CS57300
PURDUE UNIVERSITY

NOVEMBER 1, 2021

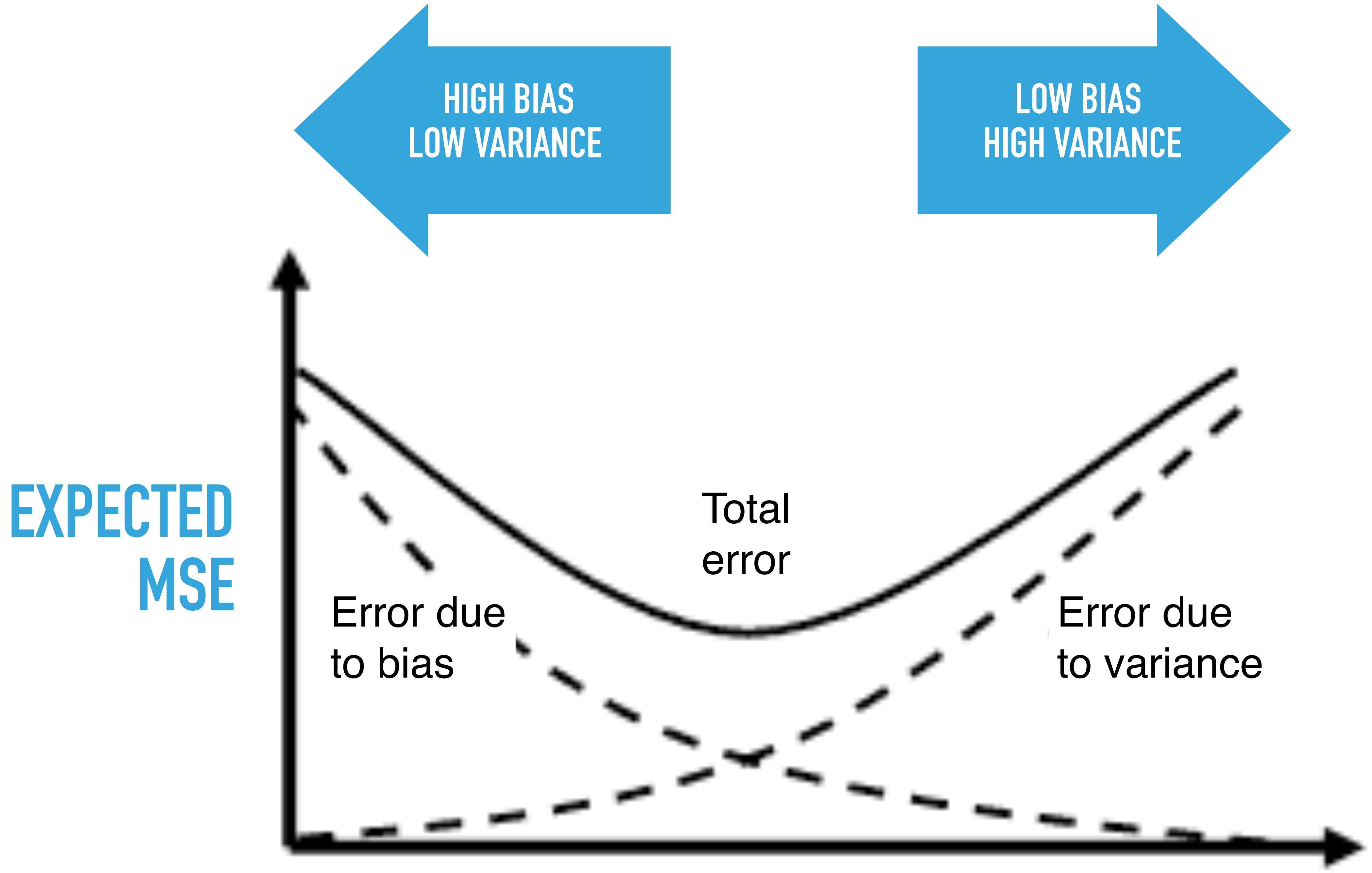
DATA MINING

ANNOUNCEMENTS

- Assignment 4 will be out today!
  - Due on November 14
  - Start early! It’s going to be more time consuming to train the models!

ENSEMBLE METHODS

BIAS/VARIANCE TRADEOFF FOR LEARNING A SINGLE MODEL



Bias-variance tradeoff:
increasing the size of the model space can **reduce bias** of the learned model, but that also tends to **increase variance**...

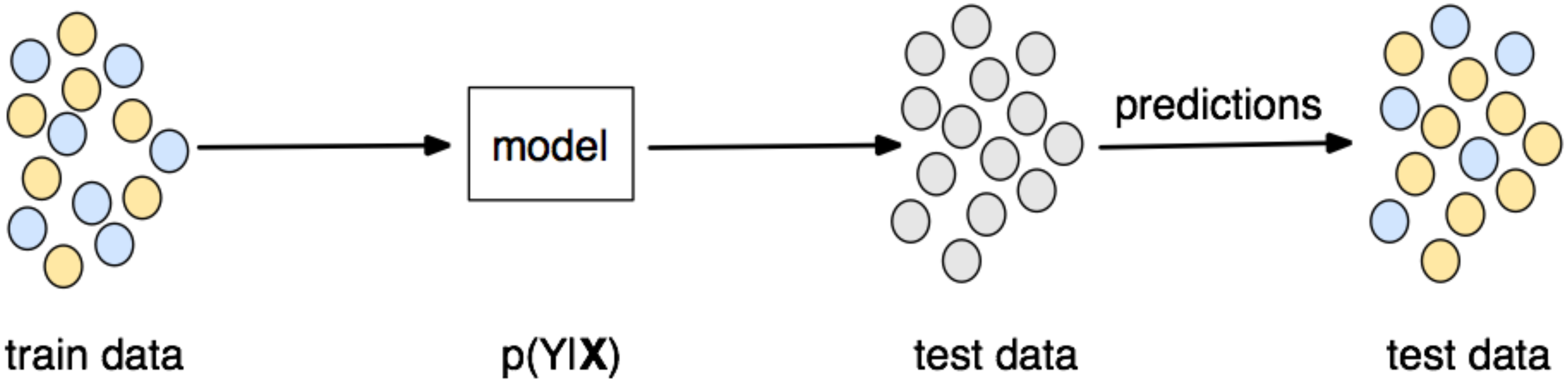
and *decreasing* the model space tends to **reduce variance** but also **increase bias**

HOW ABOUT BLENDING MULTIPLE MODELS?

