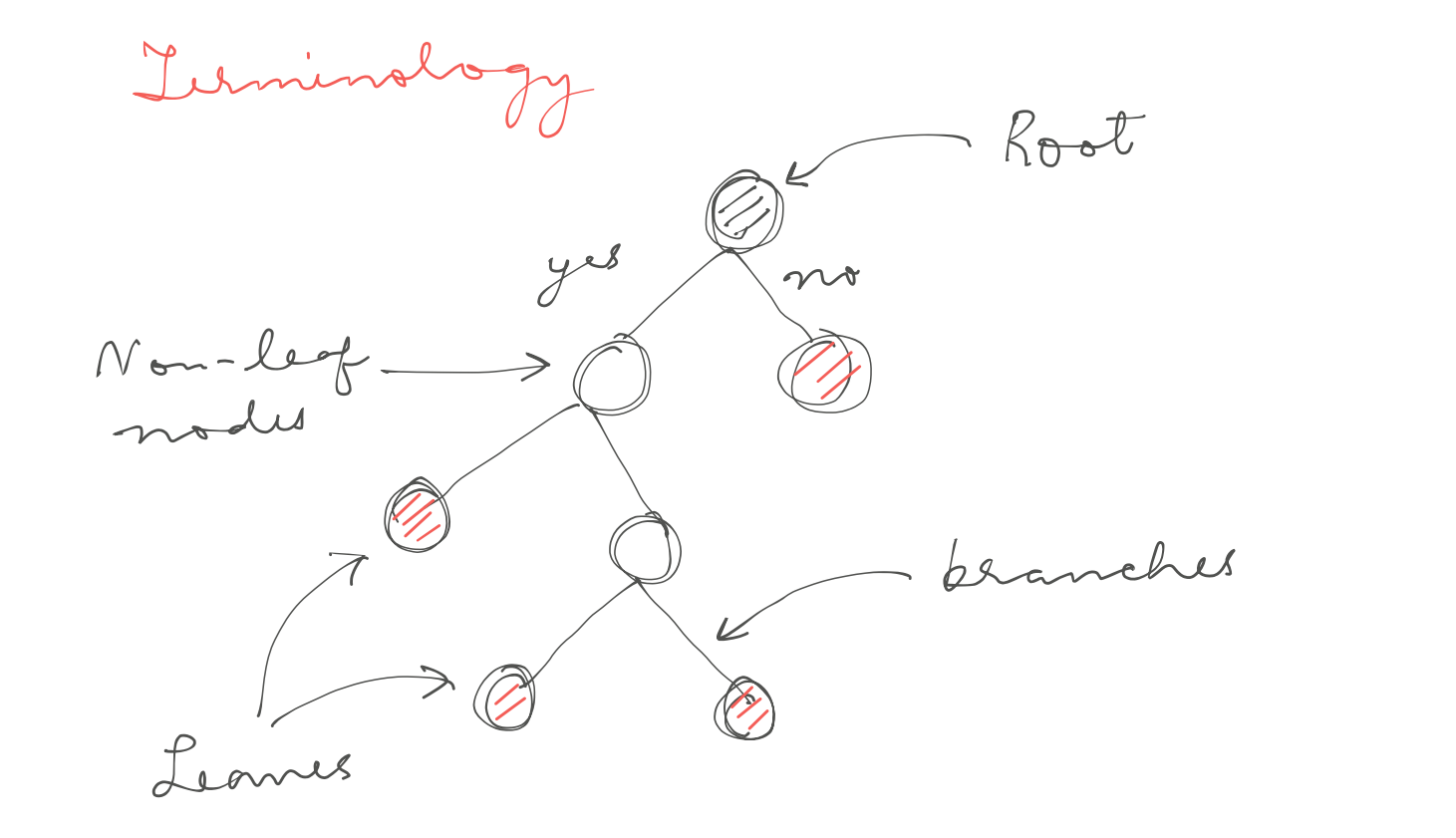
Decision Trees

Series of yes/no questions
↳ Smartphone

Terminology



Construction

Greedy splitting
↳ Classification → Impurity
↳ Regression → MSE

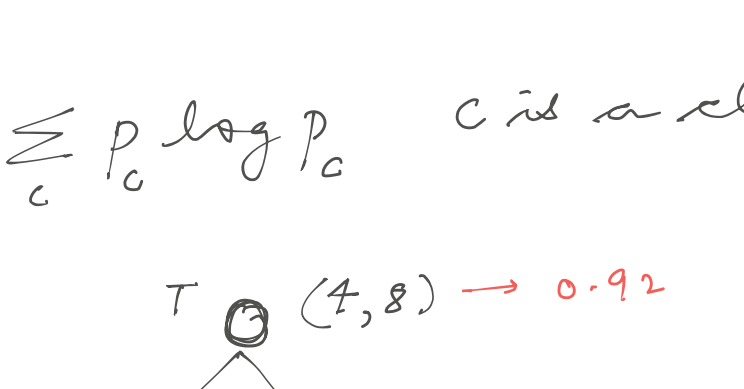
Stopping
100% training / no further improvement
Pre-pruning
↳ max depth
↳ # leaves
↳ # data points in a node
Post pruning

