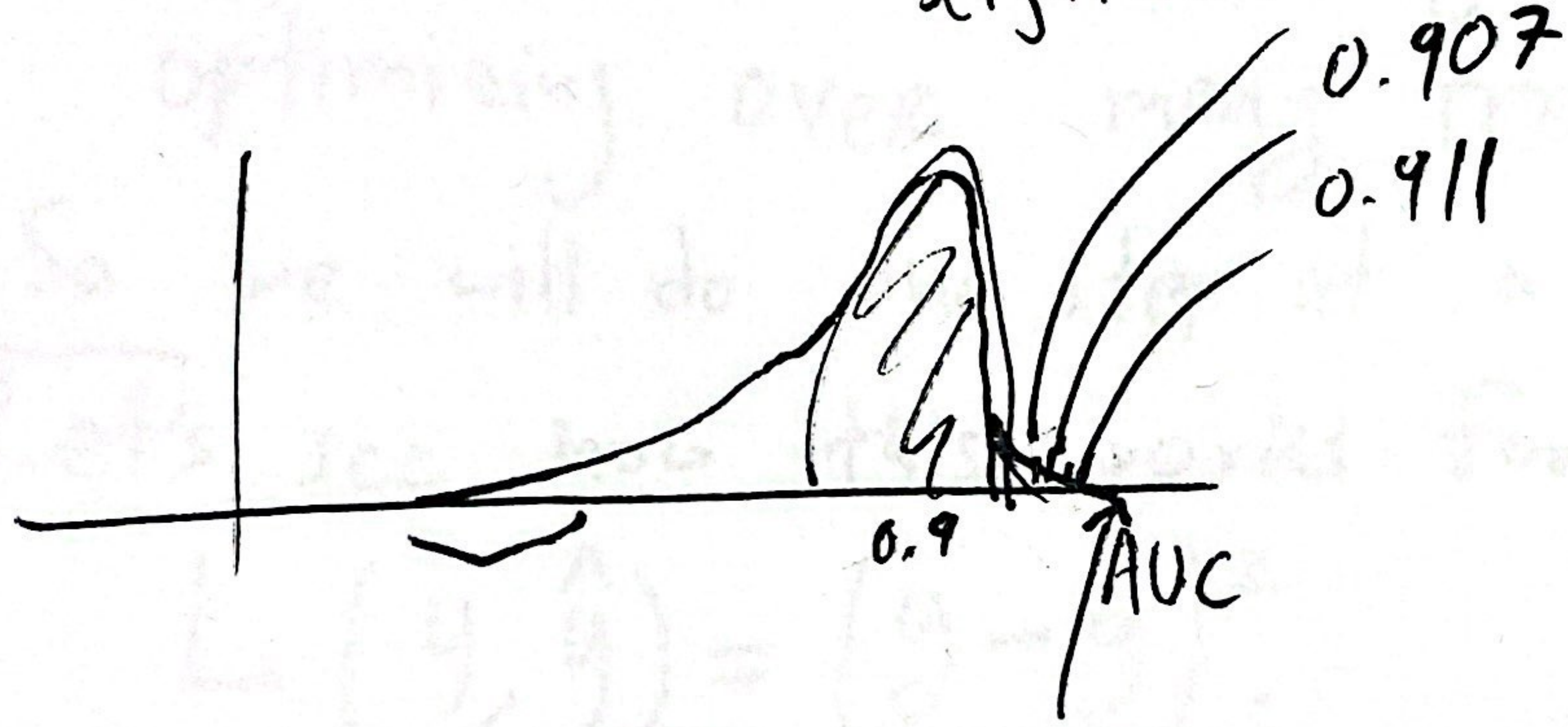
1

Business data Science.
Oct 18. Lecture no 16.

Today Boosting + Forward Stagewise Additive Modeling (FSAM).

→ Boosting and Gradient Boosting.
GBM, XGBoost, CATBoost, lightGBM.




Boosting: train many models sequentially.
→ Never change previous trained models.
→ Each new model should focus on correcting the mistakes of previous models.

Individual Models aka Weak Learners e.g. trees

$b(x, \vartheta)$

features  Parameters

$$f(x) = \sum_{m=1}^{M} b(x, \vartheta_m)$$

How to train? Jointly Fitting all $M$ trees:

$$\min_{\vartheta_1, \vartheta_2, \vartheta_3, \ldots \vartheta_M} \sum_{i=1}^{n} L\left(y_i, \sum_{m=1}^{M} b(x_i, \vartheta_m)\right)$$

- We dont know of efficient ways of jointly optimizing over many trees
- So we will do one step at a time.

