Agenda

→ DT Regression

→ Ensemble techniques

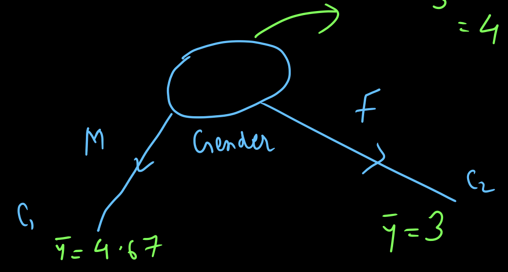
→ Bagging

→ Random Forest

DT Regression

$$\frac{2+3+4+5+6}{5} = 4$$

Gender	Education	y
F	G	2
M	NG	3
F	NG	4
M	NG	5
M	G	6



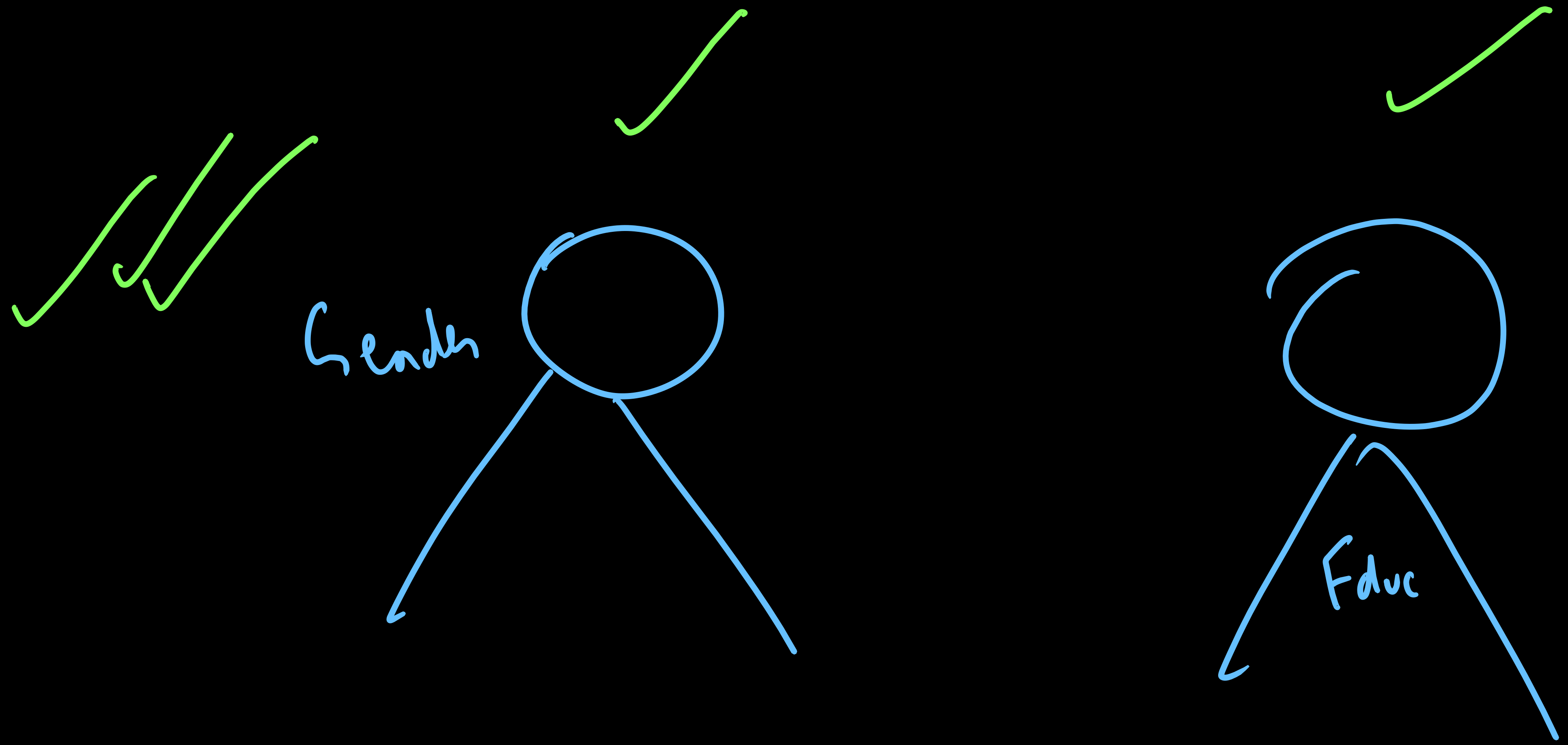
$$IG = H(P) - \left[\frac{n_1}{n} H(C_1) + \frac{n_2}{n} H(C_2)\right]$$

