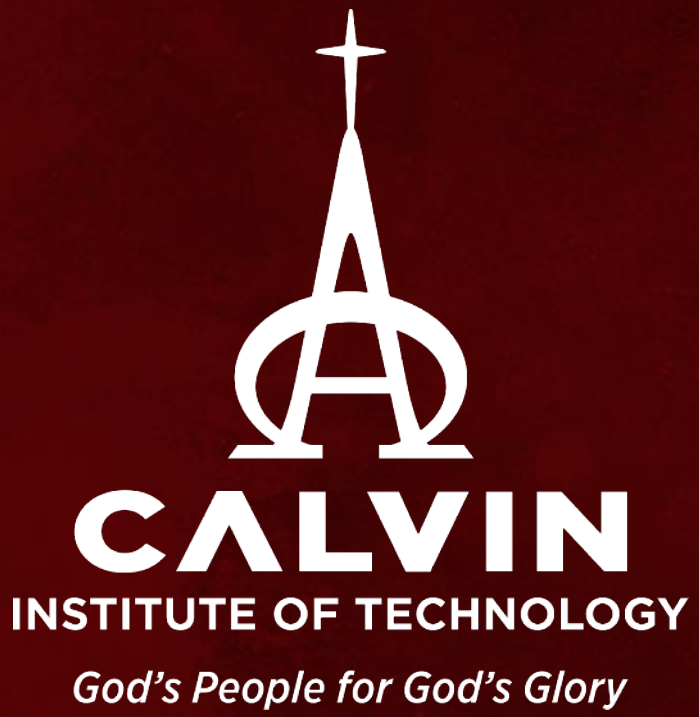


Ensemble

Hendrik Santoso Sugiarto

IBDA2032 – *Artificial Intelligence*

Capaian Pembelajaran

- Ensemble Learning
- Bagging
- Boosting
- Stacking

Ensemble

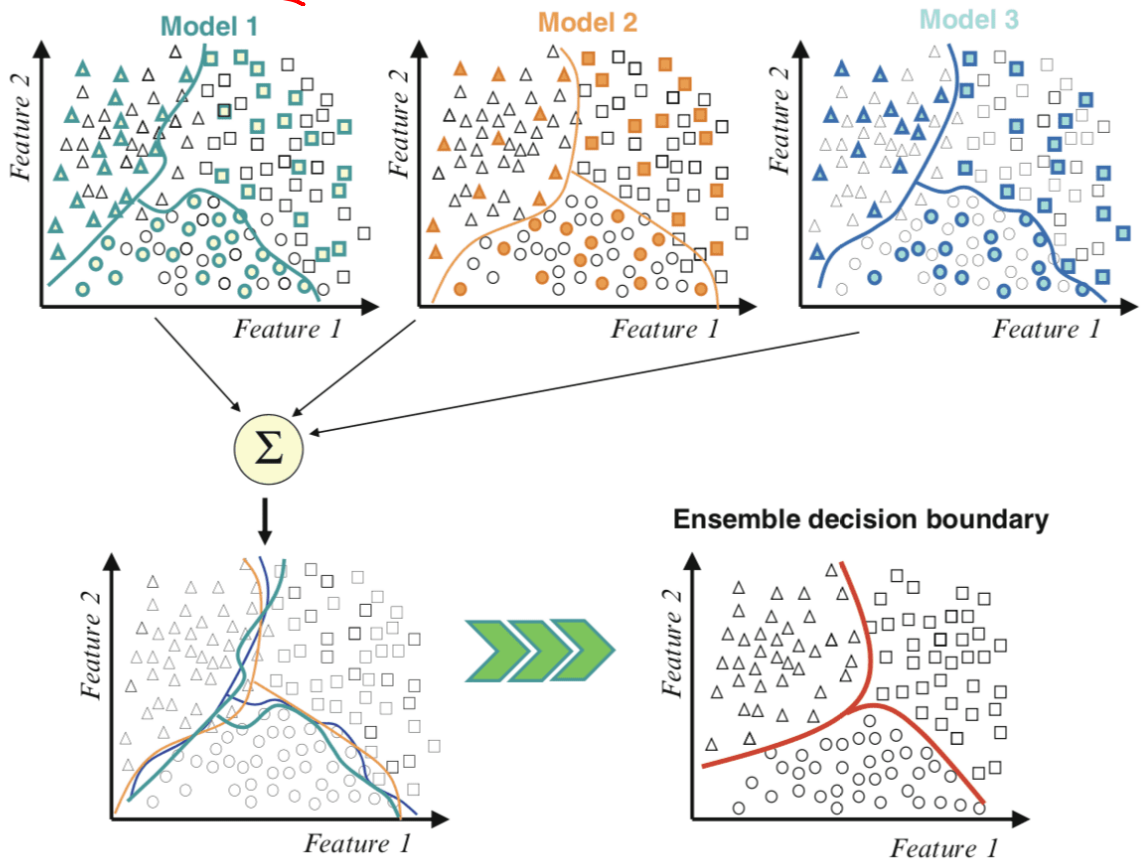
Ensemble

- Paradigma machine learning yang menggunakan perspektif kerja sama banyak model
- Dapat digunakan untuk regresi, klasifikasi, clustering
- Dapat dijadikan probabilistik

CALVIN INSTITUTE OF TECHNOLOGY

Ensemble Learning

- Ide: daripada melatih 1 jenis model, latih beberapa model dan gabungkan
- Biasanya meningkatkan performa dengan sangat banyak



