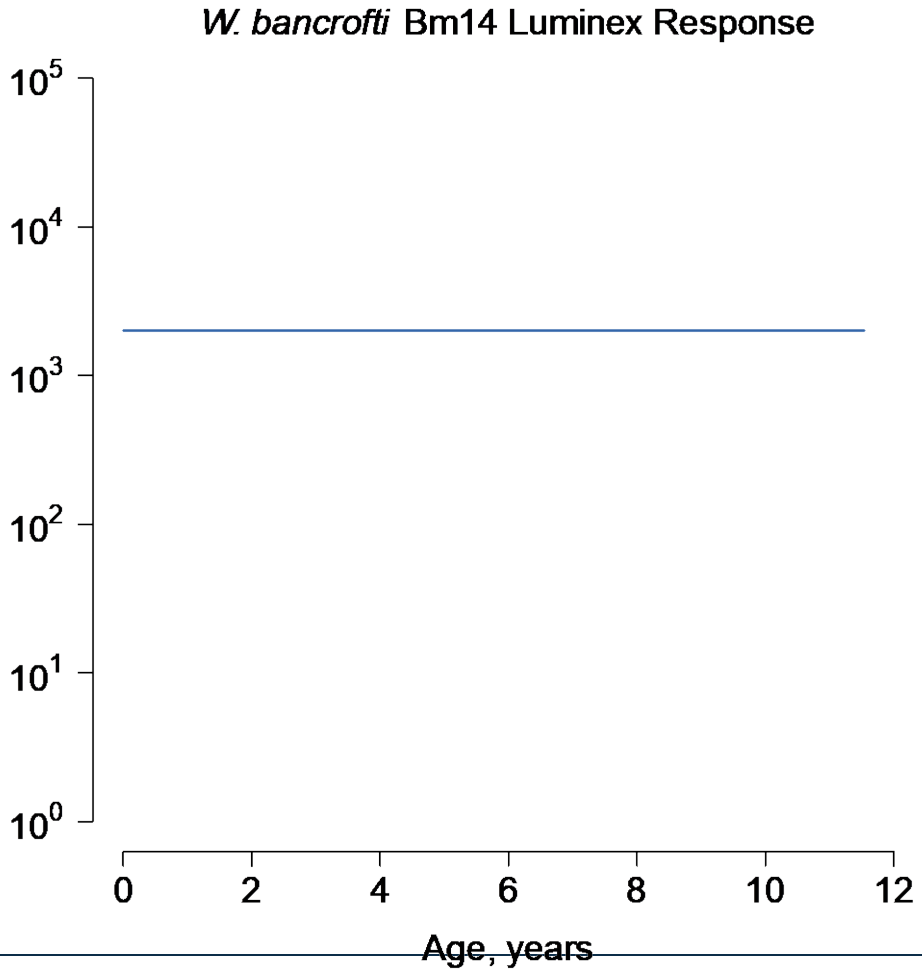
- Ensemble strengths:
  - Diversity
  - Redundancy
  - Synergy