Impurity functions

1. Entropy → Information gain
2. Gini Index ⟶ ↑

↳ Entropy

$\underline{H(P)} = -\sum_c P_c \log P_c$ c is a class



$H(T) = -p_0 \log_2 p_0 - p_1 \log_2 p_1$

$= -\frac{4}{12} \log \frac{4}{12} - \frac{8}{12} \log \frac{8}{12}$

$= 0.53 + 0.39$

$= \boxed{0.92}$

$H(L) = -\frac{4}{6} \log \frac{4}{6} - \frac{2}{6} \log \frac{2}{6}$

$= 0.92 \leftarrow$

$H(R) = -\frac{0}{6} \log \frac{0}{6} - \frac{6}{6} \log \frac{6}{6}$

$= -0 - 0$

$= 0$

Weighted average

$\longrightarrow \frac{6}{12} H(L) + \frac{6}{12} H(R)$

$= \frac{1}{2} 0.92$

$= \boxed{0.46}$

Gini Index

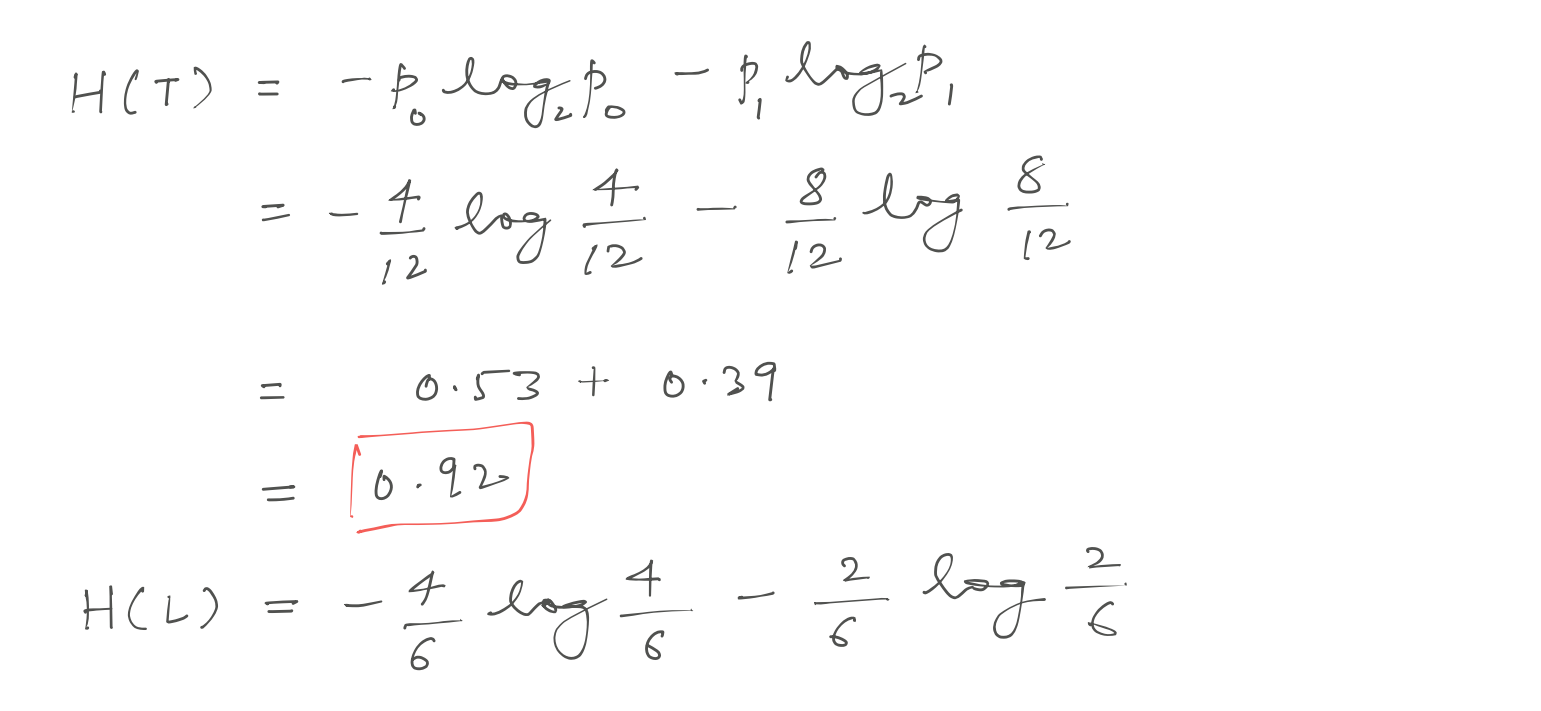
$G(P) = 1 - \sum_c P_c^2$

Stopping criteria

↳ 100% training acc.
↳ Prepruning
↳ Post-pruning
  ↳ Validation error
  ↳ CCP: Cost Complexity Pruning

Pruning based on val error

Train - Val - Test

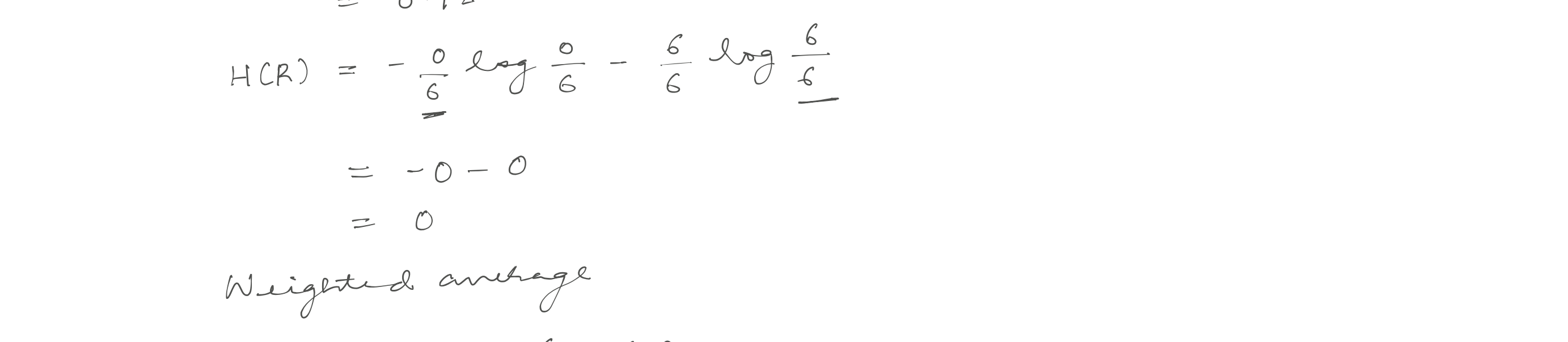


Bottom-up

CCP → Cost Complexity Pruning

$R_\alpha(T) = R(T) + \alpha |\tilde{T}|$

complexity: # leaves



weighted sum of H(P)
↳ R(T)
Total misclassification

1. Train till 100% training acc is achieved.
2. Compute the effective α (α_E) of all non-leaf nodes both of whose children are leaves.

$R_\alpha(t) = R(t) + \alpha$ T → Tree where t is root

$R_\alpha(T) = R(T) + \alpha |\tilde{T}|$

$R(t) + \alpha = R(T) + \alpha |\tilde{T}|$

$\Rightarrow R(t) - R(T) = \alpha |\tilde{T}| - \alpha$

$\Rightarrow R(t) - R(T) = \alpha (|\tilde{T}| - 1)$

$\Rightarrow \alpha = \frac{R(t) - R(T)}{|\tilde{T}| - 1}$

if binary & splitting at penultimate layer

$\alpha = R(t) - R(T)$

3. Prune the leaves of the node that has the minimum α_E.
4. Repeat this process till only the root node is left. OR Repeat till α_E is greater than a threshold.
5. Save the pruned tree at each iteration
6. After stopping → Test each pruned tree on the val set.
7. Pick the tree that performs best on the val set.