Intuisi

- Misalkan terdapat 25 classifier biasa
- Masing-masing memiliki tingkat kesalahan sekitar, $\varepsilon = 0.35$
- Asumsikan setiap classifier independent terhadap yang lain
- Maka peluang dari mayoritas classifier (13 dari 25) membuat prediksi yang salah adalah:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

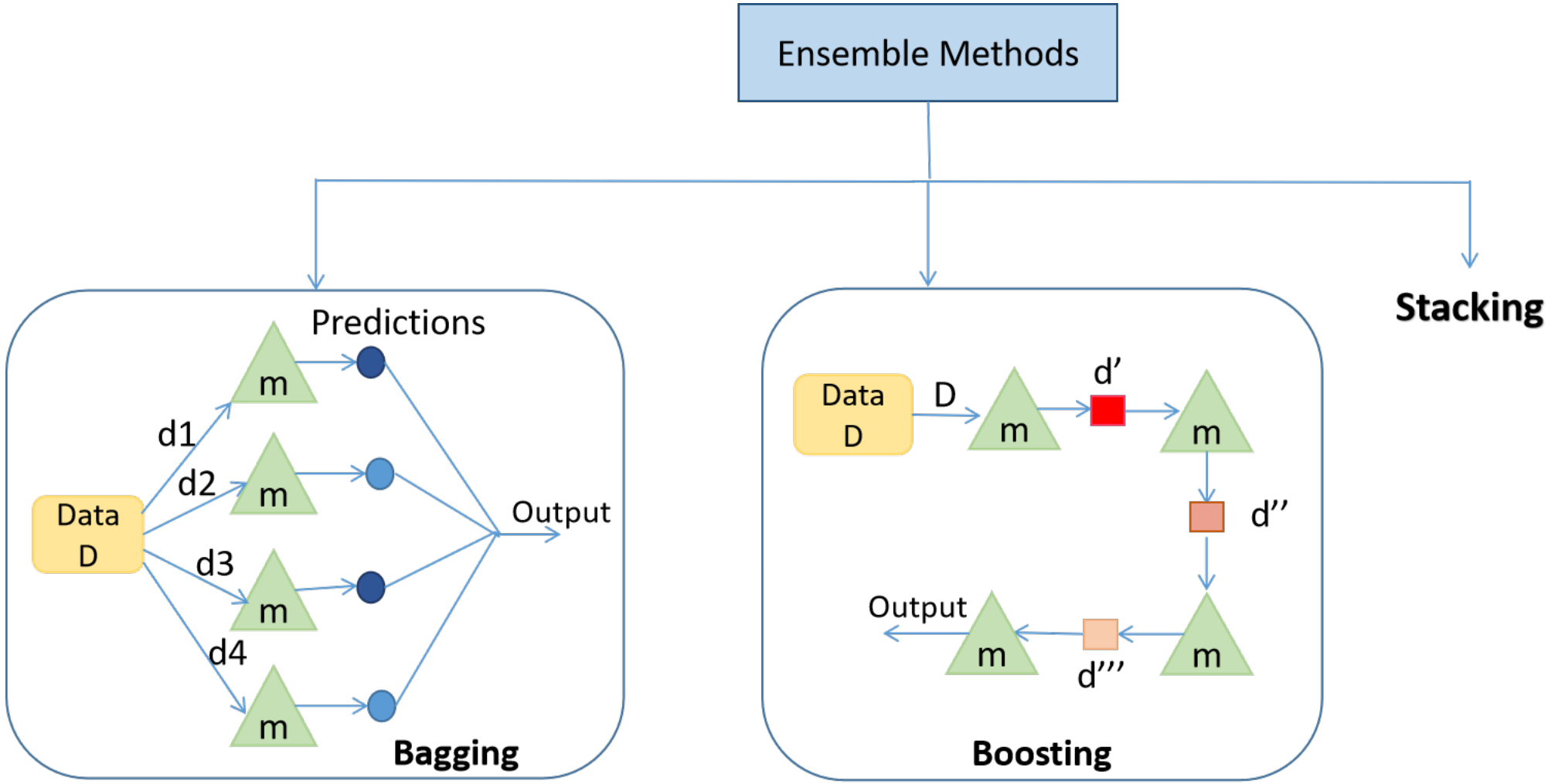
0.65

$$\sum_{i=13}^{25} \binom{25}{i} 0.35^i (1-0.35)^{25-i} = 0.06$$

0.94

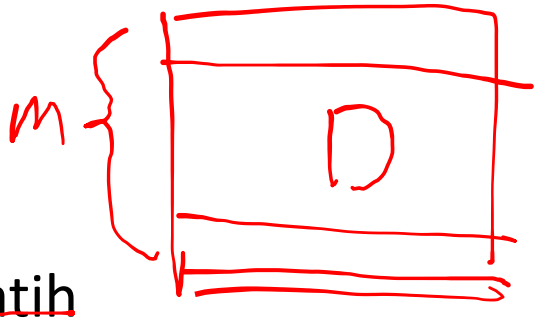
Jenis Ensemble

- Bagging
- Boosting
- Stacking
- ECOC
- Dll



Bagging

Bootstrap Sampling



- Bagging = bootstrap aggregating
- Boostrap sampling: jika terdapat himpunan D berisi m data latih
  - Buat D^i dengan mengambil m data secara random dengan replacement dari himpunan D
  - D^i akan membuang sekitar $\left(1 - \frac{1}{m}\right)^m \approx 0.37$ data dari D
- Bagging (sampling dengan replacement) vs pasting (sampling tanpa replacement)

