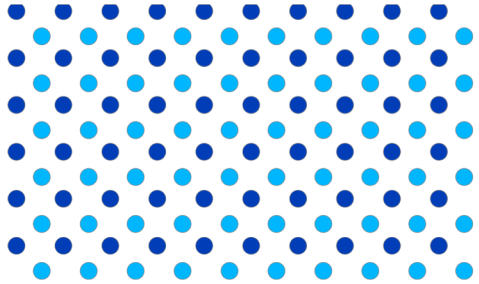
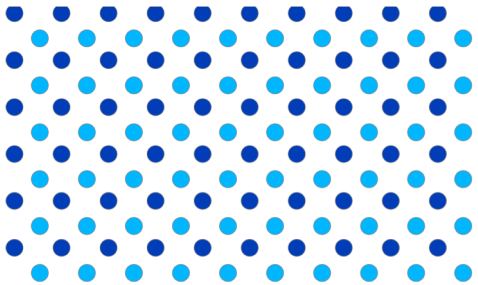


Ensemble Machine Learning with R

Poo Kuan Hoong



confeRence

About Me

Poo Kuan Hoong

- Google Developer Expert (GDE) in Machine Learning
- Principal Data Scientist – Coqnitics
- Senior Data Scientist - ASEAN Data Analytics Exchange (ADAX)
- Senior Manager Data Science – Nielsen
- Senior Lecturer – Multimedia University (MMU)

Introduction

- There are many machine learning algorithms out there
- Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone
- Based on wisdom of the crowd
- No free lunch theorem: there is no algorithm that is always the most accurate
- Generate a group of base-learners which when combined have higher accuracy

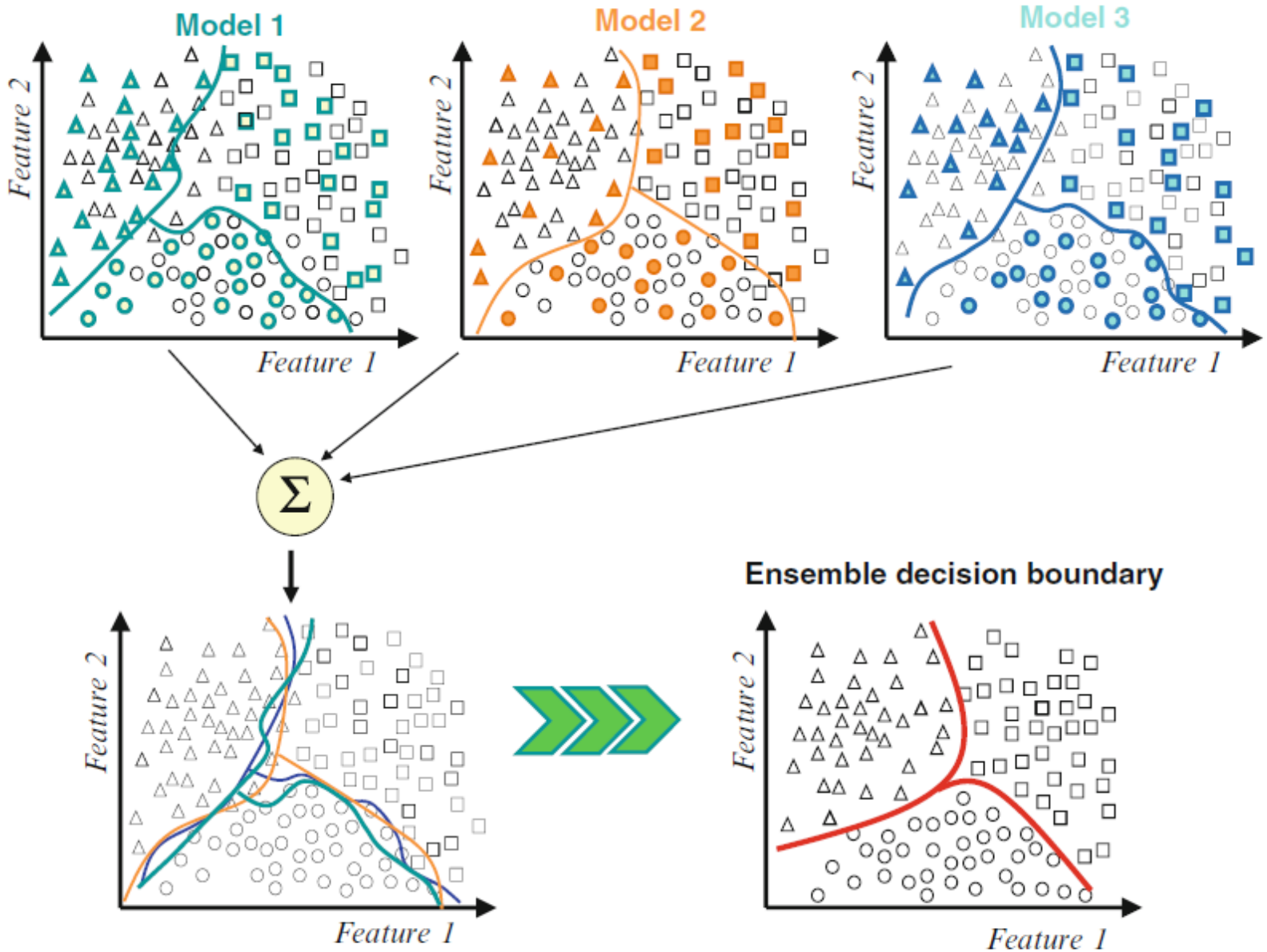
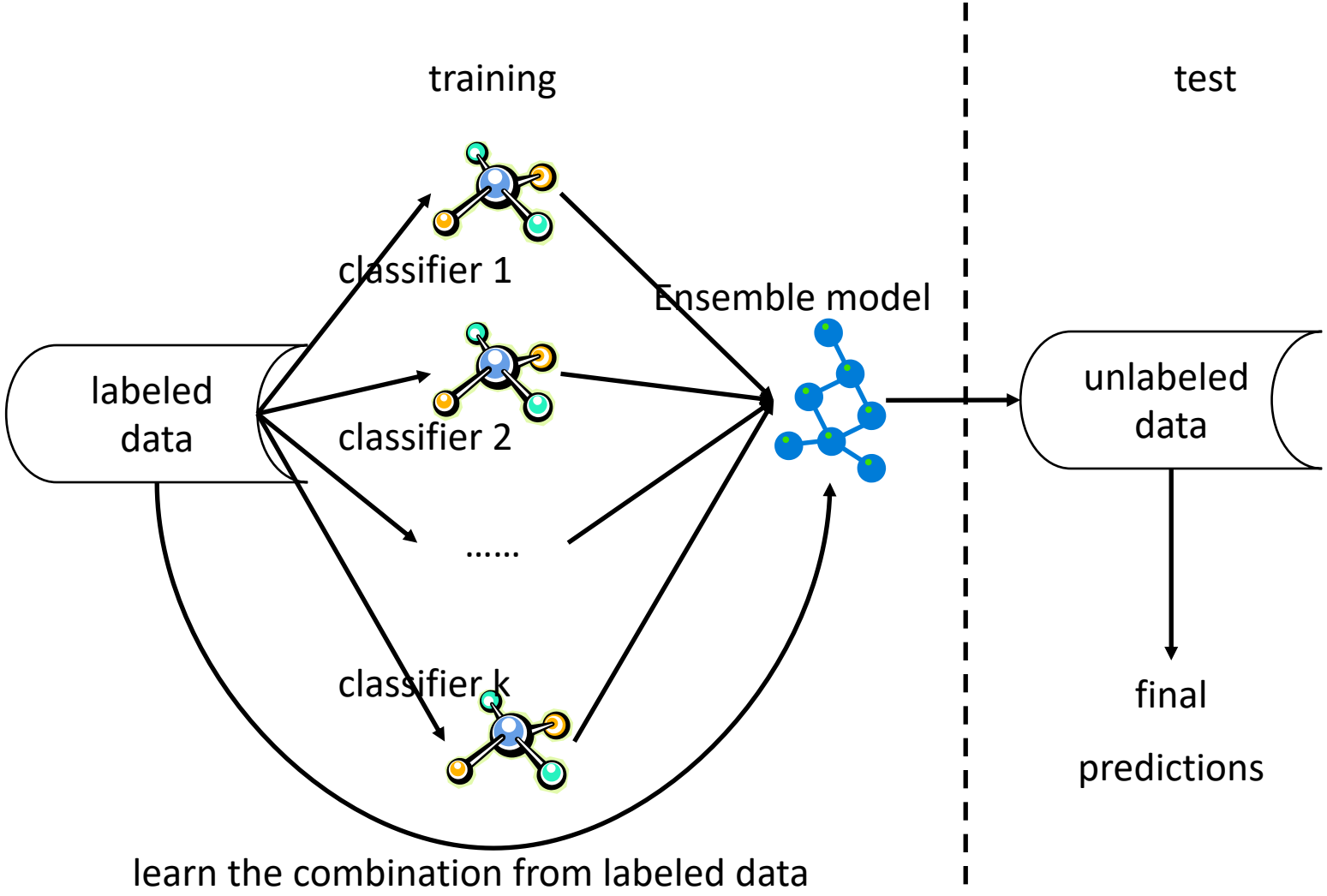


