· iloc[2:4] = 2 3

· loc[2:4] = 2,3,4

in series

· loc works with numbers

↱ Regularisation Hyperparameter Tuning of Regression.

2

Ridge and lasso Regression

→ Sum of Residuals → $\sum_{i=1}^{n} (y-\hat{y})^2$ $[\hat{y} = mx + c \rightarrow b_1 x + b_0]$

→ MSE → Cost Func → $\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)$.

- Overfit → low Bias + High variance.
- Underfit → High Bias + [redacted] variance.
  Low

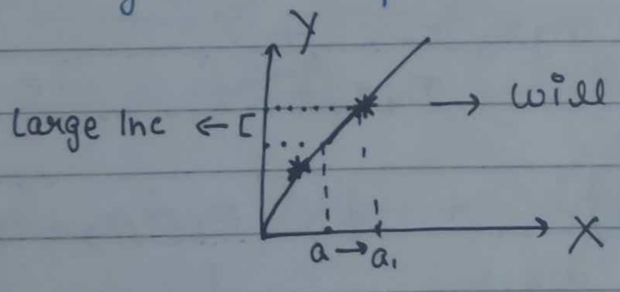
Best fit - general Model
Low Bias + Low Var.

→ We can use Ridge and lasso to solve the problem of overfitting (High variance) which will convert High variance to Low Variance.

· RIDGE Regression

$$CF = \left[\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda.(\text{slope})^2\right]$$ Reduce whole operation.

★ Problem of Steep slope will cause Overfitting.
→ moving a unit step in x - causes a large ↑ of slope



→ will cause overfitting.

[Penalize steeper (Higher) Slopes]

- λ - can be B/w 0 and +ve value

★ look for another line which can reduce the CF.

· RIDGE → Penealizing Steeper (High) slopes.
→ line with High slopes are Penealized by using CF (Reducing CF → checking for new lines with ↓m)

↑

Imp Concept

near to 0.

$\boxed{\lambda \uparrow}$ - tends to 0 line never reach 0.

Slope $\rightarrow 0$

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda (m)^2$$

$\lambda (m_1 + m_2 \ldots)$

huge

- $y = mx + c$

★ $\lambda \rightarrow$ selected by cross val

→ It will be Penalized By reducing the CF in Ridge Reg

★→ A new line will be checked which has lesser slope. - way to Penalized - looking for low slope value. - reduces test data error.

—— x —— x —— x —— x —— x ——

- Lasso Regression

$$CF \rightarrow \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = 0 + \lambda \times |\text{Slope}|$$

magnitude

→ helps in Reducing overfitting

- Line with less steepe slope

+

→ helps in feature selection.

→ $\lambda \uparrow$ the |slope| will move towards 0.

→ $y = m_1x_1 + m_2x_2 + m_3x_3 + \ldots + C_1$

$m_4x_4 + \ldots$

$$\Rightarrow CF = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \times |m_1 + m_2 + m_3 + m_4|$$

$mx+b$

moves towards 0

- So features with low Slope will be Removed and we will be left with → Important features.

Esemble learning - Boosting

(combining Diff models)

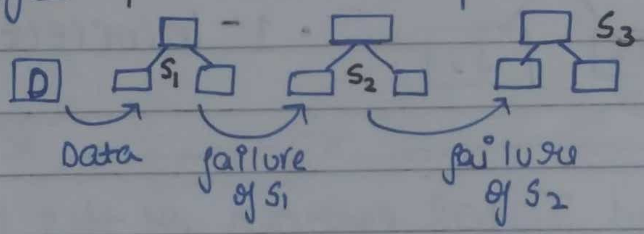
ADABOOST

→ ADABOOST → Adaptive Boosting.

a) Combines multiple weak learners into a single strong learner

b) Does not follow Bootstrap aggeration Algo But creates sequential models → models creating in a sequential manner.

c) It creates different DT in sequential manner with a "single split" (one depth - level 1). Called STUMPS (S)



[DT].

