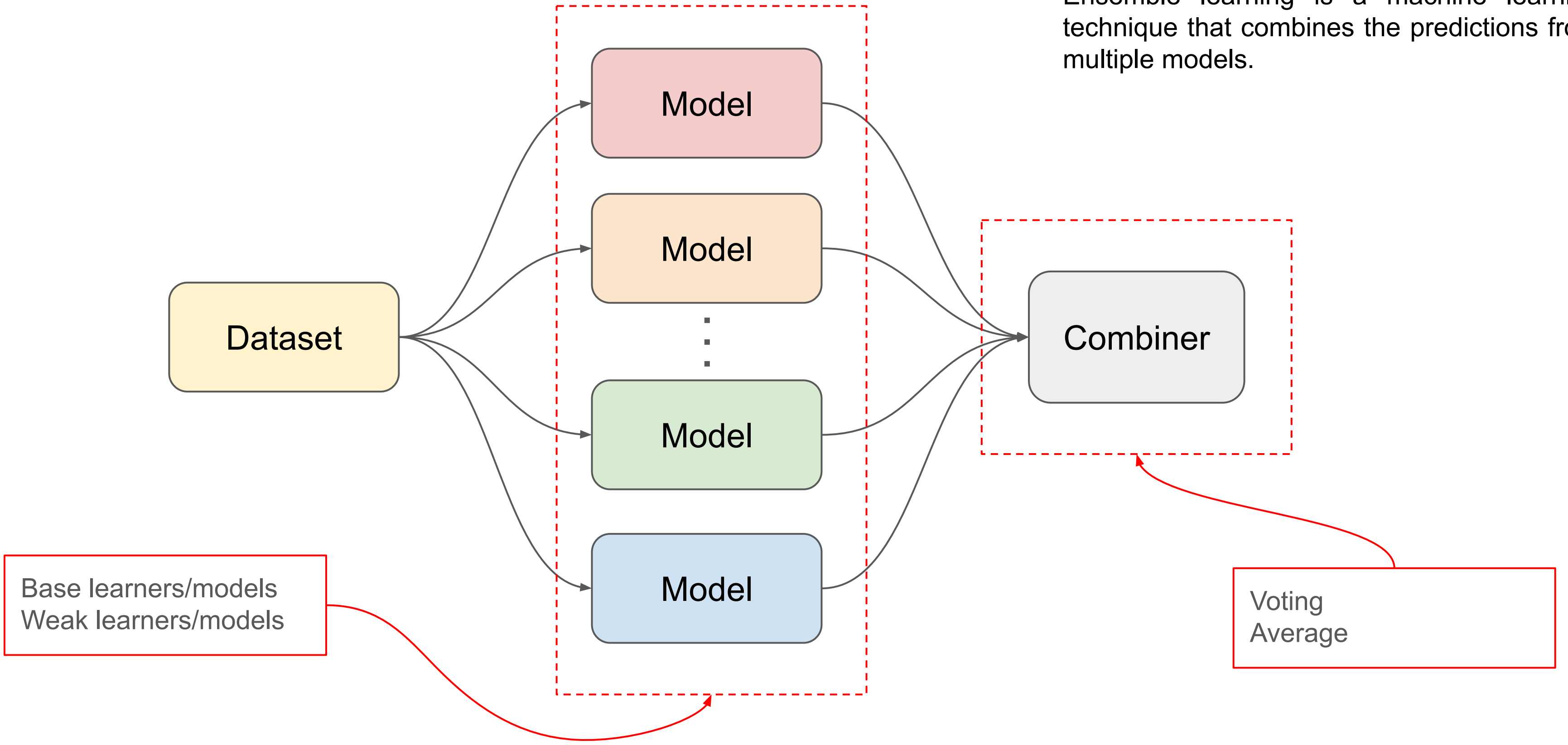
Ensemble Learning

MACHINE LEARNING

Pakarat Musikawan

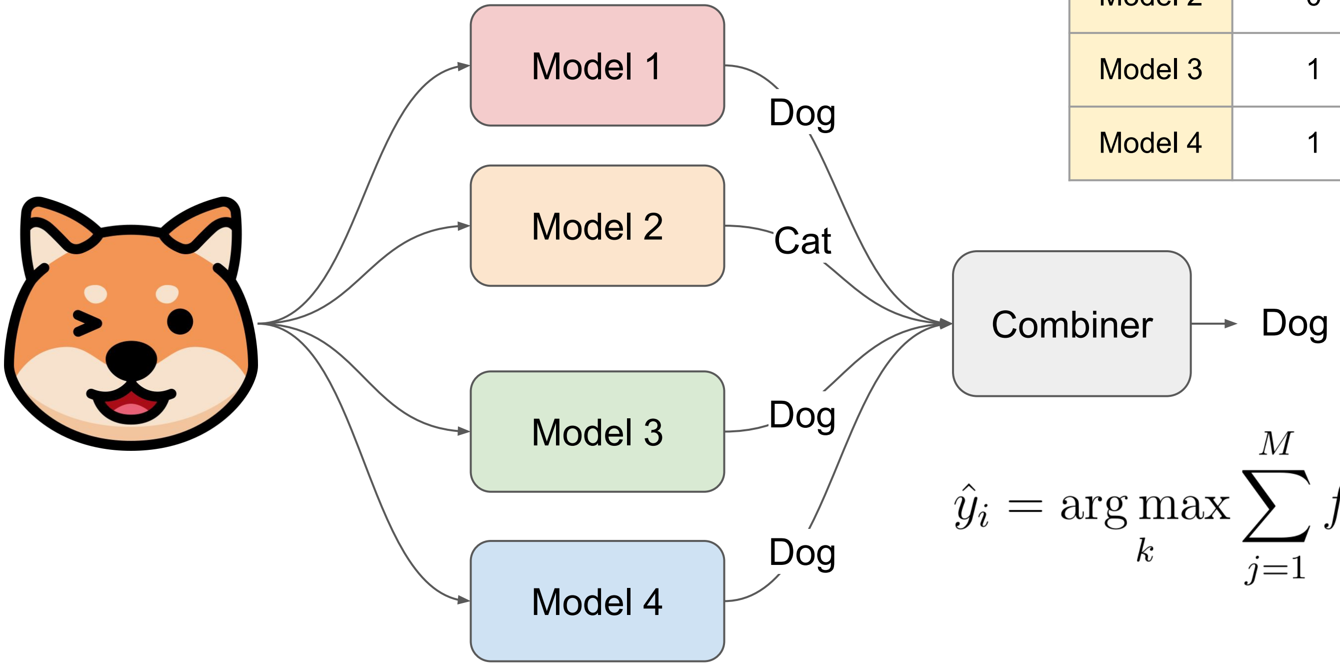
Ensemble Learning

Ensemble learning is a machine learning technique that combines the predictions from multiple models.



Ensemble Learning

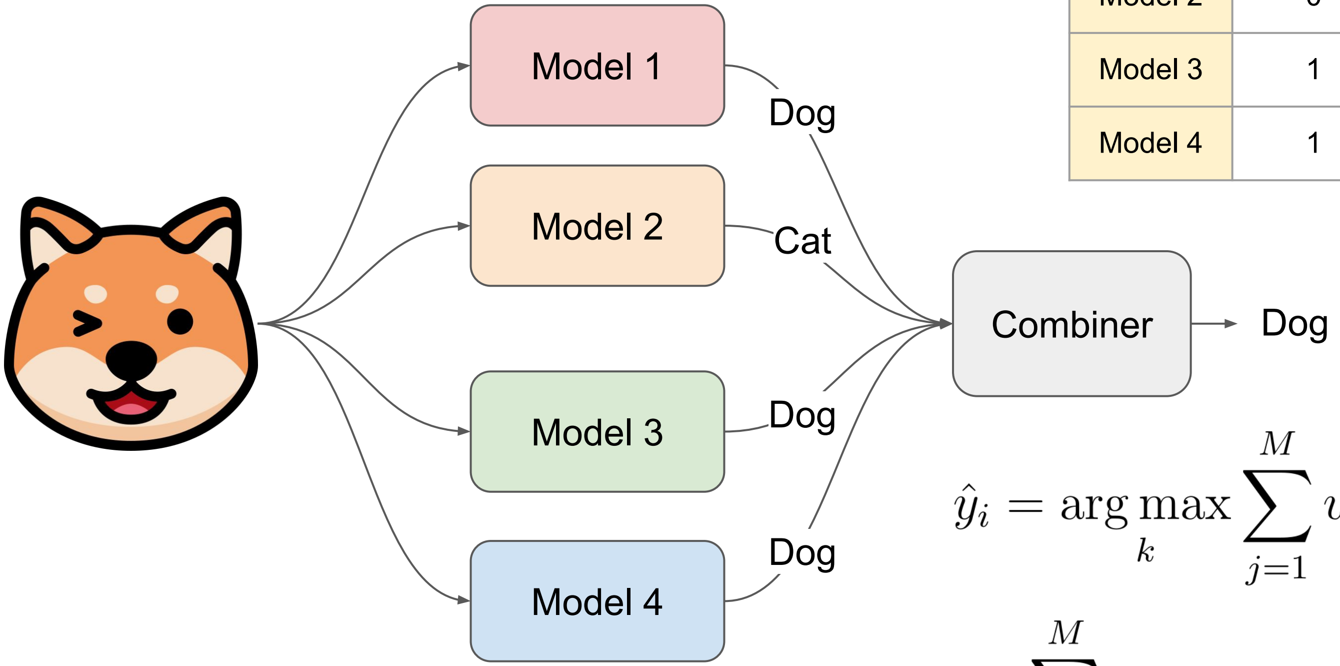
	Dog	Cat
Model 1	1	0
Model 2	0	1
Model 3	1	0
Model 4	1	0