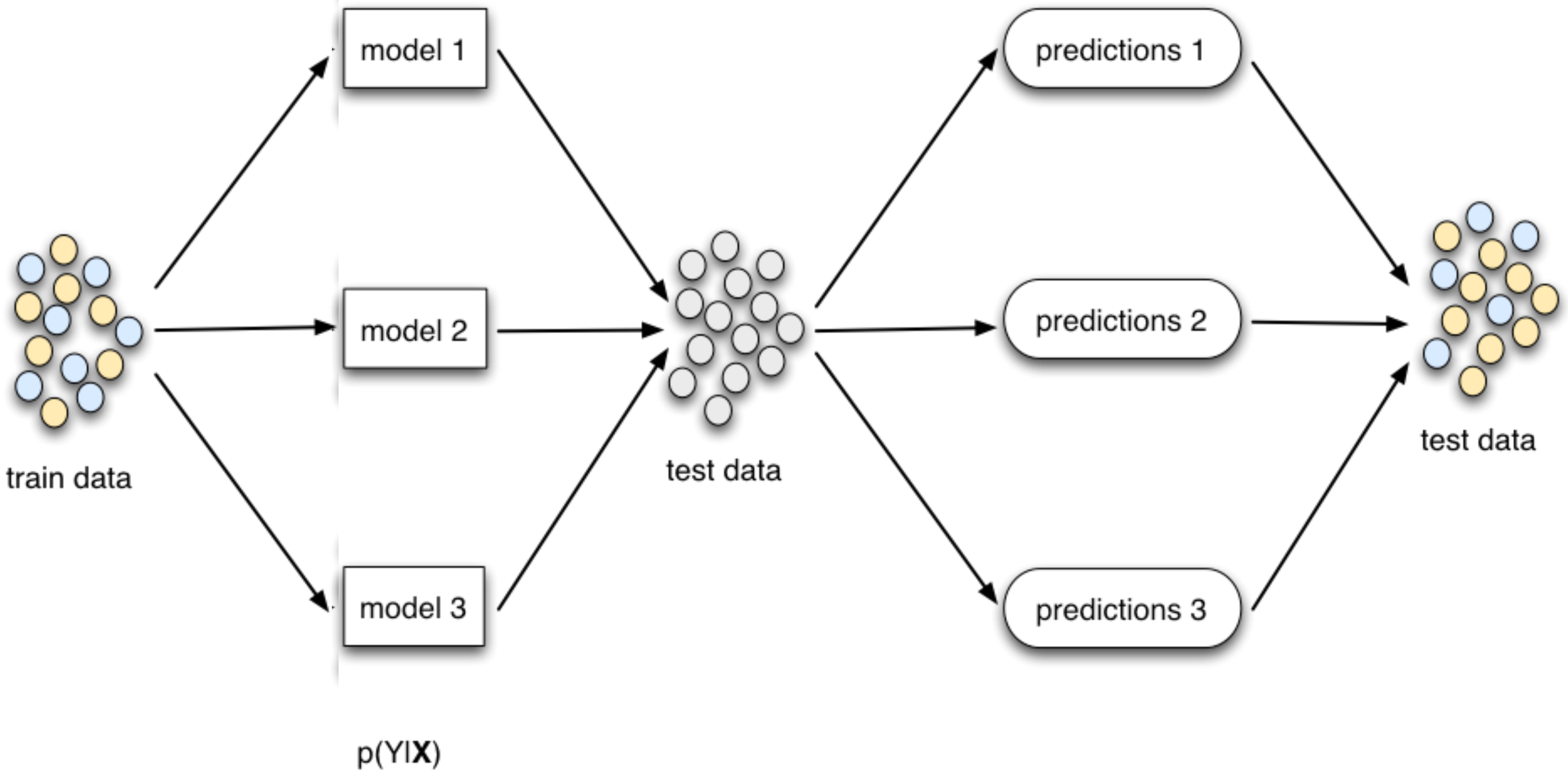
- Suppose there are N *independent* predictors $f_1(x; D), f_2(x; D), \ldots, f_N(x; D)$
- The "blended" predictor is $f(x; D) = \frac{1}{N} \sum_{i=1}^{N} f_i(x; D)$
- At data point (x*, y*), say all individual prediction has a bias of *b* and a variance of σ^2, and we have $\overline{f(x^*)} = E_D[f(x^*; D)] = \frac{1}{N} \sum_{i=1}^{N} \overline{f_i(x^*)}$
- Bias of $f(x^*; D)$: $\overline{f(x^*)} - y^* = \frac{1}{N} \sum_{i=1}^{N} (\overline{f_i(x^*)} - y^*) = b$
- Variance of $f(x^*; D)$: $\frac{1}{N^2} \sum_{i=1}^{N} Var(f_i(x^*; D)) = \frac{\sigma^2}{N}$

Variance decreases!

CONVENTIONAL CLASSIFICATION



ENSEMBLE CLASSIFICATION



BAGGING

- Is it possible to have multiple models of the same type?
- There is only one training data set, where do multiple models of the same type come from?
- Bagging: **B**ootstrap **agg**regat**ing**

BAGGING

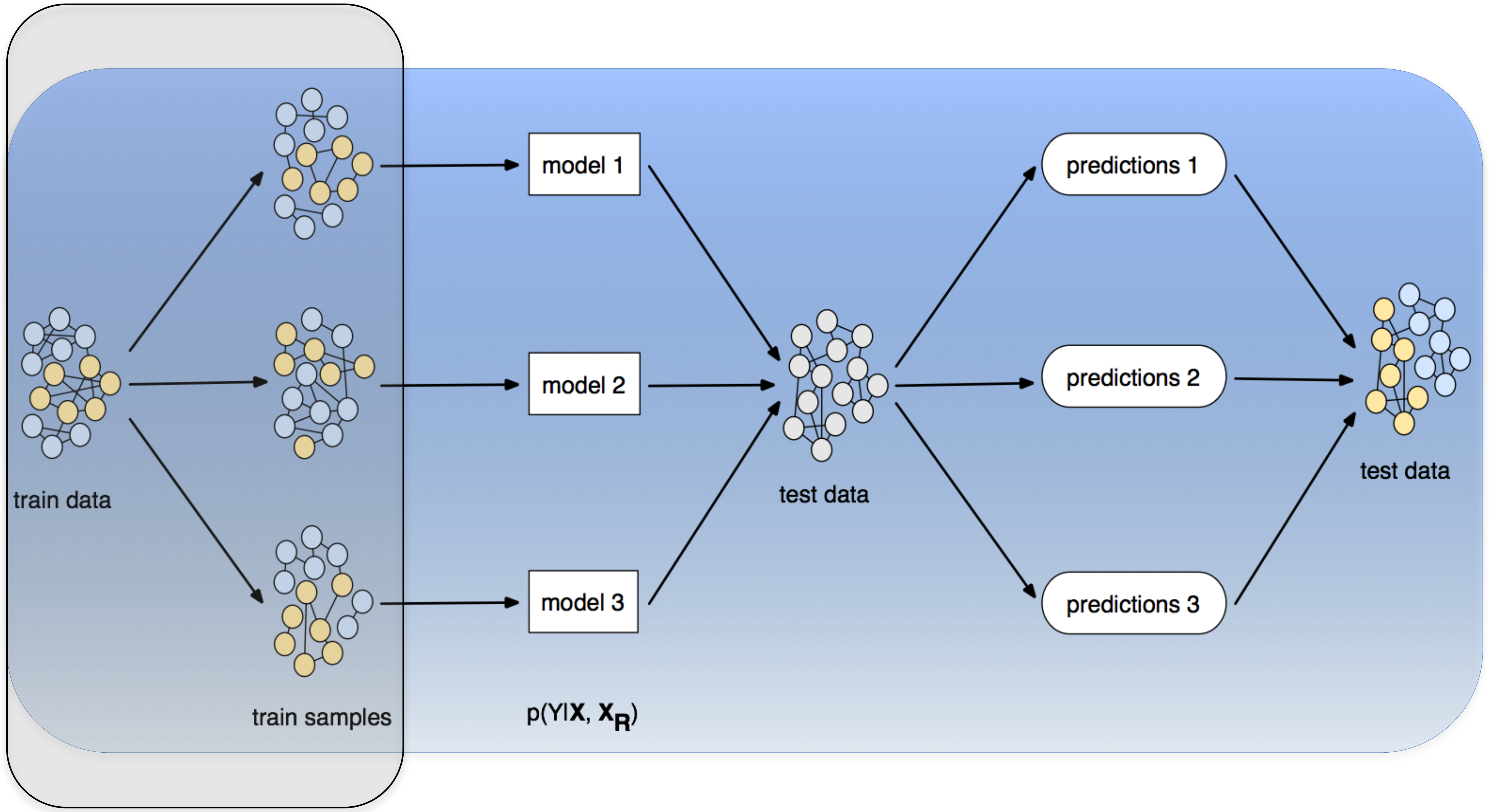
- Given a training data set $D=\{(x_1,y_1),\ldots,(x_N,y_N)\}$
- For m=1:M
  - Obtain a bootstrap sample D_m by drawing N instances ***with replacement*** from D
  - Learn model M_m from D_m
- To classify test instance *t*, apply each model M_m to *t* and use majority predication or average prediction