(i) Reduction in MSE $= MSE(P) - \frac{n_1}{n} MSE(C_1) + \frac{n_2}{n} MSE(C_2)]$

Reduction of Var $= Var(P) - \left[\frac{n_1}{n} Var(C1) + \frac{n_2}{n} Var(C2)\right]$

Max Info Gain = Max Reduction in Entropy

Max Reduction of Variance



→ Max Reduct in Variance

y	$\hat{y}$
2	4
3	4
4	4
5	4
6	4

$$MSE = \frac{1}{m} \sum \left(y^{(i)} - \bar{y}\right)^2 = \text{Variance}$$

$$MSE(P) = \frac{1}{5}\left[(2-4)^2 + (3-4)^2 + (4-4)^2 + (5-4)^2 + (6-4)^2\right]$$

$MSE(C1)$

$y^{(i)}$	$\bar{y}$
3	4.67
5	4.67
6	4.67

$$MSE(C_1) = \frac{1}{3}\left[(3-4.67)^2 + (5-4.67)^2 + (6-4.67)^2\right]$$

$MSE(C2)$ $\quad y^{(i)} -$

→ Ensembling

↳ Using mutiple model in order to improve it performance

1. Bagging
2. Boosting
3. Stacking

	y
2	3
4	3

$$MSE(C_2) = \frac{1}{2}\left[(2-3)^2 + (4-3)^2\right]$$

$$= MSE(P) - \left[\frac{n_1}{n} MSE(C_1) + \frac{n_2}{n} MSE(C_2)\right]$$

Reduction of MSE

$$= MSE(P) - \left[\frac{3}{5} MSE(C_1) + \frac{2}{5} MSE(C_2)\right]$$

Ensembling → iphone

Classification

M1 → LR

M2 → KNN

M3 → DT

Friends	Cousin	Parents/Wife
1	1	0

Mode

1

Regression

Task: Predict Cand Score

$M_1 \rightarrow 82$

$M_2 \rightarrow 90$

$M_3 \rightarrow 78$

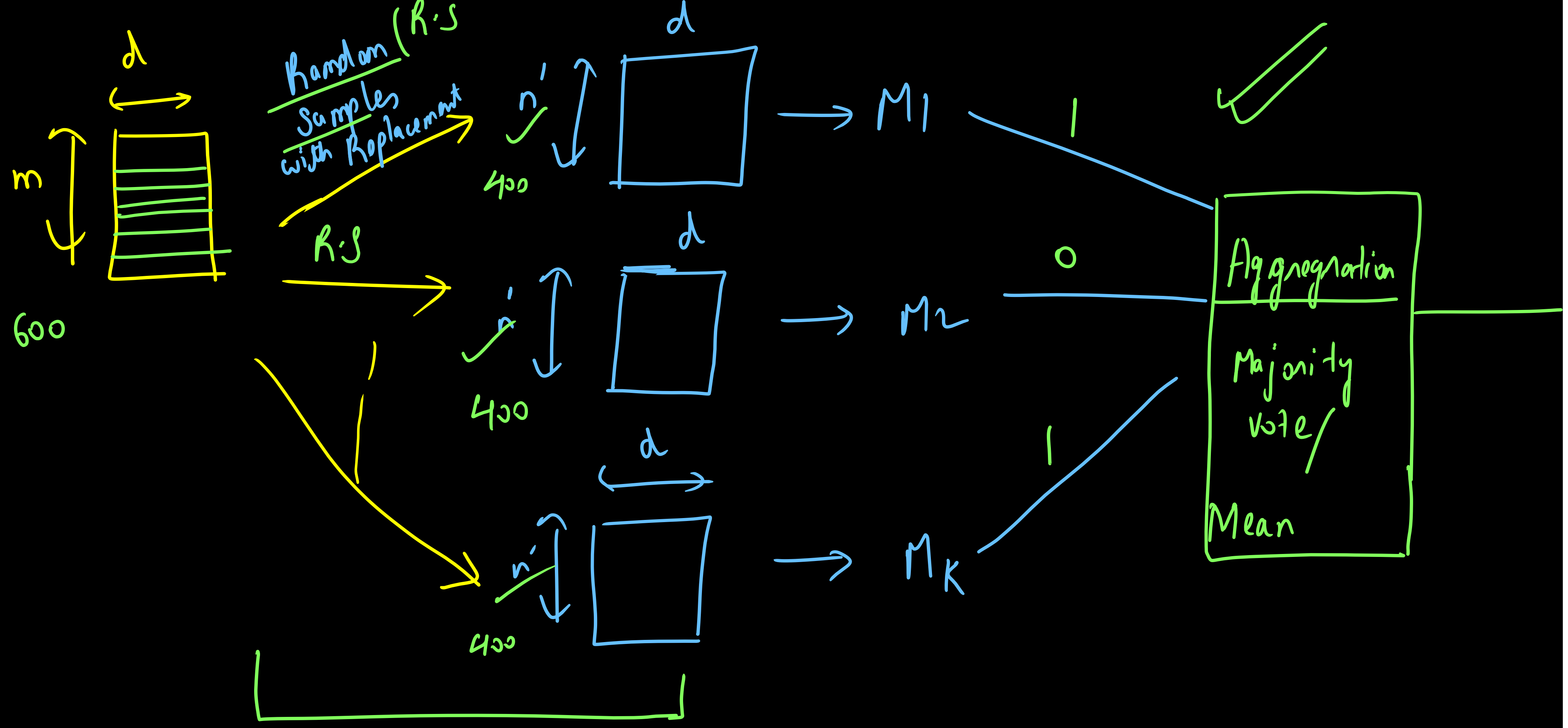
Mean/Median

$\downarrow$

83.3

$n' < m$

Bagging (Bootstrap Aggregation)