Bagging

- Sampling dengan replacement

Data Asal	1	2	3	4	5	6	7	8	9	10
Bagging 1	8	7	8	2	10	10	5	6	5	7
Bagging 2	2	9	4	1	1	3	3	7	2	5
Bagging 3	5	1	8	10	5	5	6	9	7	3

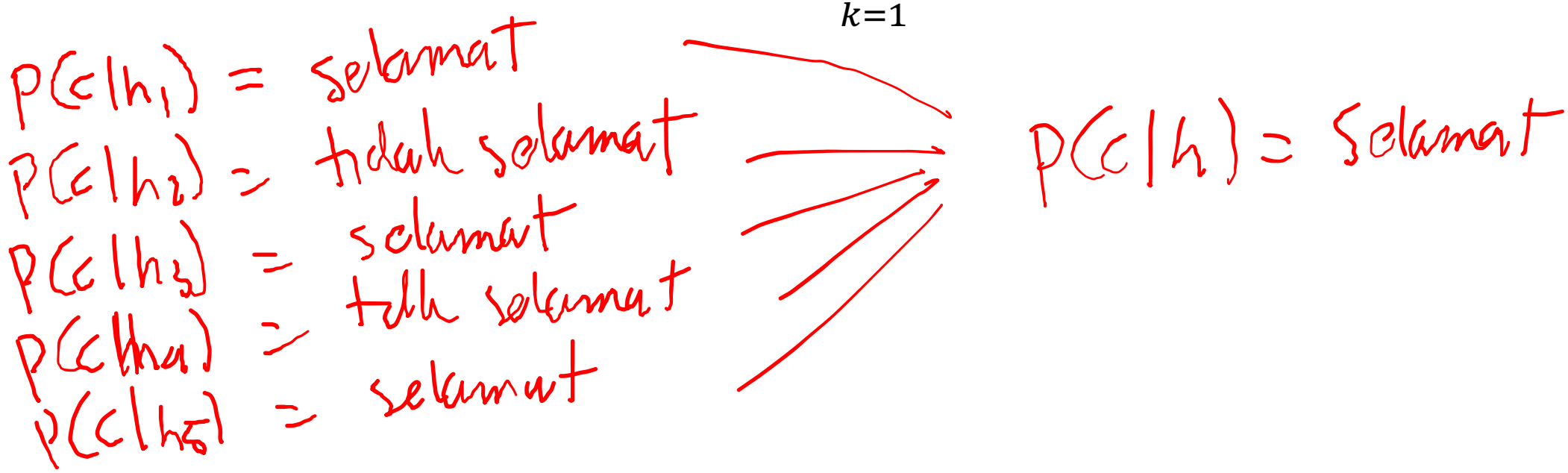
- Buat classifier untuk setiap bootstrap sample
- Setiap sample memiliki peluang $\left(1 - \frac{1}{m}\right)^m$ untuk tidak terpilih (mendekati 0.37 untuk m besar)
- Setiap sample memiliki peluang $1 - \left(1 - \frac{1}{m}\right)^m$ untuk terpilih (mendekati 0.63 untuk m besar)

CALVIN INSTITUTE OF TECHNOLOGY

Algoritma bagging

- Buat K boostrap samples $D^1, D^2, \ldots, D^K$
- Latih classifier berbeda h_k untuk setiap D^k
- Klasifikasikan data baru dengan voting

$$c^*(\boldsymbol{x}) = \arg\max_c \sum_{k=1}^{K} p(c|h_k, \boldsymbol{x})$$



Hard Voting vs Soft Voting

- Hard voting: melakukan prediksi dengan vote mayoritas dari semua prediksi model-model yang dilatih
- Soft voting: melakukan prediksi dengan rata-rata peluang prediksi dari semua prediksi model-model yang dilatih

CALVIN INSTITUTE OF TECHNOLOGY

Uji Pemahaman

- Suatu klasifikasi biner (kelas A dan B) memakai pemodelan bagging yang menggunakan 5 base learner dengan peluang prediksi kelas A, sebagai berikut:
  - Learner 1: 60%
  - Learner 2: 15%
  - Learner 3: 55%
  - Learner 4: 65%
  - Learner 5: 10%