Fig. 1.1 Variability reduction using ensemble systems

Ensemble Machine Learning – Why?

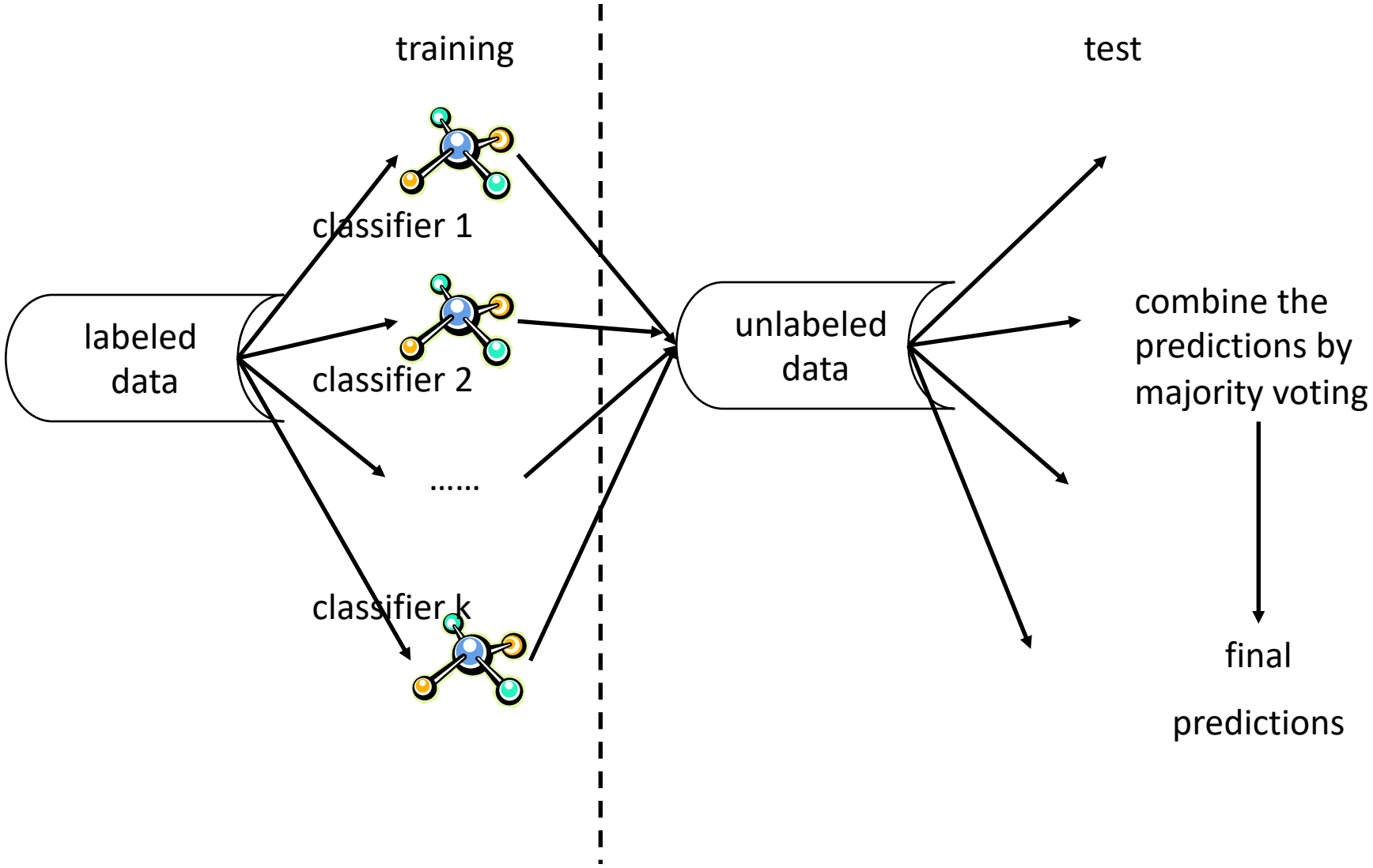
- Ensemble methods outperform individual (base) learning algorithms.
- By combining a set of individual learning algorithms using a *metalearning* algorithm, ensemble methods can approximate complex functional relationships.
- When the true functional relationship is not in the set of base learning algorithms, ensemble methods approximate the true function well.
- Ensemble methods can, even asymptotically, perform only as well as the best weighted combination of the candidate learners.