$$\hat{y}_i = \arg\max_k \sum_{j=1}^{M} f_j(\boldsymbol{x}_i)$$

Ensemble Learning

	Dog	Cat
Model 1	1	0
Model 2	0	1
Model 3	1	0
Model 4	1	0

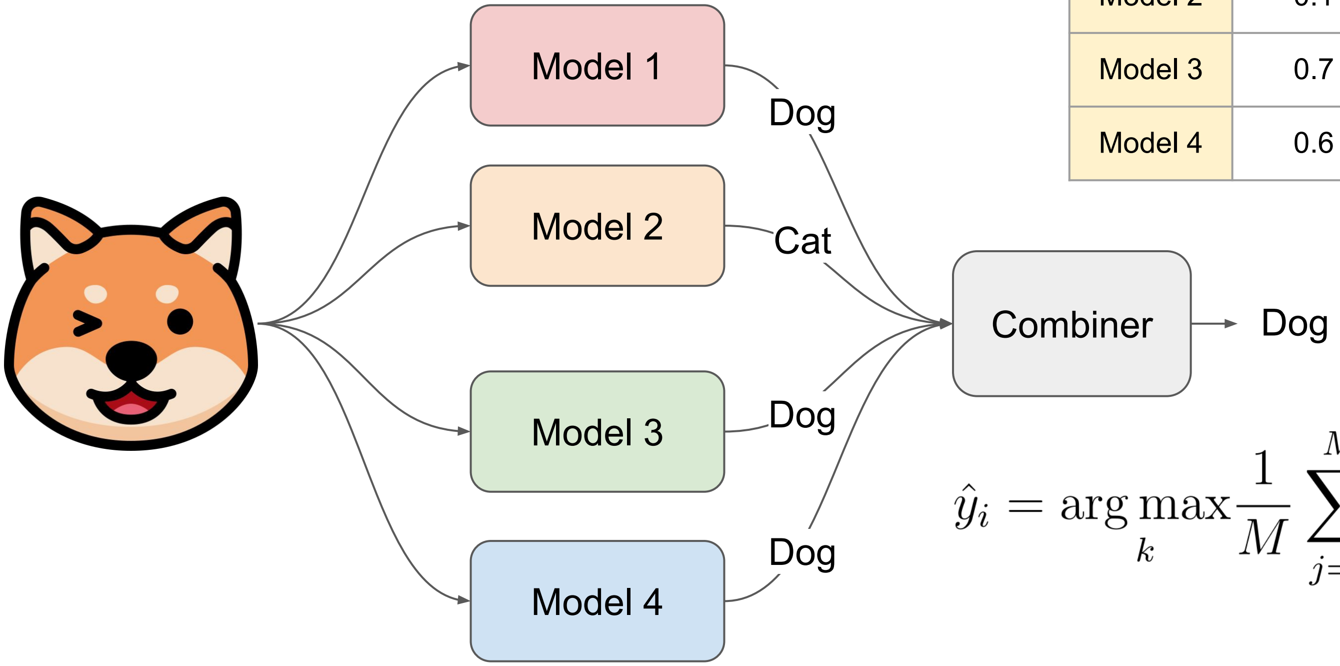


$$\hat{y}_i = \arg\max_k \sum_{j=1}^{M} w_j f_j(\boldsymbol{x}_i)$$

$$\text{s.t.} \sum_{j=1}^{M} w_j = 1$$

Ensemble Learning

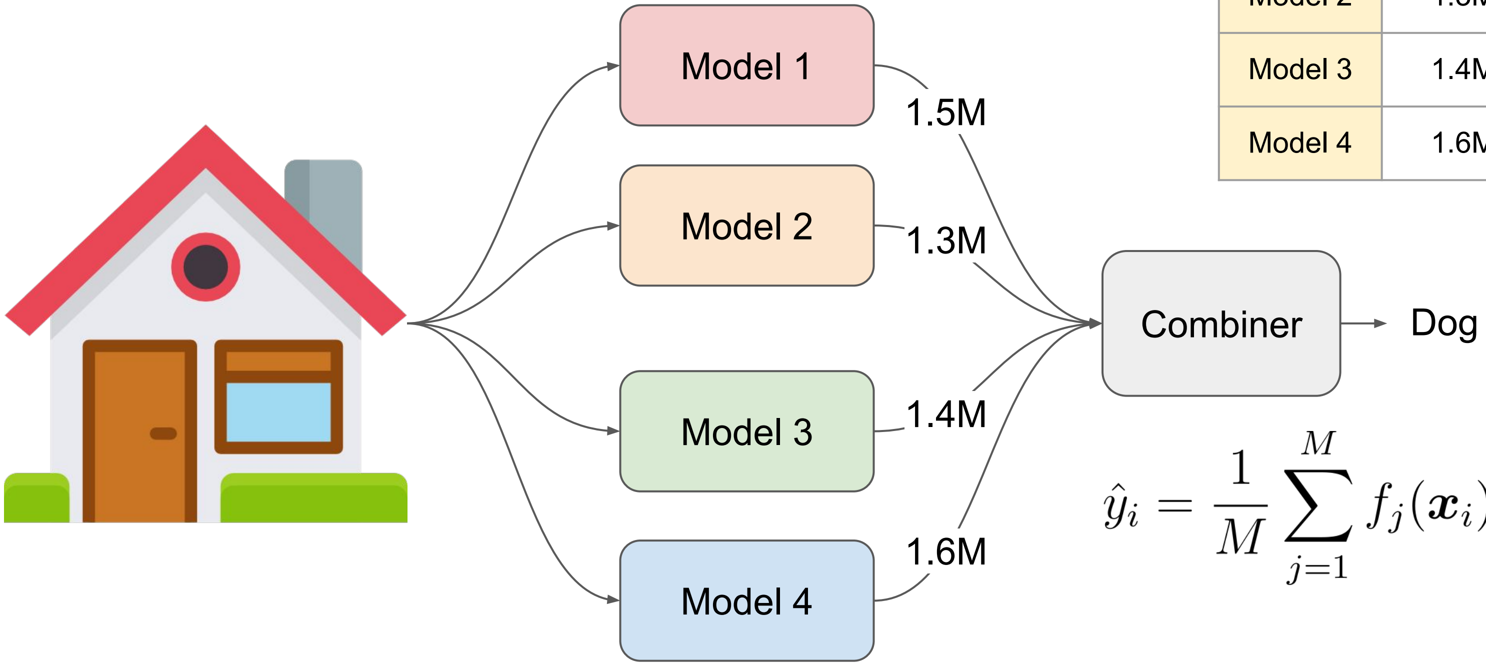
	Dog	Cat
Model 1	0.8	0.2
Model 2	0.1	0.9
Model 3	0.7	0.3
Model 4	0.6	0.4



$$\hat{y}_i = \arg\max_k \frac{1}{M} \sum_{j=1}^{M} f_j(\boldsymbol{x}_i)$$

Ensemble Learning

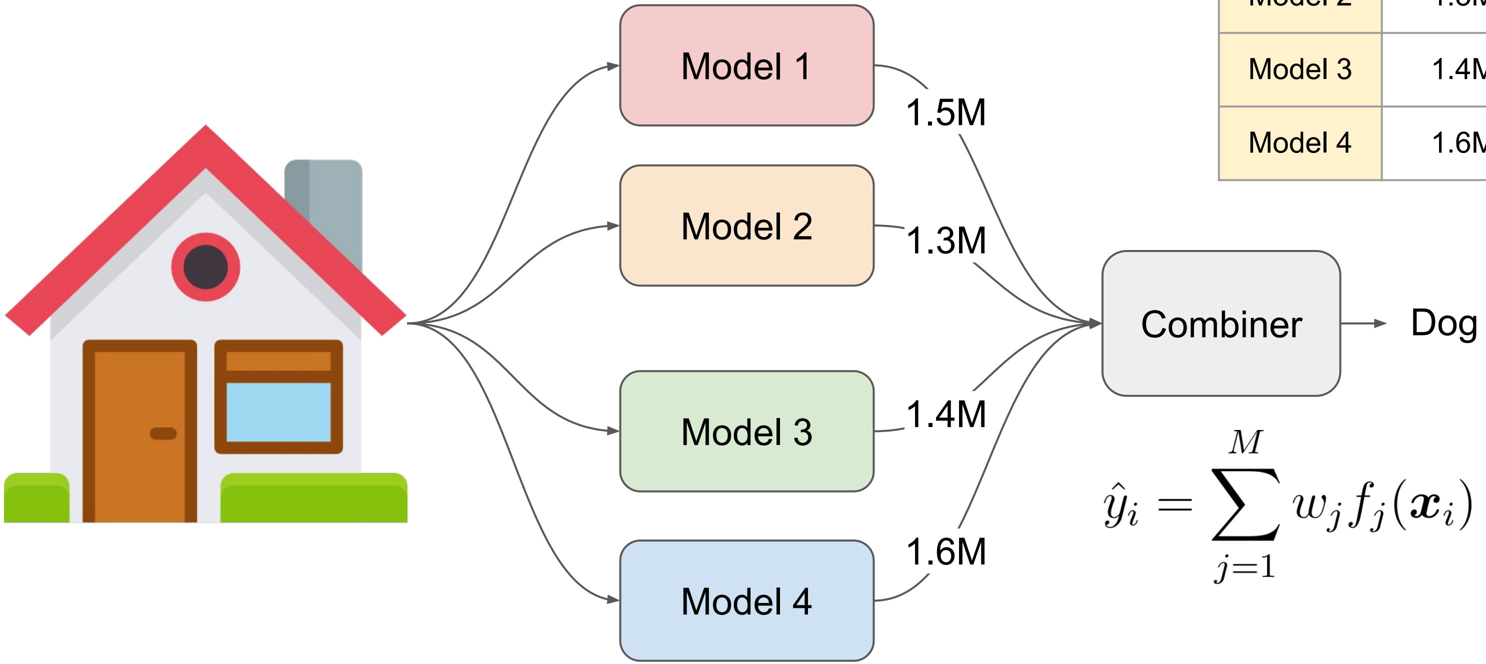
	Price
Model 1	1.5M
Model 2	1.3M
Model 3	1.4M
Model 4	1.6M



$$\hat{y}_i = \frac{1}{M} \sum_{j=1}^{M} f_j(\boldsymbol{x}_i)$$

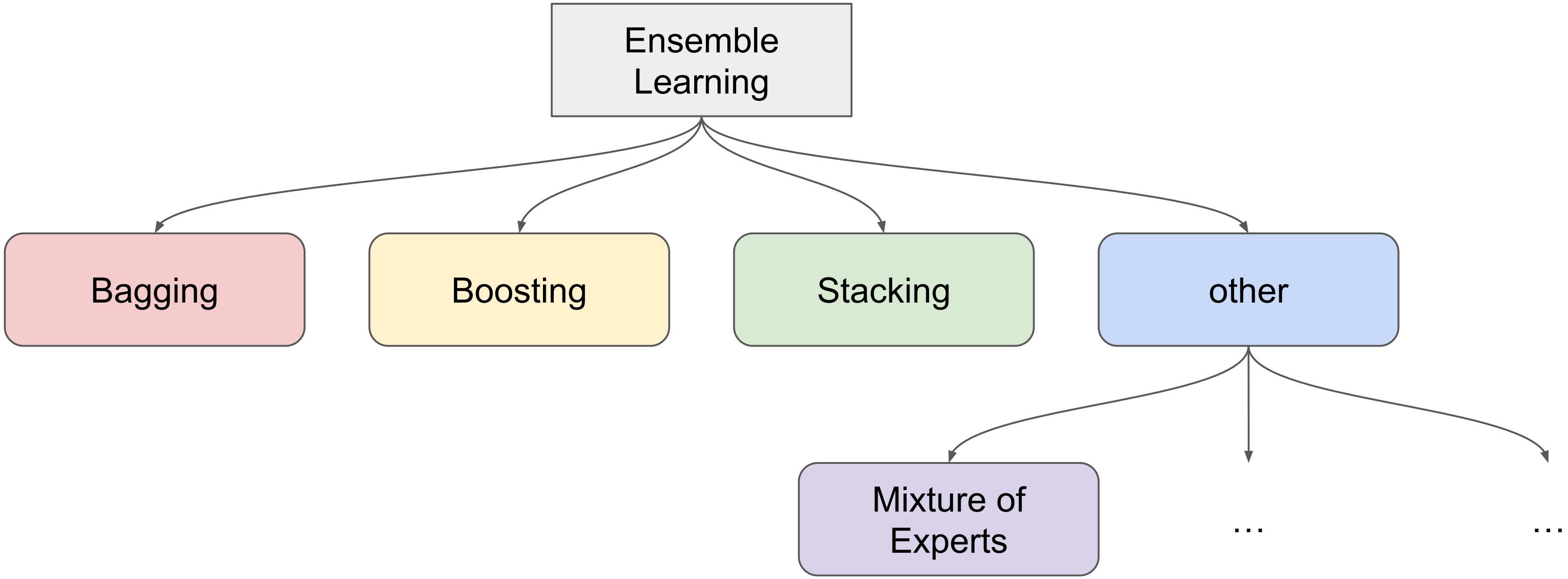
Ensemble Learning

	Price
Model 1	1.5M
Model 2	1.3M
Model 3	1.4M
Model 4	1.6M



$$\hat{y}_i = \sum_{j=1}^{M} w_j f_j(\boldsymbol{x}_i)$$

Types of Ensemble Learning



Ensemble — Bagging

- Bagging: **B**ootstrap **AGG**regat**ING**
- **Bootstraping**: a simple random sampling technique with replacement
- Bagging learns with bootstrap samples of the same size as the original data set
- Each ML-model is created for each bootstrap (Trees = Random forest)
- The predictive output of each ML-model is blended (via voting, averaging, or etc.) for the final decision

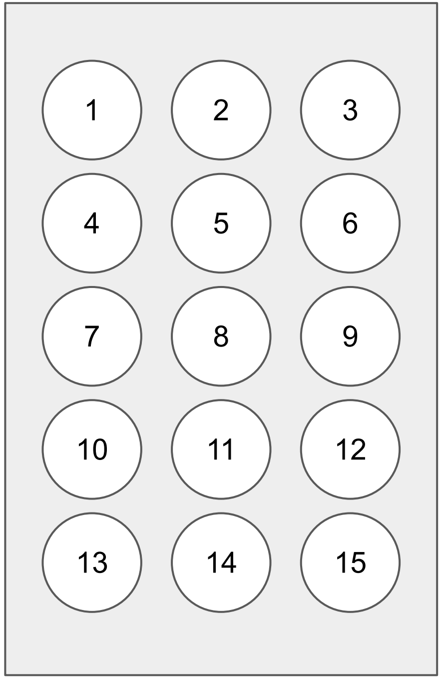
Why do we need a bootstrapping method?