Model / algorithm	CV MSE

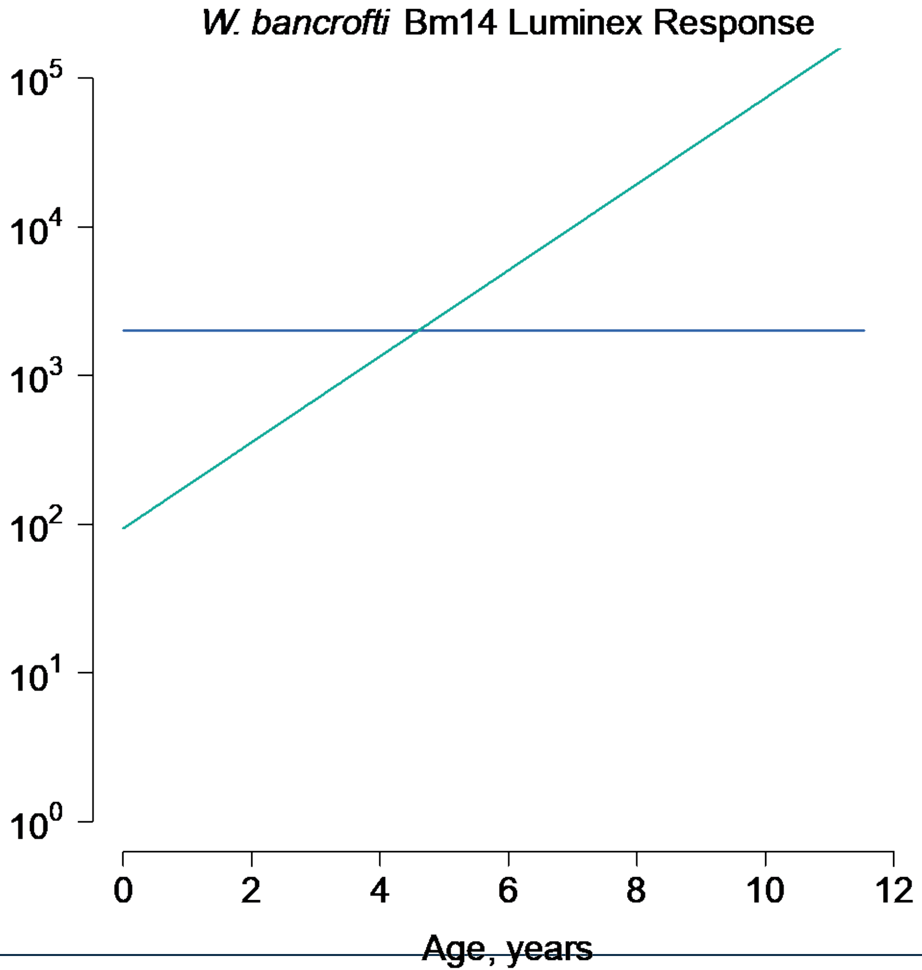


Model / algorithm	CV MSE
Mean	1.67

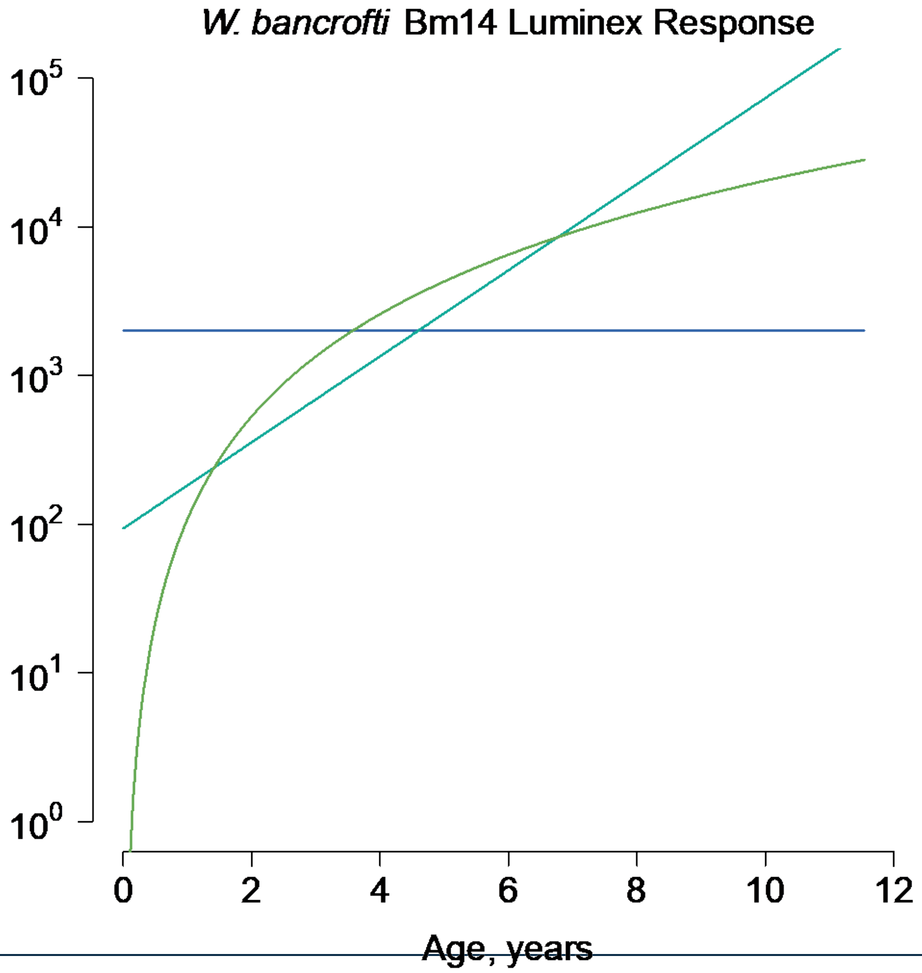
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

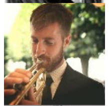
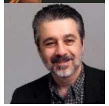


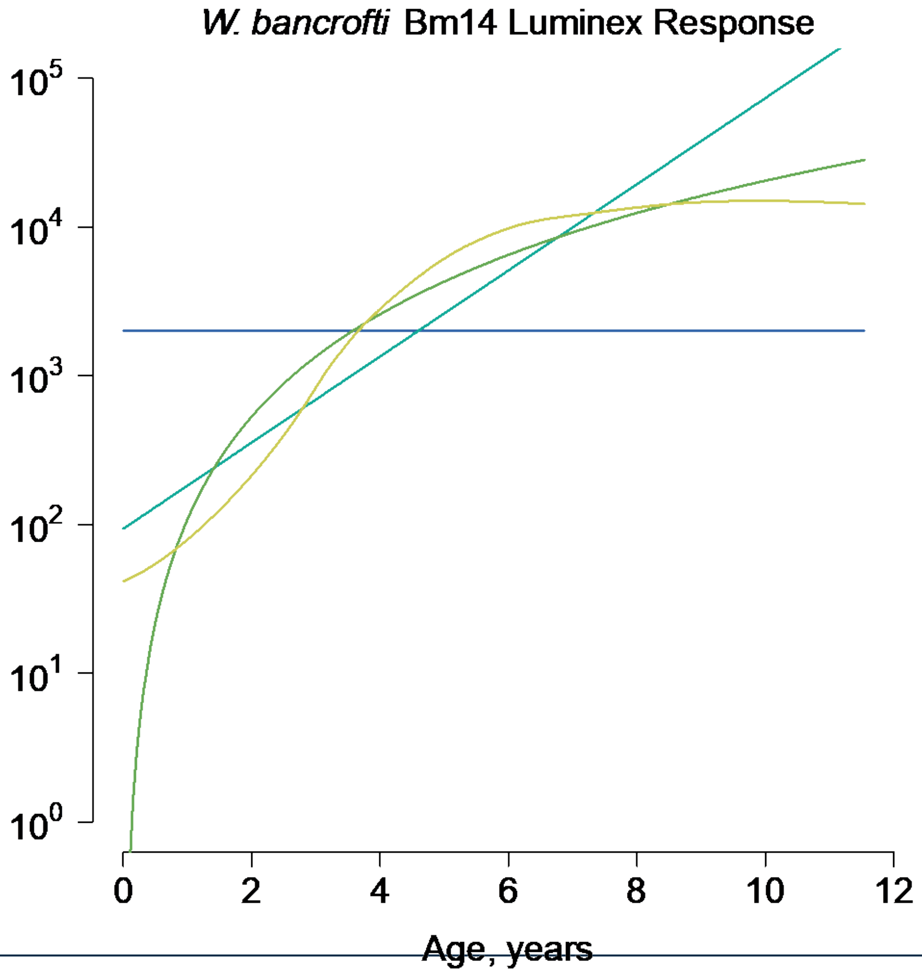
	Model / algorithm	CV MSE
	Mean	1.67
	**Linear regression**	**1.13**