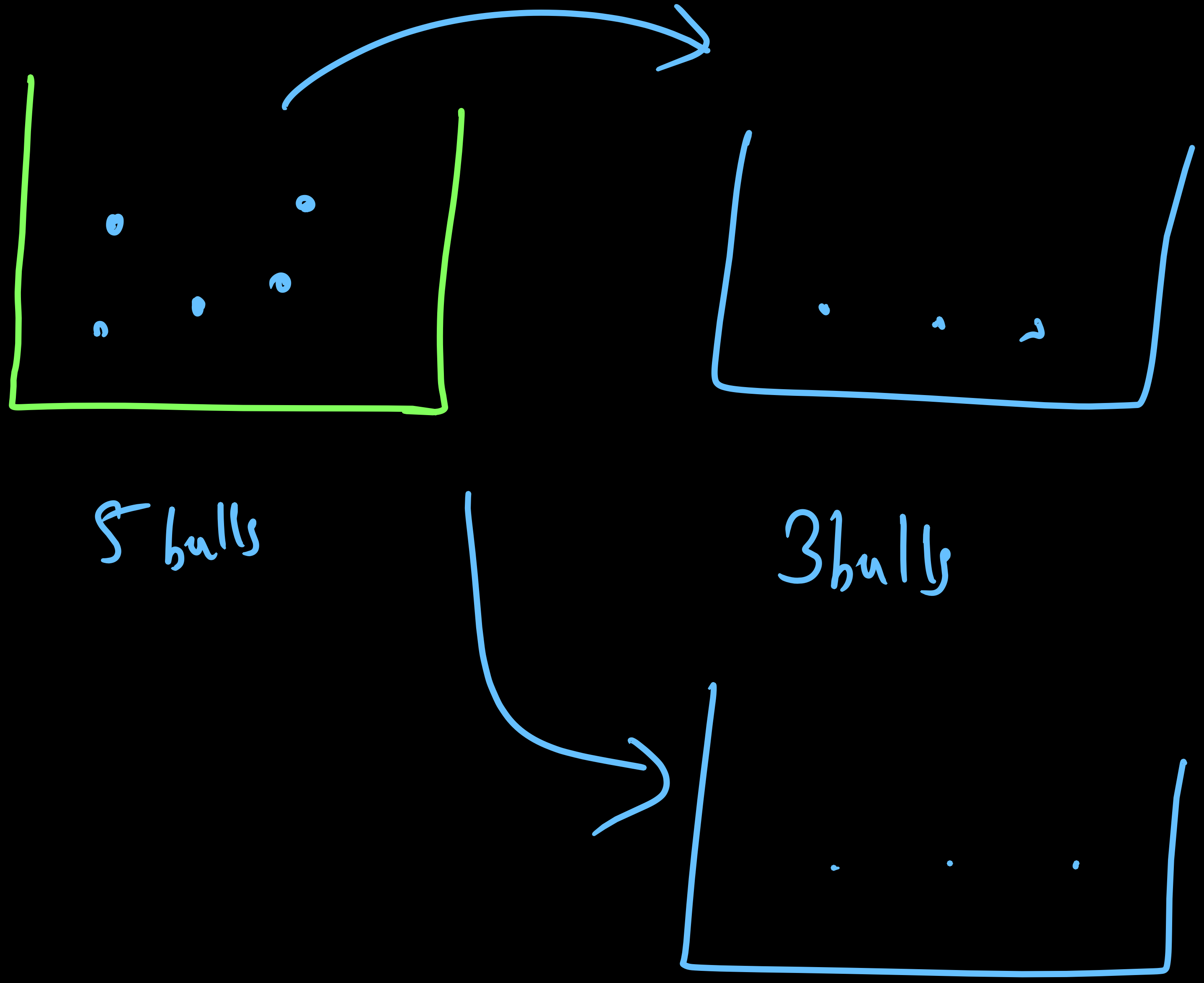


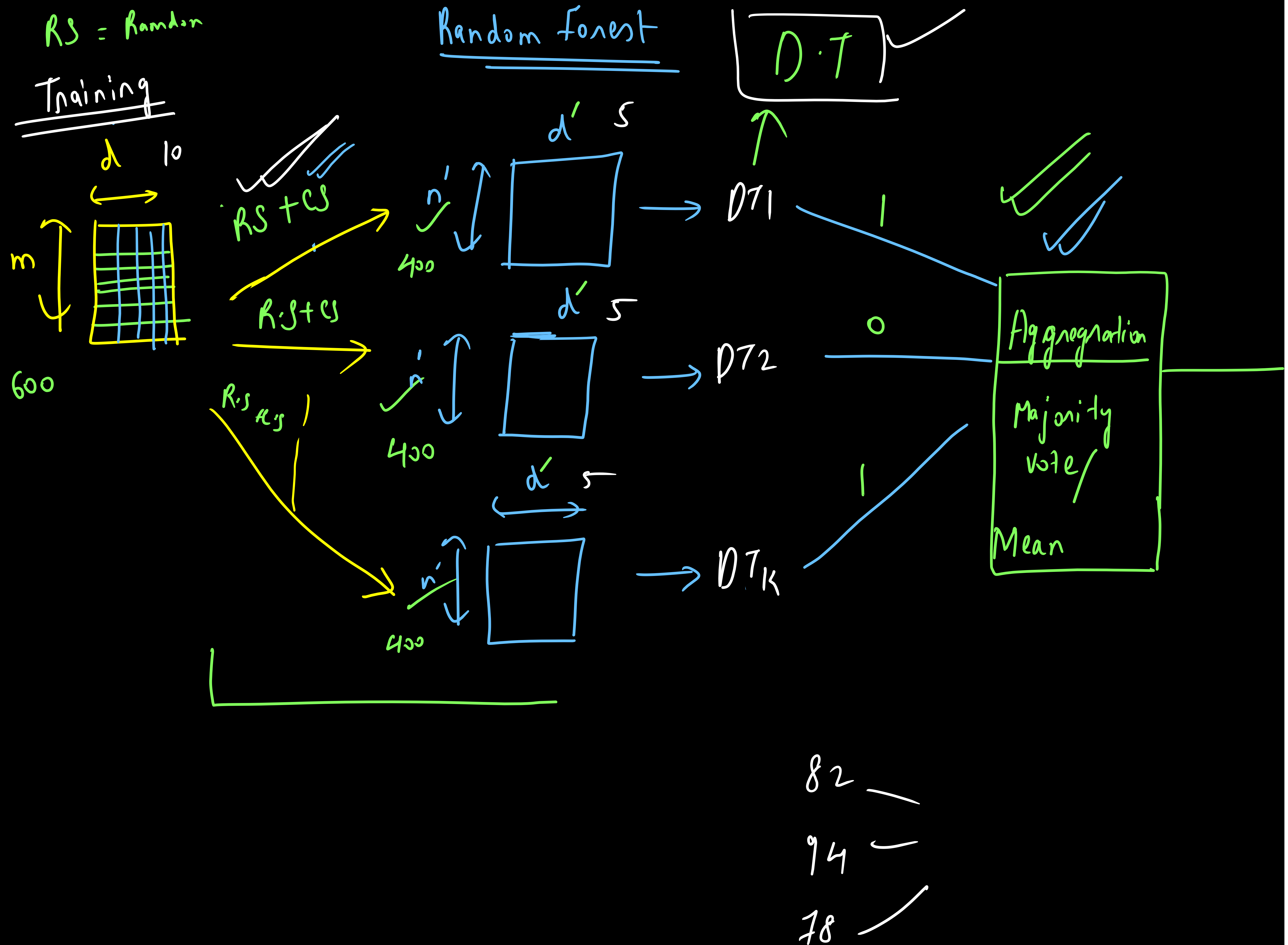
Bootstrapping $\rightarrow$ Random Sampling with Replacement