$$\{3,3,3,3,3\} = \frac{3+3+3+3+3}{5} = 3$$

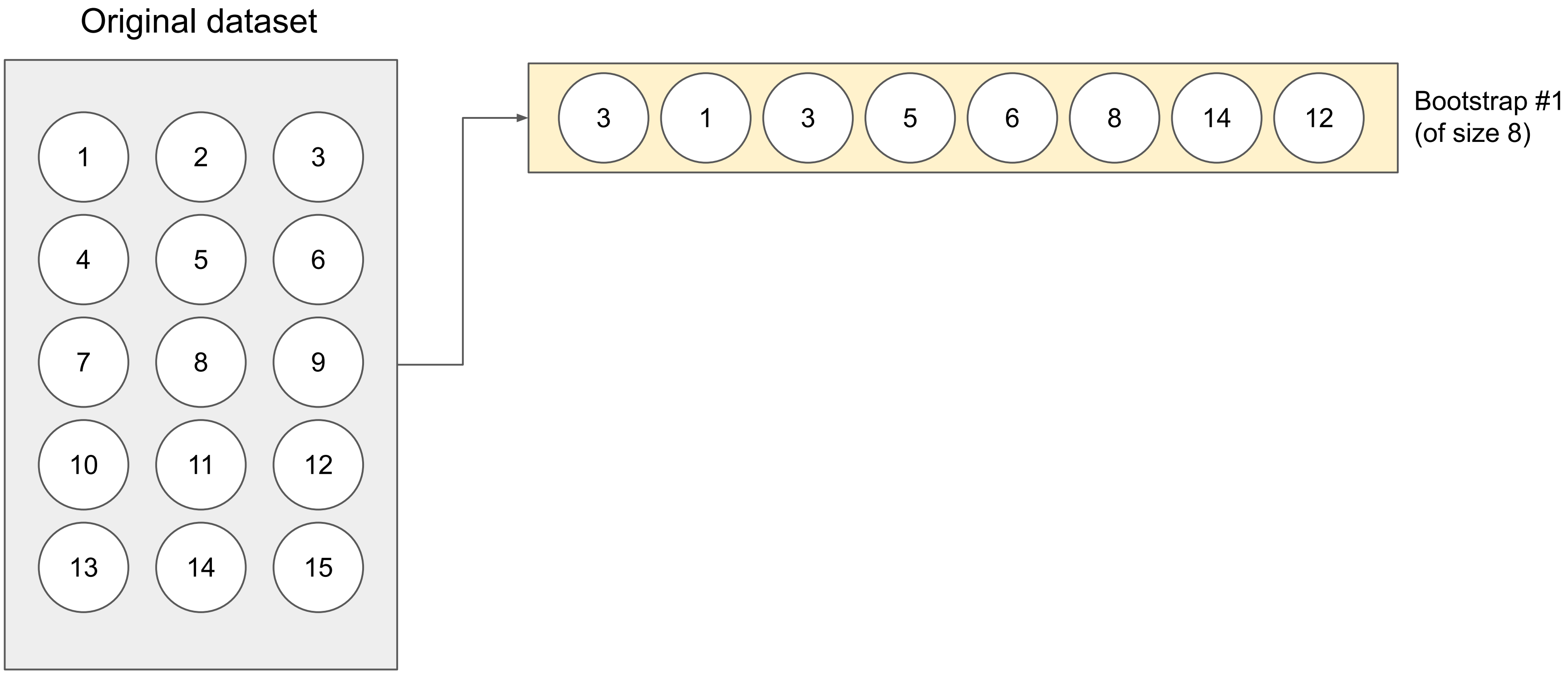
$$\{3,4,4,3,4\} = \frac{3+4+4+3+4}{5} = 3.6$$