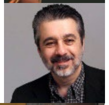


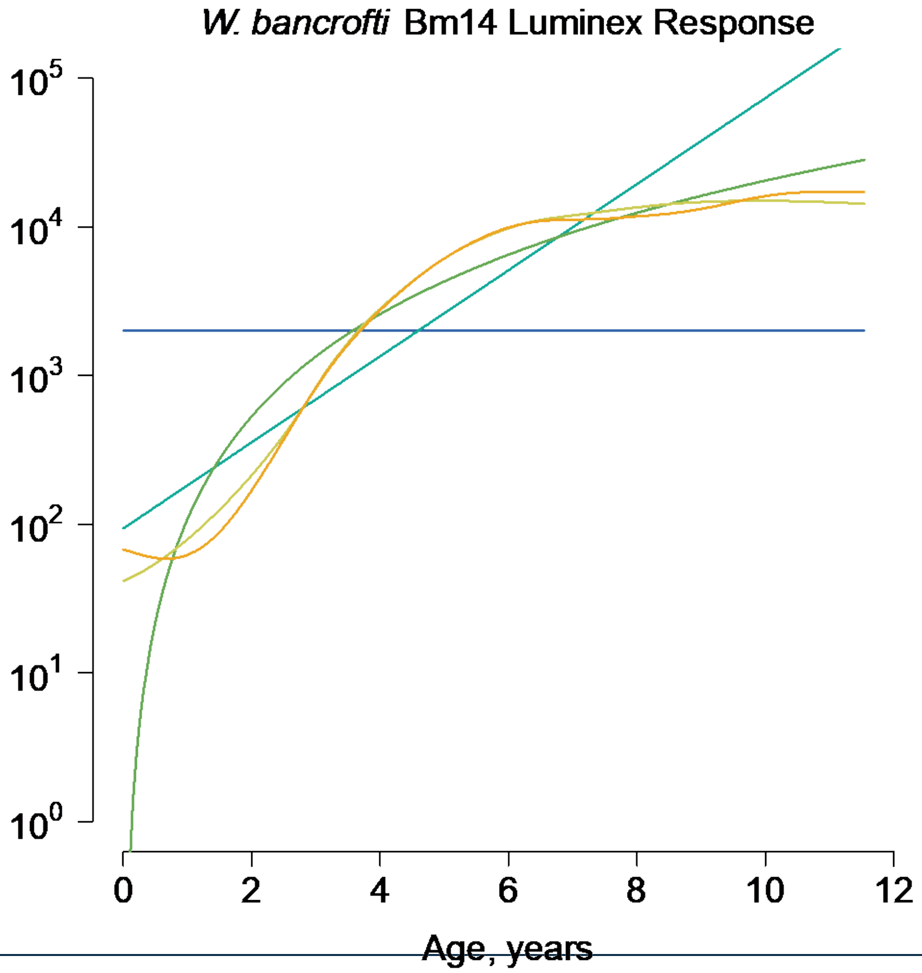
	Model / algorithm	CV MSE
	Mean	1.67
	Linear regression	1.13
	**Ab acquisition model**	**1.24**

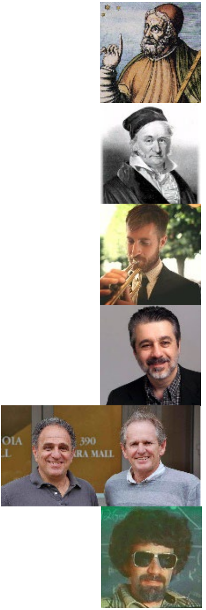


	Model / algorithm	CV MSE
	Mean	1.67
	Linear regression	1.13
	Ab acquisition model	1.24
	**LOESS**	**1.03**

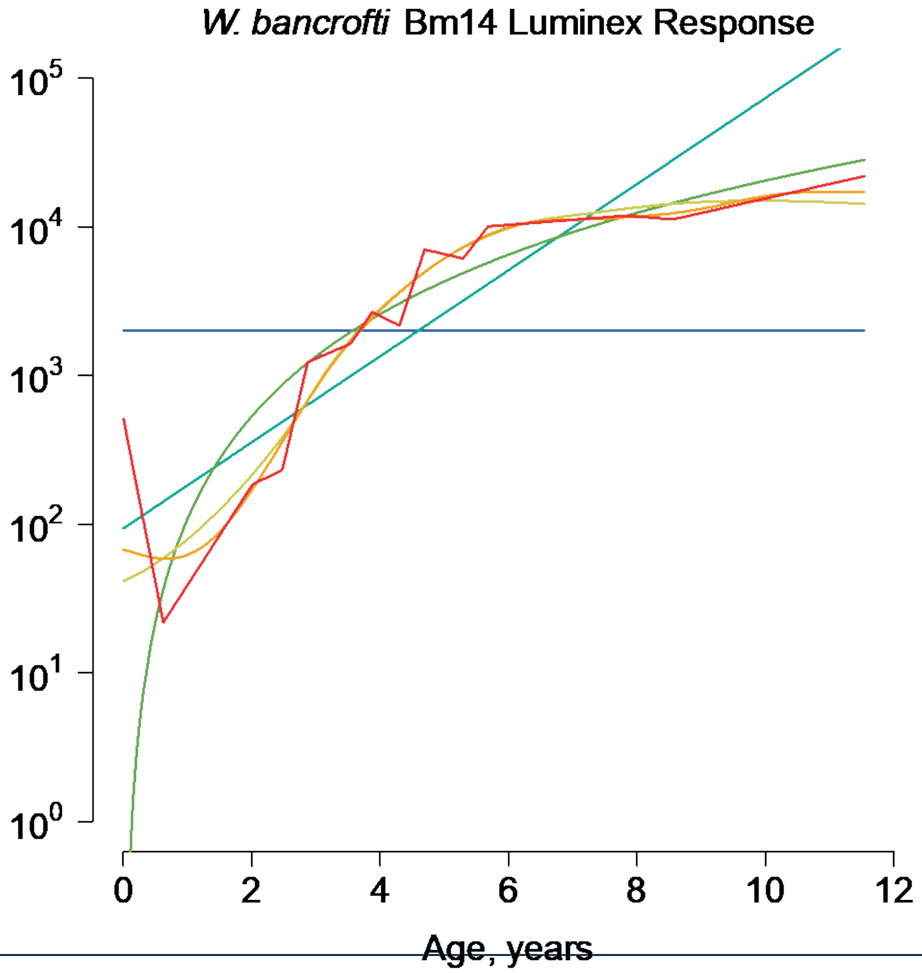


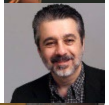
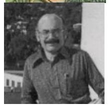
	Model / algorithm	CV MSE
	Mean	1.67
	Linear regression	1.13
	Ab acquisition model	1.24
	LOESS	1.03
	**Splines**	**1.01**

