Logistic Regression ; Weight Regularization

$2x_1 - x_2 = 0, \quad w_1 = 2, w_2 = -1$

What happens to the decision boundary when you Scale w with a positive scaling factor?

Whether it is $w_1 = 2, w_2 = -1$
$w_1 = 20, w_2 = -10$
$w_1 = 2 \times 10^5, w_2 = -1 \times 10^5$
SAME DECISION BOUNDARY

Which of these solutions is a log regression model likely to converge to?

Recall in a log. regression model:

$$P(y_i = 1 | x_i; w) = \sigma(w^T x_i) = \frac{1}{1 + e^{-w^T x_i}}$$

$\longrightarrow$ A

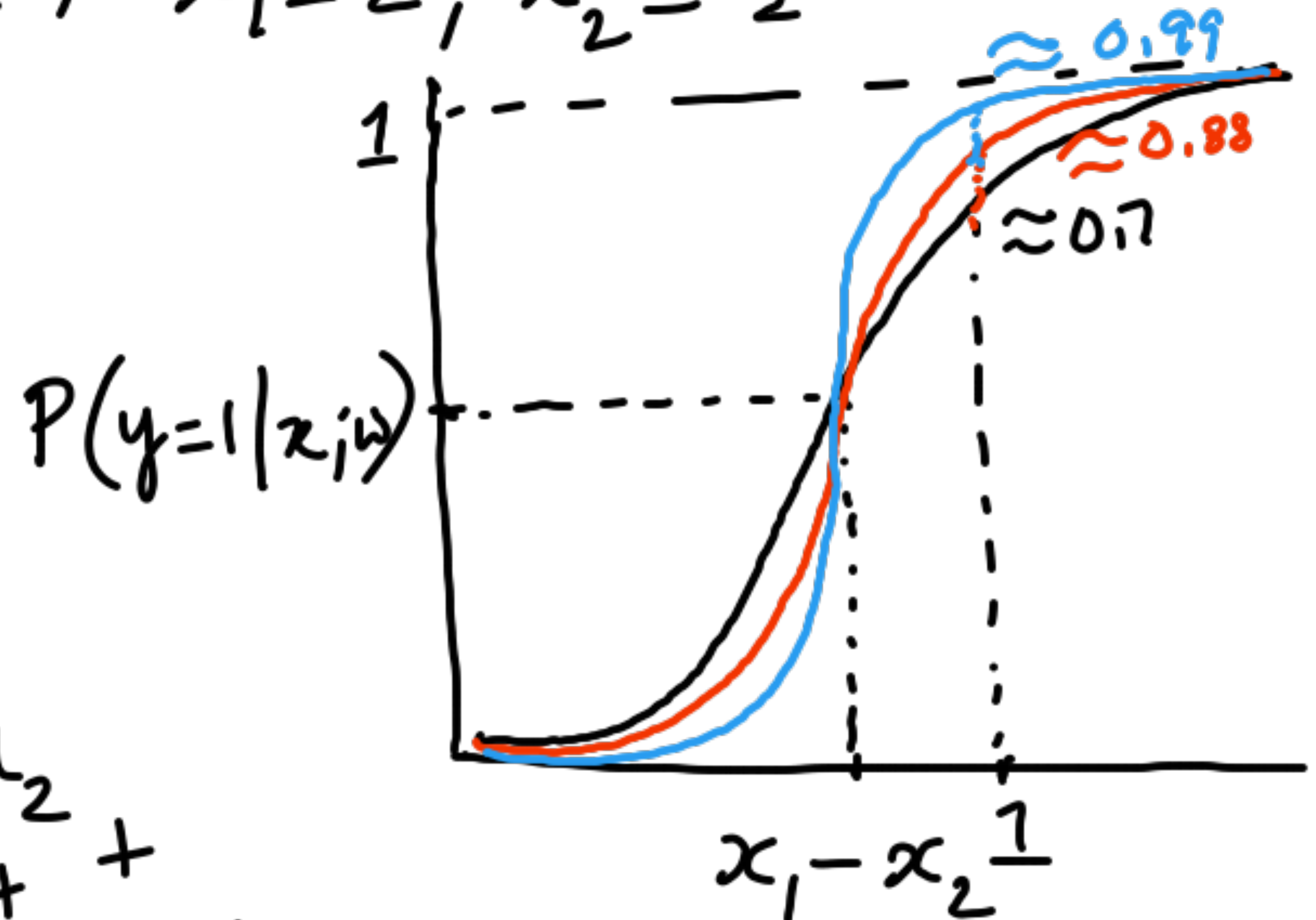
Objective function is to minimize CE loss i.e., maximize the conditional data likelihood

To maximize the prob in A, we want $w^T x_i$ to be large (so that $e^{-w^T x}$ is small)

$\Longrightarrow$ We want w values to be large

Is this a good idea?