Bias / Variance

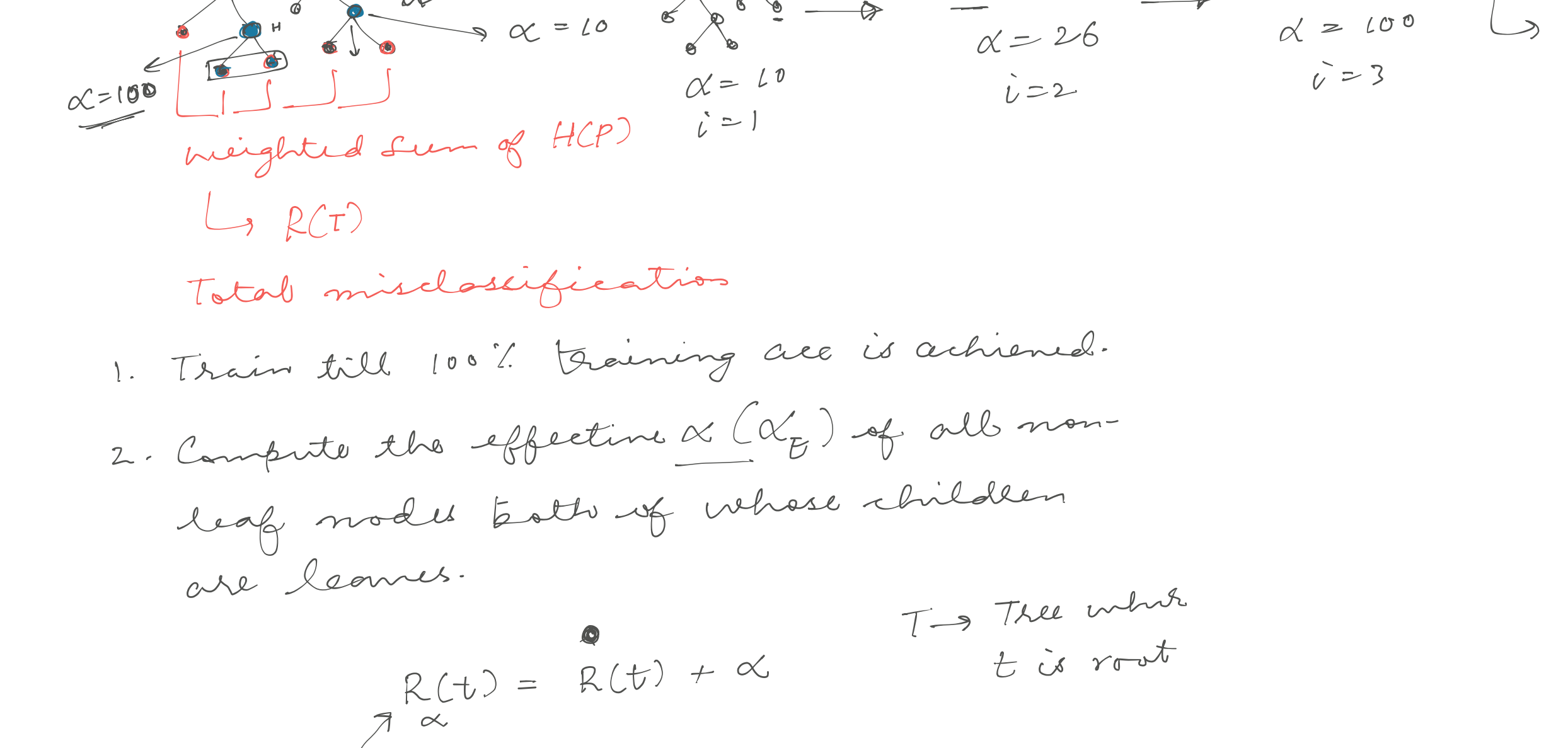
↑Bias → Your model is fundamentally incapable of performing well