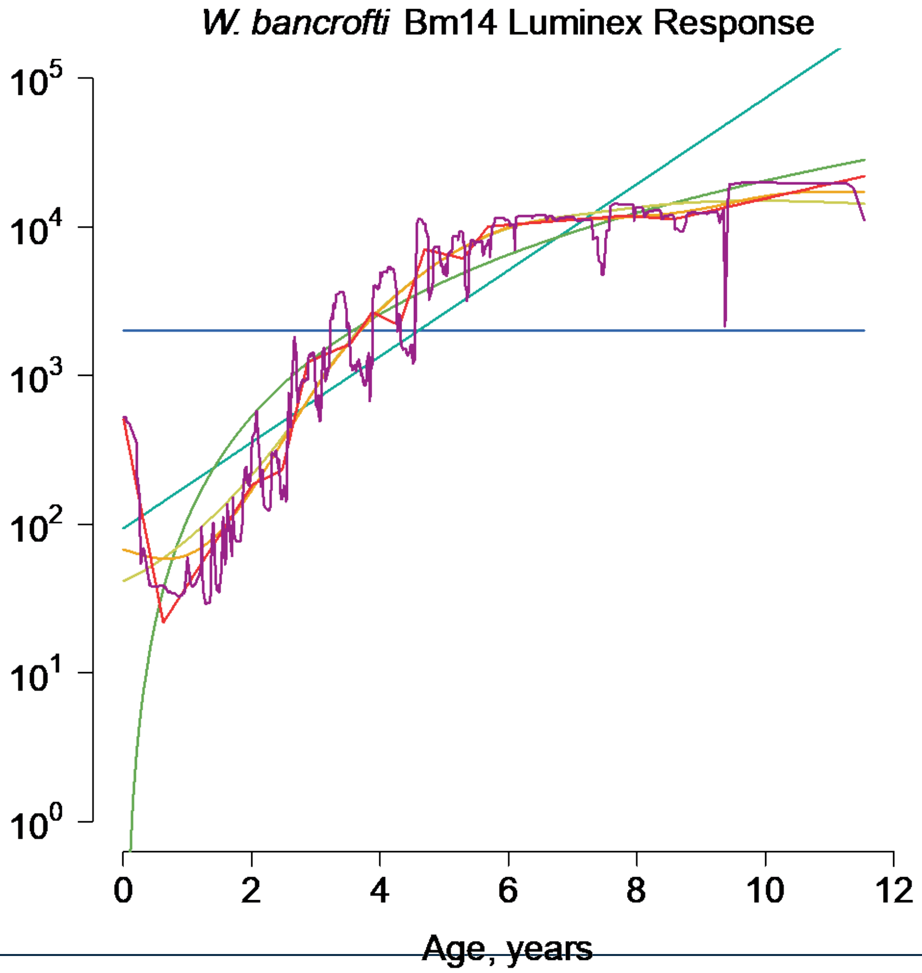




Model / algorithm	**CV MSE**
Mean	1.67
Linear regression	1.13
Ab acquisition model	1.24
LOESS	1.03
Splines	1.01
MARS	**1.02**



	Model / algorithm	CV MSE
	Mean	1.67
	Linear regression	1.13
	Ab acquisition model	1.24
	LOESS	1.03
	Splines	1.01
	MARS	1.02
	**Random Forest**	**1.04**

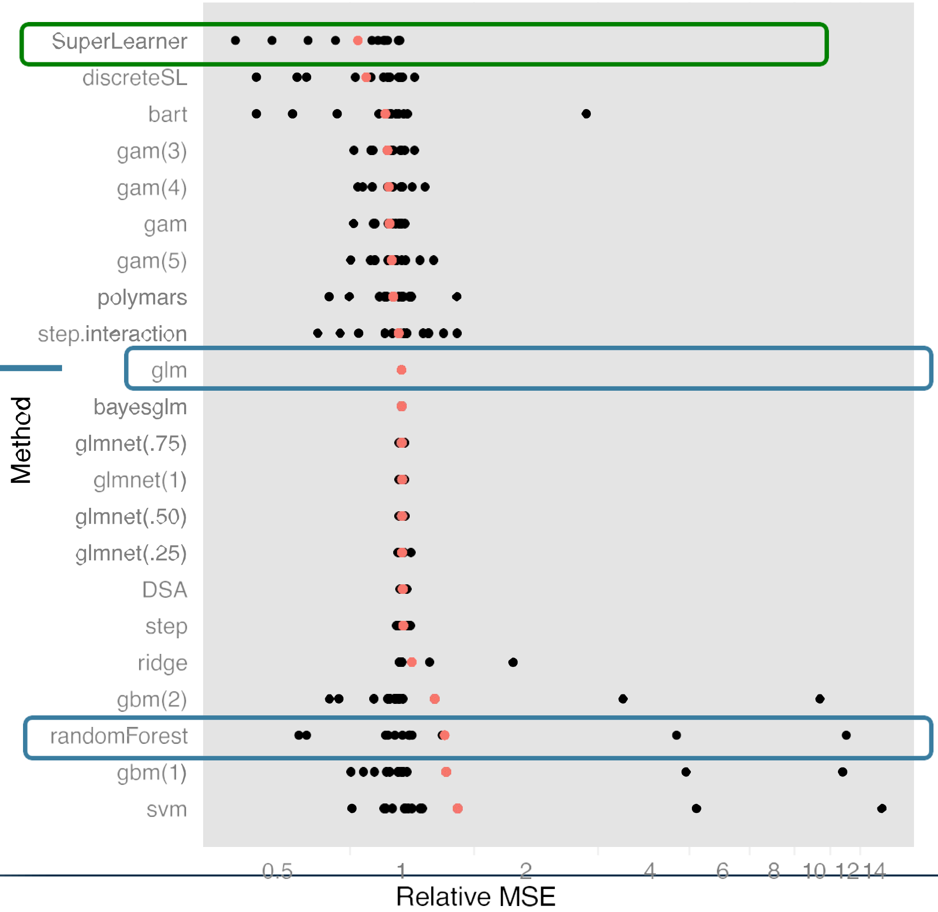


Super Learning: Building a winning team!

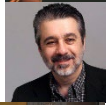
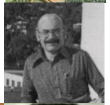
Super Learner: Best weighted combination of algorithms for a given prediction problem

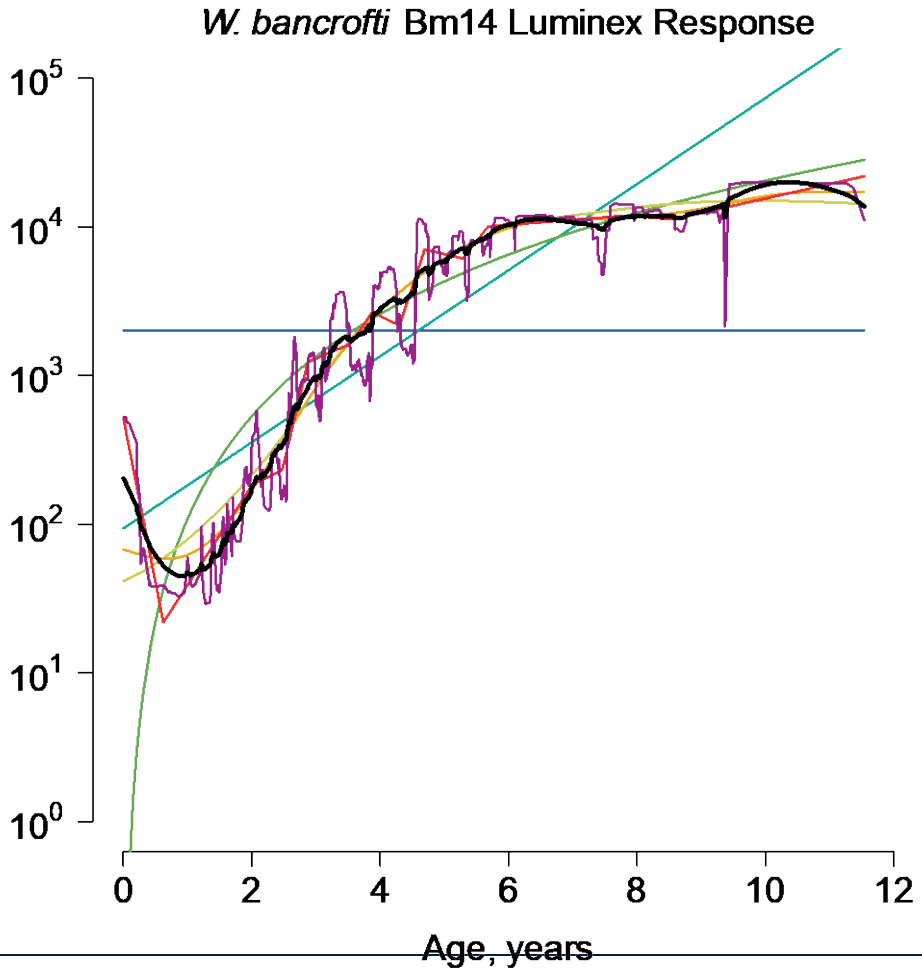
Example algorithm: Linear Main Term Regression