Ensemble of Classifiers - Learn to Combine



Algorithms: boosting, stacked generalization, rule ensemble, Bayesian model averaging……

Ensemble of Classifiers – Voting/ Consensus



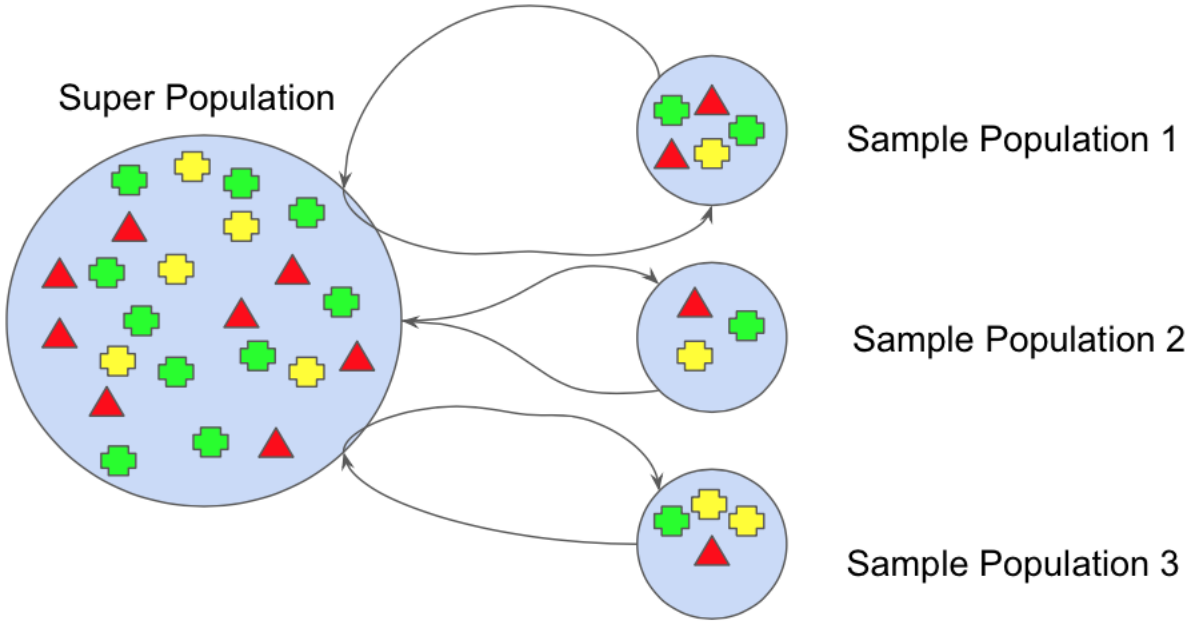
Algorithms: bagging, random forest, random decision tree, model averaging of probabilities……

Ensemble Machine Learning – Why?

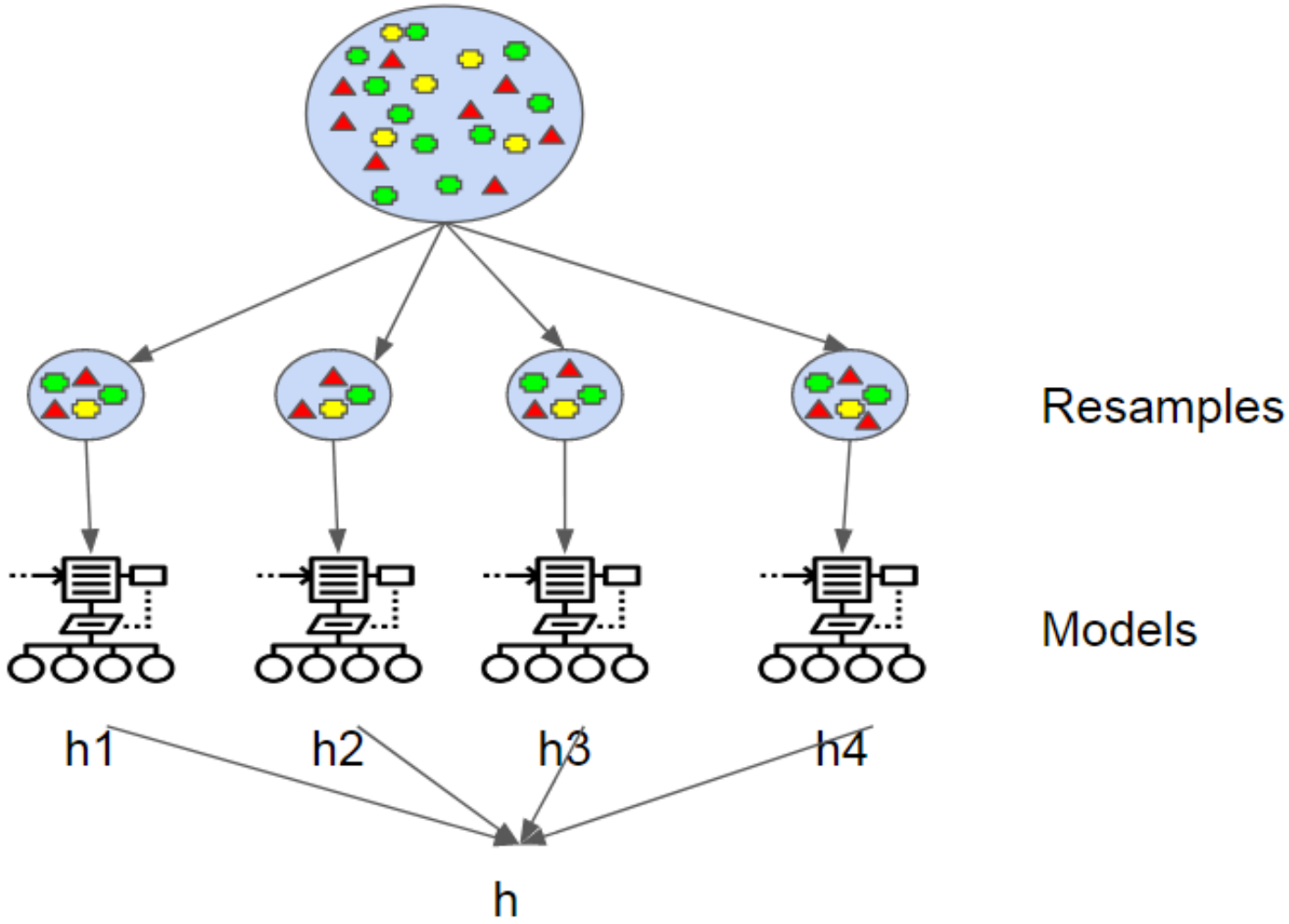
- Common strategies for performing ensemble learning:
  - **Bagging** – reduces variance and increases accuracy; robust against outliers; often used with decision trees (i.e., Random Forest).
  - **Boosting** – reduces variance and increases accuracy; not robust against outliers or noise; accommodates any loss function.
  - **Stacking** – used in combining "strong" learners; requires a *metalearning* algorithm to combine the set of learners.