3A ✓
2B

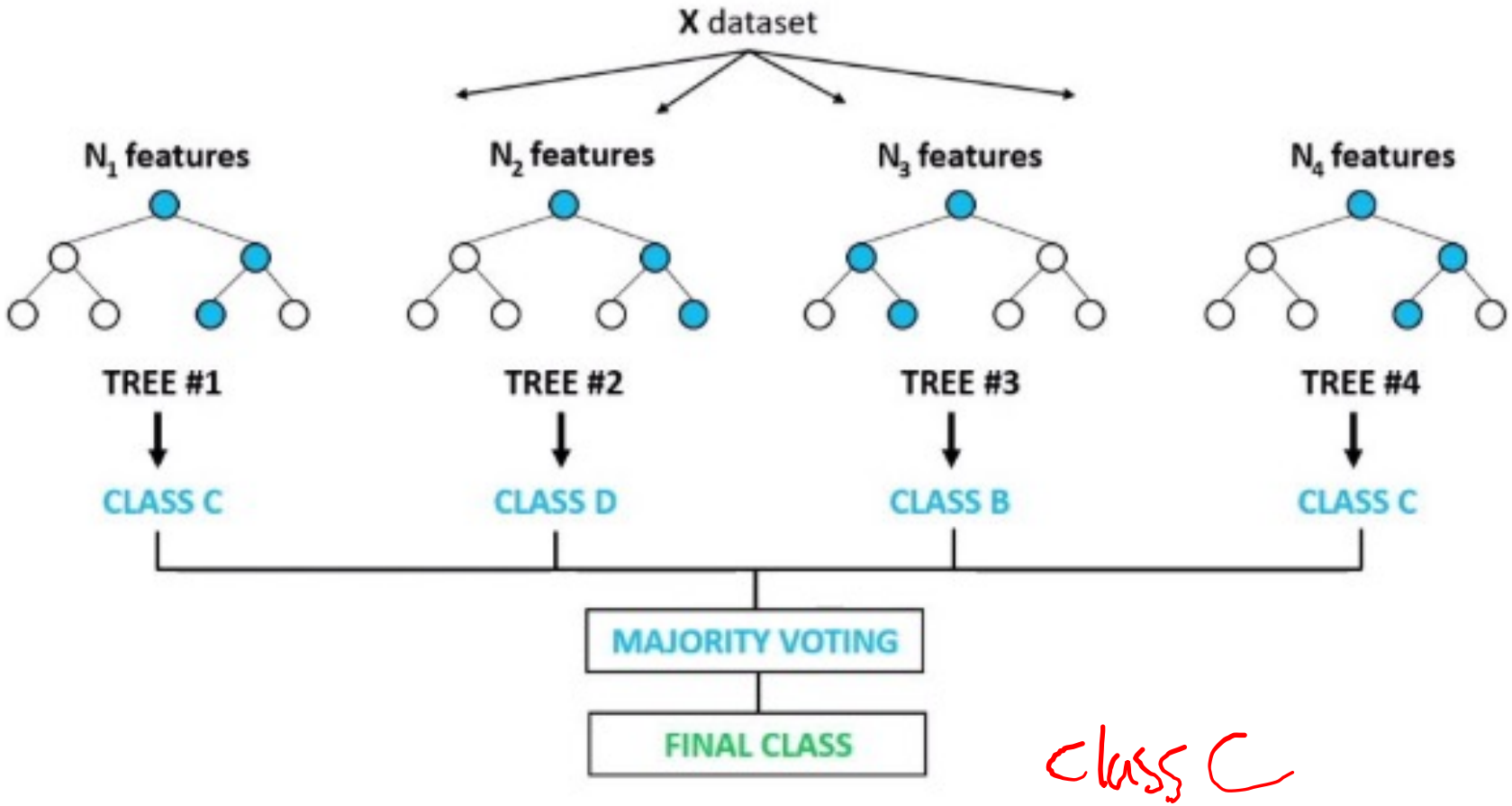
- Apakah prediksi akhir jika menggunakan hard voting?
- Apakah prediksi akhir jika menggunakan soft voting?

A
B

$$P(A) = \frac{0.6 + 0.15 + 0.55 + 0.65 + 0.1}{5} = \frac{2.05}{5} = 0.41$$

Random Forest

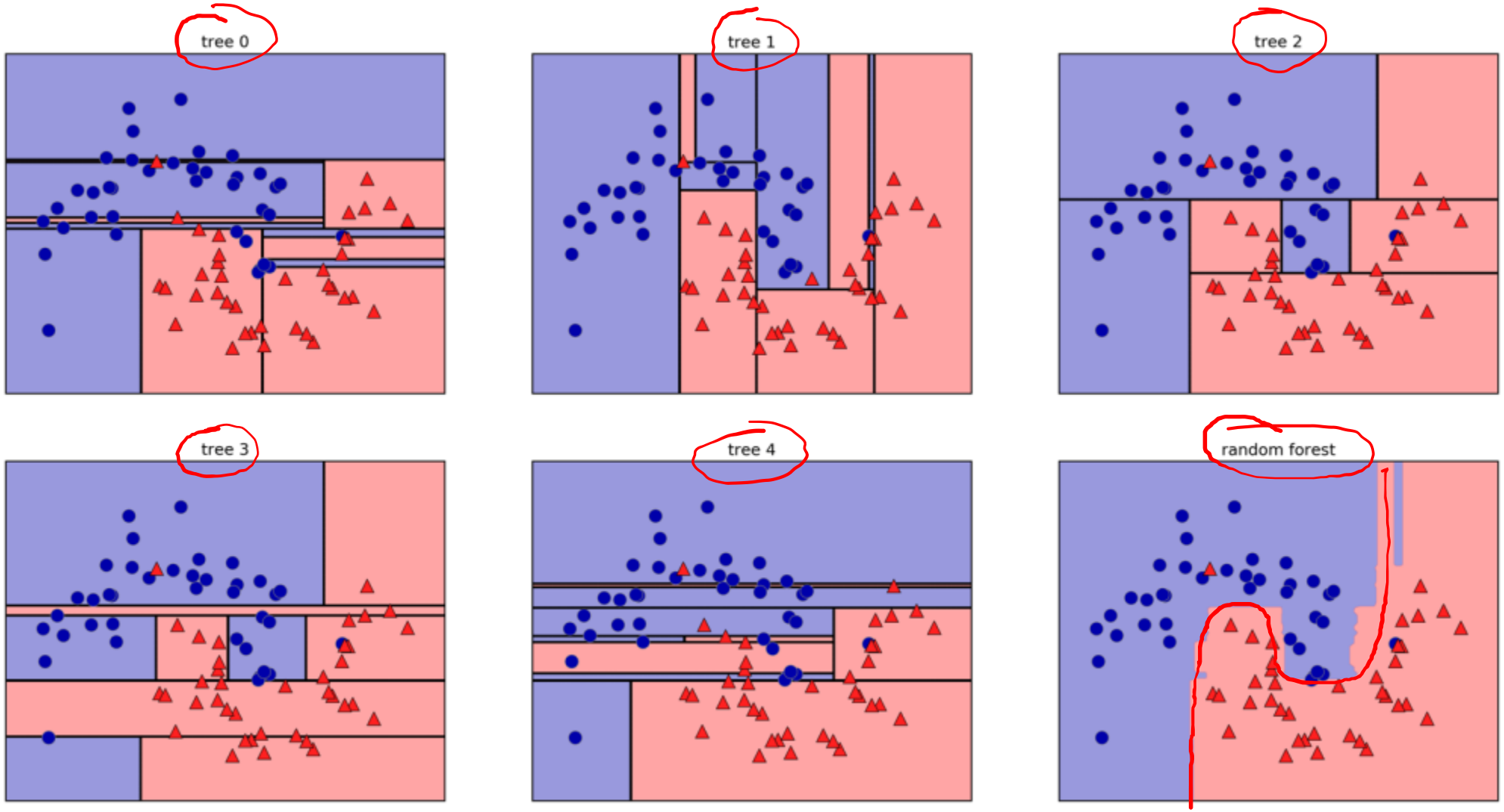
- Voting mayoritas dari banyak decision tree → prediksi random forest