Ensemble — Bootstrapping

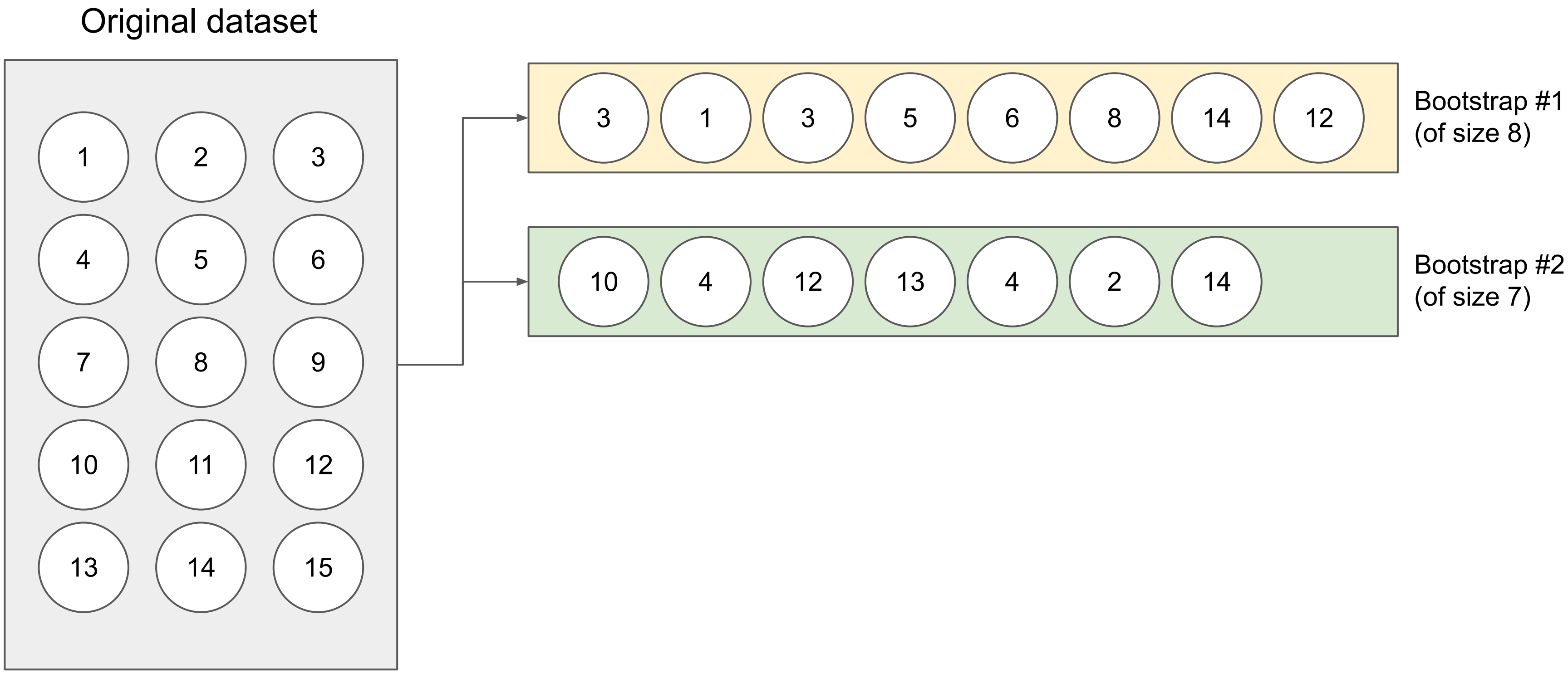
Original dataset



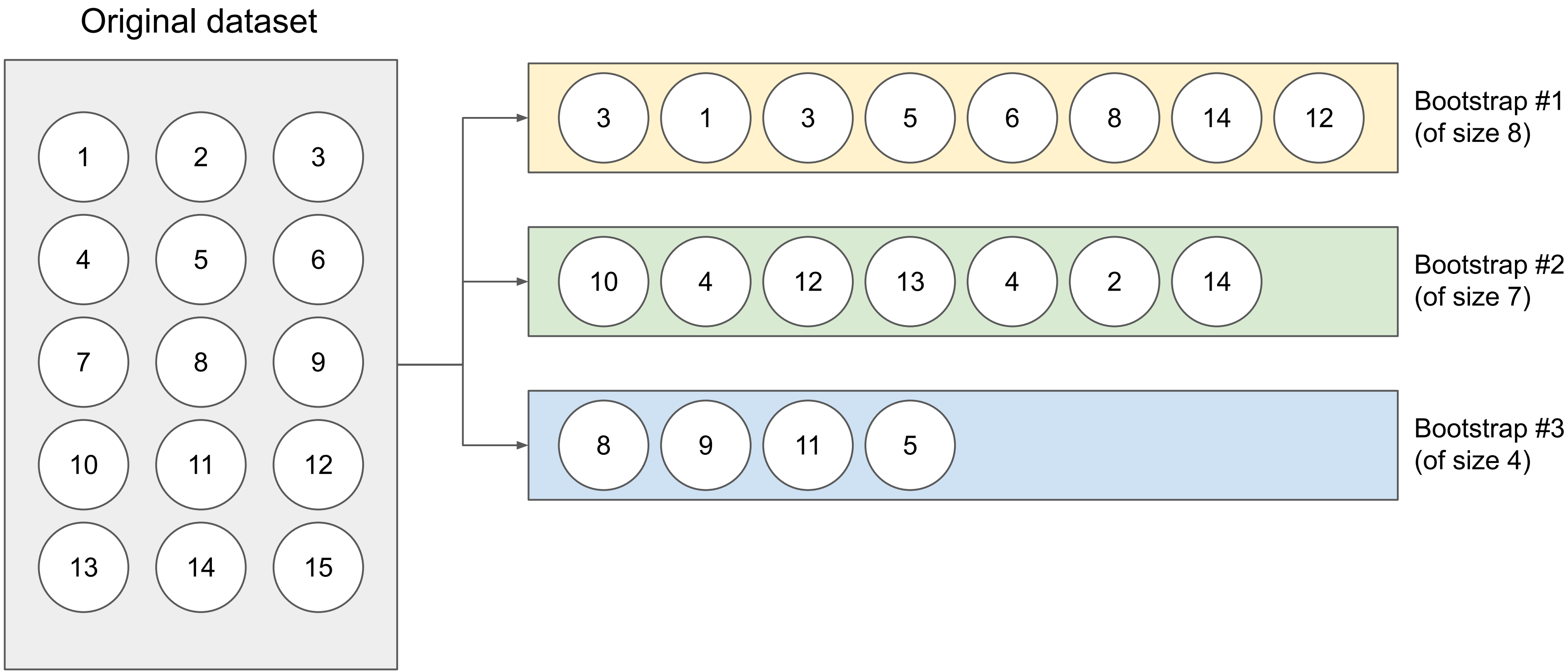
Ensemble — Bootstrapping



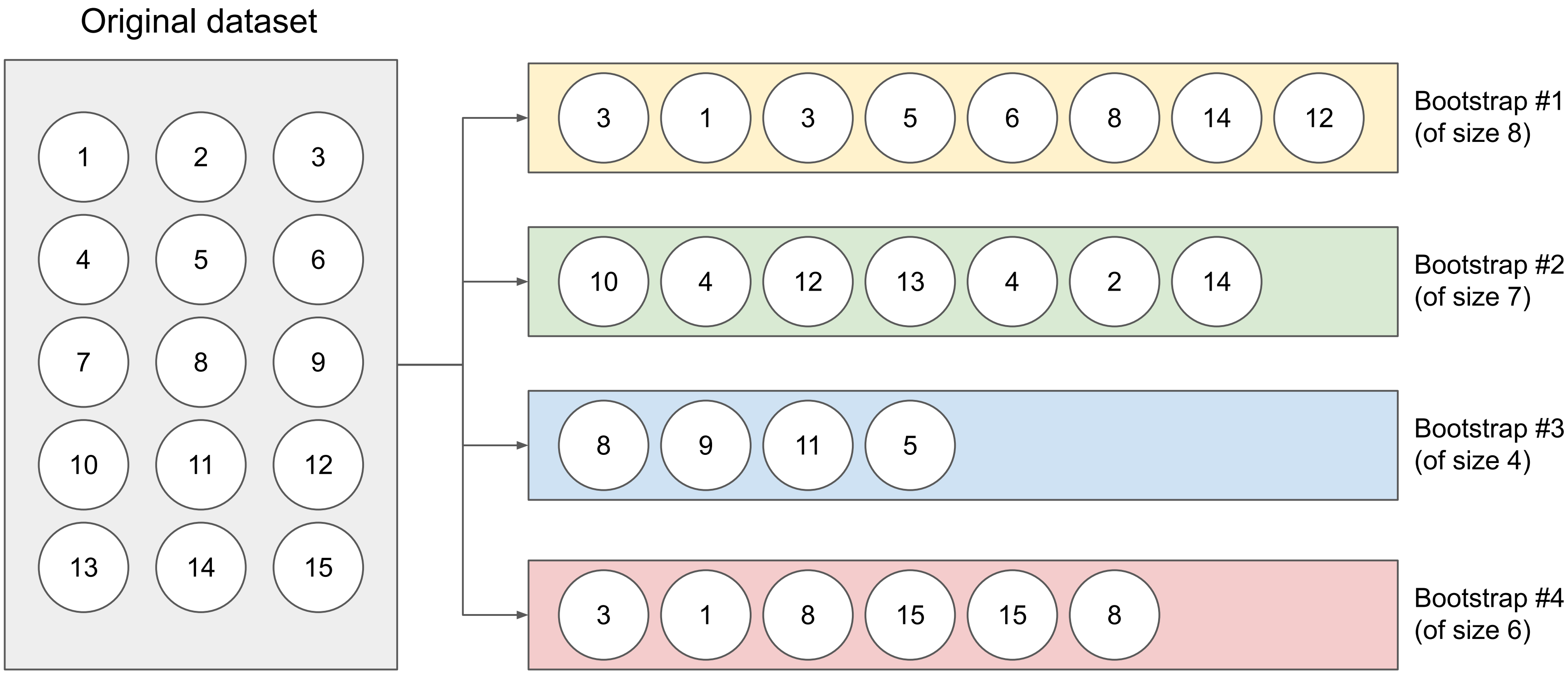
Ensemble — Bootstrapping



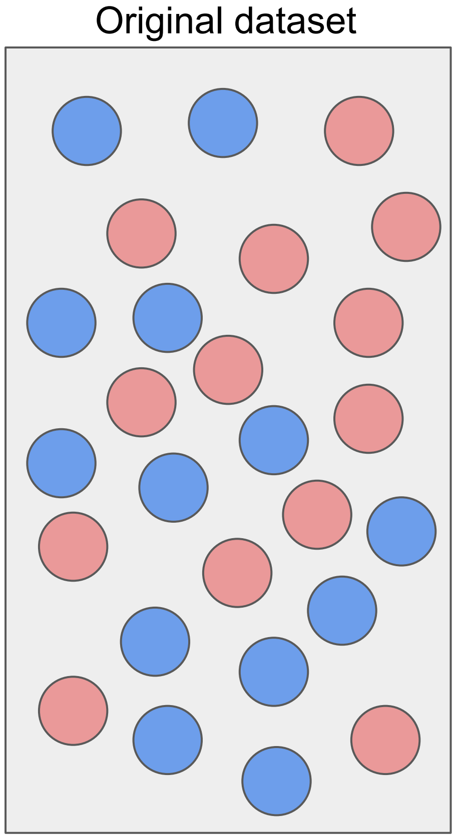
Ensemble — Bootstrapping



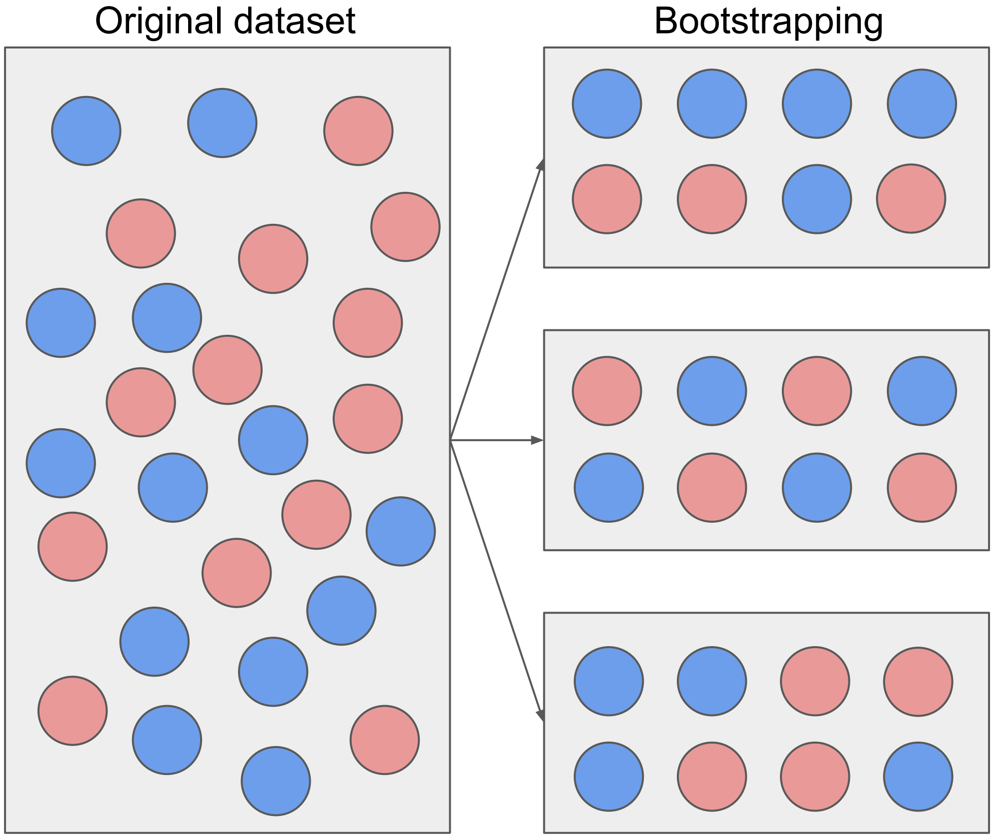
Ensemble — Bootstrapping



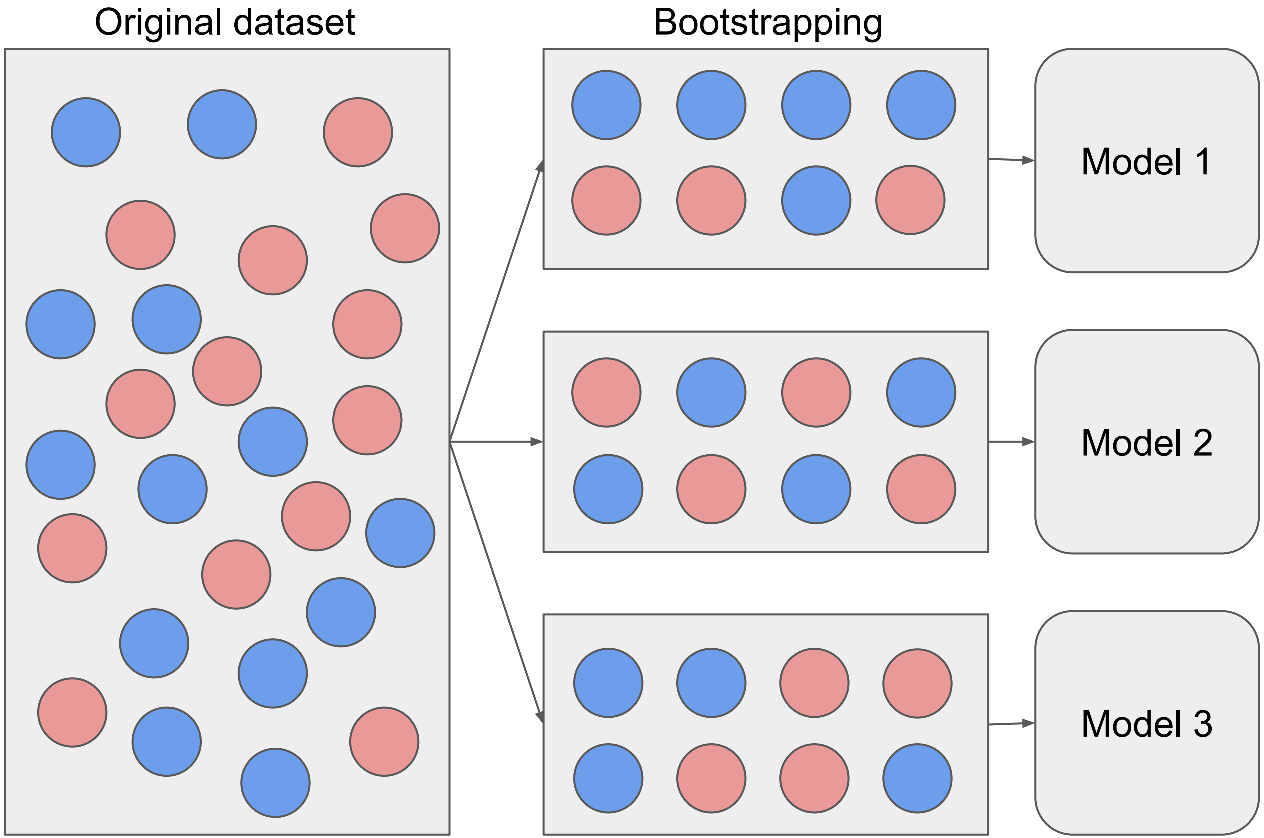
Ensemble — Bagging



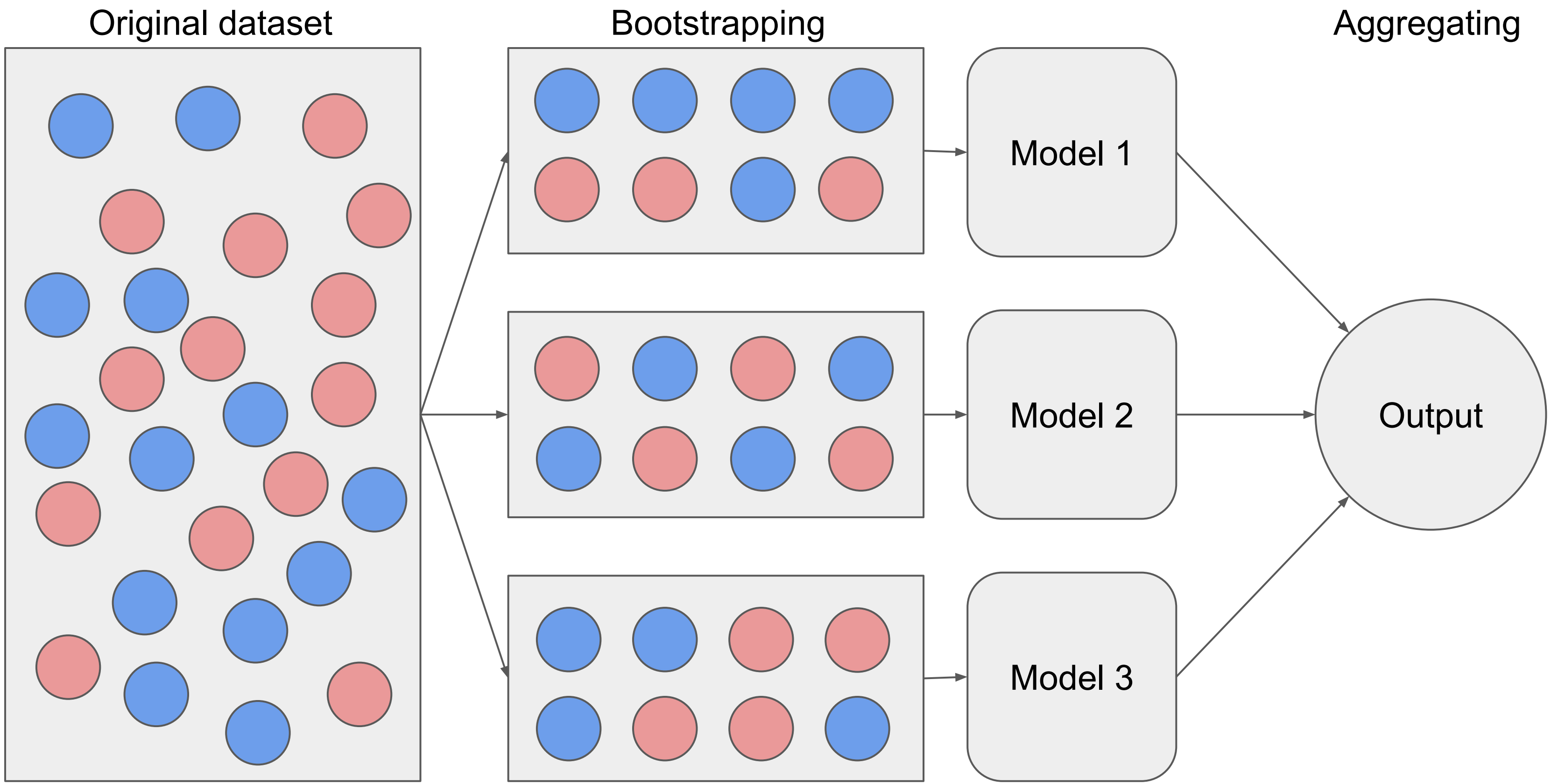
Ensemble — Bagging



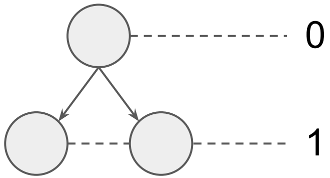
Ensemble — Bagging