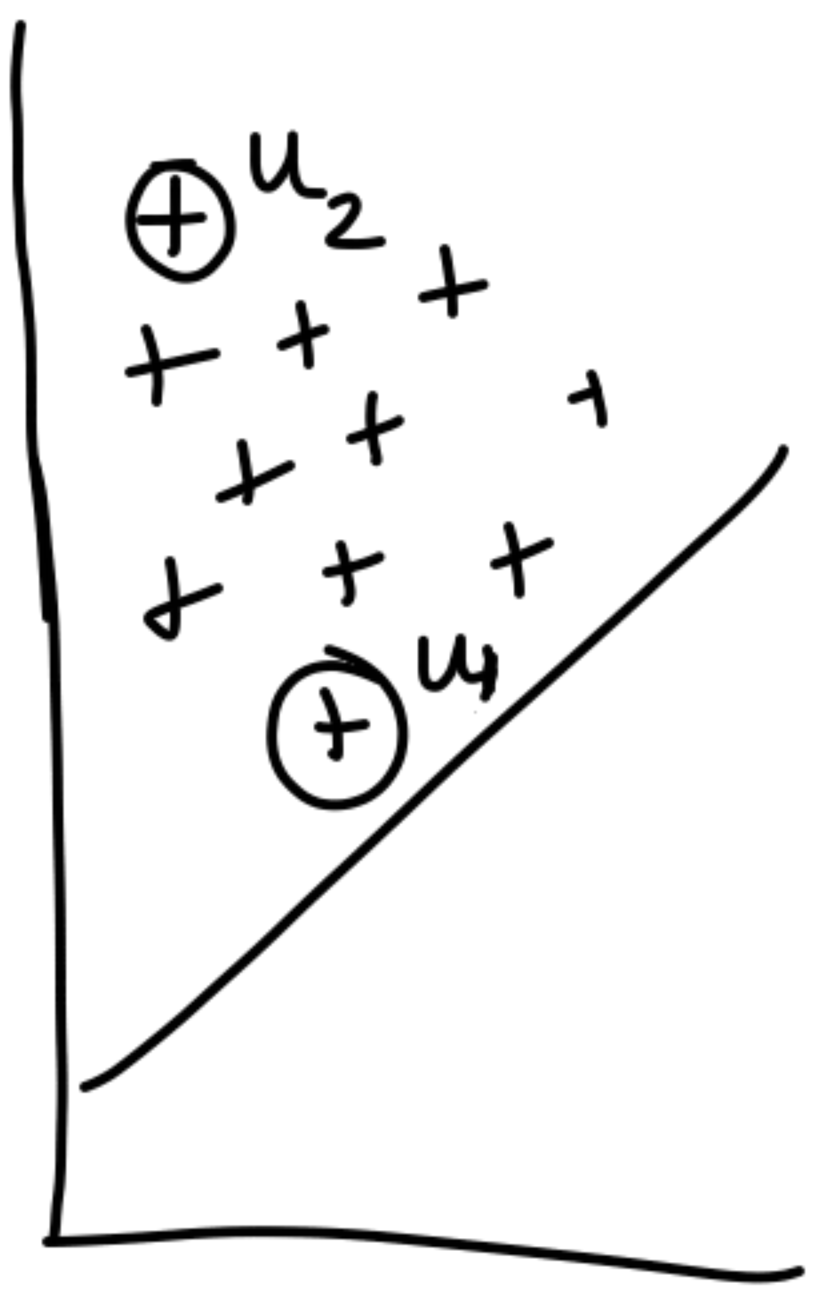
$x\ ;\ x_1 = 2,\ x_2 = 1$



$w_1 = 1,\ w_2 = 1$

$w_1 = 2,\ w_2 = 2$

$w_1 = 5,\ w_2 = 5$



Consider points u_1 and u_2 with the same label.