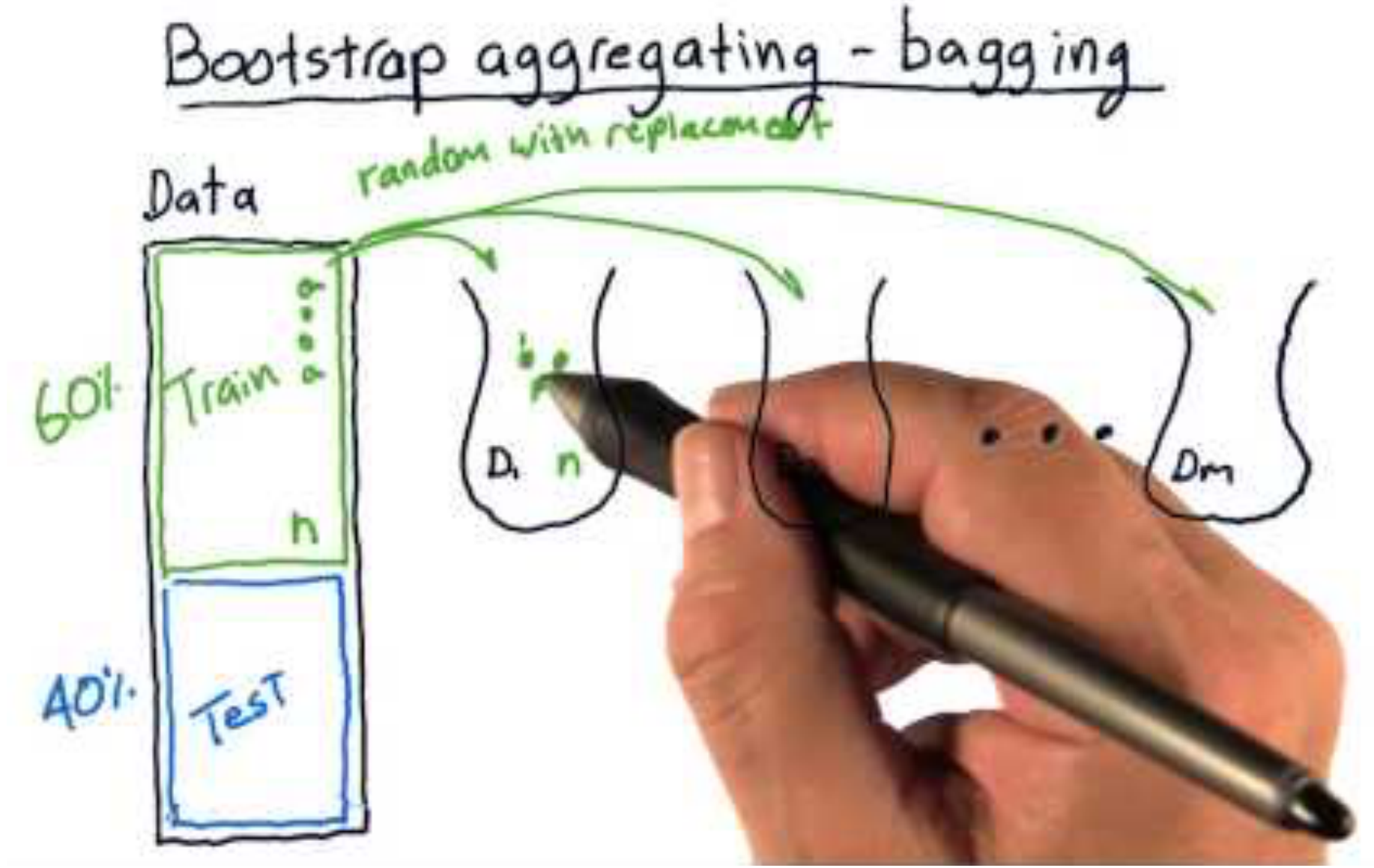
Bagging

- Take M bootstrap samples (with replacement)
- Train M different classifiers on these bootstrap samples
- For a new query, let all classifiers predict and take an average (or majority vote)
- If the classifiers make independent errors, then their ensemble can improve performance.
- Stated differently: the variance in the prediction is reduced (we don't suffer from the random errors that a single classifier is bound to make).