SAMPLE TO CREATE ALTERED TRAINING DATA

BAGGING

- Main assumption
  - Combining many *unstable* predictors in an ensemble produces a *stable* predictor (i.e., reduces variance)
  - Unstable predictor: small changes in training data produces large changes in the model (e.g., fully-grown trees)
- Models have somewhat uncorrelated errors due to difference in training sets (each bootstrap sample has ~63% of D)
- Model space: non-parametric, can model any function if an appropriate base model is used

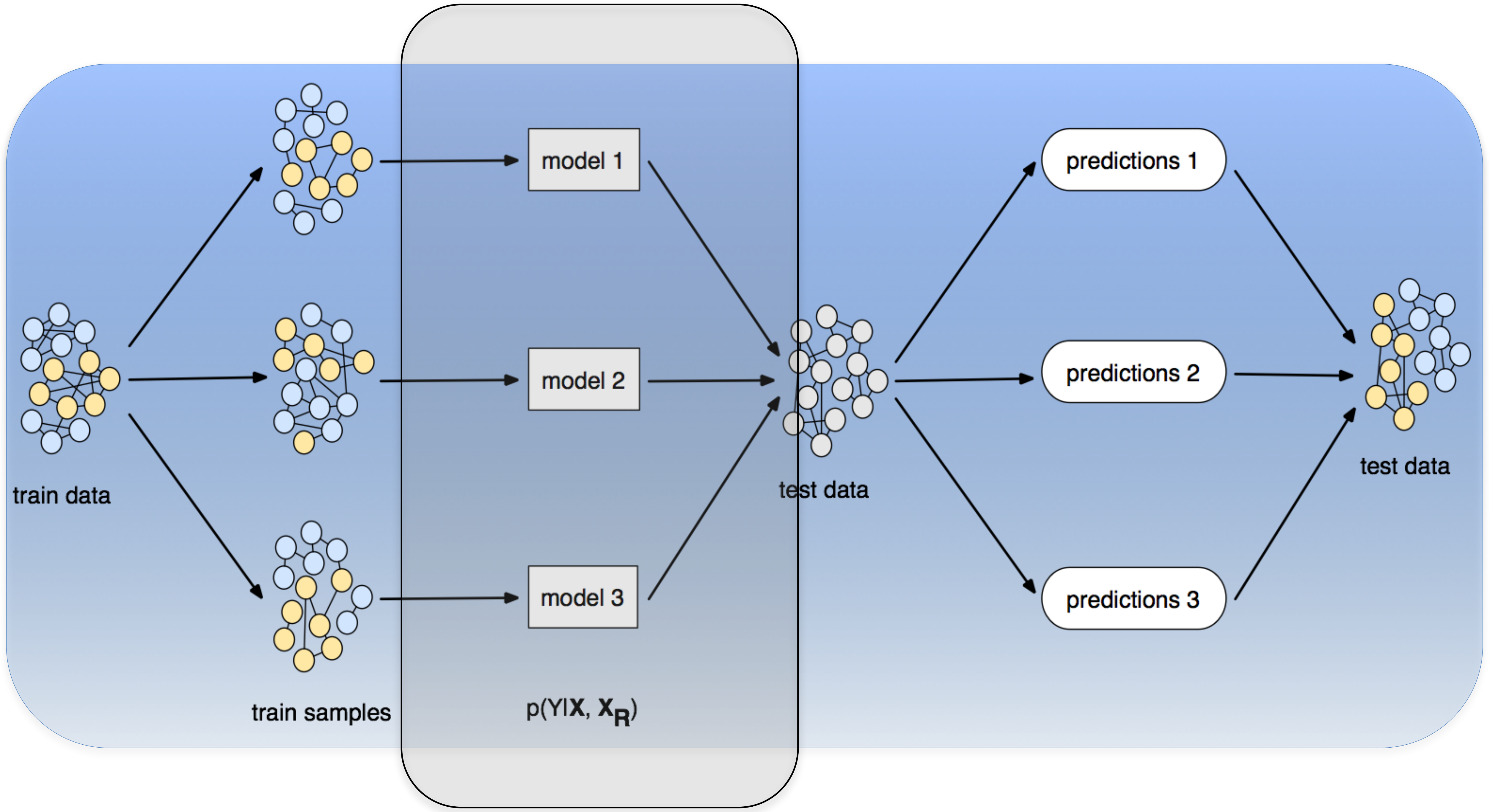
BAGGING



TREATMENT OF INPUT DATA

- sample with replacement

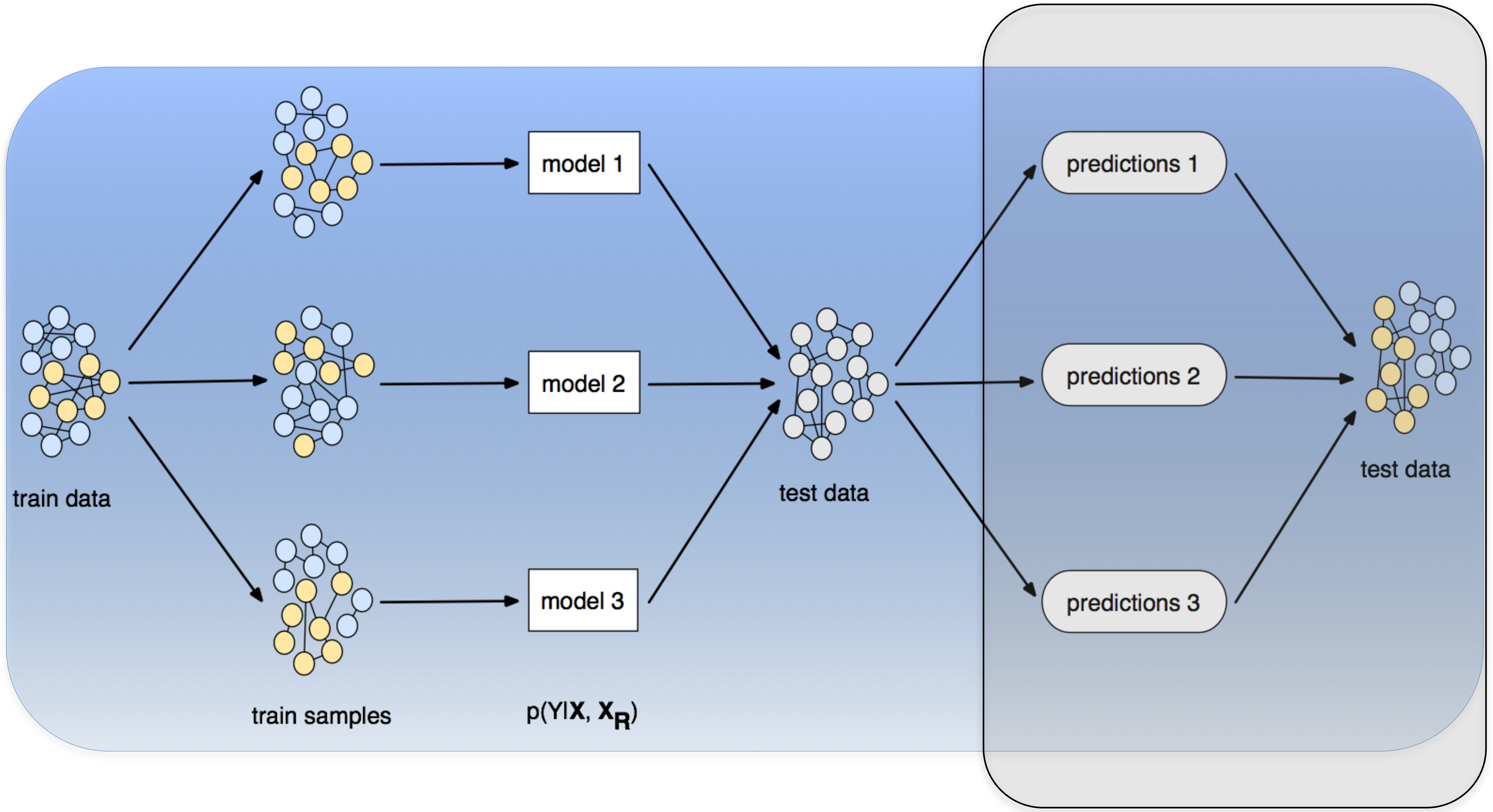
BAGGING



CHOICE OF BASE CLASSIFIER

- unstable predictor (e.g., decision tree)