We want $w^T u_2$ and consequently the $P(y=1|u_2;w) \approx 1$

[Compared to point u_1 which is closer to the decision boundary]

But with large w, the distinction between such points (e.g. u_1, u_2) becomes smaller.

Solution: Regularized logistic regression

$$w^*_{Reg\text{-}LR} = \underset{w}{\operatorname{argmin}} \sum_i -\log P(y_i \mid x_i; w) + \lambda \|w\|_2^2$$

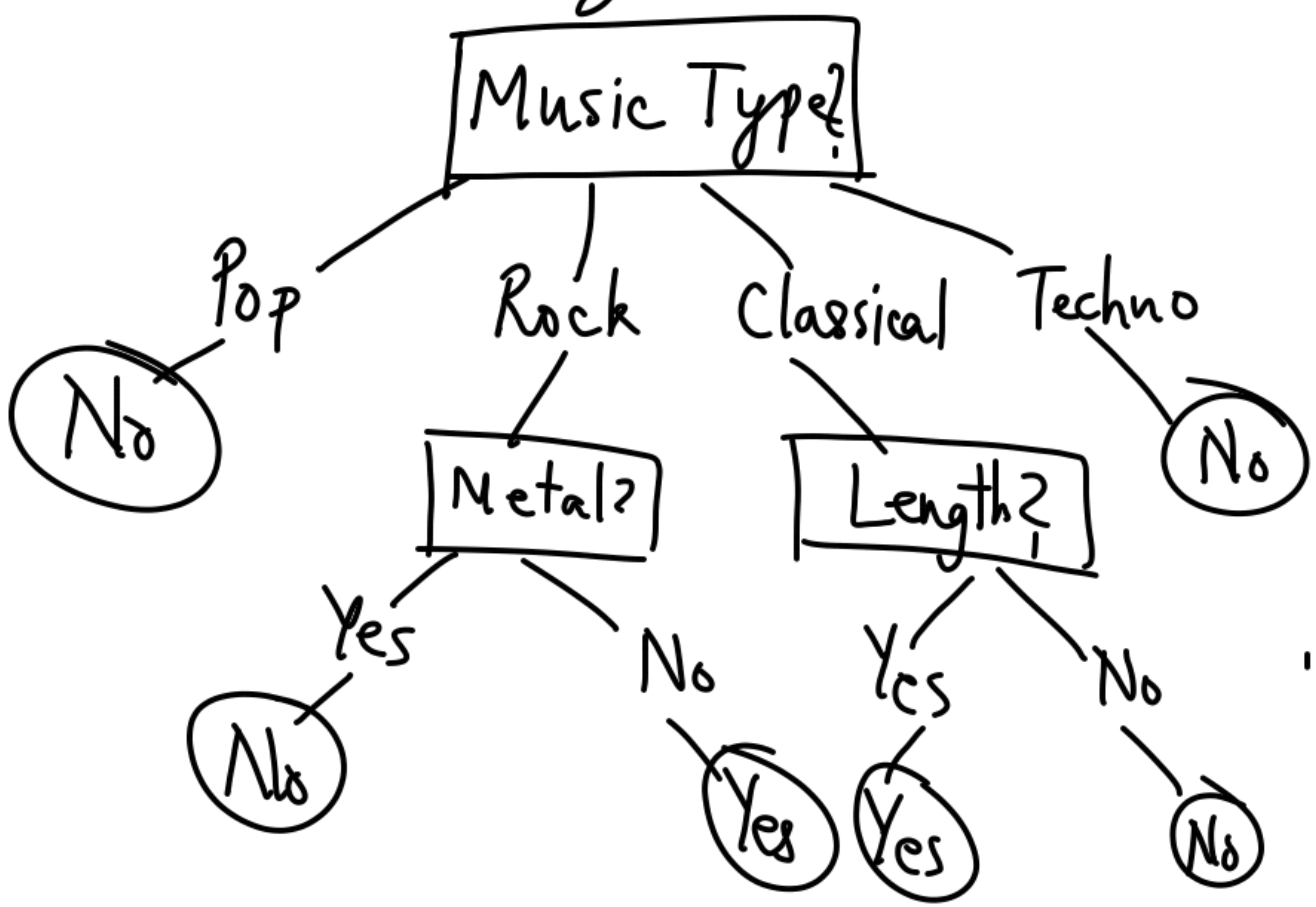
↳ L_2-regularized Log. Regression model

Desiderata for classification models:

1. Ability to learn complex decision boundaries
2. Interpretable

DECISION TREE CLASSIFIERs satisfy both criteria!