class C

Ilustrasi RF

- RF yang terdiri dari 5 pohon
- Hasil per pohon (no. 0 – 4)
- Hasil agregat RF



Performa Bagging

- Bagging decision trees
- Bootstrap 50 samples berbeda dari data asli
- Latih decision tree untuk setiap bootstrap
- Prediksi label kelas untuk data uji dengan voting
- Bagging decision tree mengalahkan 1 tree
- Bagging decision tree →random forest

Dataset	C4.5 Standard	C4.5 Bagging	Boosting Ada
breast-cancer-w	5.0	3.3	3.1
credit-a	14.9	12.1	12.6
credit-g	29.6	22.8	22.9
diabetes	28.3	21.9	22.3
glass	30.9	28.4	30.5
heart-cleveland	24.3	18.1	17.4
hepatitis	21.6	16.5	13.8
house-votes-84	3.5	3.6	4.4
hypo	0.5	0.4	0.4
ionosphere	8.1	6.0	6.0
iris	6.0	4.6	5.6
kr-vs-kp	0.6	0.5	0.3
labor	15.1	13.3	13.2
letter	14.0	10.6	6.7
promoters-936	12.8	9.5	6.3
ribosome-bind	11.2	9.3	9.1
satellite	13.8	10.8	10.4
segmentation	3.7	2.8	2.3
sick	1.3	1.0	0.9
sonar	29.0	21.6	19.7
soybean	8.0	8.0	7.9
splice	5.9	5.7	6.3
vehicle	29.4	26.1	24.8

CALVIN INSTITUTE OF TECHNOLOGY

Kelemahan Bagging

- Inefficient bootstrap sampling:
  - Setiap data punya peluang yang sama untuk tersampel
  - Tidak ada pembedaan antara data yang mudah dan data yang sulit
- Inefficient model combination:
  - Bobot yang sama untuk setiap classifier
  - Tidak ada pembedaan antara classifier akurat dan tidak