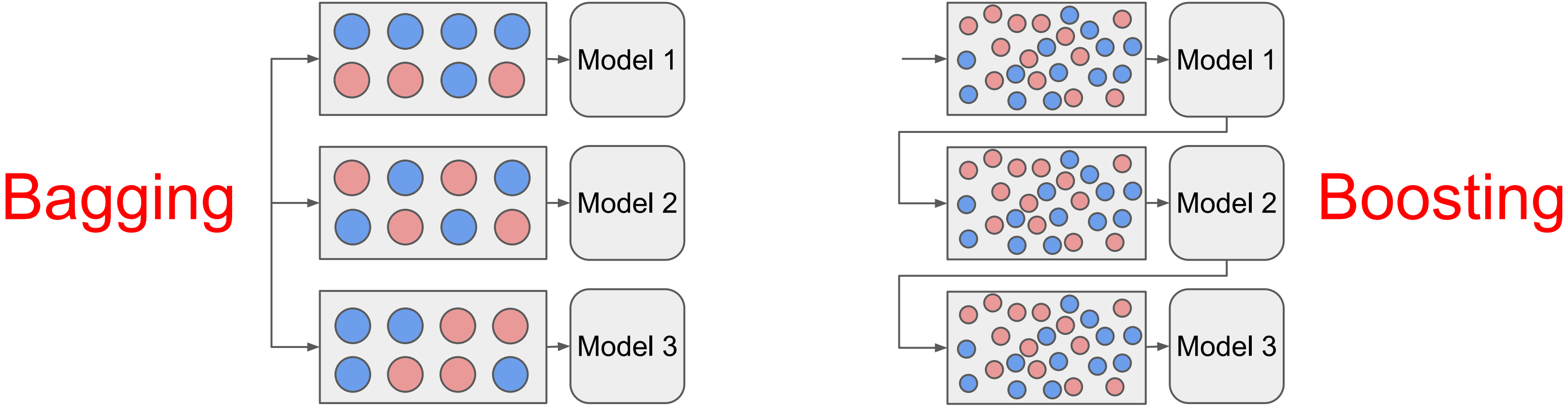
Ensemble — Bagging



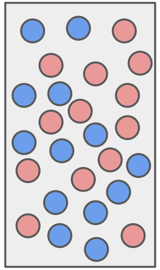
Ensemble — Boosting



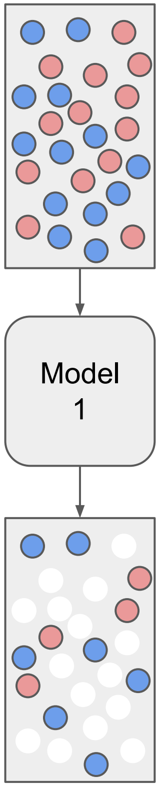
- Uses weak learners that have high bias
  - e.g., decision stumps (decision trees with depth 1)
- Unlike bagging, which trains models in parallel, boosting builds models sequentially, with each model correcting the mistakes of the previous ones
- Iterative algorithm that increases weights on hard examples
  - insight from previous iterations guides learning