A decision tree is an <u>interpretable</u> model whose predictions can be written as a "disjunction of conjunctions" across the attribute values taken by the training instances.

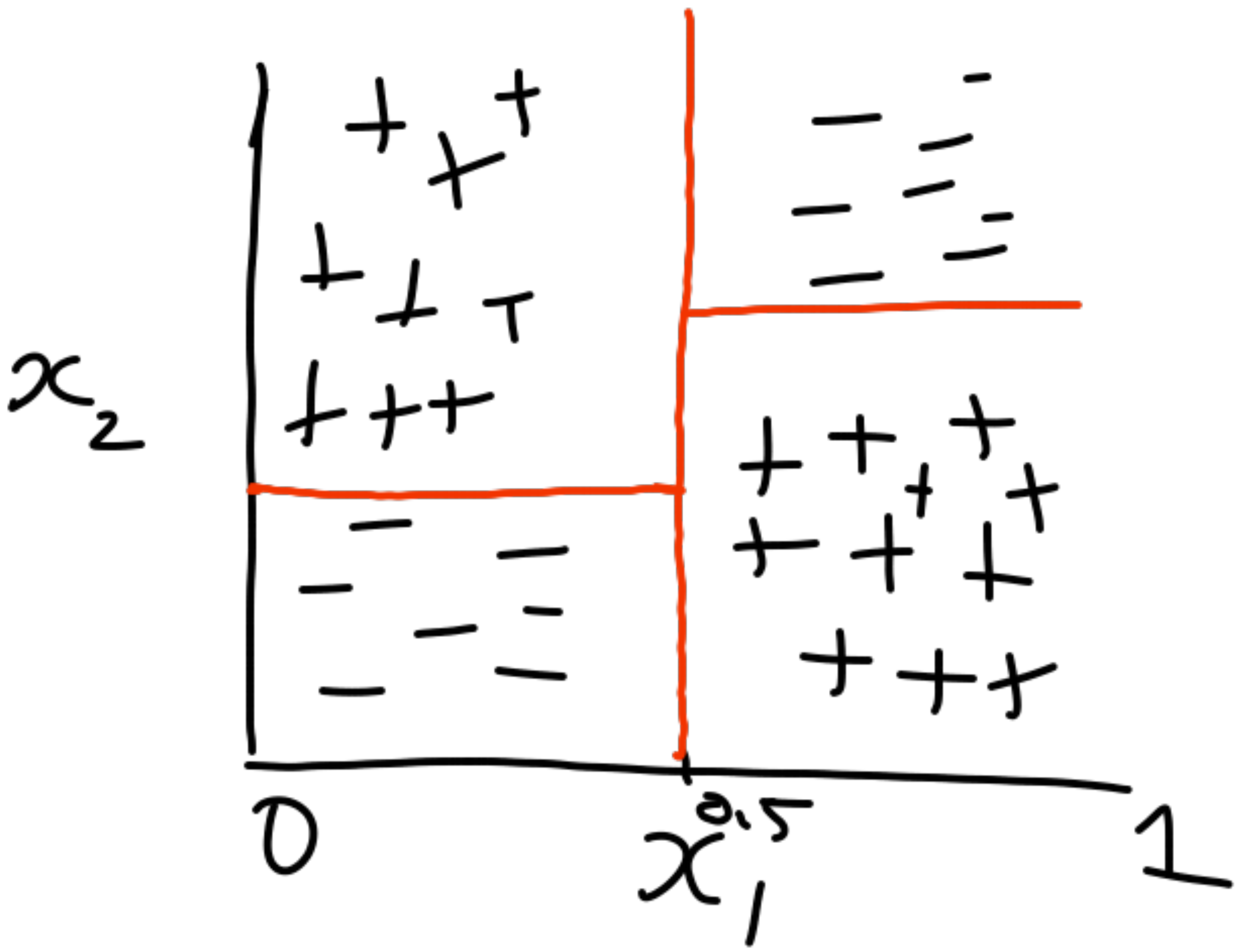