M1 , M2 - - Mk $\rightarrow$ logistic

LR LR LR $\rightarrow$ DT

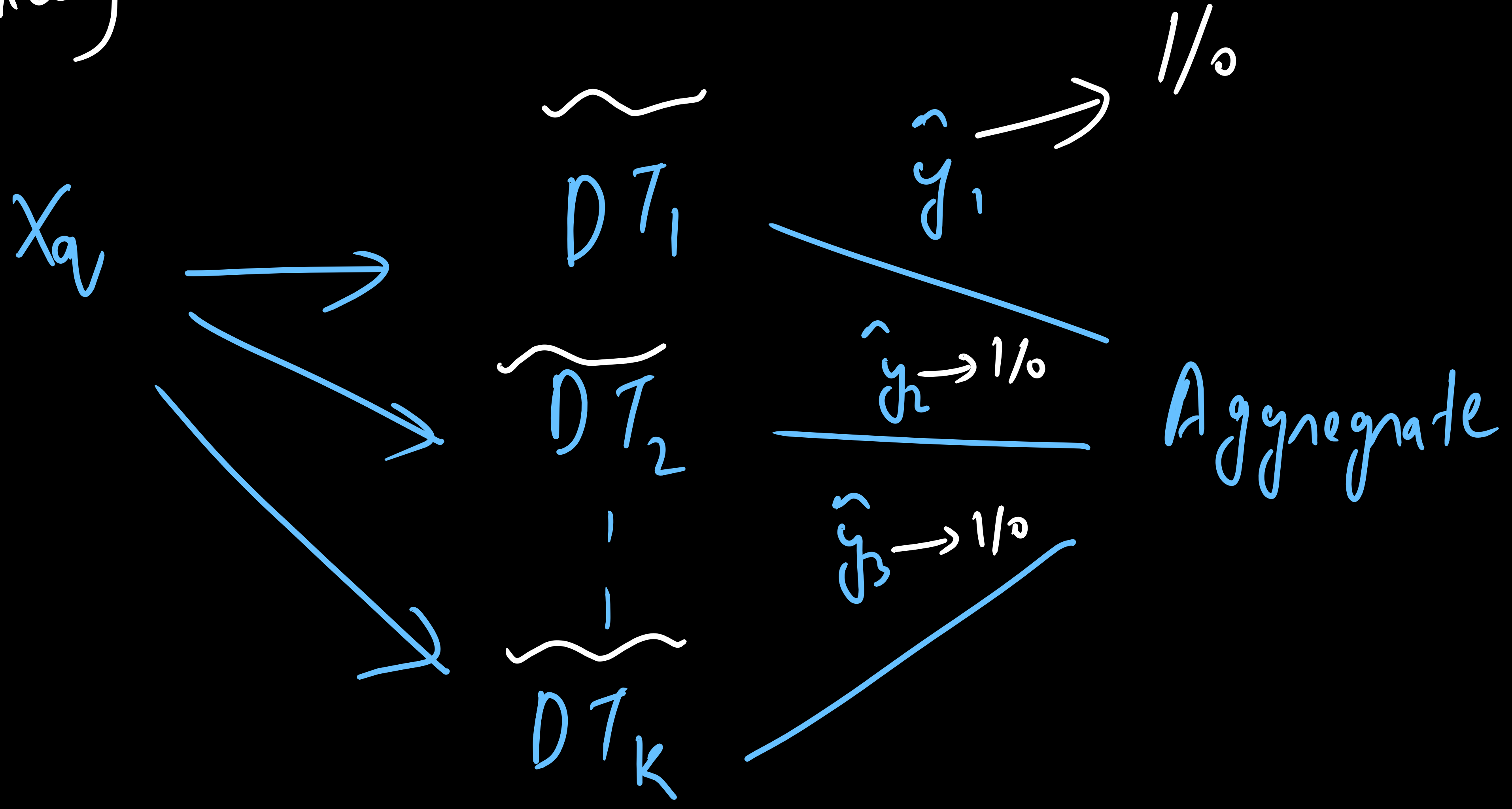
DT DT DT


5 balls
3 balls



RS = Random
Training
d 10
m
600
RS + CS
R.S + CS
R.S + C.S
Random Forest
D.T
d' 5
n'
400
d' 5
n'
400
d' 5
n'
400
DT1
DT2
DTK
1
0
1
Aggregation
Majority vote
Mean
82
94
78

Test (Inference)