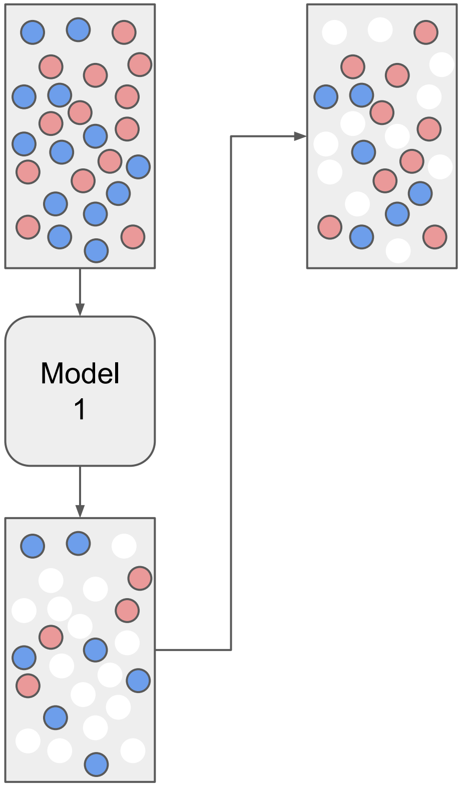
Ensemble — Boosting



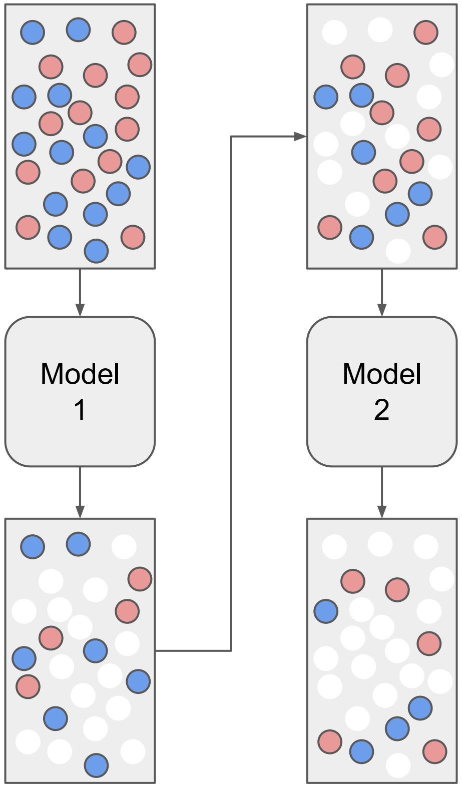
Ensemble — Boosting



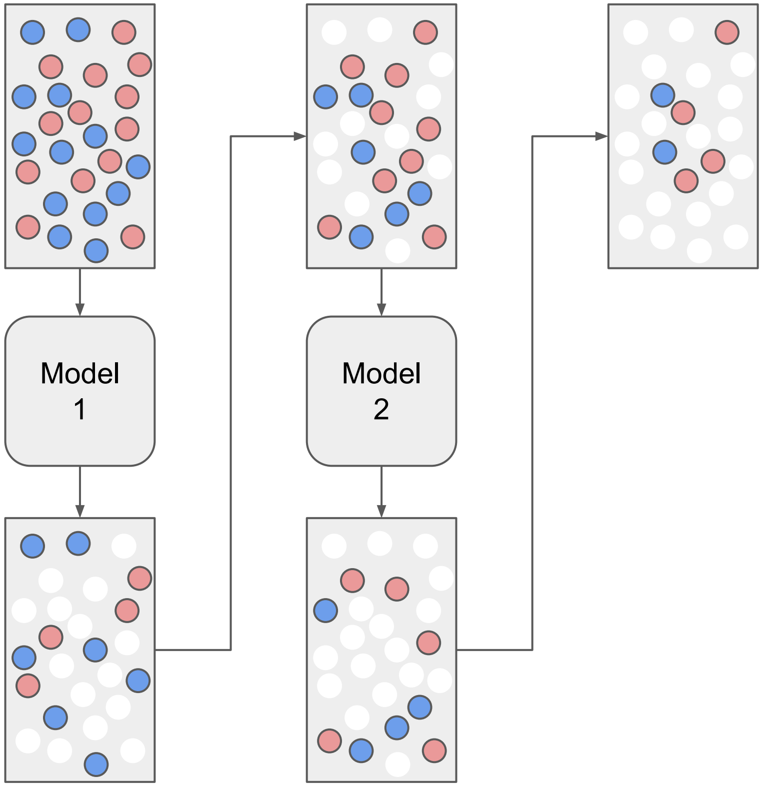
Ensemble — Boosting



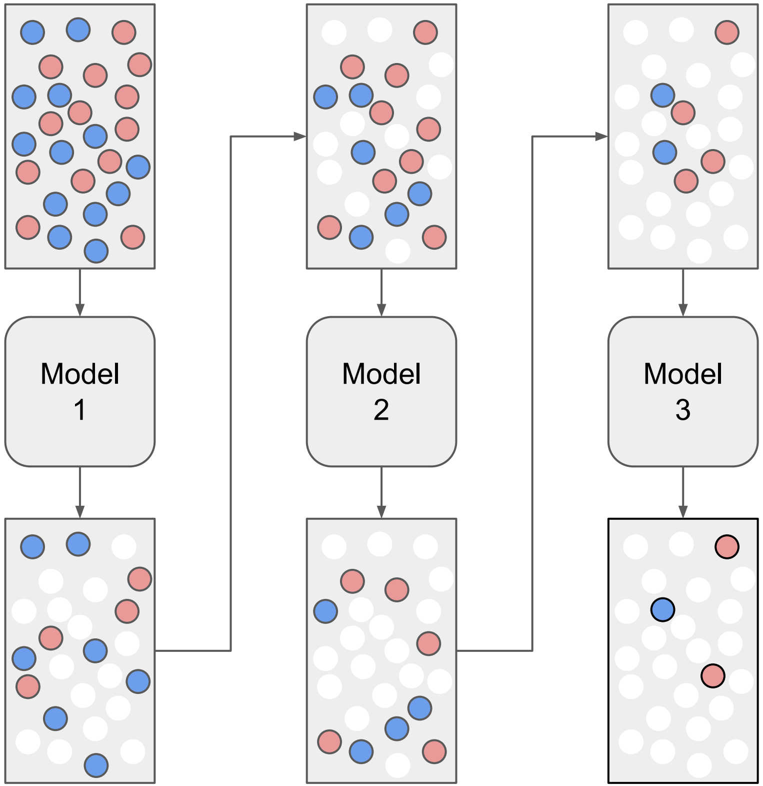
Ensemble — Boosting



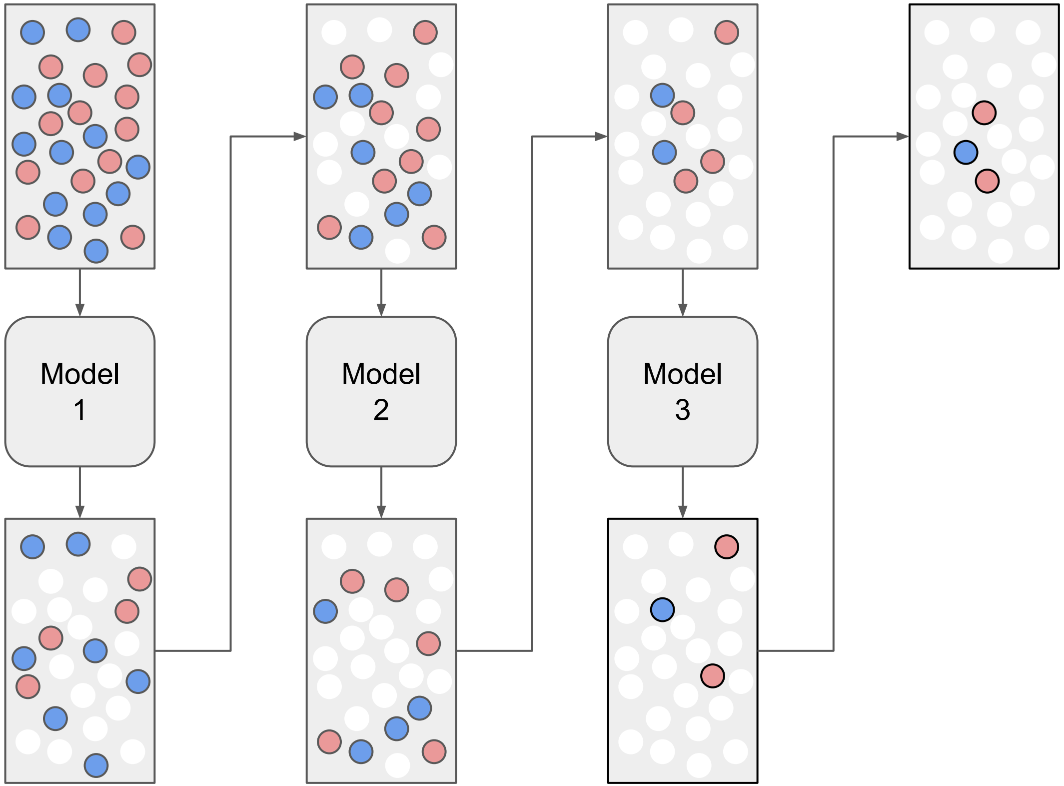
Ensemble — Boosting



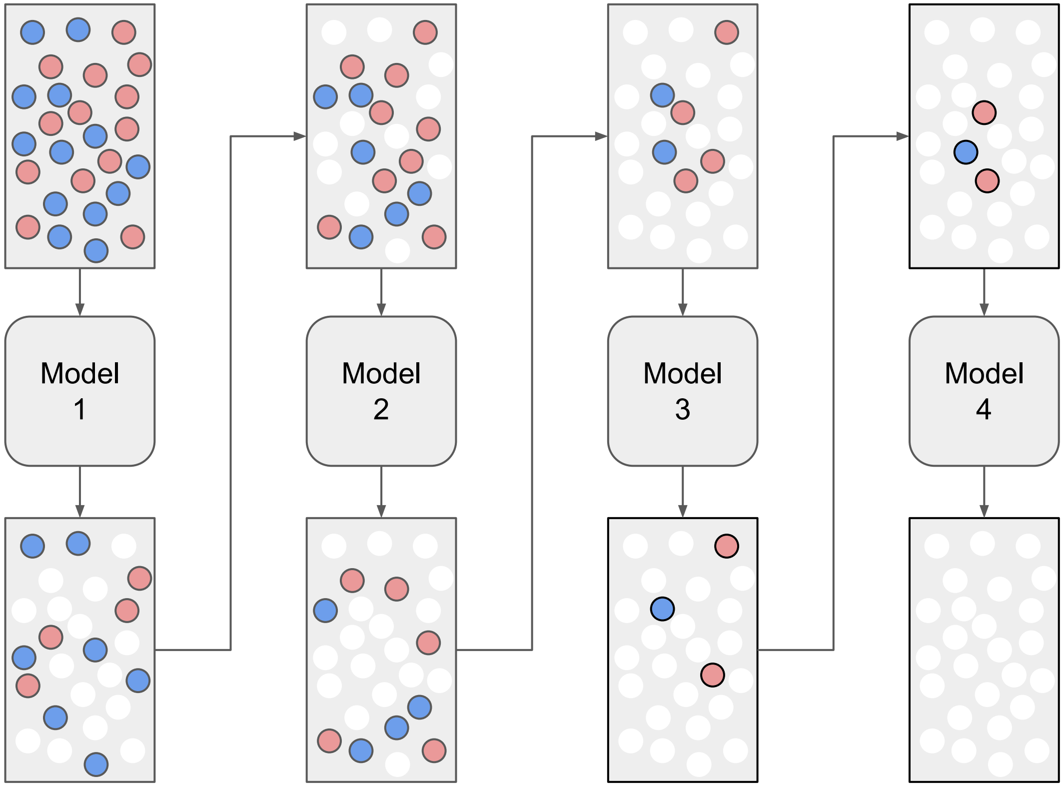
Ensemble — Boosting



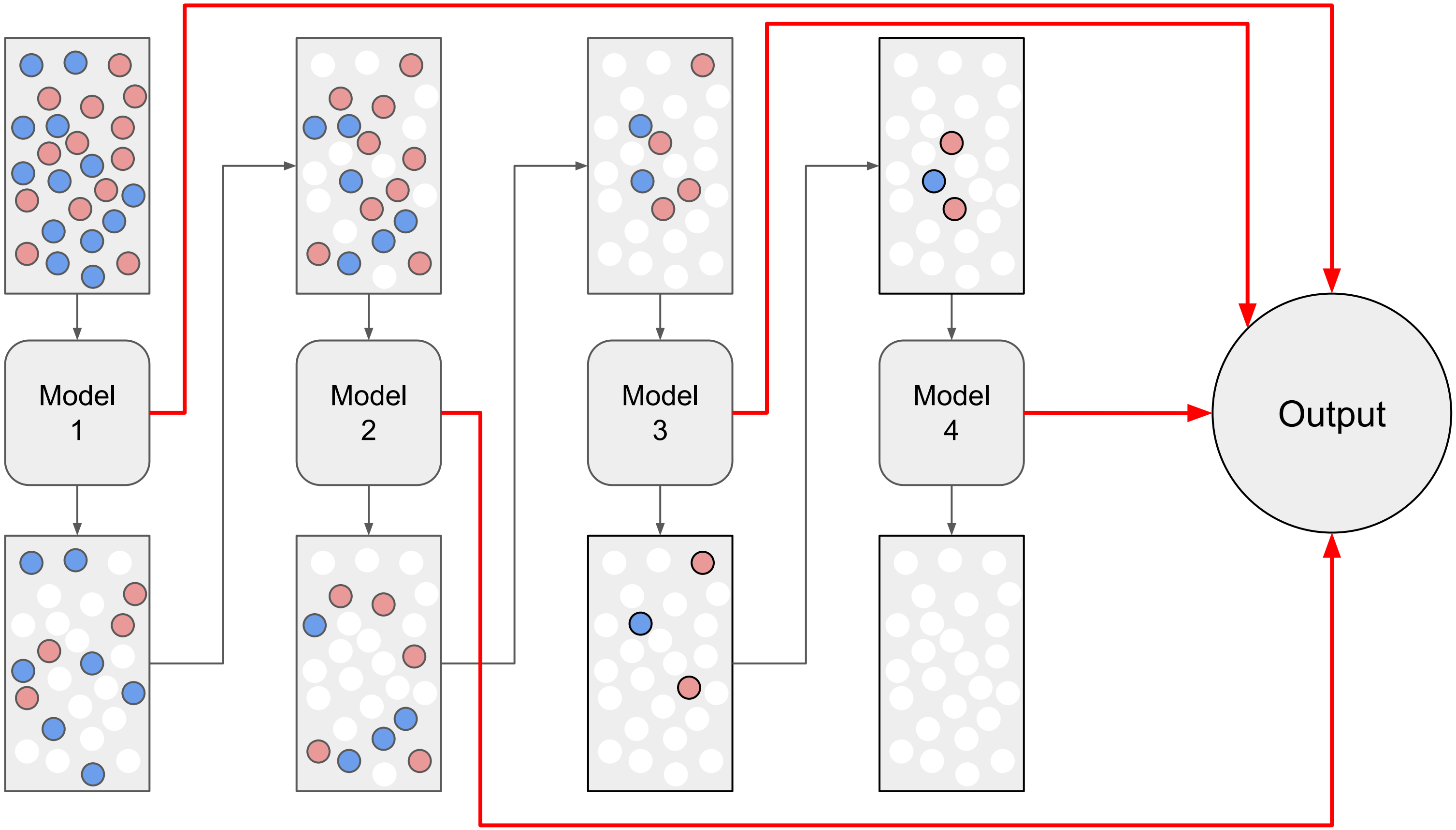
Ensemble — Boosting



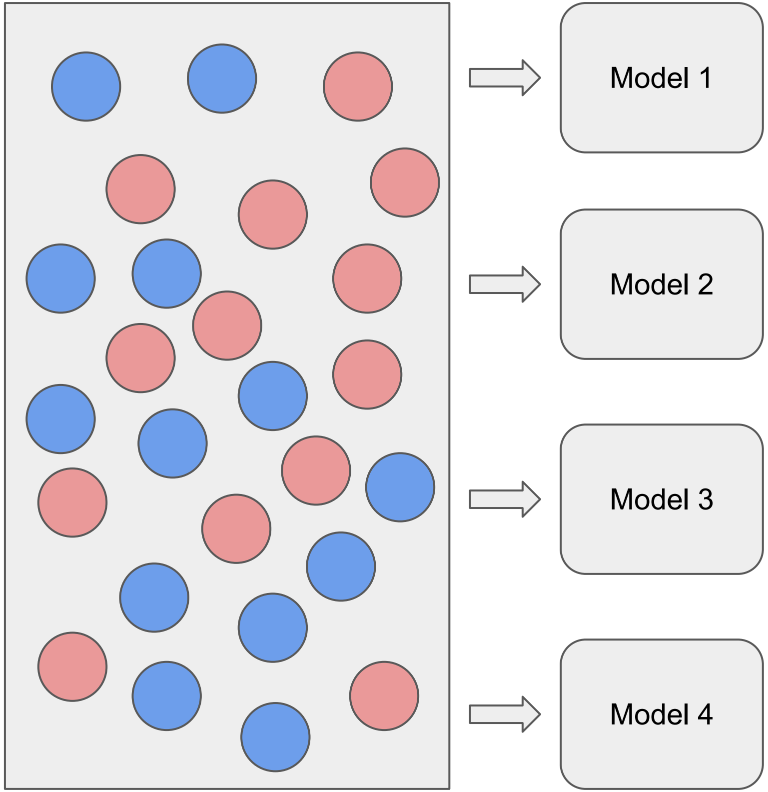
Ensemble — Boosting