Example algorithm: Random Forest

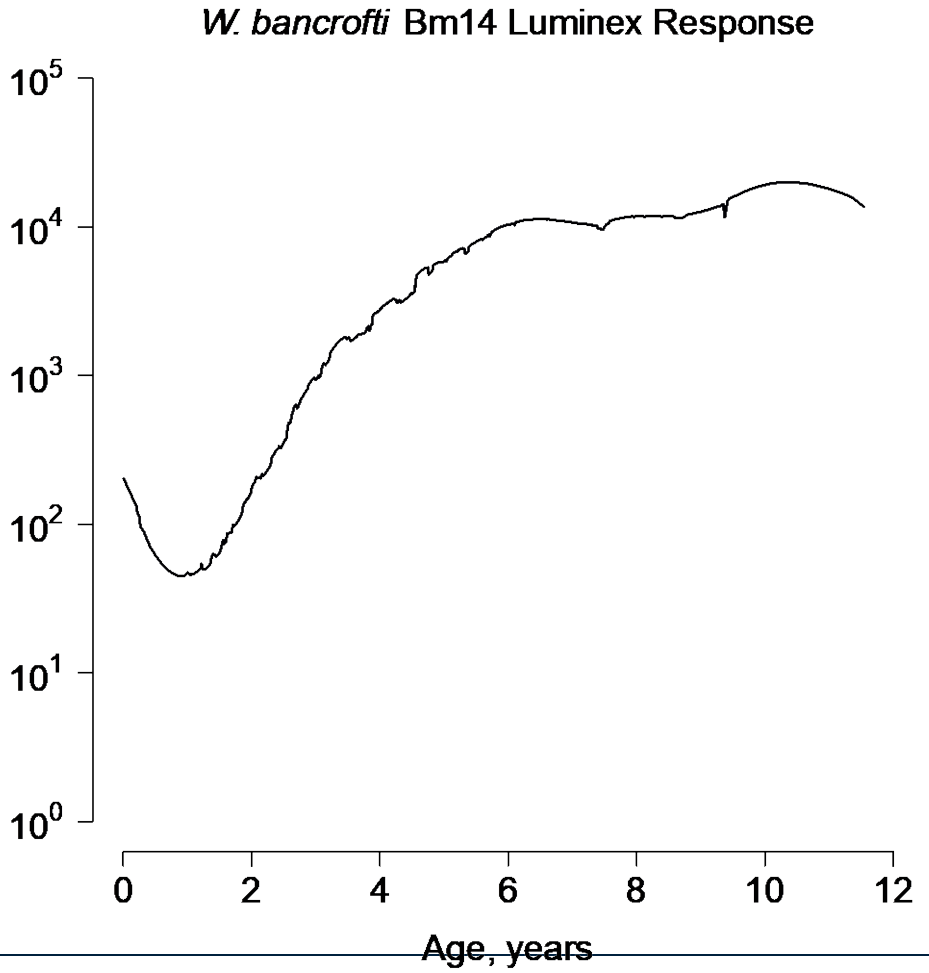


van der Laan 2011

	Model / algorithm	CV MSE
	Mean	1.67
	Linear regression	1.13
	Ab acquisition model	1.24
	LOESS	1.03
	Splines	1.01
	MARS	1.02
	Random Forest	1.04
	**Super Learner (stacked ensemble)**	**1.01**



Model / algorithm	CV MSE
Mean	1.67
Linear regression	1.13
Ab acquisition model	1.24
LOESS	1.03
Splines	1.01
MARS	1.02
Random Forest	1.04
Super Learner (stacked ensemble)	**1.01**



Superlearner: the algorithm steps

1. Pick the learners
2. Split data into cross-validation folds
3. Fit learners on cross-validation folds
4. Obtain predictions from each fitted model within folds
5. Use a metalearner to combine predictions across learners
6. Fit base learners on the full dataset
7. Use metalearner fit to weight base learners and get SuperLearner prediction
8. Predict on new data

Step 1: Pick the learners

- Better to pick a variety of different learners
- Can be guided by what’s normally used in your field + flexible ML algorithms
- You we learn more about individual algorithms tomorrow
- The **only downside** to using more learners is computation time