↑Var → Overfit

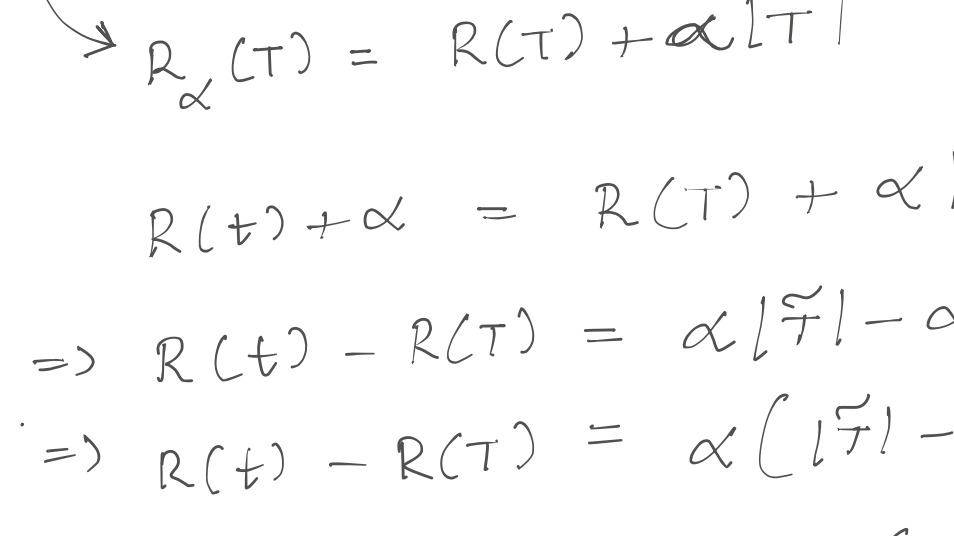
Bagging → Bootstrapped Aggregation

D → Tr / V / Test

Tr → Tree
  ↳ Use on test



Random Forests

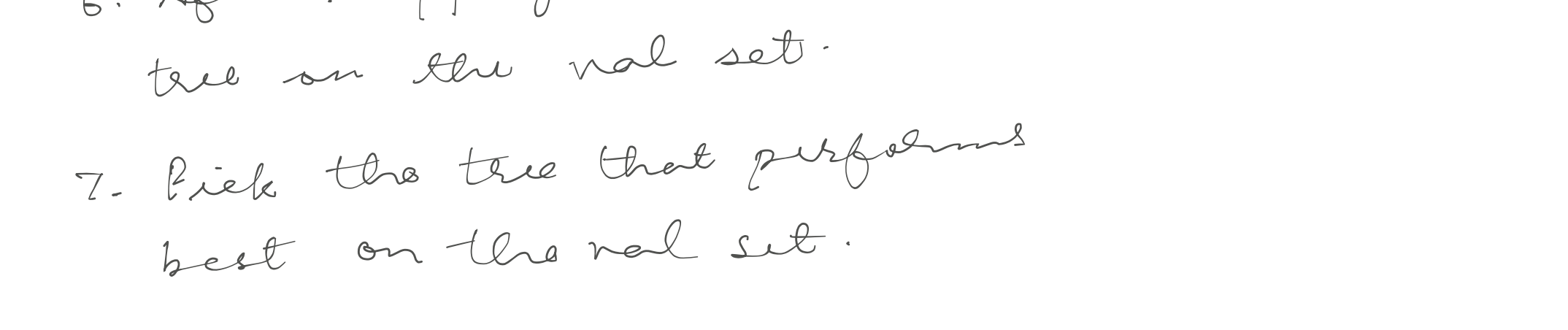


Bagging → $T_1, T_2 \ldots T_m$

Rather than greedy split on the full set of features
↓
Greedily split on a randomly sampled set of features.

Bias → Boosting

Combining multiple weak learners to create a strong learner.
↳ AdaBoost
↳ Gradient Boosting
↳ XGBoost



5 boost

D, T → clone
  ↳ say
Tree 80%

D, T₁ → say
  ↳ increase weight of misclassified instances
  ↳ reduce the w of other

Each tree gives more weightage to the previous trees's mistakes.

Voting → Soft vote
  ↳ say

Summarise

↳ Sequence of decisions
↳ Terminology
↳ Construction
  ↳ Greedy
↳ Split
  ↳ C → Impurity
  ↳ R → MSE
↳ Stopping
  ↳ 100% acc train
    ↳ Overfit
    ↳ Huge complex trees
  ↳ Prepruning
  ↳ Post-pruning
    ↳ Validation error alone
    ↳ CCP → $\alpha |\tilde{T}|$
↳ Bias / Variance
  Variance → Bagging → Sampling with replacements
  Bias → Boosting