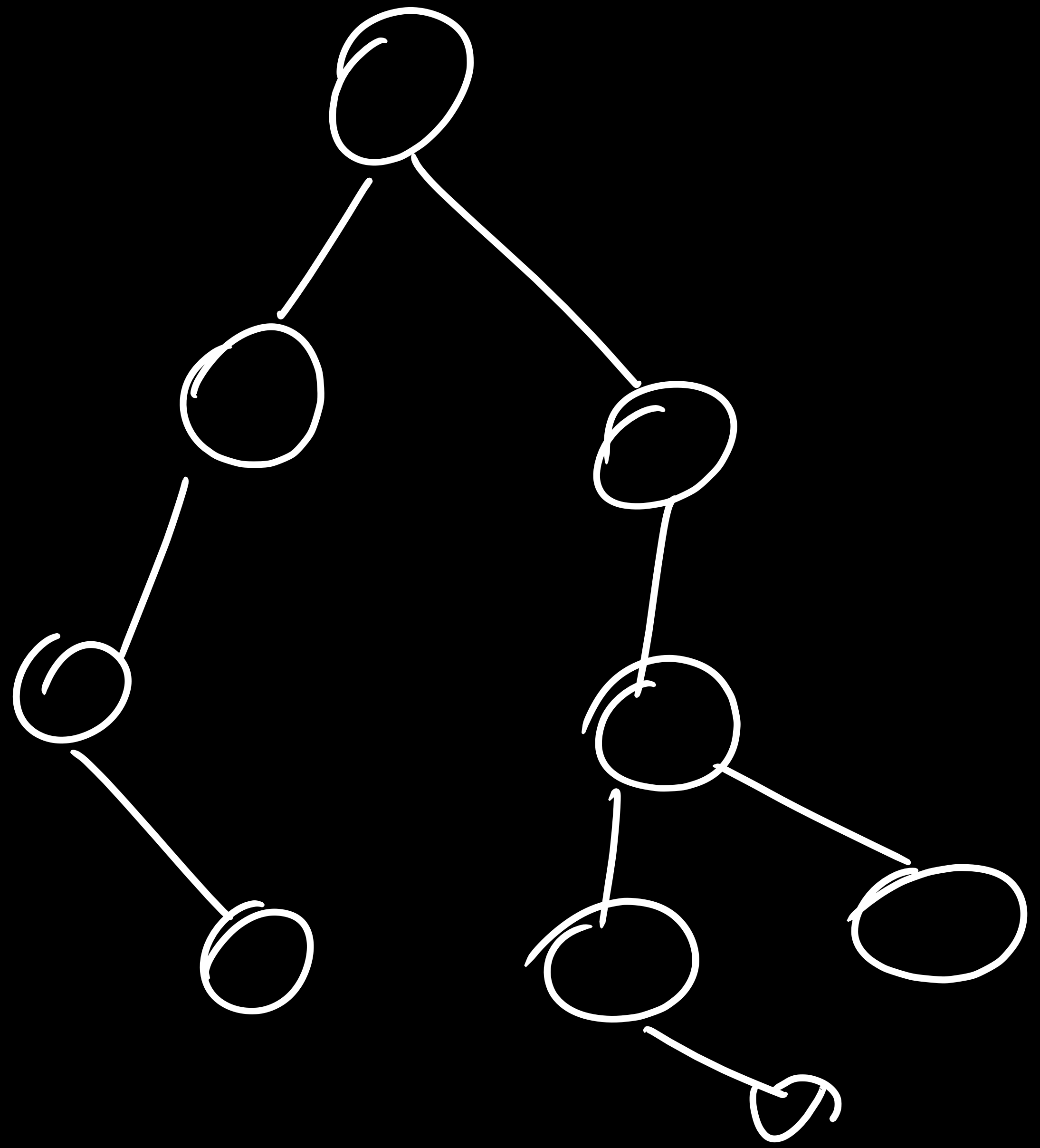
→ Break until : 22 : 27 PM

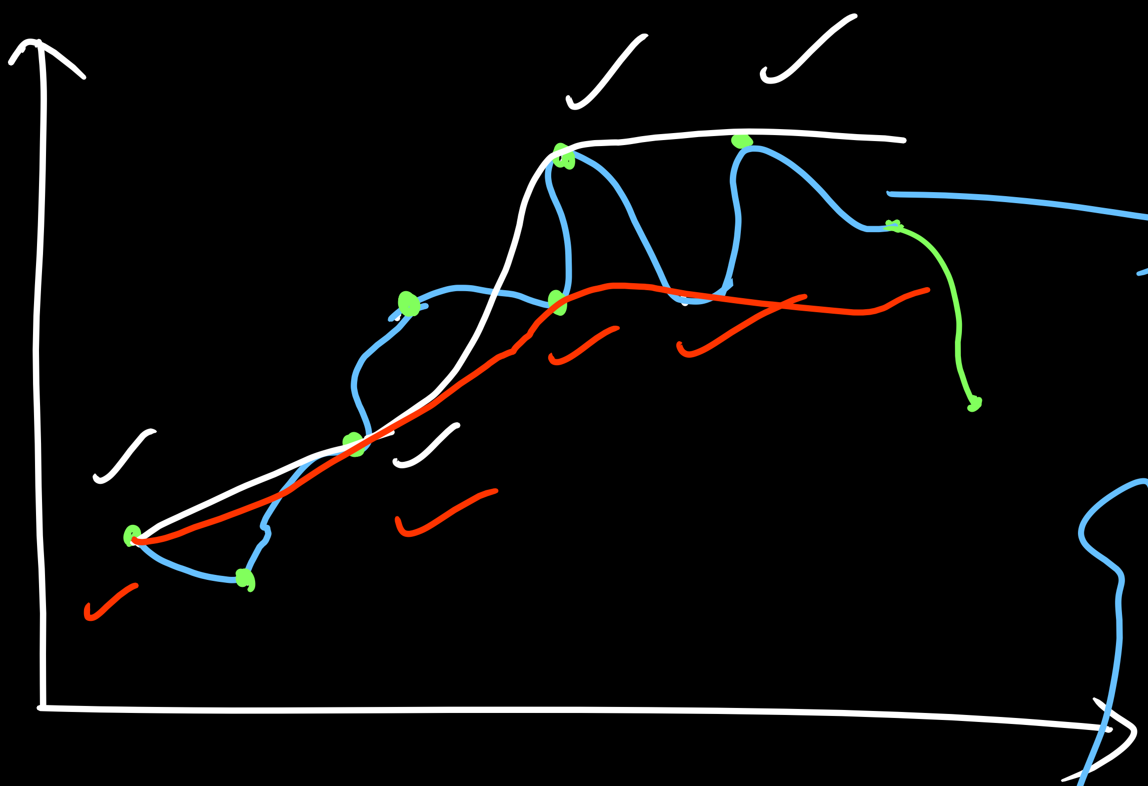
→ DT can easily overfit

↳ Low bias

&

High Variance





More complex model

K = 1 → Overfitting

K = 10 →

① Random Sampling with Replacement

② Aggregation

DT → Low bias & High Variance →

RF → Low bias & Low Variance →

Hyperparameter

1) n_estimator → No of trees

2) max_samples → RS

3) max_features → CS

4) Max_depth → depth

```
acc = []
for n_trees ____ in [1, 20, 100]:
    for depth in [1, 10, 100]:
        → model = DT(depth, n_trees)
        → model.fit(X_train, Y_train)
        ✓ acc.append(model.score(X_test, Y_test))
```