Bagging



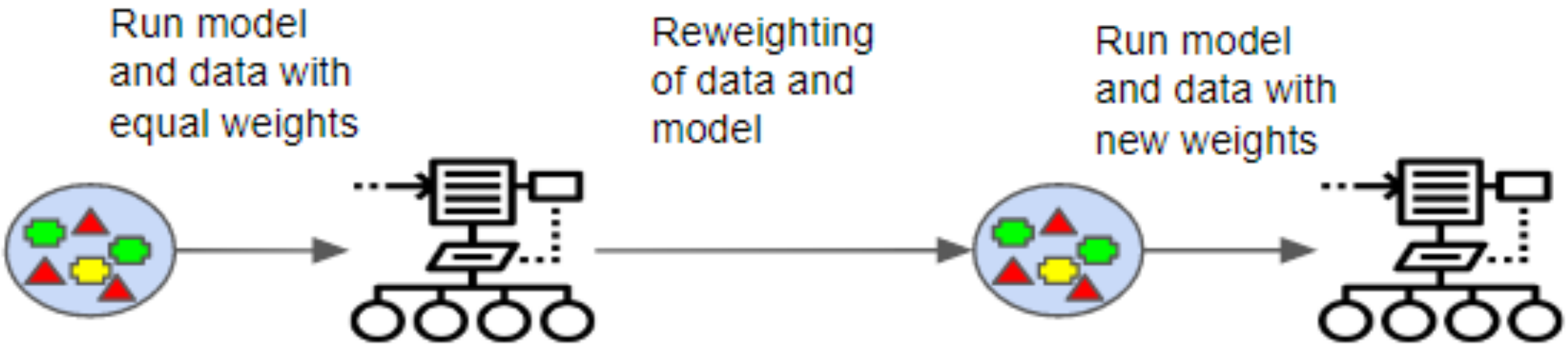


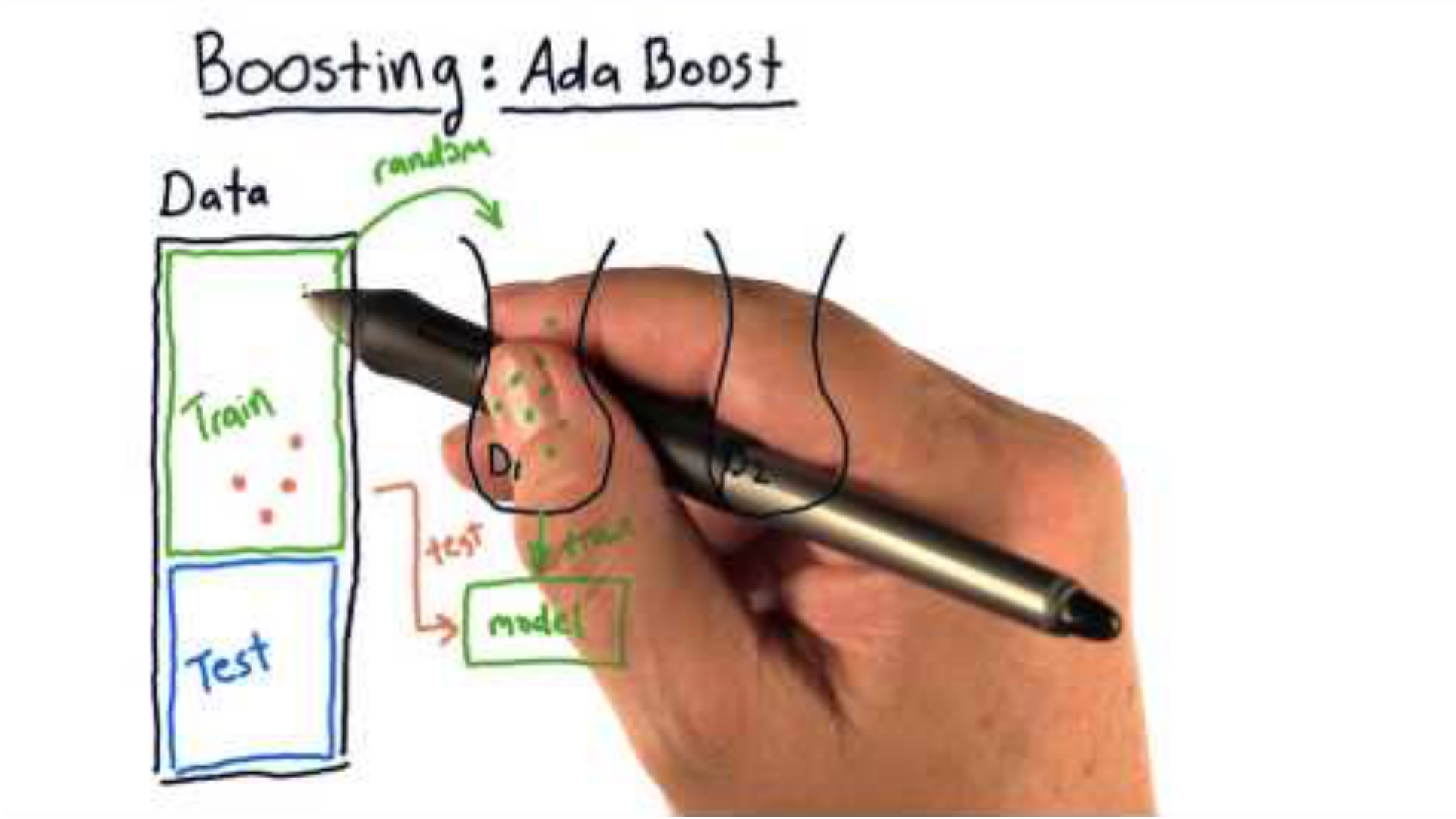
https://youtu.be/2Mg8QD0F1dQ

Boosting

- Train classifiers (e.g. decision trees) in a sequence.
- A new classifier should focus on those cases which were incorrectly classified in the last round.
- Combine the classifiers by letting them vote on the final prediction (like bagging).
- Each classifier is “weak” but the ensemble is “strong.”
- AdaBoost is a specific boosting method

Boosting





https://www.youtube.com/watch?v=GM3CDQfQ4sw

Gradient Boosting Machine (GBM)

