BOOSTING → Additive Generalised Models.

↳ Bagging → Motivation → Reduce Variance.

↳ Large trees → Low bias + High-Variance

↳ If trees are uncorrelated (good if anti-correlated)

$\frac{n \times \sigma^2}{n^2}$ ← then aggregate variance will go dn

Random forest

↳ Decorrelates the trees

↳ predictor set changes

↳ data sample bootstrapped.

→ Not possible with other predictors

Beauti of trees → Even with new set of features

→ with large depths → the tree will reduce bias

↳ ultimately it can create a linked list.

Boosting → Sequential learning

↳ Trees learn on the residual of the previous predictor

↳ concepts of sample-weights will be used

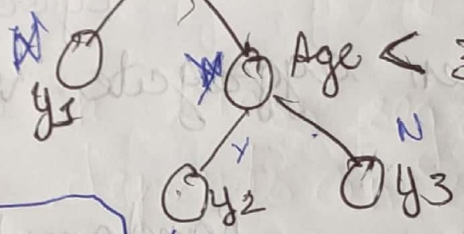
↳ $\propto$ error at ith observation (Prop)

Will be used while training the next tree.

A simple tree → if-else based non-parametric model.

2 predictor X → Age + Education.
↳ Y → Salary.

Depth = 2



Ed ≥ Bachelor → Indicator function

Age < 30 → Indicator fn.

Bivariate spline (Indicator)

$[\ f(Ed == Bach) \times f(Age < 30) \times y_2 +$
$f(Ed == Bach) \times f(Age \geq 30) \times y_3 +$
$f(Ed \,!= Bach) \times y_1\ ]$ → This whole is on Non-parametric F applied on (X_1, X_2)
Age ↲ ↳ Education

→ weighted prediction (At last)

Multiple such F → Random Forest

↳ sequential F's with different $\boxed{Y}$ — Target vars.

Boosting.

ADA

⇒ Boosting → A sequence of weak learners that train of modified data-set
=

↳ Additive models + Forward stagewise modelling
↳ does not affect the older models.

Loss is not just non or/ parametrically solved (like done is LR or trees) but also incorporates sequential contribution of previous function.

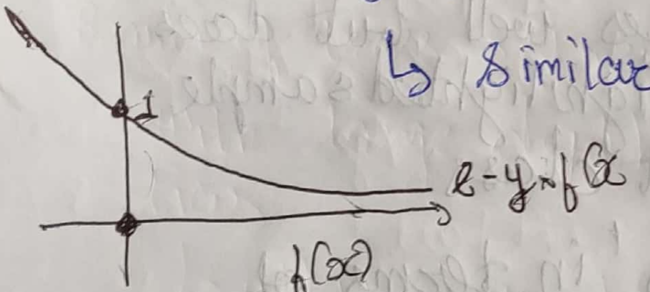
$\mathcal{L}(y, f) \rightarrow \mathcal{L}(y, f_{i-1} + w_i f_i)$ → weight of new tree

↳ Square loss for Reg-
Binomial for Classi-

f_i → Are tree based structures/classifiers

Loss $= e^{(-y \times f(x))}$ → Defined.

↳ Similar to $-y_i \log(f(x_i))$



Say y true = 1.

↳ give 1 for

→ Robust to outlier, scaling, nans

Why tree → Can deal with mix data-type.

↳ Can provide low bias with higher depths.

↳ Interpretable + Non-parametric.

Splitting criterion

↳ Entropy / gini

↳ Weighted gini

Parent ent → Fixed

↓

Will select the splitter that gives the max dip in the entropy (weighted entropy)

↳ Depends on class segregation

↳ Also depends on sample wts.

$\sum$ Number x sample wt ↳ $\frac{N_{node-left}}{N_{parent}}$

↓

Interpretation.

If a splitter segregates well but does not classify the highlighted sample all alone in the right class

Then there will be loss in terms of increase in entropy

↳ That splitter will never be selected