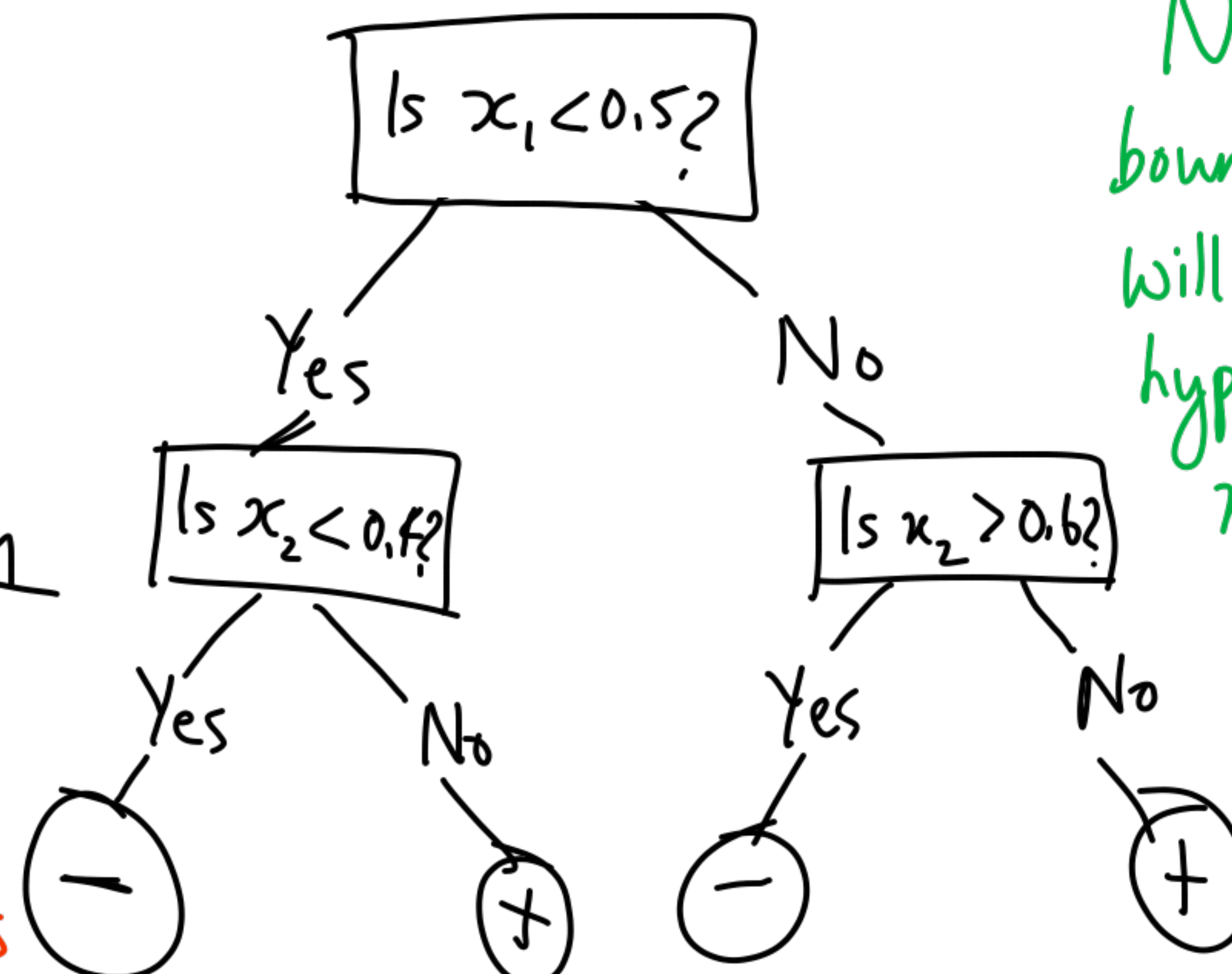
Path in this DT is a conjunction of attributes;

(Type = Rock $\wedge$ Metal = No)

$\therefore$ Tree is a <u>disjunction of conjunctions</u>

What is the decision boundary of a DT?