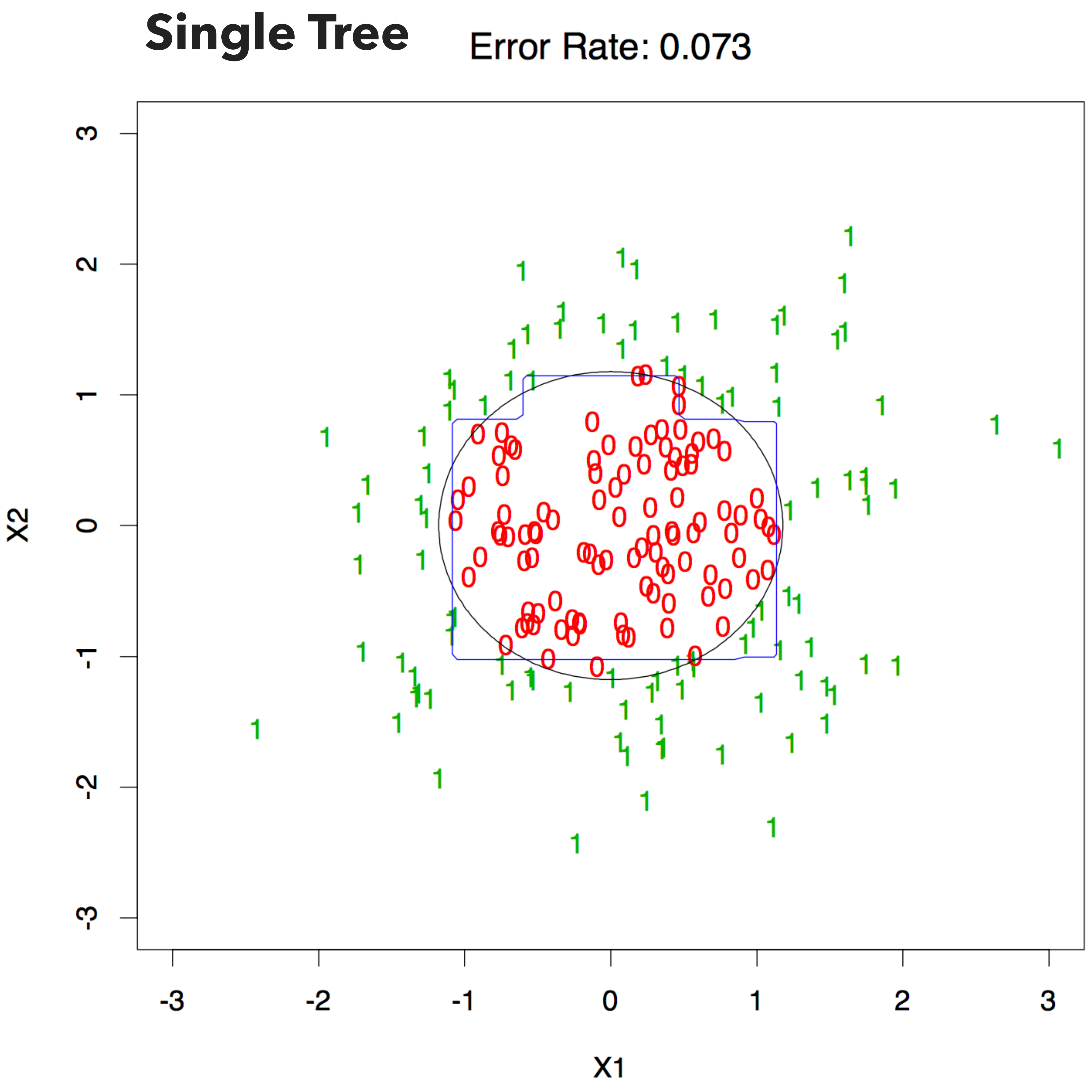
BAGGING

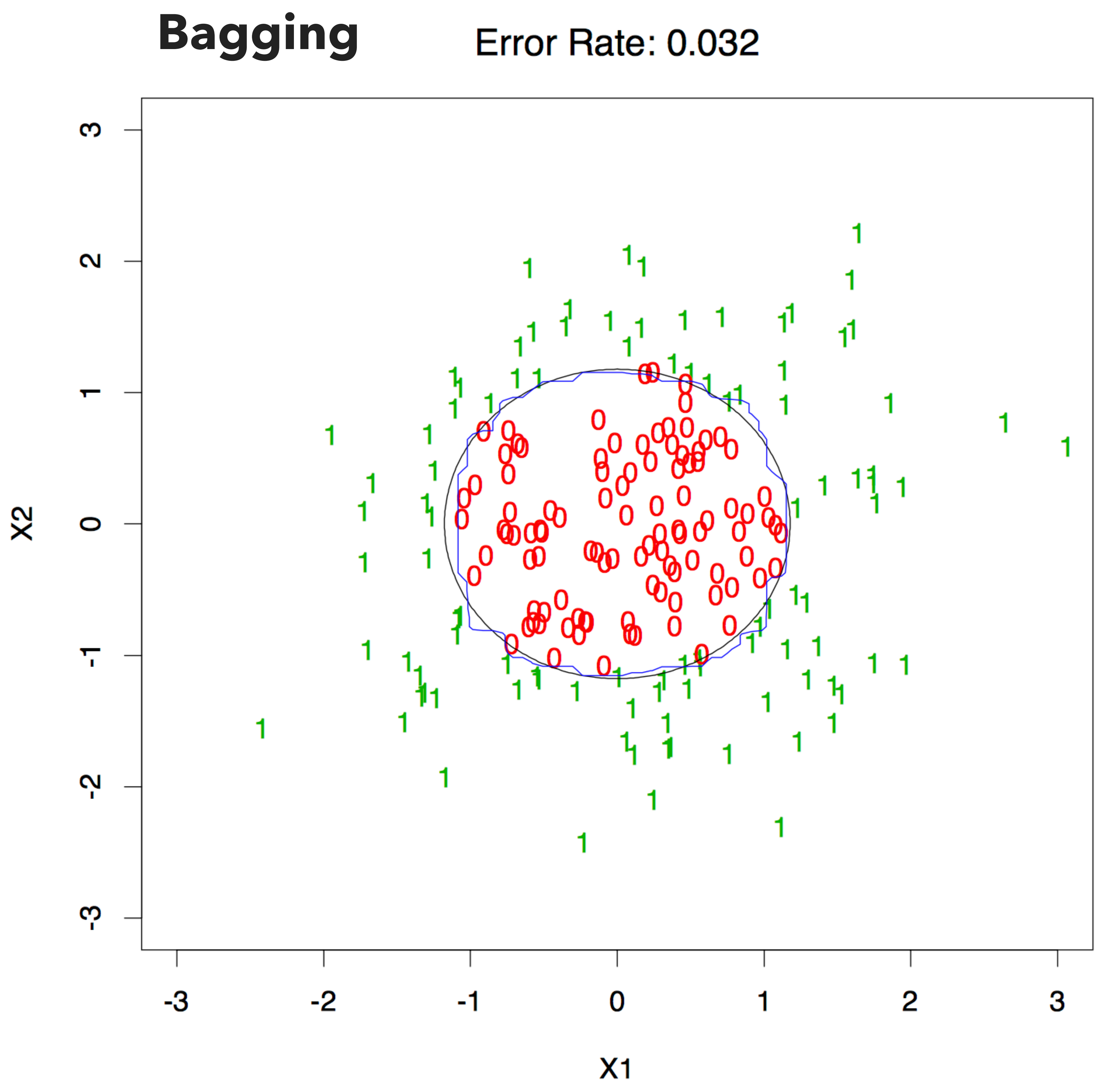


PREDICTION AGGREGATION

- averaging / majority voting

DECISION BOUNDARY WITH SINGLE TREE VS. BAGGING





Source: T. Hastie, Stanford

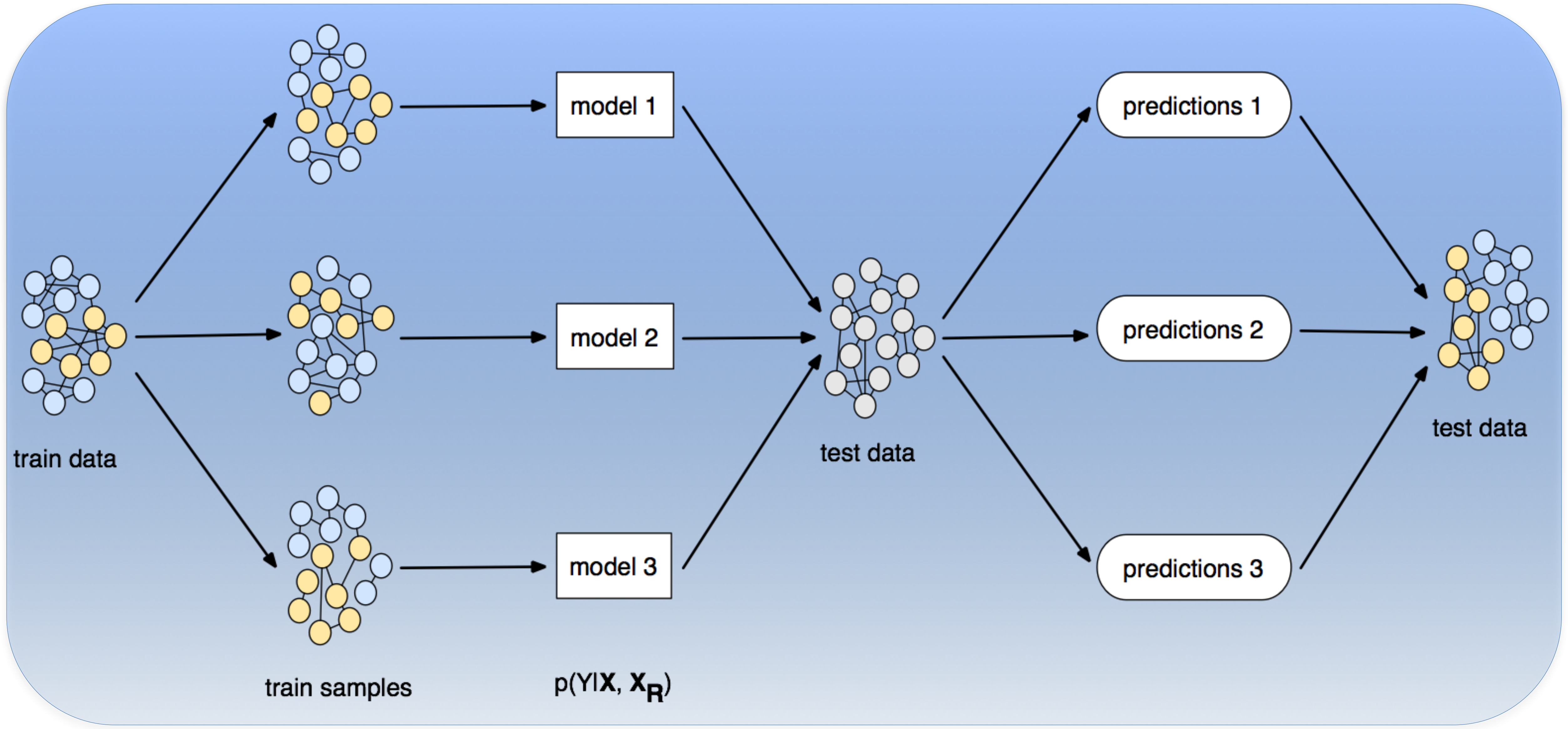
LIMITATIONS OF BAGGING

- A bag of M trees typically will lead to a reduction in variance that is smaller than 1/M
- Because the M models are correlated to some degree…
- Solution: further decrease the correlation between models…

RANDOM FORESTS

