SUPERVISED LEARNING

BINARY CLASSIFICATION

$\{x_1, \ldots, x_n\}$ $x_i \in \mathbb{R}^d$ (i)

$y_1, \ldots, y_n$ $y_i \in \{0,1\}$ / $\{-1,+1\}$

Goal: $h : \mathbb{R}^d \to \{0,1\}$

LOSS / ERROR

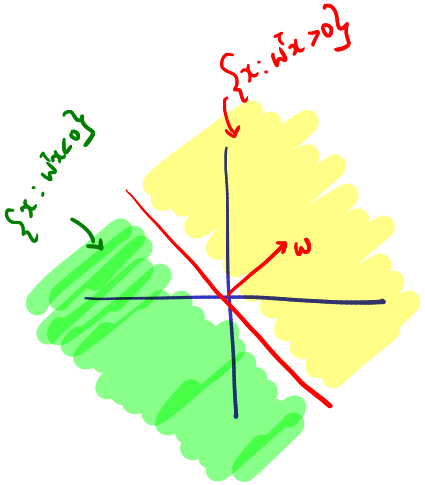
$$\text{Loss}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(h(x_i) \neq y_i\right)$$

$$\mathbb{1}(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{o/w} \end{cases}$$

0-1-loss

$$\min_{h \in \mathcal{H}_{\text{linear}}} \sum_{i=1}^{n} \mathbb{1}\left(h(x_i) \neq y_i\right)$$

NP-hard problem in general

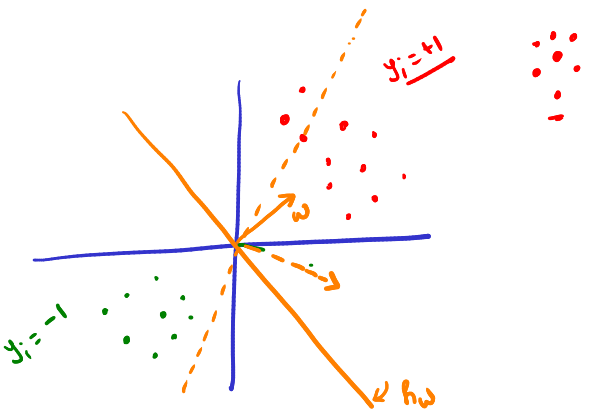


$$\mathcal{H}_{\text{linear}} = \left\{ h_w : h_w(x) = \text{Sign}(w^T x) \right\}$$

$$\text{Sign}(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{o/w} \end{cases}$$

- Can we use linear regression to solve classification problem?

$\{(x_1, y_1) \ldots (x_n, y_n)\}$ → [Lin Reg] → $w \in \mathbb{R}^d$ → $h_w : \mathbb{R}^d \to \{0,1\}$



Conclusion

Regression is SENSITIVE to location of the data points and not just the "side" on which the data lies w.r.t separator.

SIMPLE ALGORITHMS FOR CLASSIFICATION

- Given a test point $x_{test} \in \mathbb{R}^d$,
  find the closest point x^* to x_{test} in the training set

- Predict $y_{test} = y^*$

ISSUE: can get affected by outliers

Fix: Ask more neighbours

(K-NN) (K- Nearest Neighbours)

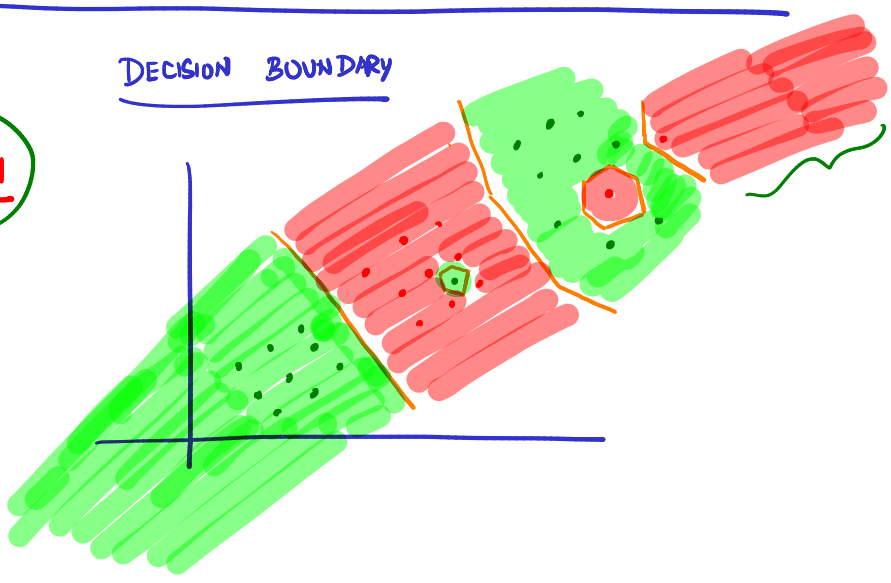
- Given x_{test}, find the K - closest points in the training set - $(x_1^*, x_2^*, \ldots, x_k^*)$