Note: Decision boundary for DTs will be linear hyperplanes if the nodes are a linear combination of 2 or more attributes (e.g., $x_1 + x_2 > 0$)

Decision boundary of DTs are axis-parallel hyperplanes [if the nodes are functions of a single attribute)

Finding an optimal DT

Finding the optimal DT by optimizing an objective/loss function over the attributes is NP-hard!

DT construction is typically greedy. Here's the basic template:

Step 1: Start with an empty node

Step 2: Pick the best attribute to split on

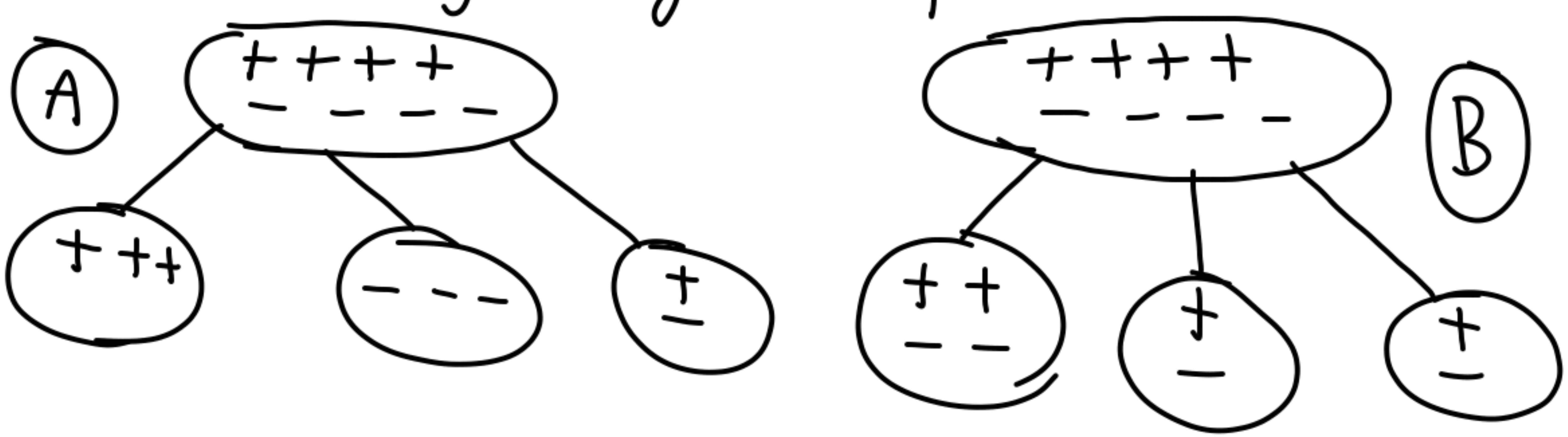
Step 3: Repeat step 2 recursively on each node till a stopping criterion is met

Two Qs:

1. Choosing the "best" attribute to split on
2. Choosing a Stopping criterion

Q1 Picking the "best" attribute

Consider the following two splits:



Which split is better?

A is better because two of the resulting data subsets are homogenous in label

A is a better split because the tree depth is smaller compared to B. We like smaller trees since they tend to generalize better [lower overfitting].