- Random forests are a variant that aims to improve on bagged decision trees by reducing the correlation between the models
  - Each tree is learned from a bootstrap sample (same as before)
  - For each tree split, a random sample of *k* features is drawn first, and **only** those features are considered when selecting the best feature to split on (typically $k=\sqrt{p}$ or $k=\log p$, p is the total number of features)

RANDOM FORESTS



TREATMENT OF INPUT DATA

- sampling with replacement

CHOICE OF BASE CLASSIFIER

- decision tree (limited attributes are considered at each node)

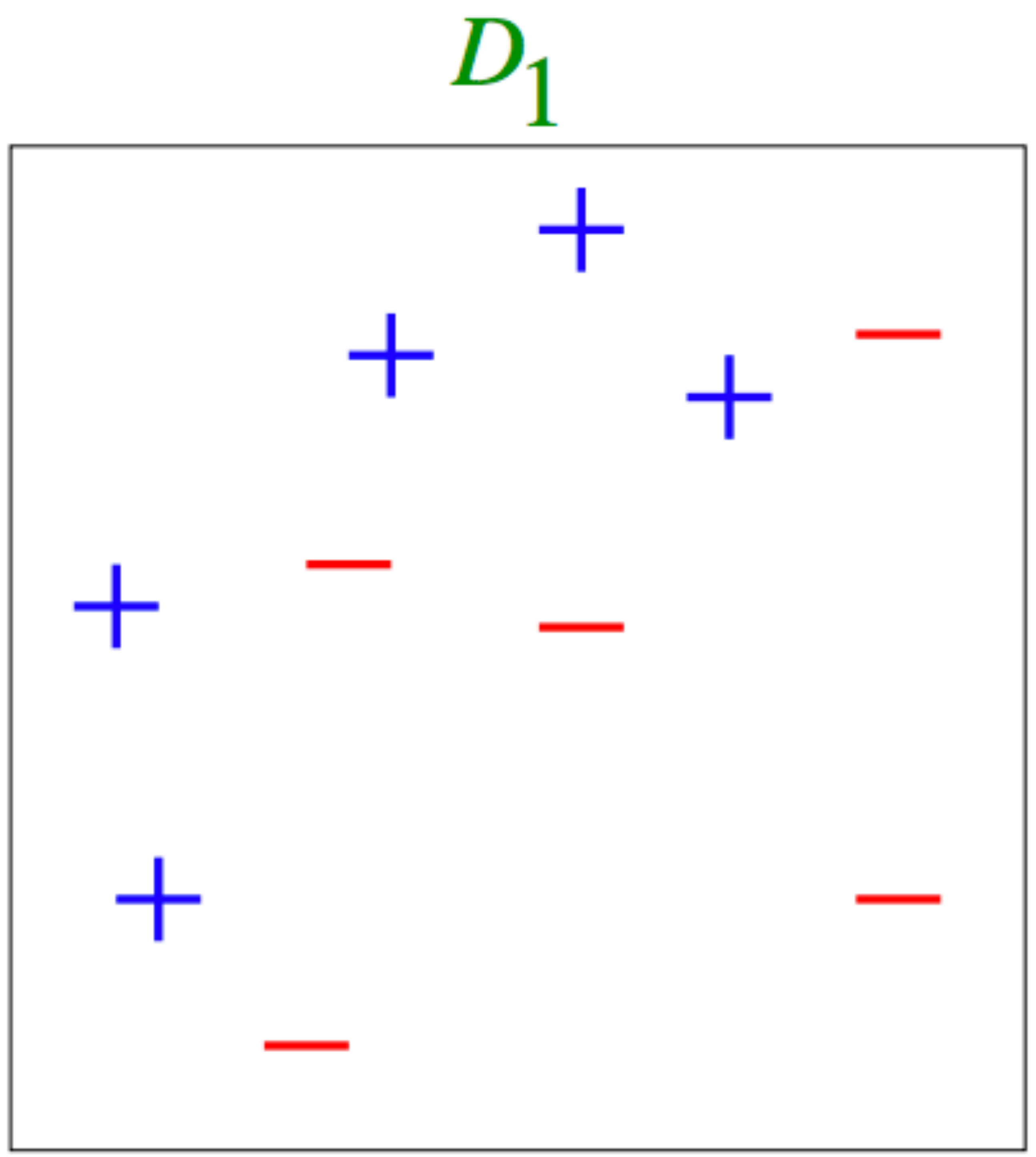
PREDICTION AGGREGATION

- averaging/majority voting

BOOSTING

- Bagging and random forests share the same idea of combining multiple models that are trained on bootstrapped samples of the training data
  - Mimic learning the model from different training data
  - Each model has an equal amount of say (i.e., equal weights) in influencing the aggregated prediction
- Boosting
  - Combine multiple "complementary" models
  - Aggregate model predictions by considering how accurately each model can predict