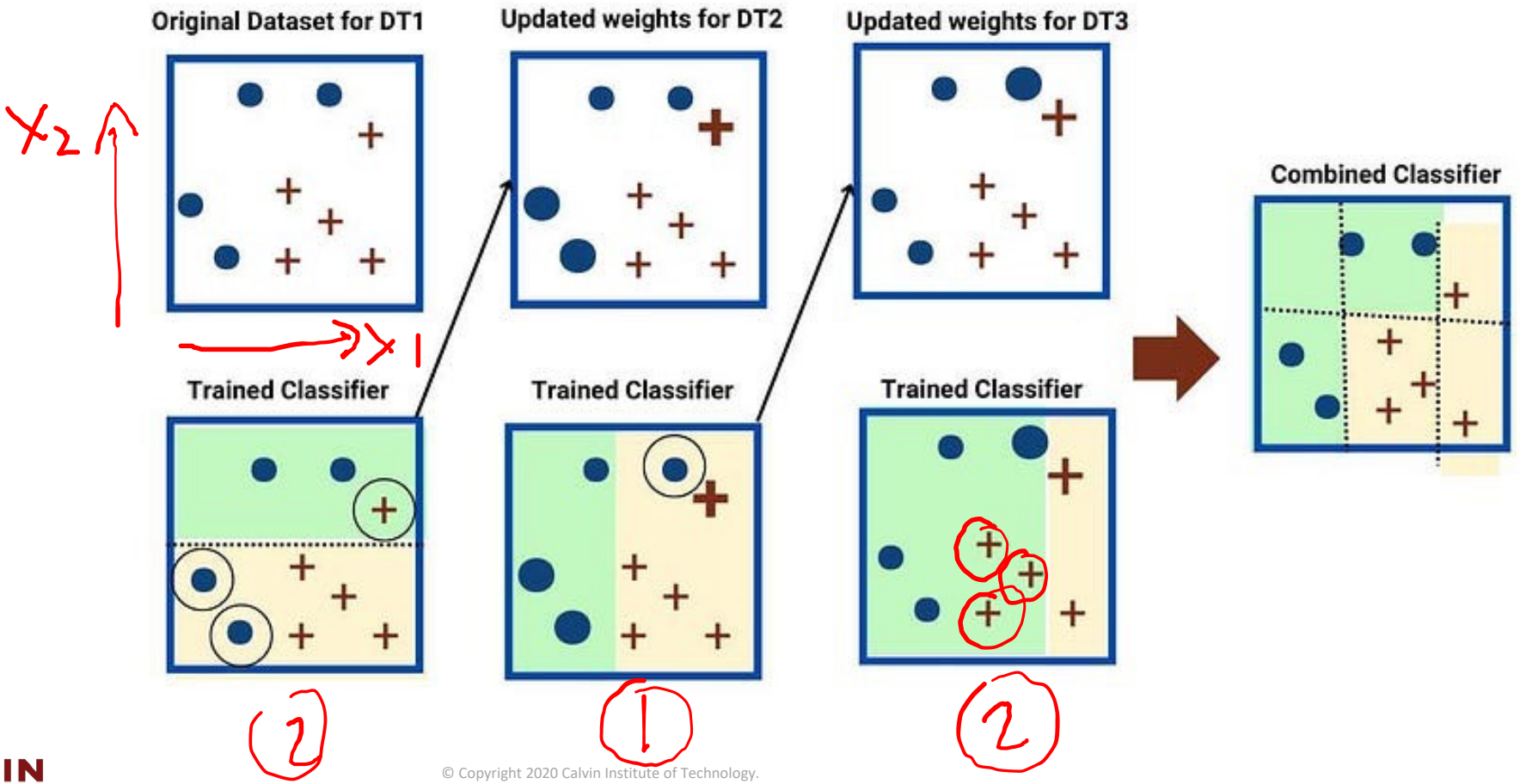
Memperbaiki Efisiensi Bagging

- Strategi sampling yang lebih baik
  - Fokus pada data yang sulit diprediksi
- Strategi pengkombinasian model yang lebih baik
  - Model yang akurat harus diberi bobot yang lebih besar

Boosting

Boosting

- Boost = memperkuat → perkuat model yang bagus melalui data yang sulit



Contoh

- Data yang terklasifikasi salah akan dinaikan bobotnya
- Data yang terklasifikasi benar akan dikurangi bobotnya
- Contoh: data 10 susah diklasifikasi→bobotnya dinaikan sehingga peluang tersampel lagi lebih besar pada round berikutnya

Data Asal	1	2	3	4	5	6	7	8	9	10
Boosting 1	8	7	8	2	10	10	5	6	5	7
Boosting 2	2	9	4	10	10	3	10	7	2	5
Boosting 3	5	10	8	10	10	10	6	9	7	3

Varian Algoritma Boosting

- Adaboost (adaptive boosting)
- Gradient boosting
- XGboost (extreme gradient boosting)
- LightGB

Performa Boosting

- AdaBoosting decision tree
- Hasilkan 50 decision trees dengan adaboosting
- Kombinasi linear dengan bobot adaboosting
- Secara umum:
  - Adaboost ≈ bagging > C4.5
  - Adaboost butuh lebih sedikit classifier dibanding bagging