Step 2: Split data into cross-validation folds

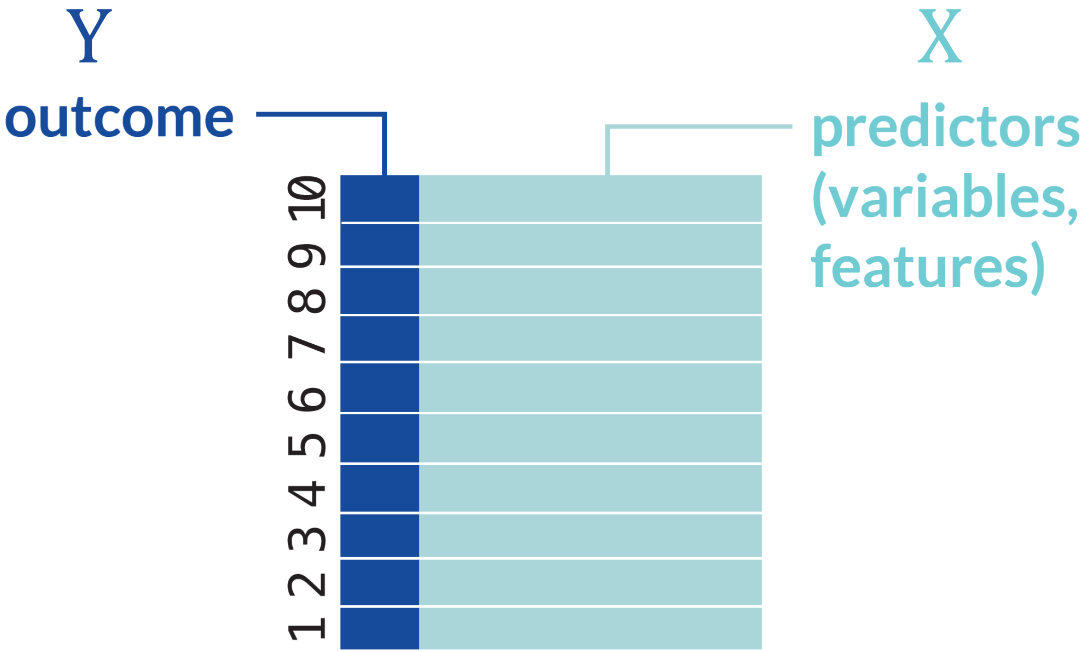
Example data

- ID variable identifying observation, individual or cluster
- Set of X predictors
- Y outcome

Simulated data set					
id	x1	x2	x3	x4	Y
1.000	2.287	1.000	1.000	1.385	5.270
2.000	−1.197	0.000	0.000	0.000	−1.197
3.000	−0.694	0.000	0.000	0.000	−0.694
4.000	−0.412	0.000	1.000	−0.541	−0.928
5.000	−0.971	0.000	0.000	0.000	−0.971
6.000	−0.947	0.000	1.000	−0.160	−1.107

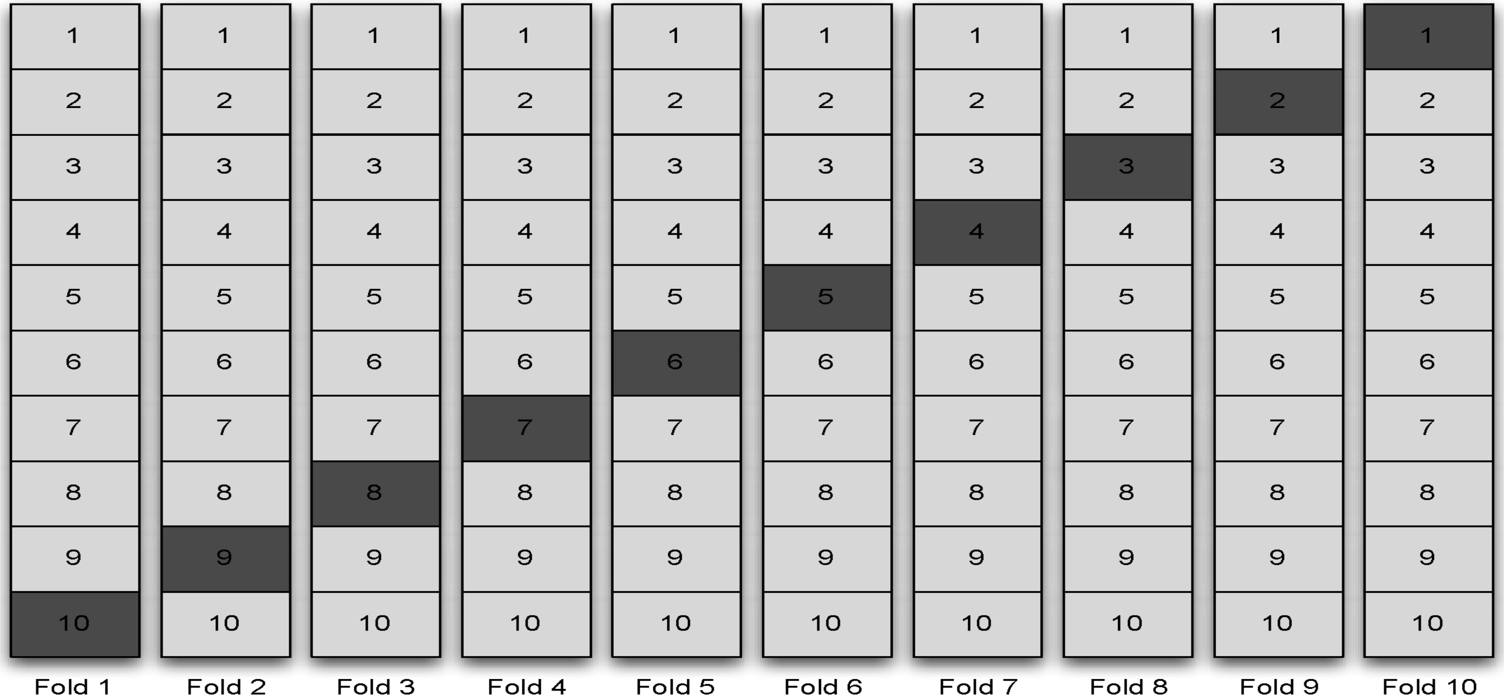
Step 2: Split data into cross-validation folds

Create indices for each of the K folds of size N/K



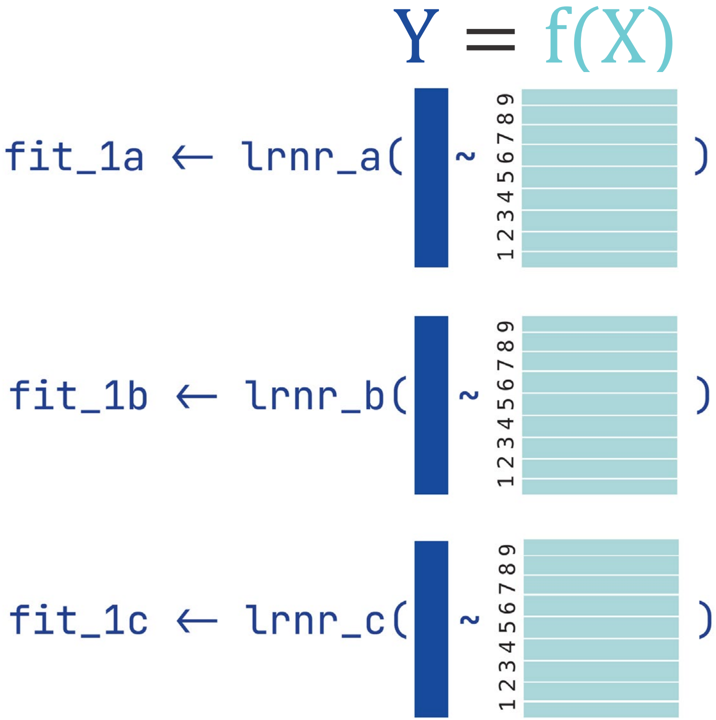
https://www.khstats.com/blog/sl/superlearning