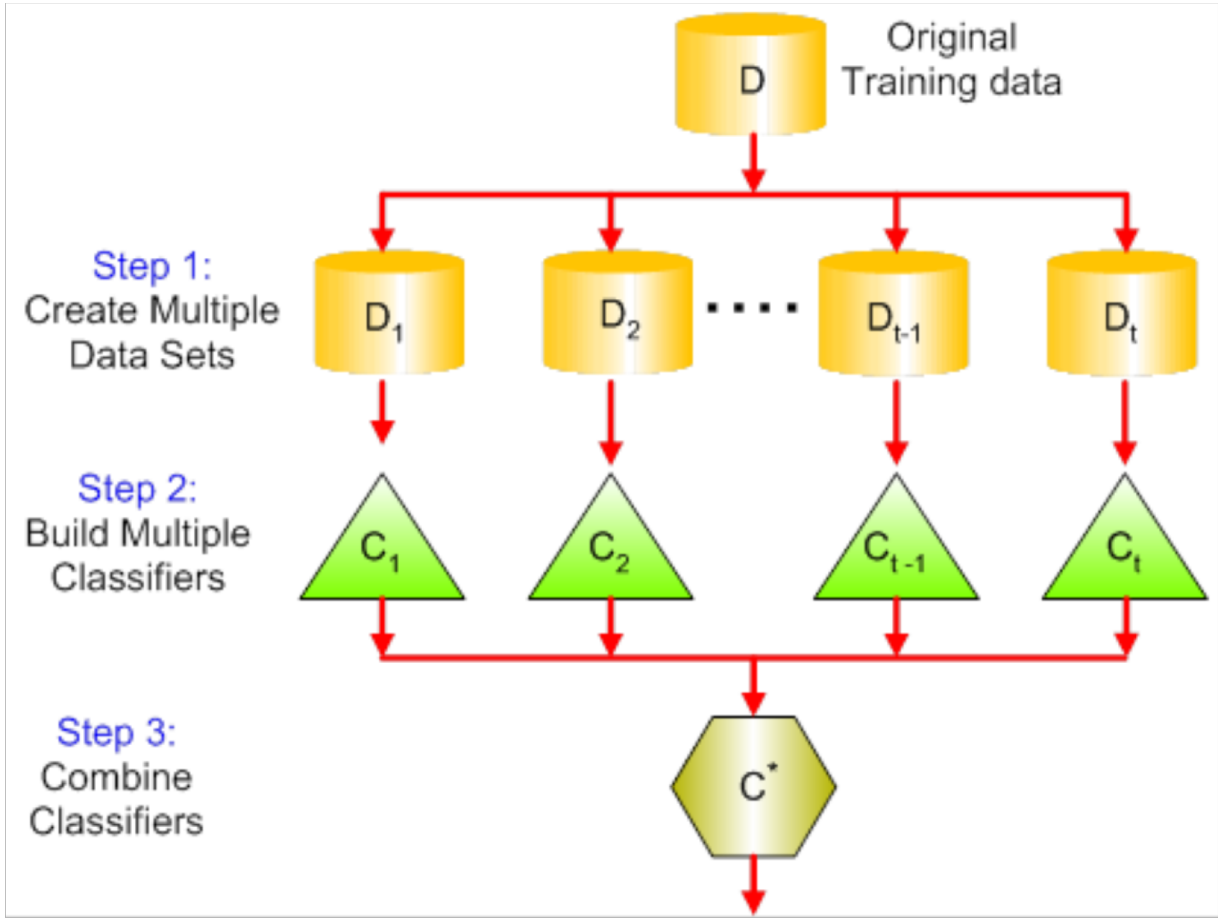
- A variant of boosting algorithm
- Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

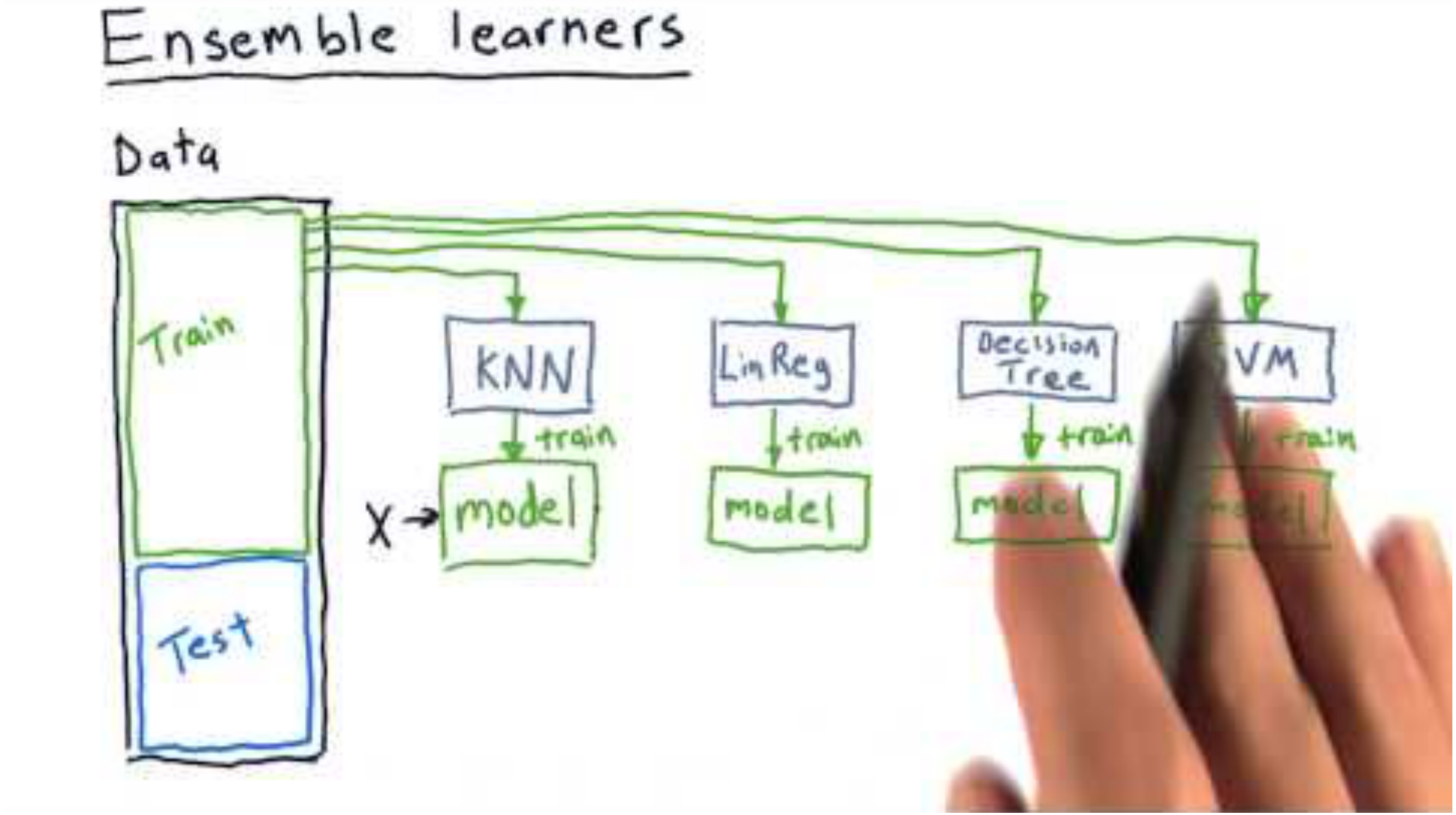
XGBoost

- https://xgboost.readthedocs.io/en/latest/
- eXtreme Gradient Boosting algorithm
- A variant of gradient boosting machine (GBM), a tree based model
- xgboost used a more regularized model formalization to control over-fitting, which gives it better performance.
- Better support for multicore processing which reduces overall training time.
- A model used by many winning teams in Kaggle competition