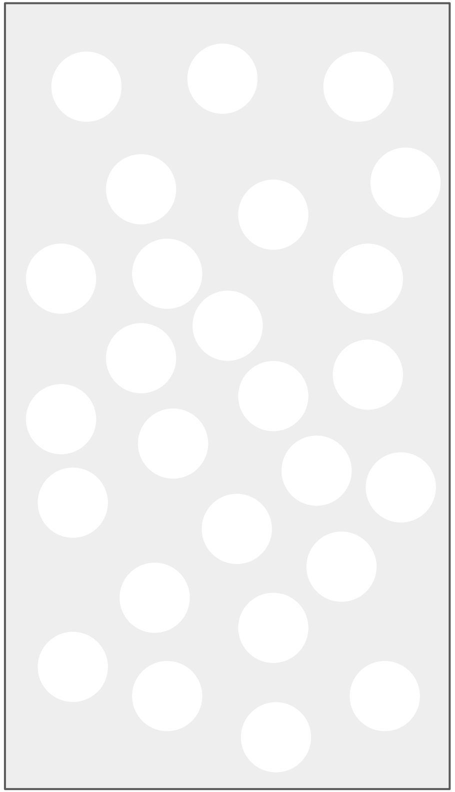


Ensemble — Boosting

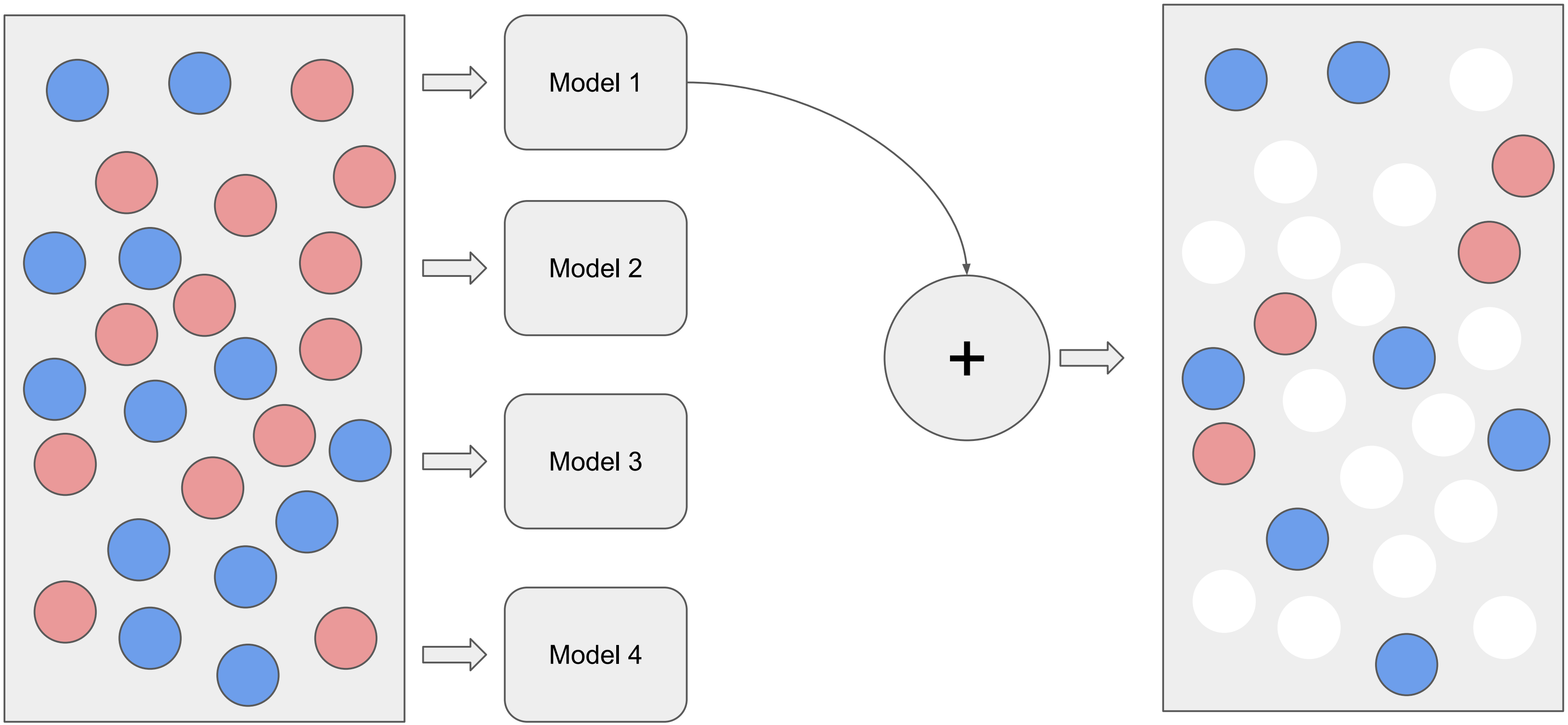


Ensemble — Boosting

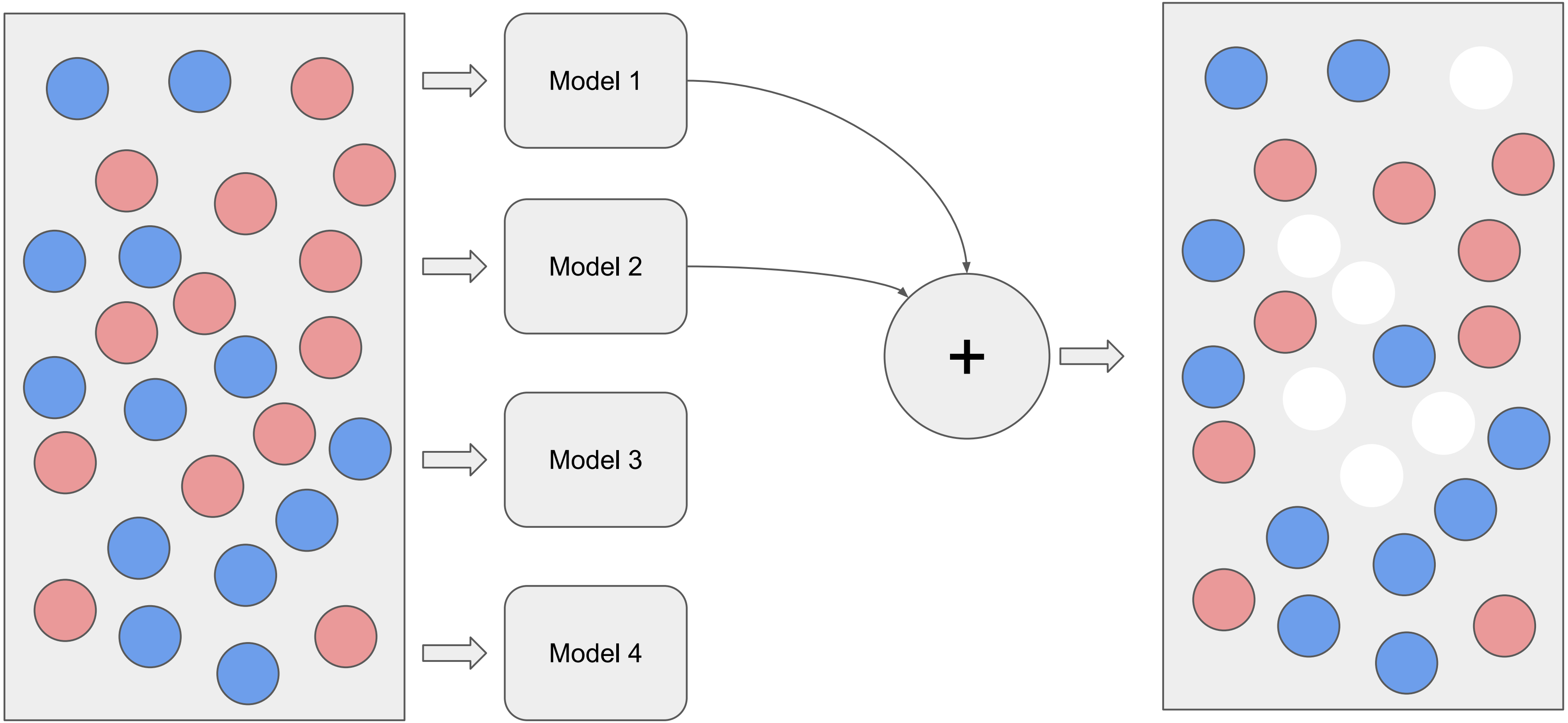




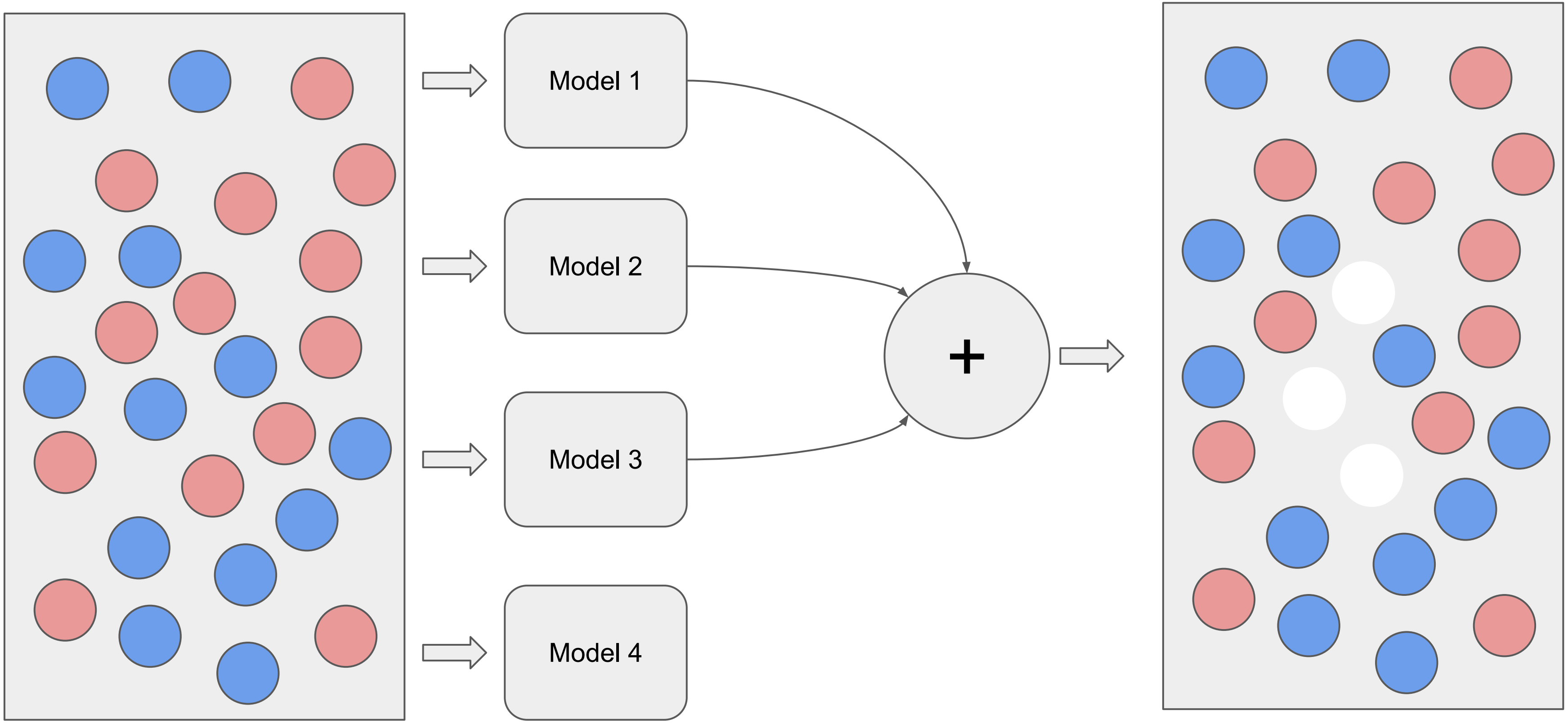
Ensemble — Boosting



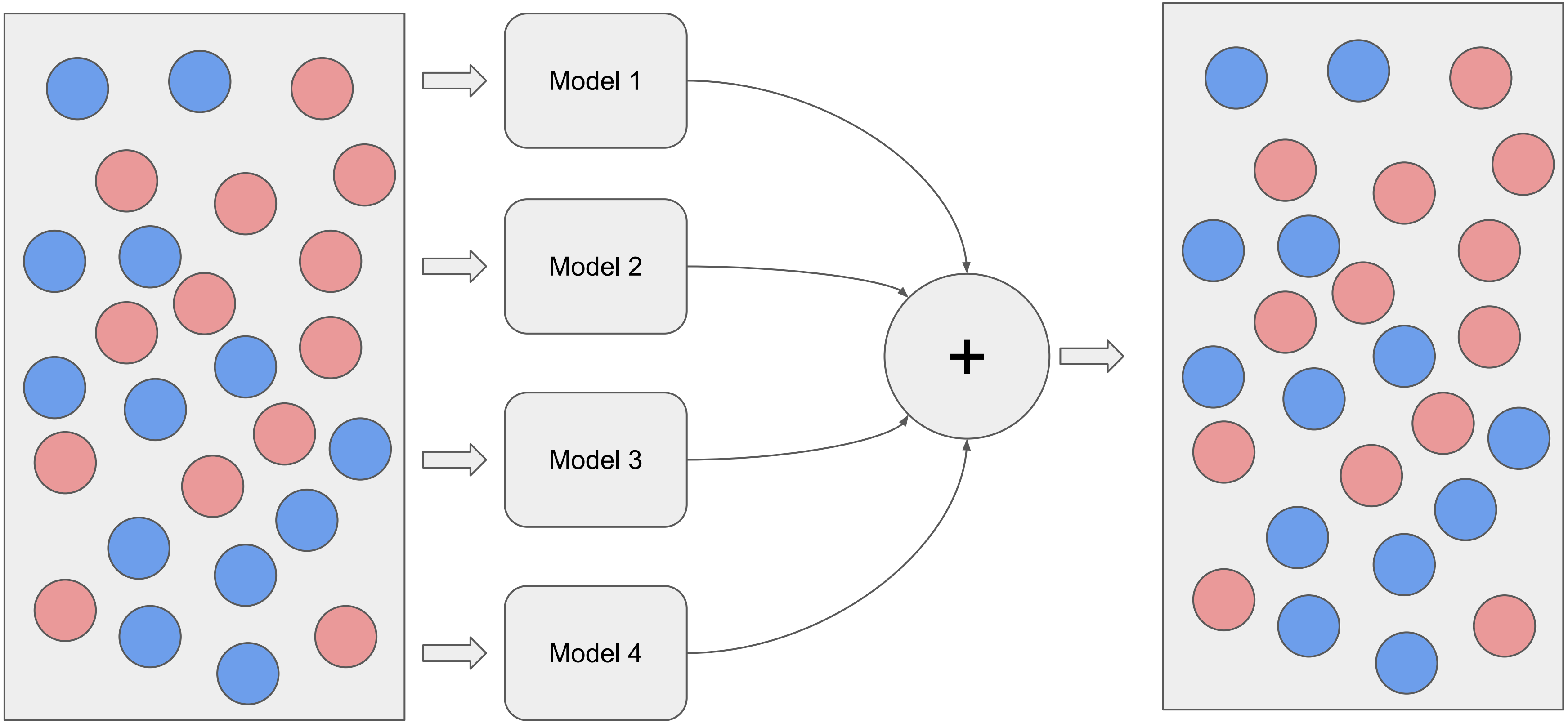
Ensemble — Boosting



Ensemble — Boosting



Ensemble — Boosting



AdaBoost: **ADA**ptive **BOOST**ing

Let $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ be *N* training data, *M* is an ensemble size