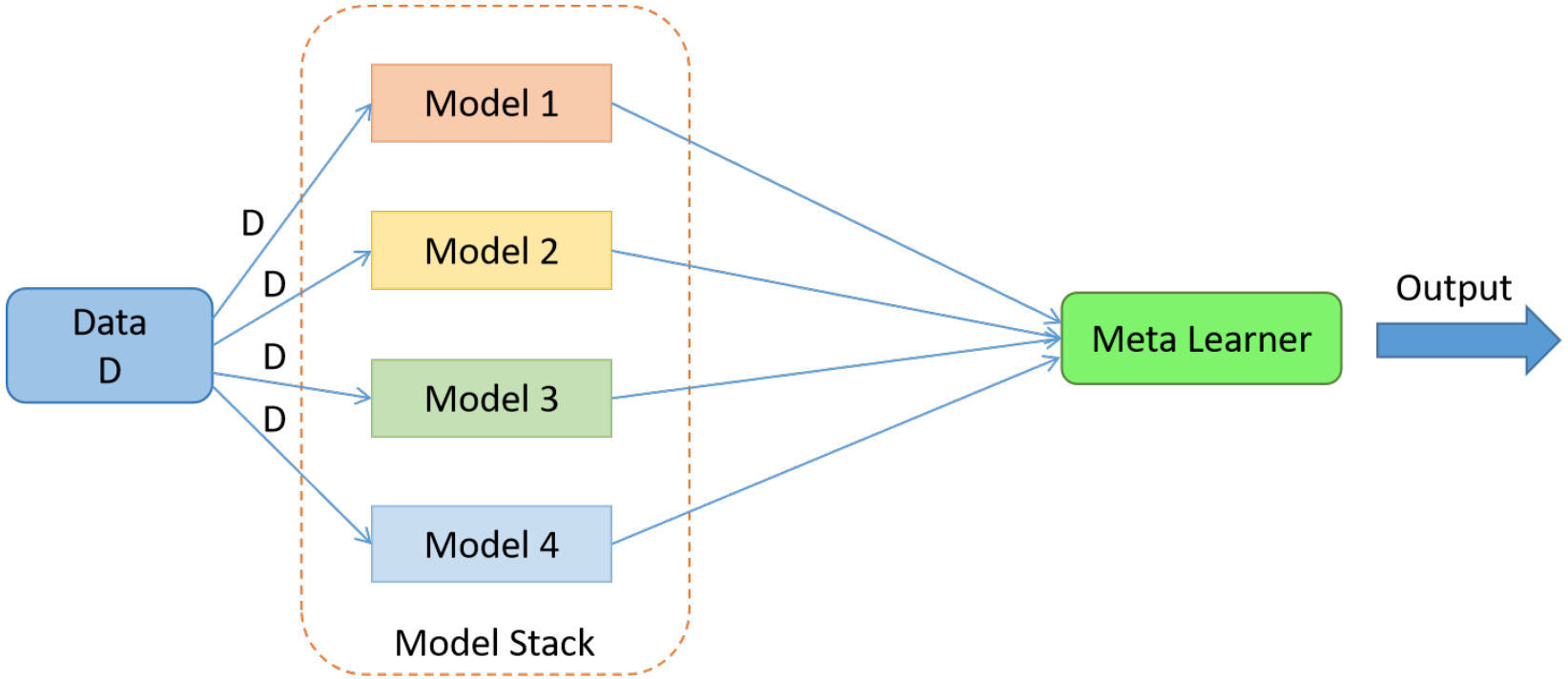
	C4.5		
Dataset	Standard	Bagging	Boosting Ada
breast-cancer-w	5.0	3.3	3.1
credit-a	14.9	12.1	12.6
credit-g	29.6	22.8	22.9
diabetes	28.3	21.9	22.3
glass	30.9	28.4	30.5
heart-cleveland	24.3	18.1	17.4
hepatitis	21.6	16.5	13.8
house-votes-84	3.5	3.6	4.4
hypo	0.5	0.4	0.4
ionosphere	8.1	6.0	6.0
iris	6.0	4.6	5.6
kr-vs-kp	0.6	0.5	0.3
labor	15.1	13.3	13.2
letter	14.0	10.6	6.7
promoters-936	12.8	9.5	6.3
ribosome-bind	11.2	9.3	9.1
satellite	13.8	10.8	10.4
segmentation	3.7	2.8	2.3
sick	1.3	1.0	0.9
sonar	29.0	21.6	19.7
soybean	8.0	8.0	7.9
splice	5.9	5.7	6.3
vehicle	29.4	26.1	24.8