Stacking

- Stacking, also called Super Learning or Stacked Regression, is a class of algorithms that involves training a second-level “metalearner” to find the optimal combination of the base learners.
- Unlike bagging and boosting, the goal in stacking is to ensemble strong, diverse sets of learners together.





https://www.youtube.com/watch?v=Un9zObFjBH0

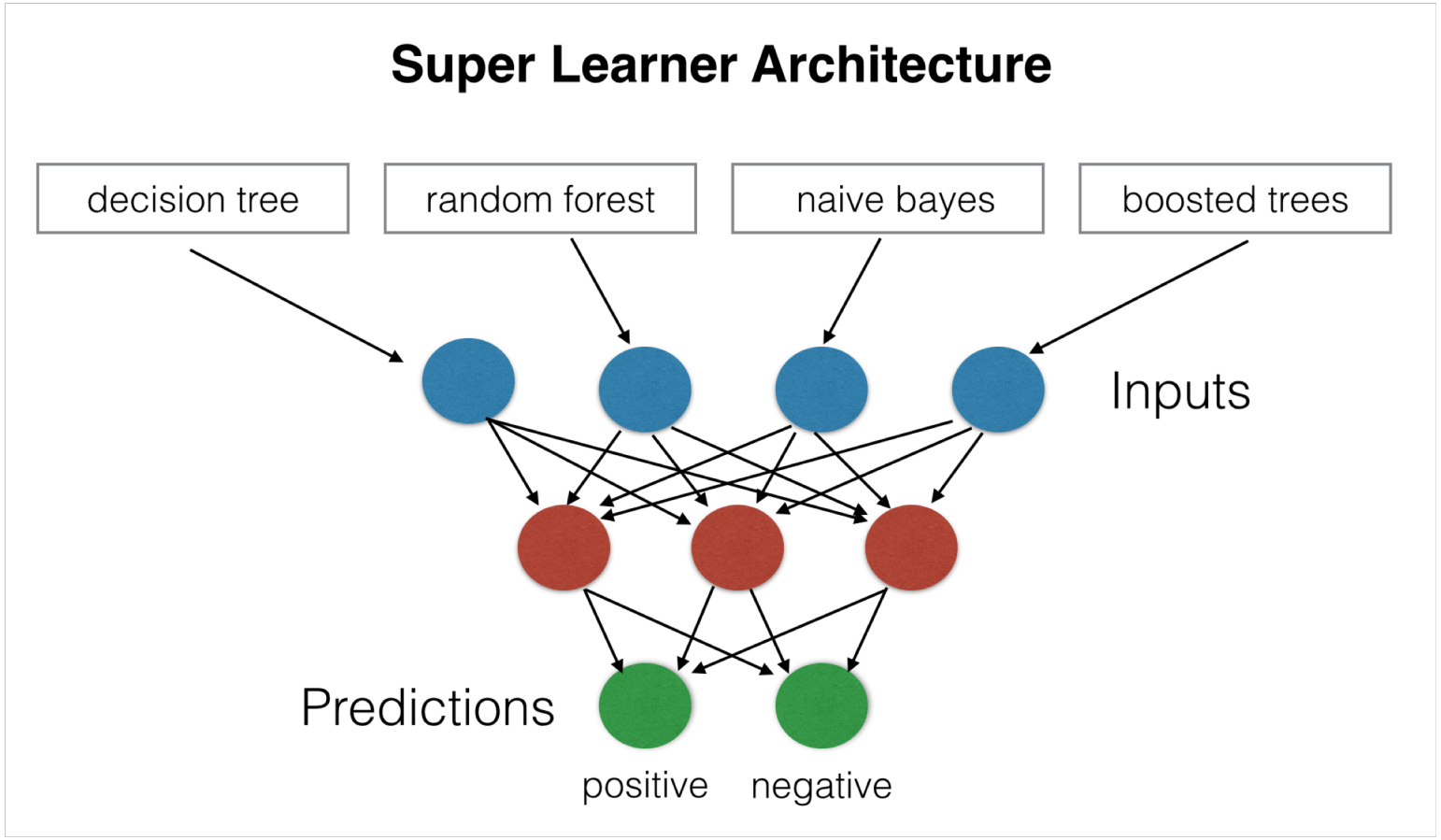
Random Forest

- **Random forest** (or **random forests**) is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees.
- Can be used for Classification or Regression (CART)

https://www.youtube.com/watch?v=6yICuCnlh5Q

SuperLearner Package

- Implements the super learner prediction method and contains a library of prediction algorithms to be used in the super learner.
- https://cran.r-project.org/web/packages/SuperLearner/index.html



SuperLearner Package

- The package is available at: https://github.com/ecpolley/SuperLearner
- Need to install R packages `nnls` and `quadprog` before installing `SuperLearner`

SuperLearner Package

Table: Main functions in the **SuperLearner** package

Function	Description
`SuperLearner`	fits super learner
`CV.SuperLearner`	cross-validate in super learner
`listWrappers`	returns list of wrappers in package
`write.SL.template`	prediction wrapper template
`write.screen.template`	screening wrapper template
`write.method.template`	method wrapper template

SuperLearner Package

SuperLearner

```
fitSL <- SuperLearner(Y = Y,
                      X = X,
                      SL.library = c('SL.glm'),
                      family = gaussian(),
                      method = 'method.NNLS',
                      verbose = TRUE,
                      cvControl = list(V = 10))
```

SuperLearner Package

Table: Main arguments for `SuperLearner`

Name	Description	Req.	Default
`Y`	outcome	Y	–
`X`	data.frame for fit	Y	–
`newX`	data.frame for predict	N	`X`
`SL.library`	library of algorithms	Y	–
`cvControl`	list for CV control	N	–
`control`	optional controls	N	–
`verbose`	detailed report	N	FALSE
`family`	error distribution	N	gaussian
`method`	loss function & model	N	NNLS
`id`	cluster id	N	–
`obsWeights`	observations weights	N	–

SuperLearner Package

- `Y` and `X` are the data used to fit each algorithm (the learning data)
- `newX` is not required but can be a helpful shortcut.
- `newX` will not be used to fit the models.