M_1

M_2

$\vdots$

M_K

$\rightarrow$ Grid Search

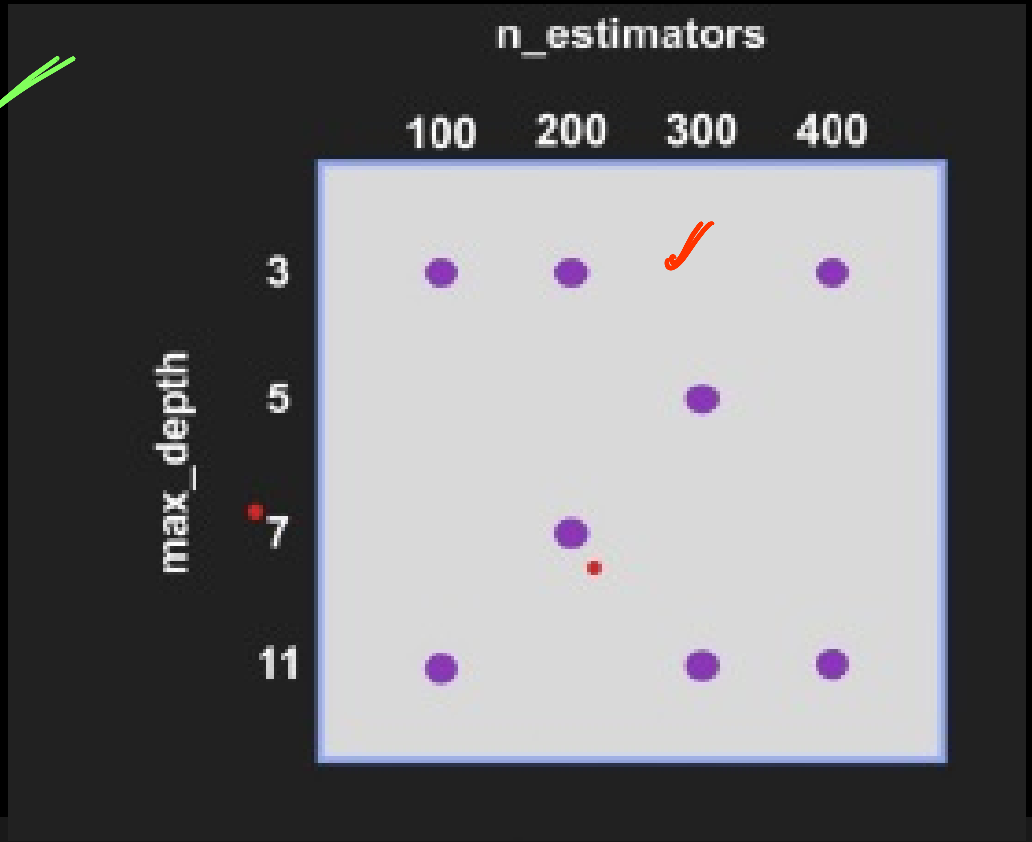
$n \times m \times d$

m

n

$\rightarrow$ Time Consuming

→ Randomized Search



→ Faster

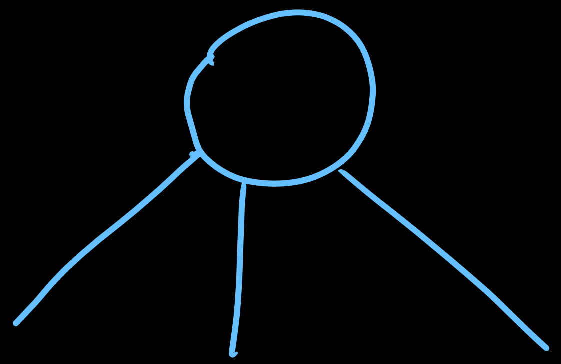
→ May not give best result

→ Baseyian Optimization (Hyperopt, Optuna)

↳ Fast like Random Search

↳ Accuracy like Grid Search

	Y	-R	-G	-B
	R	1	0	0
	G	0	1	0
	B	0	0	1



Sklearn