V-Fold Cross Validation



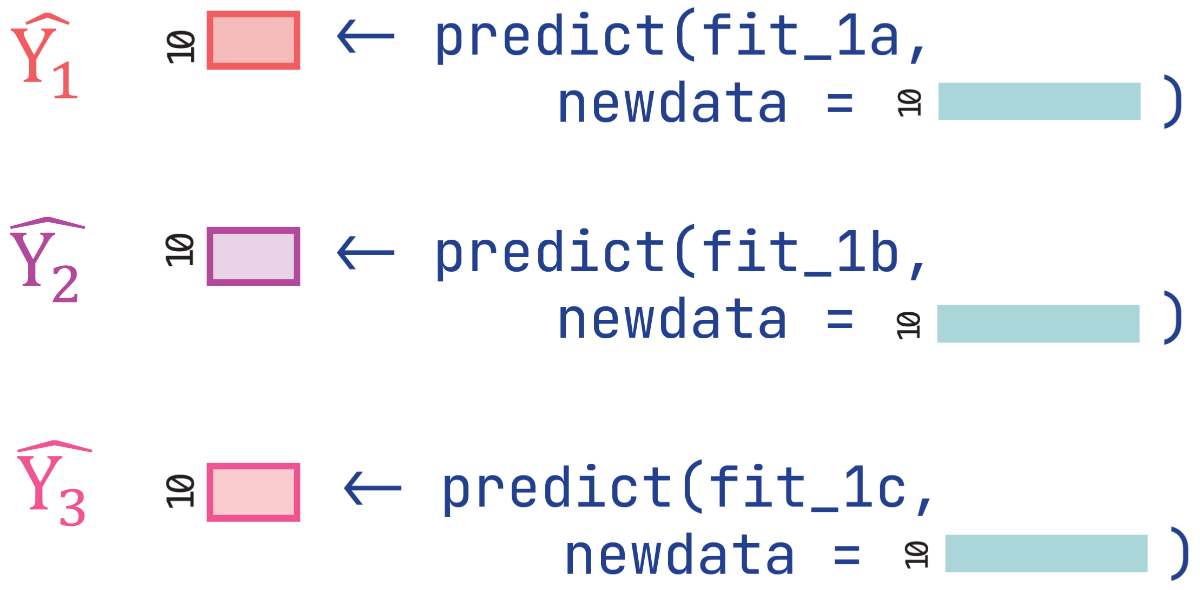
Step 3: Fit learners on first cross-validation folds

Fit model for each learner on the training data.
Here, in the first CV fold, data fold 10 is held out and 3 learners are fit to folds 1-9.



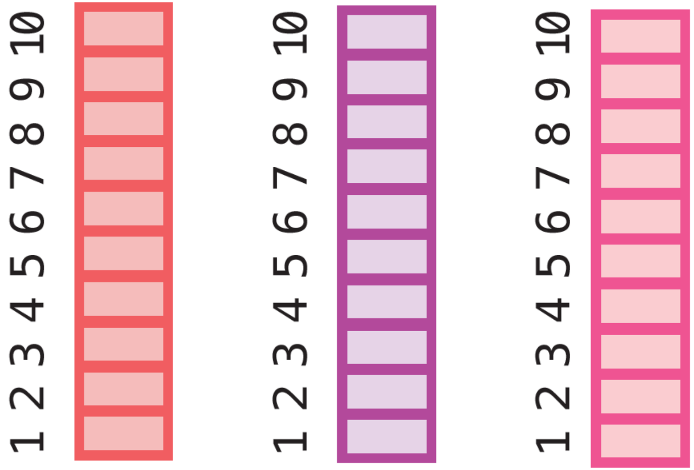
Step 4: Obtain predictions from each fitted model within folds

Get the predictions for each held out fold.



Step 4: Obtain predictions from each fitted model within folds

Repeat for the entire dataset, so now there are outcome predictions for every row of data

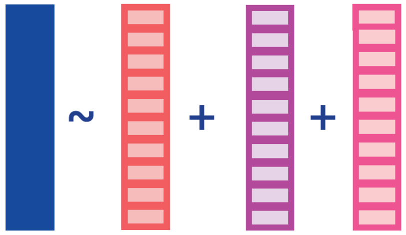


Step 5: Use a metalearner to combine predictions across learners

- This is a function combining predictions across learners to minimize the loss function.
  - Example: CV-MSE
- Could use something as simple as a linear regression, predicting the true outcome from the set of learner-specific prediction.
  - Coefficients become the weights of each algorithm
- Non-negative least squares is the R package default, but variety of options.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$Y = \beta_1\widehat{Y_1} + \beta_2\widehat{Y_2} + \beta_3\widehat{Y_3}$$



```
SL_fit ← meta_lrnr(  ~  +  +  )
```

Step 5: Use a metalearner to combine predictions across learners

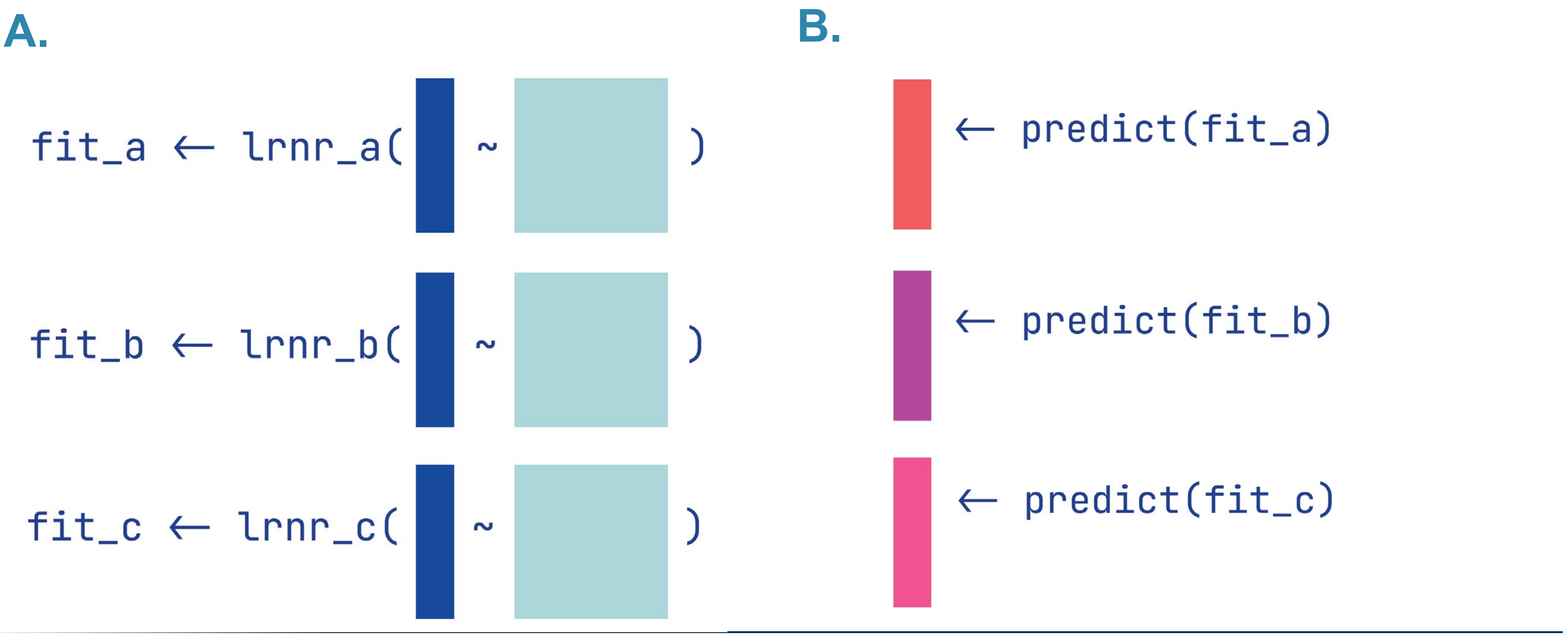
- You can compare the risk of different learners and the weights chosen by the metalearners using the SuperLearner R package

```
Call:
SuperLearner(Y = obs$y, X = x_df, family = gaussian(),
SL.library = c("SL.ranger",
   "SL.glmnet", "SL.earth"))
```