Example with X and `newX`:

```
fit <- glm(Y ~ ., data = X)
out <- predict(fit, newdata = newX)
```

SuperLearner Package

- There are two types of algorithms that can be used in `SL.library`:
  - Prediction algorithms. Algorithms that take as input X and Y and return a predicted Y value.
  - Screening algorithms. Algorithms designed to reduce the dimension of X. They take as input X and Y and return a logical vector indicating the columns in X passing the screening. Screening algorithms can be coupled with prediction algorithms to form new prediction algorithms.

SuperLearner Package

Algorithm	Description	Package
`glm`	linear model	**stats**
`randomForest`	random Forest	**randomForest**
`bagging`	bootstrap aggregation of trees	**ipred**
`gam`	generalized additive models	**gam**
`gbm`	gradient boosting	**gbm**
`nnet`	neural network	**nnet**
`polymars`	polynomial spline regr.	**polspline**
`bart`	Bayesian additive regr. trees	**BayesTree**
`glmnet`	elastic net	**glmnet**
`svm`	support vector machine	**e1071**
`bayesglm`	Bayesian glm	**arm**
`step`	stepwise glm	**stats**

Summary

- Ensemble Methods combine several machine learning algorithms and harness the combined result
- Ensemble methods improve overall result by reducing the variance

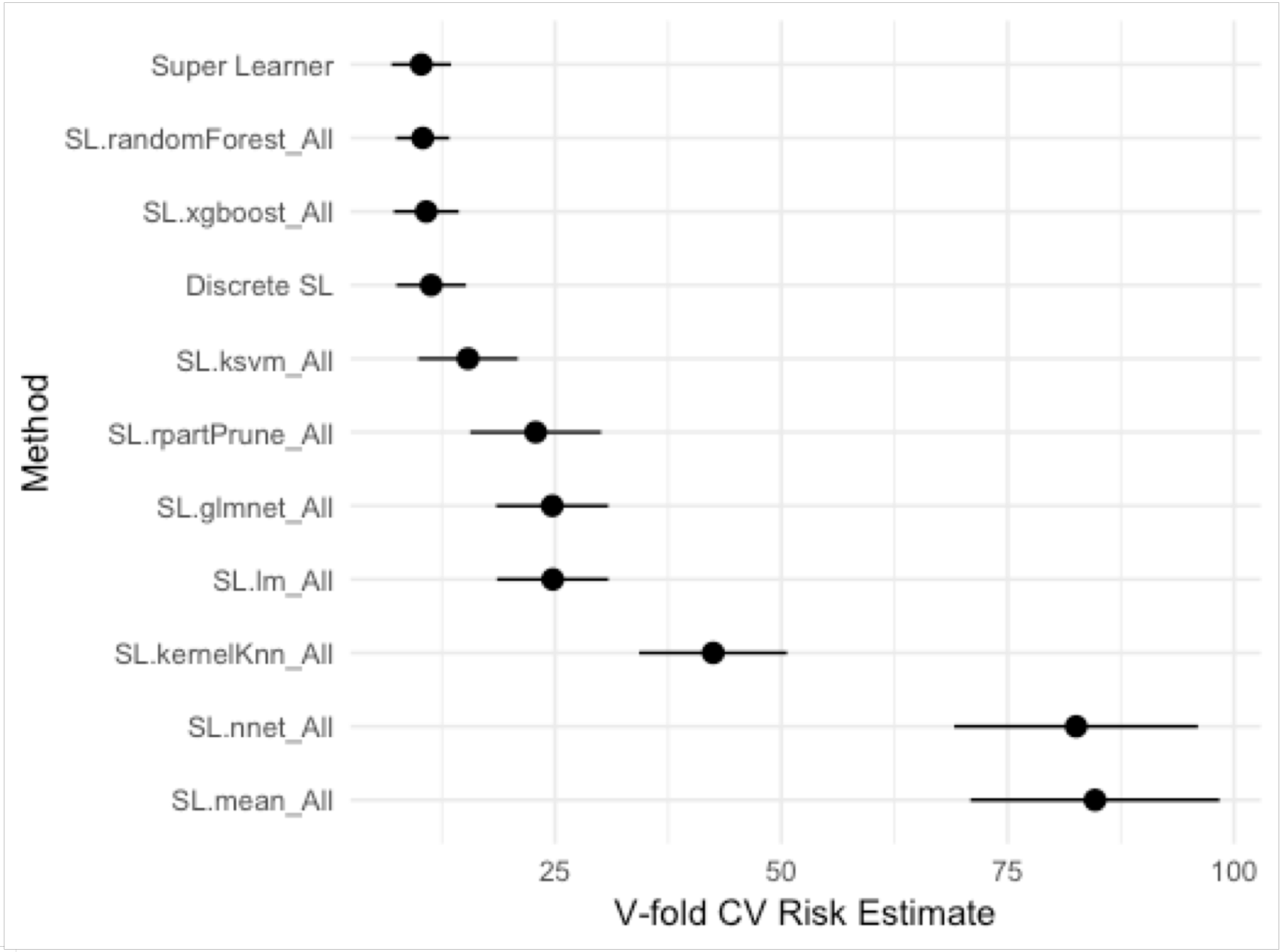
DEMO

Boston Dataset

- This dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass
- There are **14** attributes in each case of the dataset. They are:CRIM - per capita crime rate by town
  - **ZN** - proportion of residential land zoned for lots over 25,000 sq.ft.
  - **INDUS** - proportion of non-retail business acres per town.
  - **CHAS** - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
  - **NOX** - nitric oxides concentration (parts per 10 million)
  - **RM** - average number of rooms per dwelling
  - **AGE** - proportion of owner-occupied units built prior to 1940
  - **DIS** - weighted distances to five Boston employment centres
  - **RAD** - index of accessibility to radial highways
  - **TAX** - full-value property-tax rate per $10,000
  - **PTRATIO** - pupil-teacher ratio by town
  - **B** - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
  - **LSTAT** - % lower status of the population
  - **MEDV** - Median value of owner-occupied homes in $1000's

Result

- The final super learner prediction model is the weighted combination of the library algorithms where the estimates of the weights can be found with coef(fitSL).


Super Learner
SL.randomForest_All
SL.xgboost_All
Discrete SL
SL.ksvm_All
SL.rpartPrune_All
SL.glmnet_All
SL.lm_All
SL.kernelKnn_All
SL.nnet_All
SL.mean_All
Method
25
50
75
100
V-fold CV Risk Estimate

Q&A

Thanks!

Questions?

@kuanhoong

https://www.linkedin.com/in/kuanhoong

kuanhoong@gmail.com

confeRence