CALVIN INSTITUTE OF TECHNOLOGY

Stacking

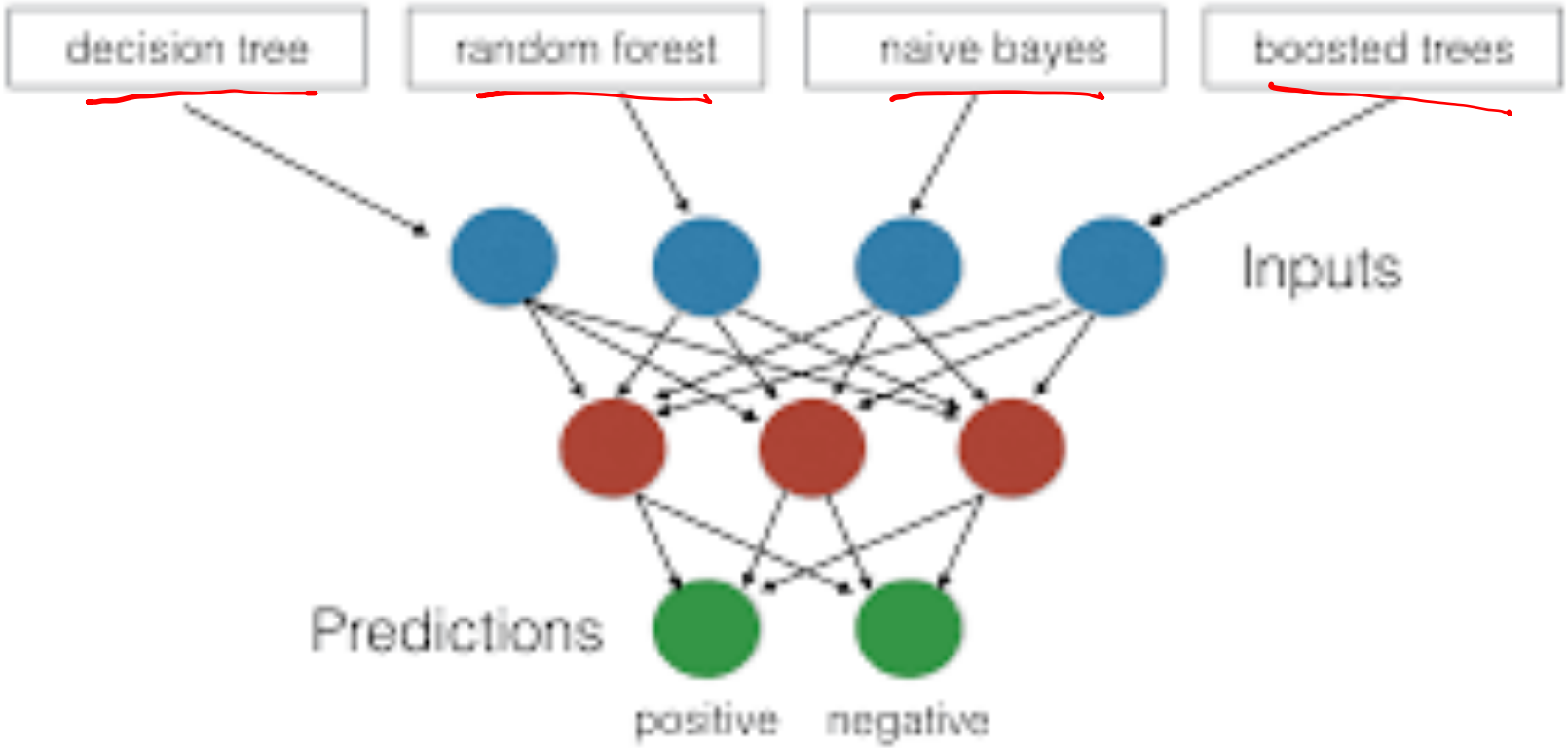
Stacking

- Gunakan bebagai jenis model yang berbeda (misalnya decision tree, naïve bayes, regresi logistik)
- Meta-learner: inputs = prediksi dari tiap model
- Proses pelatihan menggunakan cross-validation

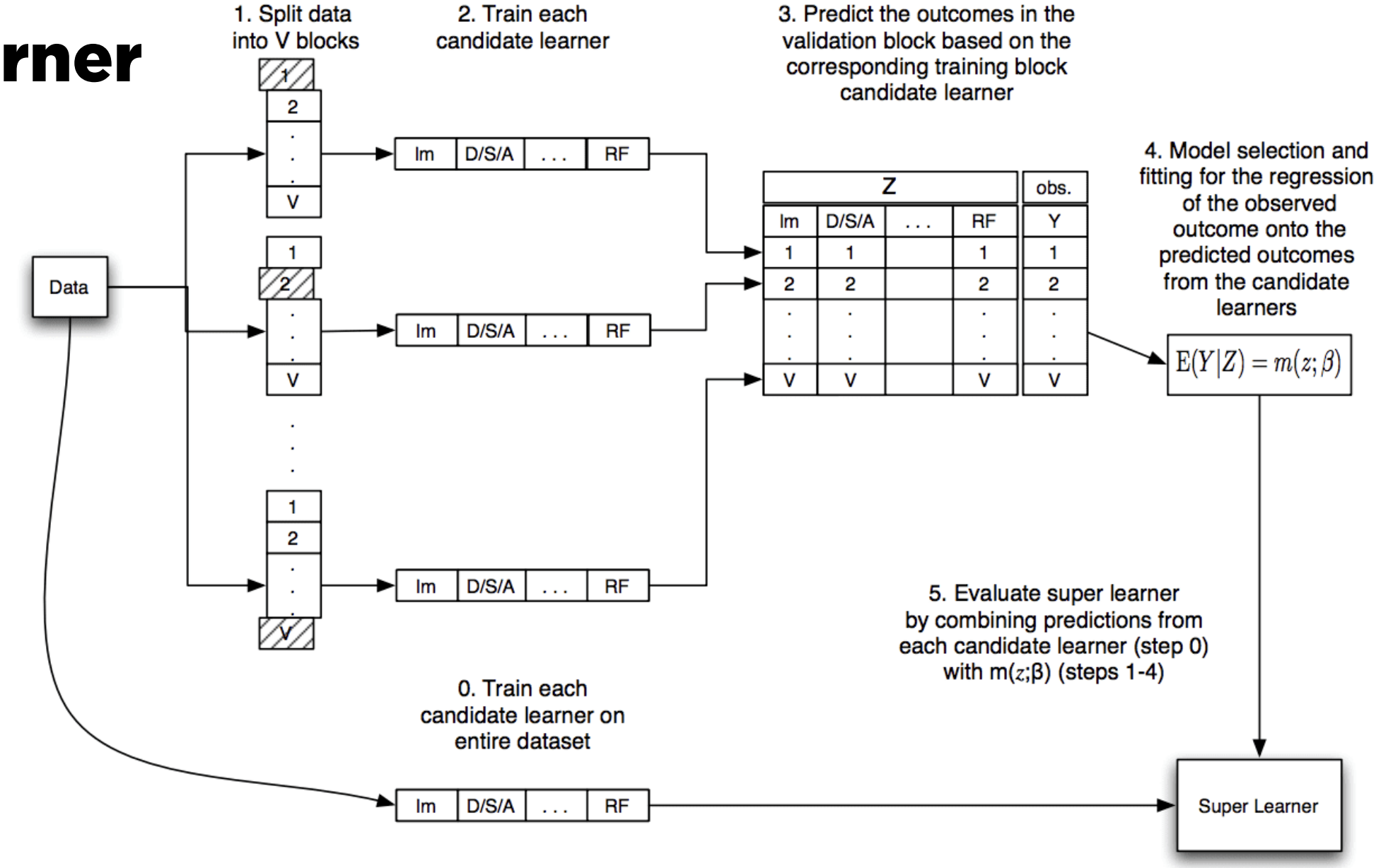


Superlearner

- Superlearner adalah generalisasi stacking dengan menggunakan out-of-fold selama proses k-fold validation
- https://machinelearningmastery.com/super-learner-ensemble-in-python/



Superlearner



Tuhan Memberkati