- Initialize the observation weights $\{w_1, \ldots, w_N\}$
- For $j = 1 \ldots M$:
  - Fit a model to the training data using weights $\{w_1, \ldots, w_N\}$
  - Compute

$$e_j = \left(\sum_{i=1}^N w_i \mathbb{I}[y_i \neq f_j(\boldsymbol{x}_i)]\right) / \left(\sum_{i=1}^N w_i\right)$$

$$\alpha_j = \frac{1}{2} \ln\left((1 - e_j)/e_j\right)$$

$$w_i \leftarrow \frac{w_i \exp\left(-\alpha_j y_i f_j(\boldsymbol{x}_i)\right)}{\sum_{i=1}^N w_i \exp\left(-\alpha_j y_i f_j(\boldsymbol{x}_i)\right)}$$

- Output function

$$F(\boldsymbol{x}_i) = \mathrm{sign}\left(\sum_{j=1}^M \alpha_j f_j(\boldsymbol{x}_i)\right)$$

$$\boxed{y_i \in \{-1, 1\}}$$

$$\boxed{\mathrm{sign}(u) = \begin{cases} 1, & \text{if } u > 0 \\ 0, & \text{if } u = 0 \\ -1, & \text{if } u < 0 \end{cases}}$$

$$\boxed{\mathbb{I}[y_i \neq f_j(\boldsymbol{x}_i)] = \begin{cases} 1, & \text{if } y_i \neq f_j(\boldsymbol{x}_i) \\ 0, & \text{if } y_i = f_j(\boldsymbol{x}_i) \end{cases}}$$

AdaBoost: ADAptive BOOSTing

$\boldsymbol{y}$	$\hat{\boldsymbol{y}}$	$\boldsymbol{w}$
+1	+1	0.25
+1	+1	0.25
-1	+1	0.25
-1	-1	0.25

$$e_j = \frac{w_3}{w_1 + w_2 + w_3 + w_4} = \frac{0.25}{1} = 0.25$$

$$\alpha_j = \frac{1}{2} \ln \left(\frac{1 - e_j}{e_j} \right) = \frac{1}{2} \ln \left(\frac{1 - 0.25}{0.25} \right) \approx 0.5493$$

$$w_1 = w_2 = w_4 = 0.25 \times \exp(-0.5493 \times 1 \times 1) \approx 0.1443$$

$$w_3 = 0.25 \times \exp(-0.5493 \times -1 \times 1) \approx 0.4330$$

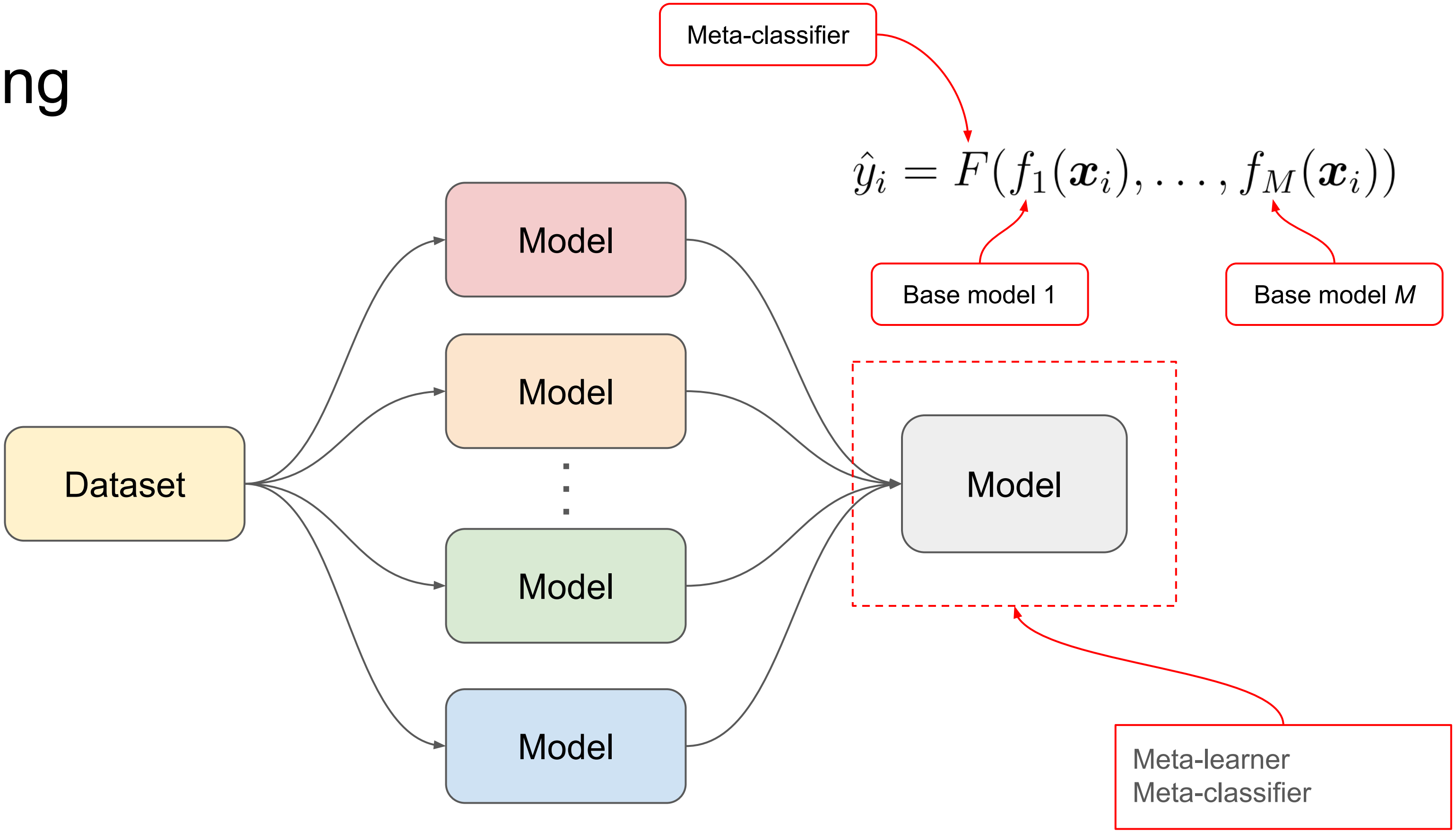
$$w_1 \leftarrow \frac{0.1443}{0.1443 + 0.1443 + 0.4330 + 0.1443} \approx 0.1666$$

$$w_2 \leftarrow \frac{0.1563}{0.1563 + 0.1563 + 0.4375 + 0.1563} \approx 0.1666$$

$$w_3 \leftarrow \frac{0.4375}{0.1563 + 0.1563 + 0.4375 + 0.1563} \approx 0.5000$$

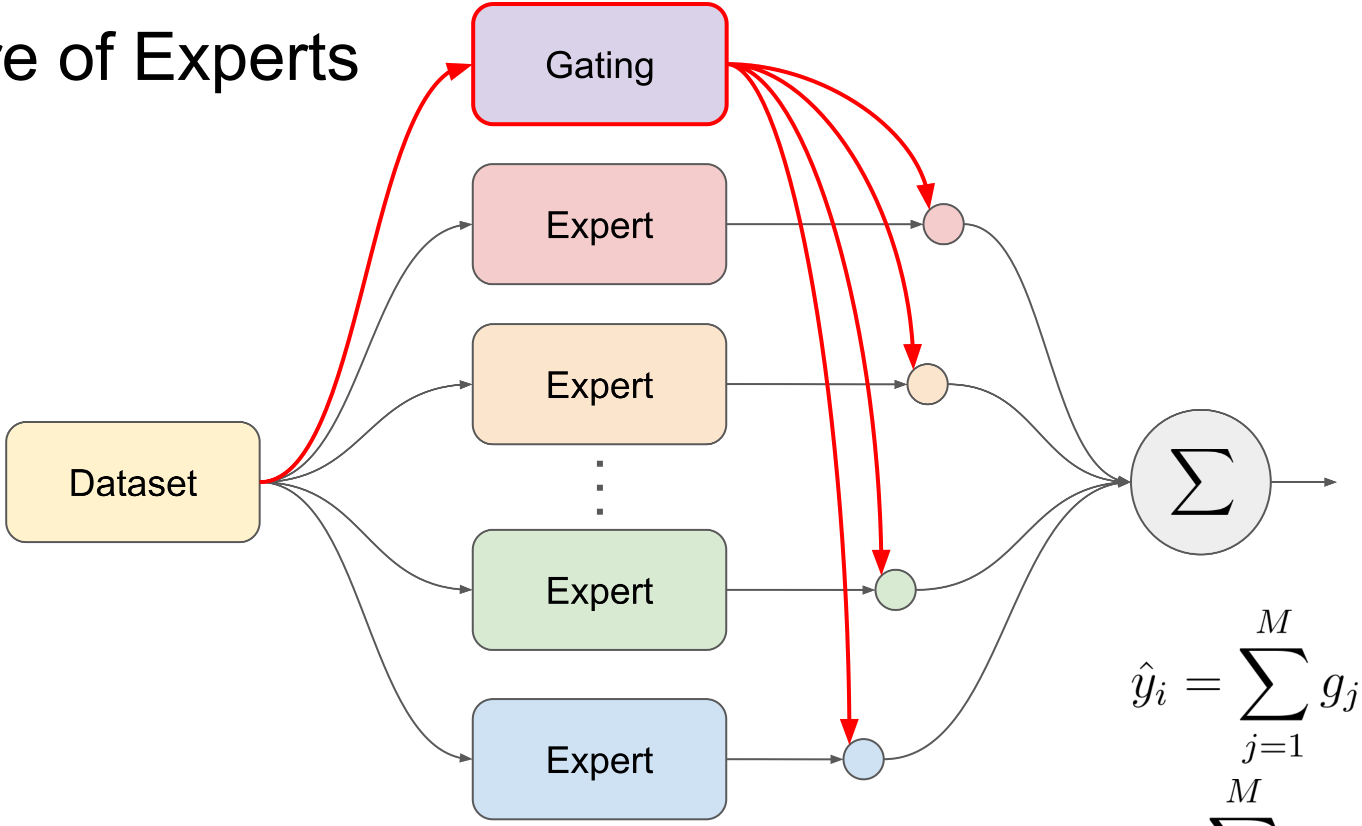
$$w_4 \leftarrow \frac{0.1563}{0.1563 + 0.1563 + 0.4375 + 0.1563} \approx 0.1666$$

Stacking



Meta-classifier
$$\hat{y}_i = F(f_1(\boldsymbol{x}_i), \ldots, f_M(\boldsymbol{x}_i))$$
Base model 1
Base model M
Dataset
Model
Model
Model
Model
Model
Meta-learner
Meta-classifier

Mixture of Experts



$$\hat{y}_i = \sum_{j=1}^{M} g_j(\boldsymbol{x}_i) f_j(\boldsymbol{x}_i)$$

$$\text{s.t.} \sum_{j=1}^{M} g_j(\boldsymbol{x}_i) = 1$$

Workshop-1

จงคำนวนเพื่อหาผลลัพธ์ของโมเดล Ensemble โดยใช้สมการที่กำหนดให้

	Apple	Orange	Berry
Model 1	0.8	0.1	0.1
Model 2	0.1	0.7	0.2
Model 3	0.7	0.2	0.1
Model 4	0.6	0.3	0.1
Model 5	0.1	0.1	0.8

$$\hat{y}_i = \arg\max_k \sum_{j=1}^{M} w_j f_j(\boldsymbol{x}_i)$$

$$w_1 = 0.3$$

$$w_2 = 0.05$$

$$w_3 = 0.3$$

$$w_4 = 0.3$$

$$w_5 = 0.05$$

Workshop-2

จงคำนวนเพื่อหาผลลัพธ์ของโมเดล Stacking โดยใช้ Logistic Regression ที่กำหนดให้เป็น Meta-Classifier

Base model 1		Base model 2	
0.8	0.2	0.7	0.3
0.1	0.9	0.2	0.8
0.7	0.3	0.6	0.4
0.6	0.4	0.7	0.3
0.5	0.5	0.3	0.7