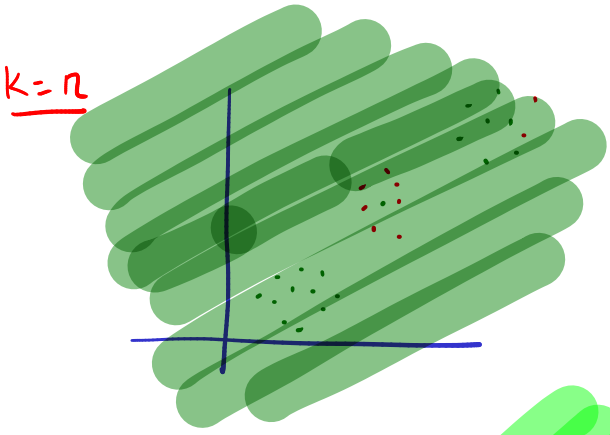
- PREDICT y_{test} = majority $(y_1^*, y_2^*, \ldots, y_k^*)$

DECISION BOUNDARY

K=1

SENSITIVE TO OUTLIERS!

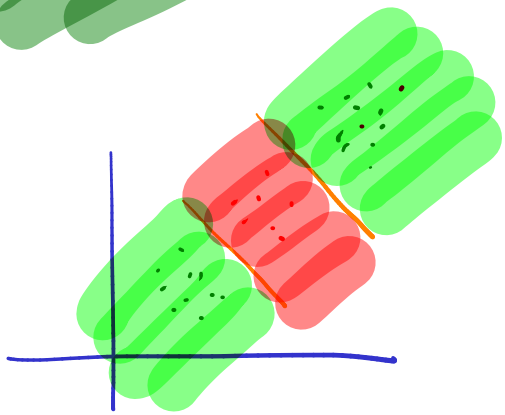




$k = n$

Every point is predicted $y = -1$ (green).

Too Smooth!



k^*.

A good choice.

Choosing k

→ Can treat as a HYPER PARAMETER

→ Smaller the k, complicated the decision boundary

→ Soln: CROSS-VALIDATE for k

ISSUES with k-NN

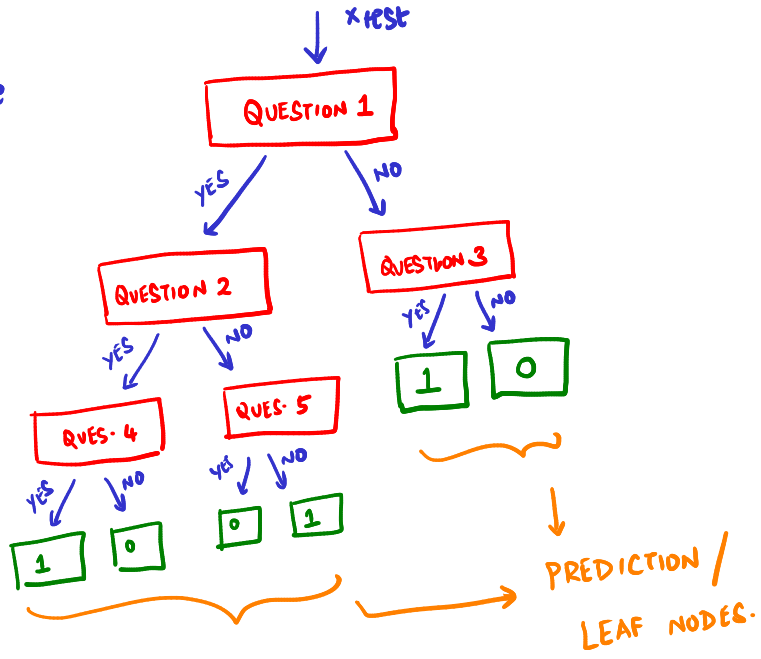
- Choosing a distance function
- PREDICTION is COMPUTATIONALLY EXPENSIVE.
- No MODEL is LEARNT.
  - Cannot throw away data after "LEARNING"