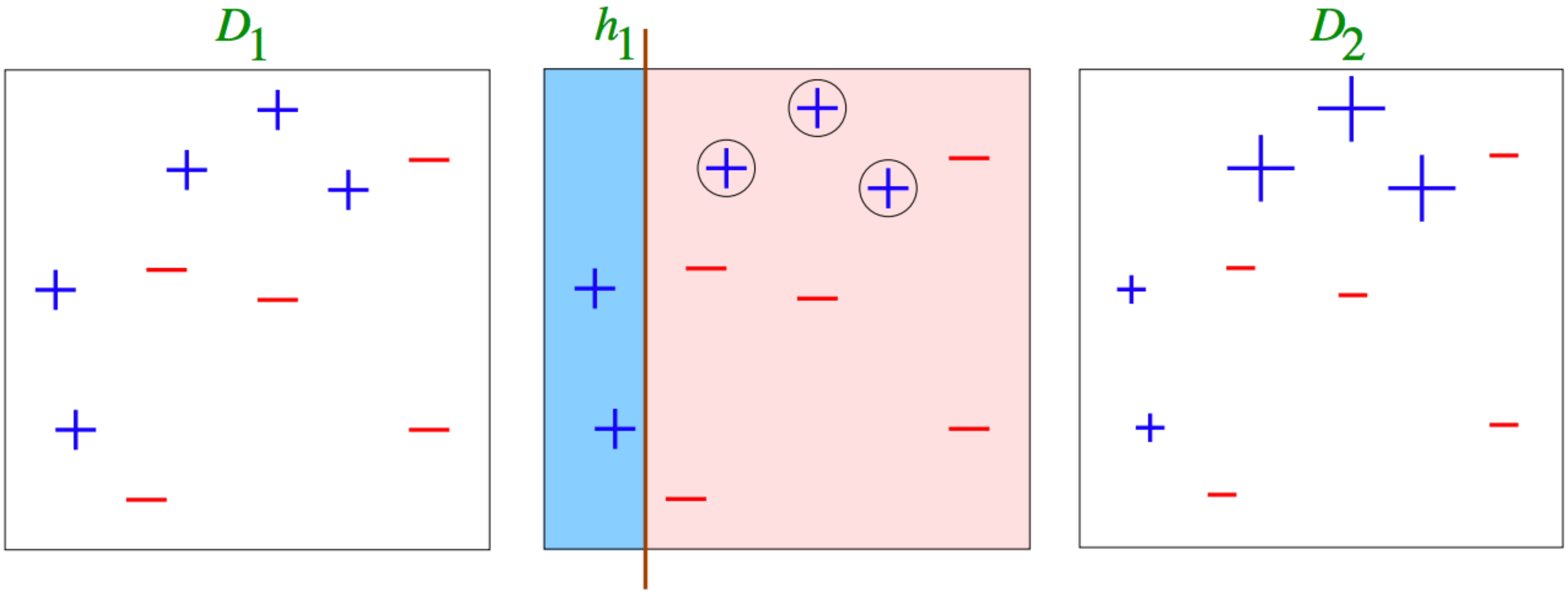
BOOSTING EXAMPLE



Model: Decision stump

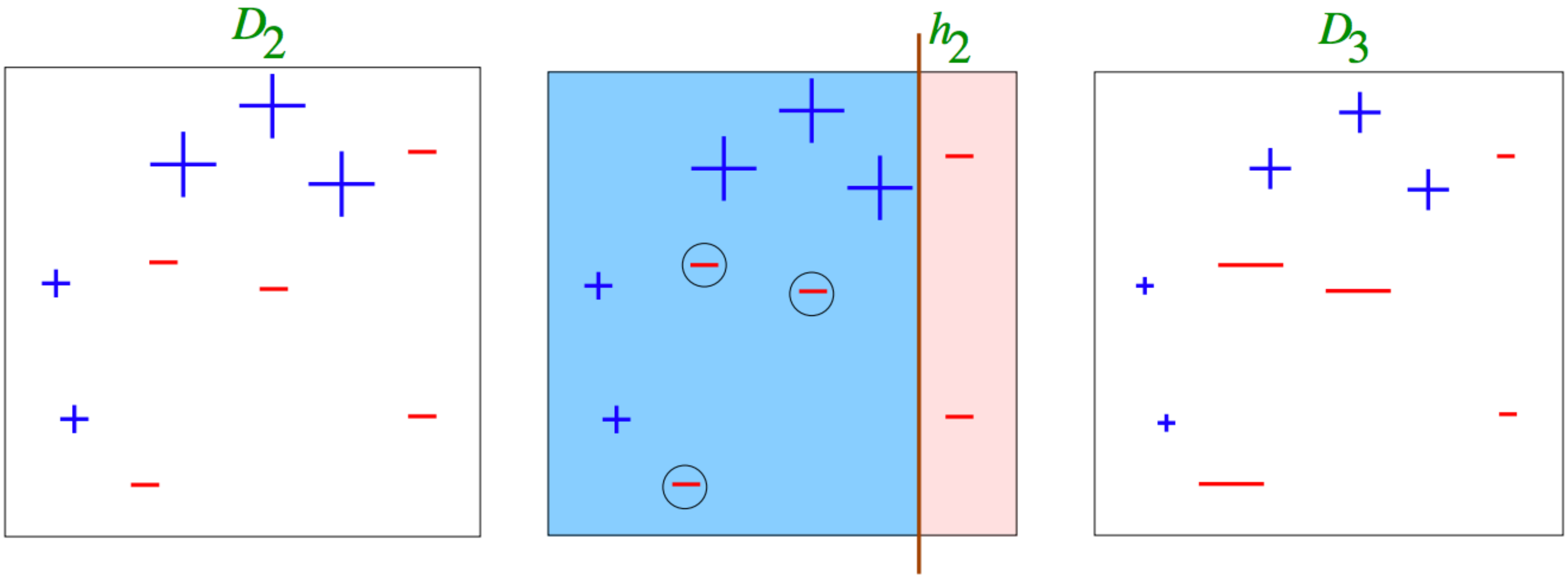
If $x_i > c$, then "+"; otherwise "-"