Let's see how this works for $\ell_2$ Loss (MSE)

$$L(y, \hat{y}) = (y - \hat{y})^2.$$

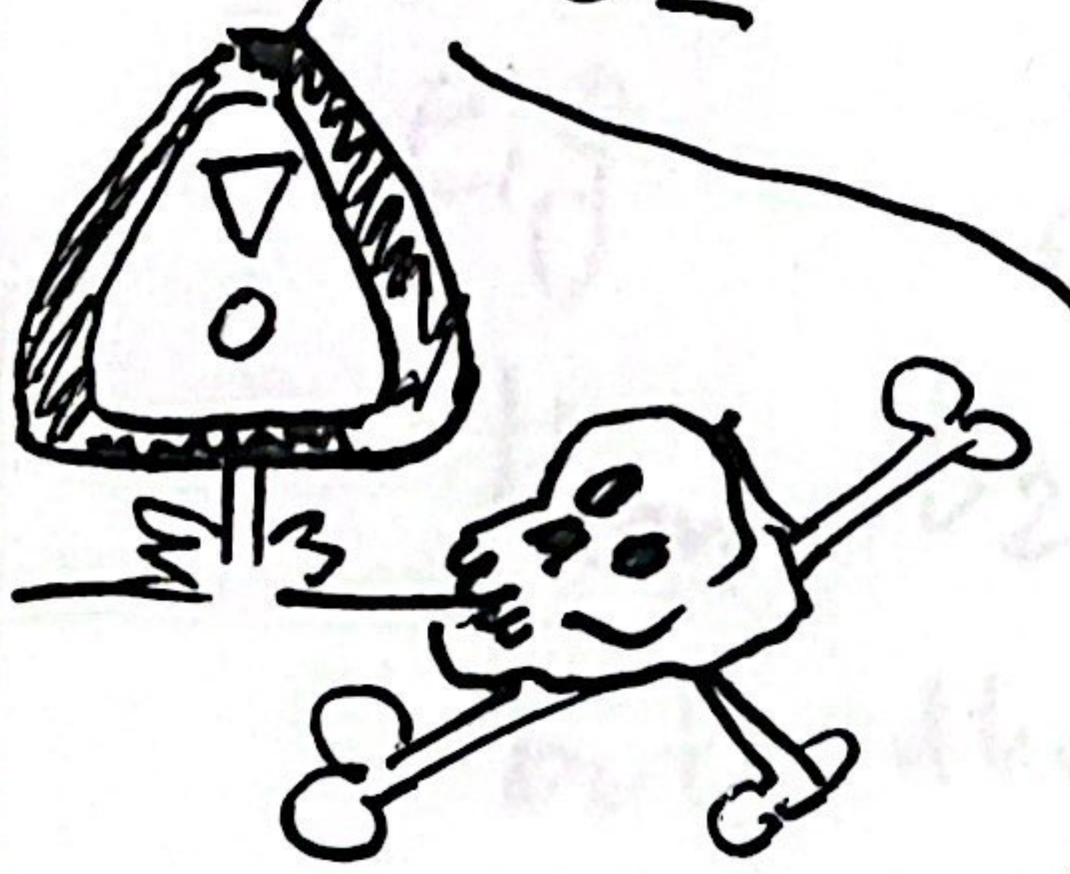
Step 1: train tree 1: $\min_{\vartheta_1} \sum_{i=1}^{n} L(y_i, b_1(x_i, \vartheta_1))$

$$\min_{\vartheta_1} \sum_{i=1}^{n} (y_i - b_1(x_i, \vartheta_1))^2$$

At Step 2, train $b_2$ so that $b_1 + b_2$ minimizes loss.

$$\min_{\theta_2} \sum_i L\Big(y_i, \overbrace{b_1(x_i,\theta_1)}^{\text{fixed}} + b_2(x_i,\theta_2)\Big).$$

$\left\{\begin{array}{l} b_1(x_i,\theta_1) \\ b_2 \end{array}\right.$



$$\min_{\theta_2} \sum_{i=1}^{n} \Big(\underbrace{y_i - b_1(x_i,\theta_1)}_{r_1(x_i)} - b_2(x_i,\theta_2)\Big)^2$$

(Mistakes OR Residuals of $b_1$.)

$$\min_{\theta_2} \sum_{i=1}^{n} \big(r_1(x_i) - b_2(x_i,\theta_2)\big)^2.$$

$\text{model2.fit}(x, \underbrace{r_1}_{y - \text{model1.predict}(x)})$

Would this work for binary classification?

No! Because for Regression $y - b_1(x)$ tells you how wrong the previous model was.

c

4. Fitting to erros of previous model does not work for Binary classification

e.g $y=0$ but $b_1(x)=1$.

then $b_2(x)$ should try to output $-1$ but that is not a binary classifier anymore.

---

For Binary Classification $y \begin{cases} +1 \\ -1. \end{cases}$ (Not (0,1)!

Define the margin $y \cdot f(x)$ for a binary classifier

$$L_{01}(y, f(x)) = \begin{cases} 0 & \text{if } f(x)>0,\ y=+1 \\ 0 & \text{if } f(x)<0,\ y=-1 \\ 1 & \text{if } f(x)>0,\ y=-1 \\ 1 & \text{if } f(x)<0,\ y=+1 \end{cases}$$

---

$$L_{01}(y, f(x)) = \mathbb{1}(\operatorname{sgn}(f(x)) \neq y)$$

$$L(y, f(x)) = \exp(-y \cdot f(x))$$