DECISION TREES

INPUT: Dataset $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ $\forall i$ $x_i \in \mathbb{R}^d$, $y_i \in \{+1, 0\}$

OUTPUT: DECISION TREE

DECISION tree



PREDICTION: Given x_{test}, traverse through the tree to reach a leaf node.

y_{test} = value in leaf node.

QUESTION: A question is a (feature, value) pair.

Eg: height $\leq$ 180cm ?

(f_3) θ

How to measure "goodness" of a Question?

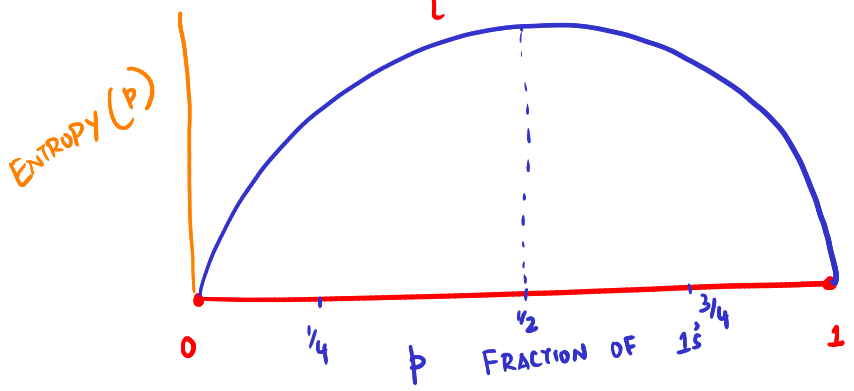
DATASET $D = \{(x_1, y_1) \ldots (x_n, y_n)\}$

$f_k \leq \theta$?

YES / NO

$D_{yes} = \{(x_1, y_1), (x_{10}, y_{10}), \ldots\}$ $\{(x_2, y_2), (x_3, y_3) \ldots\}$ D_{no}

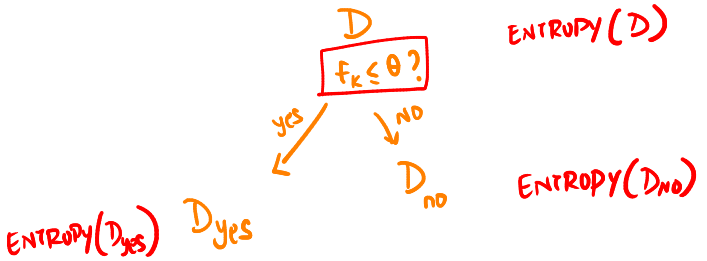
- Need is a measure of "Impurity" for a set of labels $\{y_1, \cdots, y_n\}$



$$\text{ENTROPY}(\{y_1, \cdots, y_n\}) = \text{ENTROPY}(p)$$

$$= -\left(p \log p + (1-p) \log (1-p)\right)$$

[convention $\log(0) = 0$]