BOOSTING EXAMPLE: ROUND 1

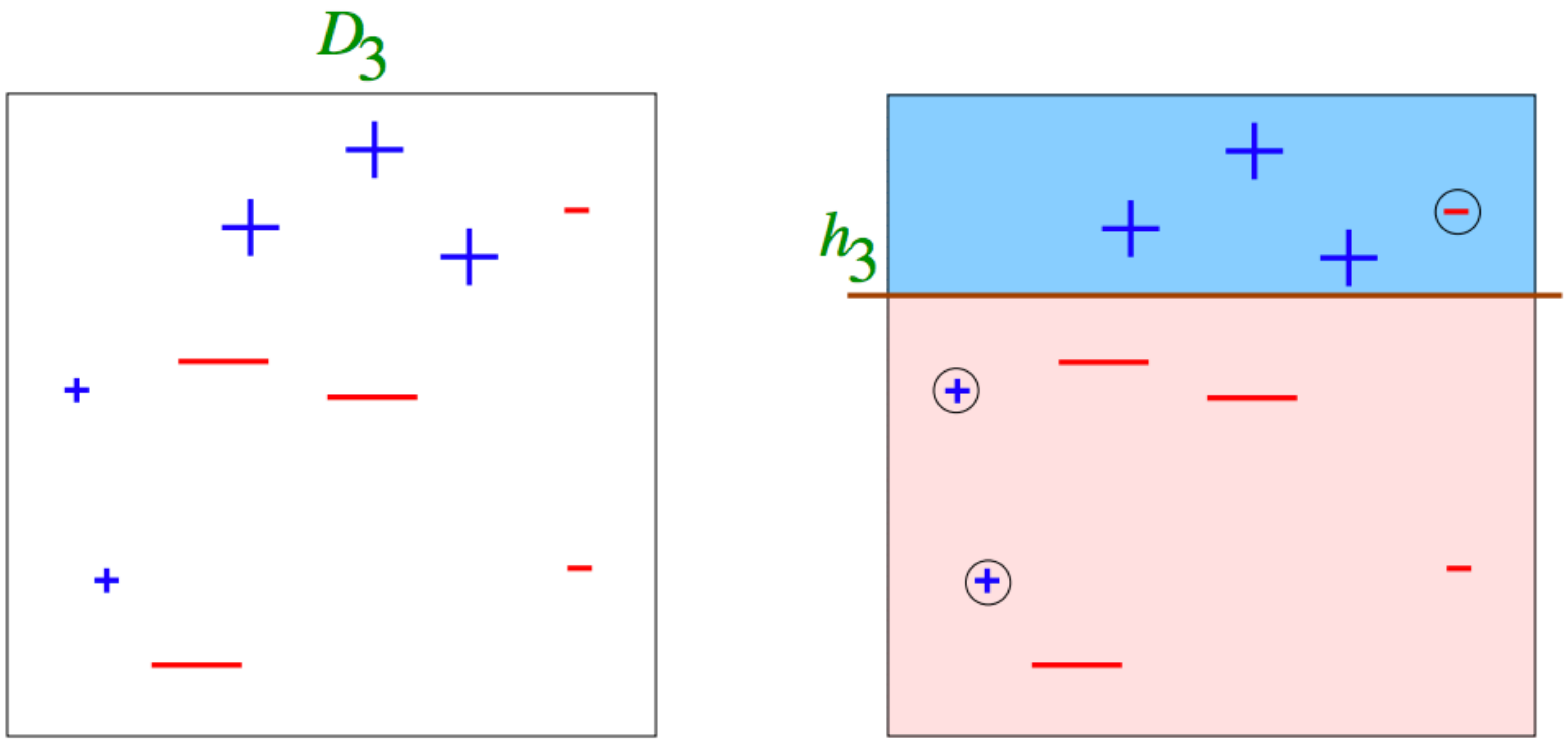


Construct "complementary" models? Re-weighting!

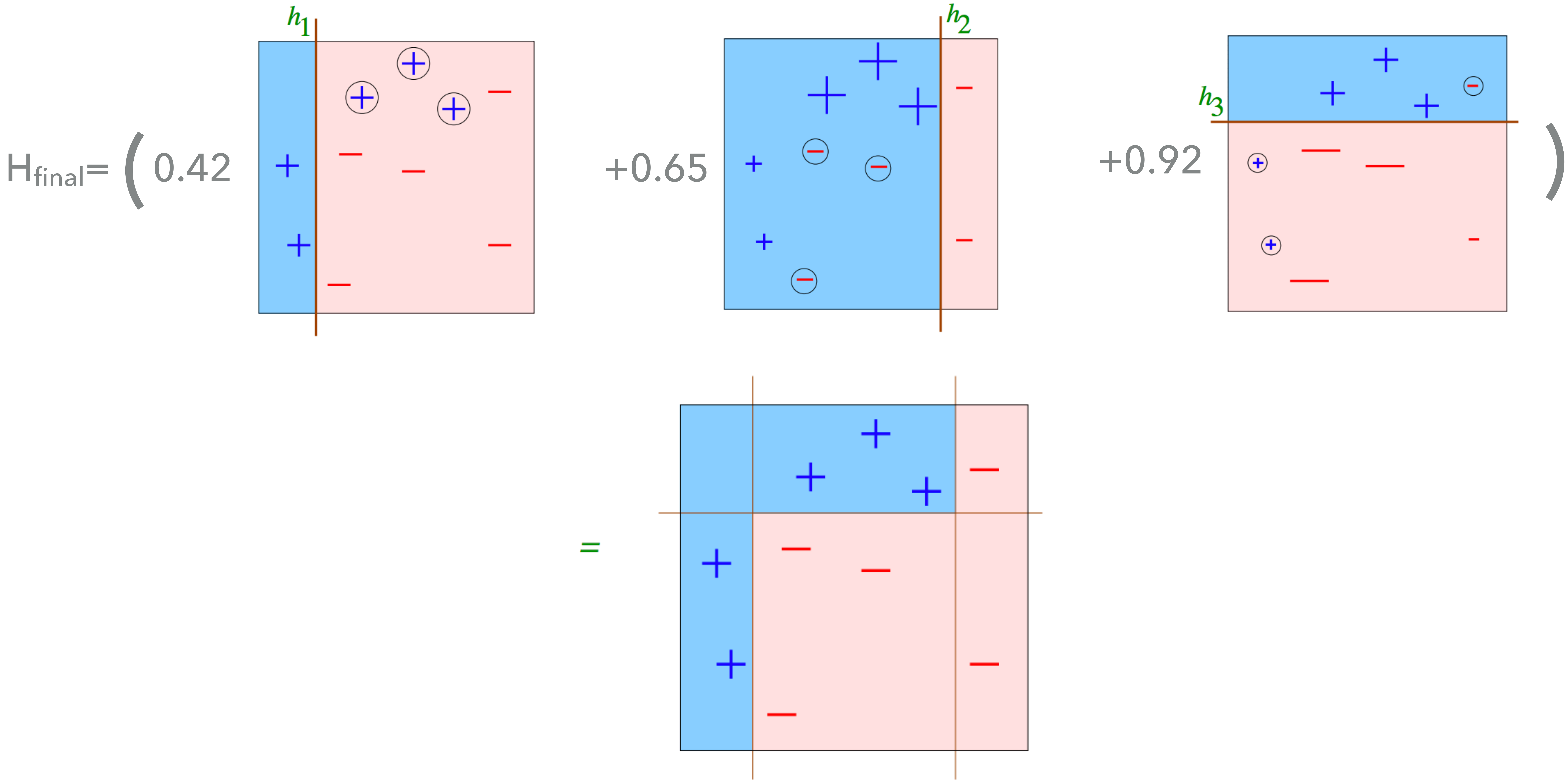
BOOSTING EXAMPLE: ROUND 2



BOOSTING EXAMPLE: ROUND 3



BOOSTING EXAMPLE: AGGREGATING



ADABOOST

- Given N training examples $(x_1, y_1), \ldots, (x_N, y_N)$, assign every example in with an equal weight $D_1(i)=1/N$
- For t=1:T
  - Learn model $h_t(x)$ to minimize the weighted error: $\epsilon_t = \text{Pr}_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i=1}^{N} D_t(i) \mathbb{I}(h_t(x_i) \neq y_i)$
  - Set the weight of this model: $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$
  - Update training example weights: up-weight the examples that are incorrectly classified and downright examples that are correctly classified: $D_{t+1}(i) = \frac{D_t(i)exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ where $Z_t = \sum_{i=1}^{N} D_t(i)exp(-\alpha_t y_i h_t(x_i))$ is a normalization factor
- To classify new test instance x', apply each model $h_t(x)$ to x' and take weighted vote of predictions

$$H(x') = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t(x'))$$