	Risk (MSE)	Coef (weight)
SL.ranger_All	0.013672503	0.1606329
SL.glmnet_All	0.097257031	0
SL.earth_All	0.003181357	0.8393671

Step 6: Fit base learners on the full dataset and get predictions

A. Get a single model fit (not cross-validated) for each learner on the full data
B. Get full-data predictions



Step 7: Use metalearner fit to weight base learners and get SuperLearner prediction

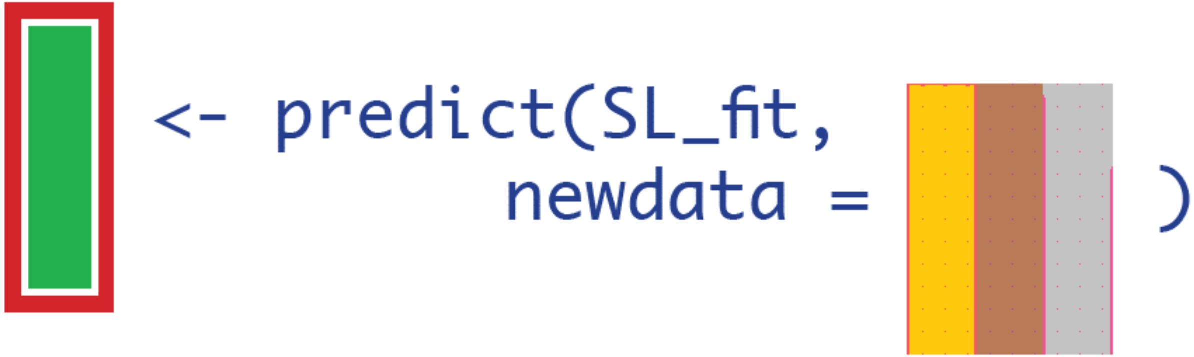
Combine the learner-specific predictions with the estimated weights to get a single predicted outcome per observation

$$\widehat{Y_{SL}} = \beta_1 \widehat{Y_1} + \beta_2 \widehat{Y_2} + \beta_3 \widehat{Y_3}$$

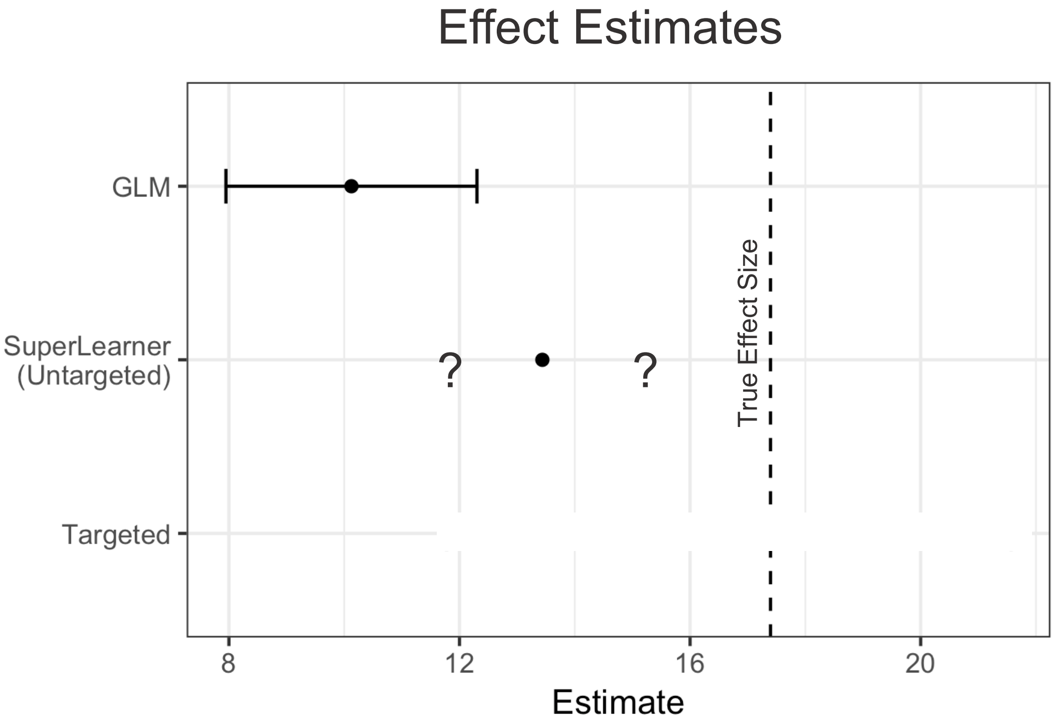
```
← predict(SL_fit,
          newdata =        )
```

Step 8: (Optional) Evaluate SuperLearner fit

- Repeat steps 5 and 6 on any new data
- This is how to use SuperLearner as a clinical screening/prediction tool



Results: removing bias compared to regression approach



GLM did not learn the correct outcome mechanism, so its estimate is very biased

SuperLearner does a better job of estimating the outcome mechanism, but does not allow valid inference

Questions?