No. of STUMPS = No. of features/Columns in Dataset

weigths are assigned. - Working

f_1	f_2	f_3	output	Sample weight
→	R_1	←	——	1/7
→	R_2	←	——	1/7
→	R_3	←	——	1/7
→	R_4	←	——	1/7
→	R_5	←	——	1/7
→	R_6	←	——	1/7
→	R_7	←	——	1/7

$\boxed{\text{Total samples} = 7} = n.$

STEP 1

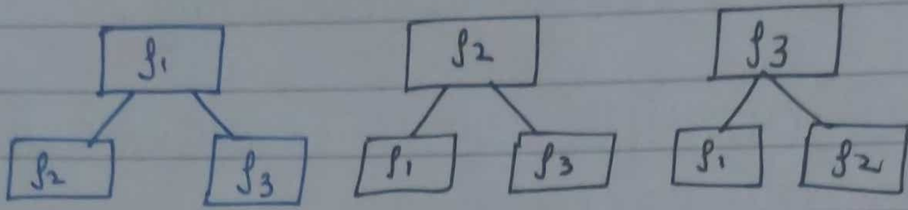
Sample weigths are assigned to Records.

$\frac{1}{n}$ ← samples

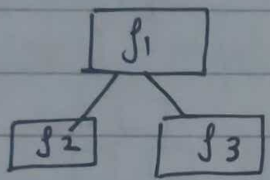
DT (1 one split)

STEP-2 → Create 1st STUMP sequentially.

→



→ STumPs of all features are created – Using Entropy or gini Index the DT of that feature is selected as 1st Base Model.



→ • 6-correct
• 1-Incorrect → failure.

STEP-3 → Find Total error of the Incorrect – failure

→ T → Total Incorrect
N → Total sample
TE → T/N

[TE → Perf → update weig]

→ TE → 1/7.

STEP-4 → Performance of STUMP

$$\Rightarrow P = \frac{1}{2} \log_e \left(\frac{1-TE}{TE} \right)$$

$$P = 0.896$$

Step-5 UPDATE WEIGHT – incorrect weights ↑
→ correctly weights ↓

→ For incorrect

→ UPDATE = new weight = Old weight × e^{P}

For correct

→ new weight = new weight × e^{-P}

→ $\frac{1}{7} \times e^{0.896} = 0.349$

→ $\frac{1}{7} \times e^{-0.896} = 0.05$

	UW	Normalized	
1/7	0.05	0.07	→ 0 – 0.07
1/7	0.05	0.07	0.07 – 0.14
1/7	0.05	0.07	0.14 – 0.28
Inc ← 1/7	0.349	0.51	0.28 – 0.58
1/7	0.05	0.07	0.58 → 0.65
1/7	0.05	0.07	0.65 – 0.72
1/7	0.05	0.07	0.72 – 0.79
	0.68	1	

Sum (UW ≠ 1)
L Normalized

$\text{Norm} = \frac{UW}{\text{Sum}(UW)}$

— × — × — × —

STEP-6 Create Buckets / Intervals of Norm weight to create new Dataset for 2nd Stump.

STEP 7 → Generate N iterations to select Random value from old Data set. and check where it lies and that row interval is given to new sample set

L mostly wrong classified are sent to new sample set.

——— x ——— x ———

For 2nd - 3rd All other steps are repeated.

——— x ——— x ——— x ———

GENERAL

STEPS OF

ADA BOOST →

General steps

1] We assign equal sample weigth to each Observation. (1/n)

2] We will create M Decision STumps for M number of features To select our 1^{st} Base DT Model

3] Out of all M $\boxed{DS}$ we select one Best DT using entropy or giniIndex (lower - more pure).

4] The 1^{st} Base Model will work and we will check its failure (incorrectly classified)

5] For 'T' number of failure we will calculate TOTAL ERROR.

- Total observation = N
- Total error ⇒ T/N

6] Using TE calculate Performance of 1^{st} Decision stump.

- Performance (P) ⇒ $\frac{1}{2} * \log_e \left(\frac{(1-TE)}{TE} \right)$

7] UPDATE The WEIGHTS

- weights of failure / incorrect Classified Observatio

$$= \boxed{\text{old weight} * e^{P}} = \text{value}$$

the weights are increased

- weights of correctly classified are updated and the value is reduced

  └ $\boxed{\text{old weight} * e^{-P}}$

[Step 8]. Check if sum of updated weights == 1

If NOT - Then Normalized the updated weights.

$$\text{Normalized weight} = \frac{\text{Each updated weight}}{\text{Sum(updated weight)}}.$$

[Step 9] → CREATE A New Dataset Based on Normalized Weights. and make next Decision STUMP.

a) Divide Normalized weights in Buckets or intervals.
   └ Data lies in region (Divided in Regions).

→ Then we run N no. of interations, On each iteration it will calc Random number Ranging B/w 0 and 1 and this no. is compared to the Buckets / intervals.

└ in ~~which~~ which interval the Random No, lies, that record will be selected for sample Date set. (N observations).

→ whole Process Again runs for M $\boxed{DS}$ | Final sea Tree is considered final.