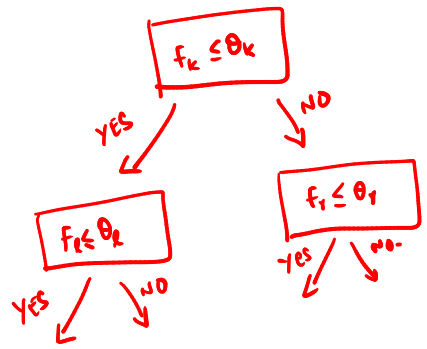
$$\text{INFORMATION GAIN}(\text{feature}, \text{value}) =$$

$$\text{ENTROPY}(D) - \left[\gamma \ \text{ENTROPY}(D_{yes}) + (1-\gamma)\ \text{ENTROPY}(D_{no})\right]$$

$$\gamma = \frac{|D_{yes}|}{|D|}$$

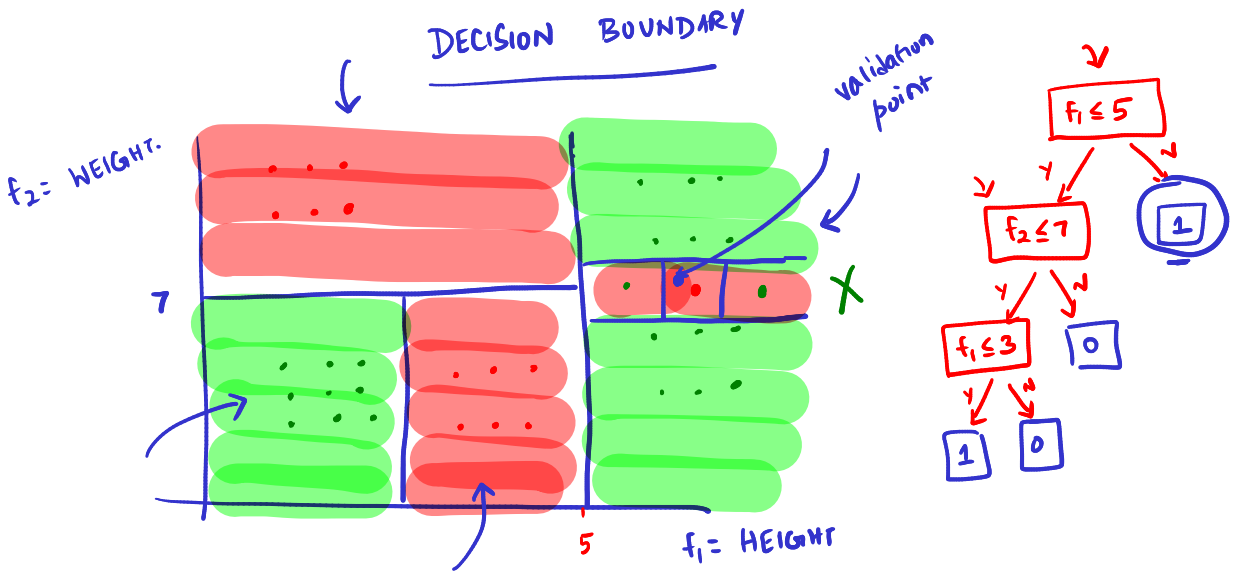
ALGORITHM - DECISION TREE

- DISCRETIZE each feature in [min, max] range
- Pick the Question that has the largest Information gain.
- Repeat the procedure for Dyes, Dno



POINTS

- Can stop growing a tree if a node becomes "SUFFICIENTLY" pure.
- DEPTH of the tree is a hyperparameter
- There are alternate measures for "goodness" of a Question
  → GINI INDEX


DECISION BOUNDARY
validation point
f_2 = WEIGHT.
7
5
f_1 = HEIGHT
$f_1 \leq 5$
Y
N
$f_2 \leq 7$
1
$f_1 \leq 3$
0
1
0

TYPES OF MODELING

DISTRIBUTION D OVER $X \times Y$ ← $\{+1, -1\}$ (FEATURE SPACE)

$\{x_1, \ldots, x_n\}$
$y_1, \ldots, y_n$

UNKNOWN BUT FIXED

$(x_i, y_i) \sim D$ ← LINK BETWEEN TRAIN and TEST.

CLASSIFICATION

- GENERATIVE MODEL
- DISCRIMINATIVE MODEL

GENERATIVE MODEL

MODEL

- $P(x, y)$

↳ NEXT.

DISCRIMINATIVE MODEL

MODEL

- $P(y/x)$

Eg: K-NN
DECISION-TREES

↳ $P(y=1/x) = 1$ if decision tree for x says 1.

$P(y=1/x) = 1$ if majority of neighbours say 1
$= 0$ otherwise