Intuition: We want splits to result in nodes that are homogenous in their labels

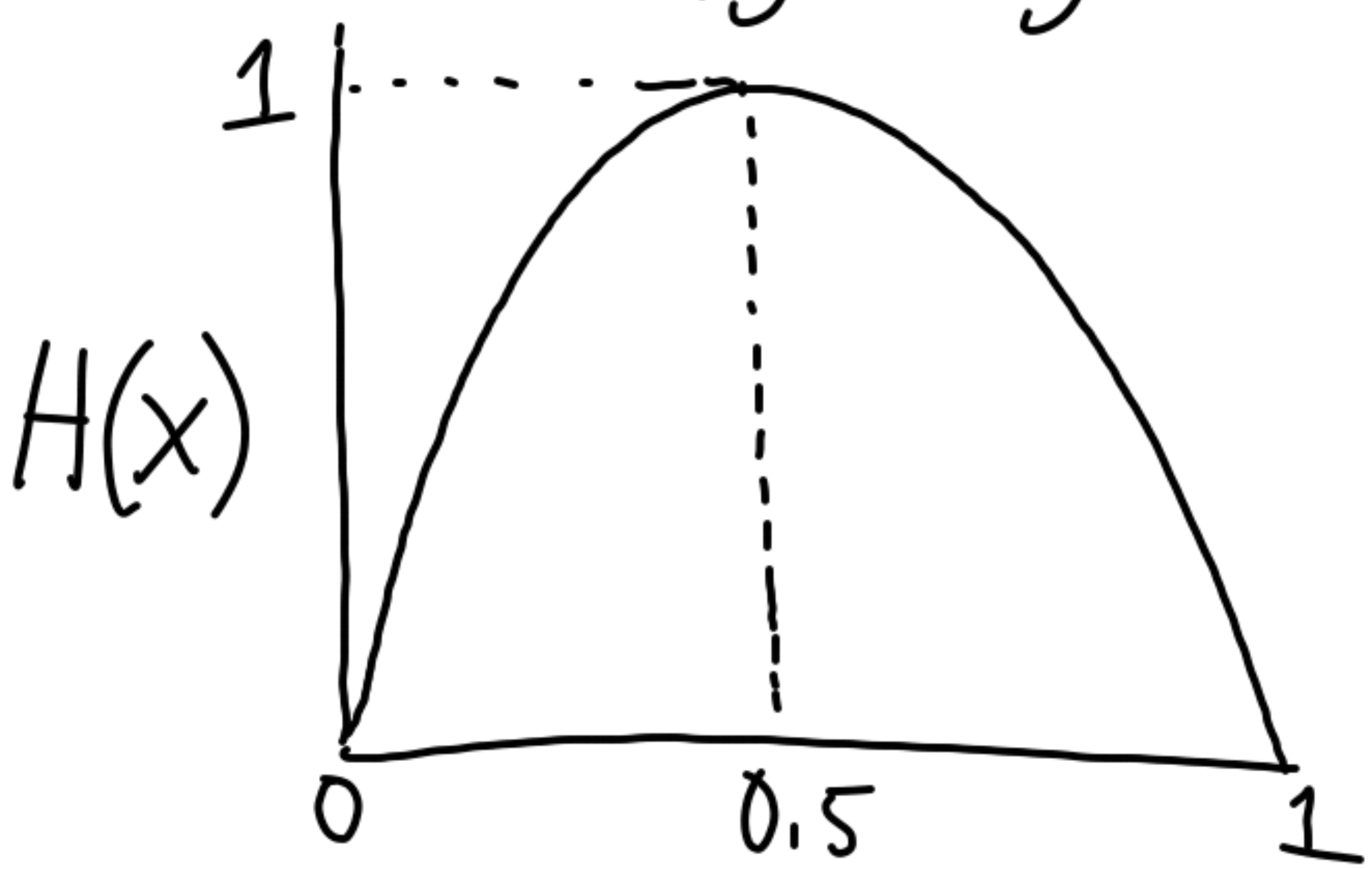
How do we quantify this intuition?

<u>ENTROPY</u>: Entropy of a random variable X is a measure of uncertainty in X

For a random variable X that takes values $x \in \mathcal{X}$, entropy $H(X)$ can be defined as: $H(X) = \sum_{x \in \mathcal{X}} -P(X=x) \log_2 P(X=x)$

base 2 to measure entropy in bits

Illustrate entropy using a coin toss. X is a binary r.v. taking values 0 (H) and 1 (T)



High entropy indicates that the underlying distribution is nearly uniform

Low entropy indicates low uncertainty meaning there are well-defined modes in the underlying distribution

Entropy of a dataset S

$\Rightarrow$ Entropy of the underlying label distribution

$$\Rightarrow H(S) = \sum_{i=1}^{K} -p_{S,i} \log p_{S,i} \quad \left[K \text{ labels overall} \right]$$

where $p_{S,i}$ is the probability that a random sample from S will have label i

($p_{S,i}$ is the relative count of # of instances with label i)

A good splitting criterion for DTs is "INFORMATION GAIN"

Consider an attribute "a" that can take values from $V(a)$, and a dataset S. Let S_γ be the subset of S with all instances having its "a" attribute labeled as γ.

$$\boxed{\mathrm{Gain}(S,a) = H(S) - \sum_{\gamma \in V(a)} \frac{|S_\gamma|}{|S|} H(S_\gamma)}$$