Decision Trees - 1

13 February 2025 19:53

Attrition Rate of Airtel

Attrition → Yes ⇒ left

→ No ⇒ stayed

The company now has two options - ① Retain ② Recruite

What analysis can we do on this data?

① Probability of leaving of an employee - classification - Log. Reg.

② Factors' contribution towards attrition - Model's Interpretation

$$\vec{\omega} \cdot \vec{x} + \omega_0 = 0 \Rightarrow \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n + \omega_0 = 0$$

salary

if $\omega_i > \omega_j$ mean feature 'i' has higher impact on attrition than feature 'j'

Home work: standard procedure of creating an ML model

① Acquire the data - Data Ingestion Pipeline

② Pre processing - ⓐ clean

- ↳ Duplicates
- ↳ Missing values

ⓑ Encoding

ⓒ Feature scaling

- ↳ standardize | Normalize

ⓓ Rebalancing data

ⓔ Treatment of outliers

(e) Treatment of outliers

(f) Feature Engineering

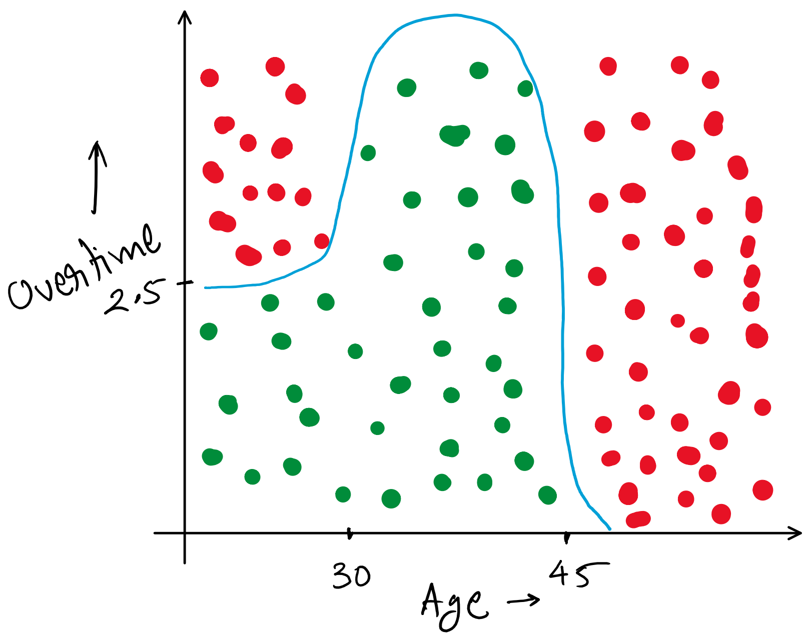
└ Reduce the dimensionality

└ add new features (more relevant)

(g) EDA

(3) ML model

★ How about solving this problem with a different approach.



→ This data is not linearly seperable. ∴ LR won't work

→ Can we use polynomial Logistic Reg.?

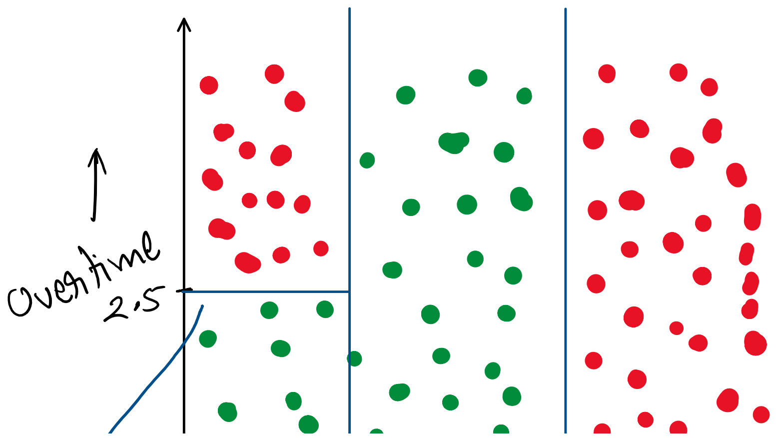
↳ Complex

∴ More chances to overfit

→ KNN?

↳ Slow for big datasets

→ But Airtel has ~87,000 employees for which KNN might become slow.

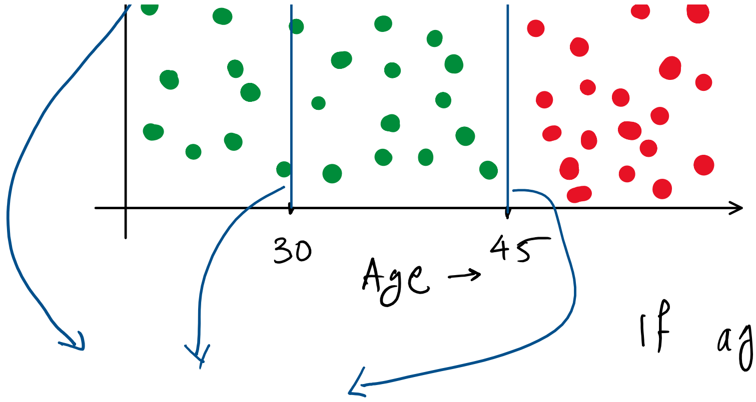


Let's ask a few questions to our data:

age $\geq$ 45 ⇒ Leave

30 $\leq$ age < 45 ⇒ stay

age < 30 ⇒ OT > 2.5 ⇒ Leave



age < 30 ⇒ OT > 2.5 ⇒ Leave

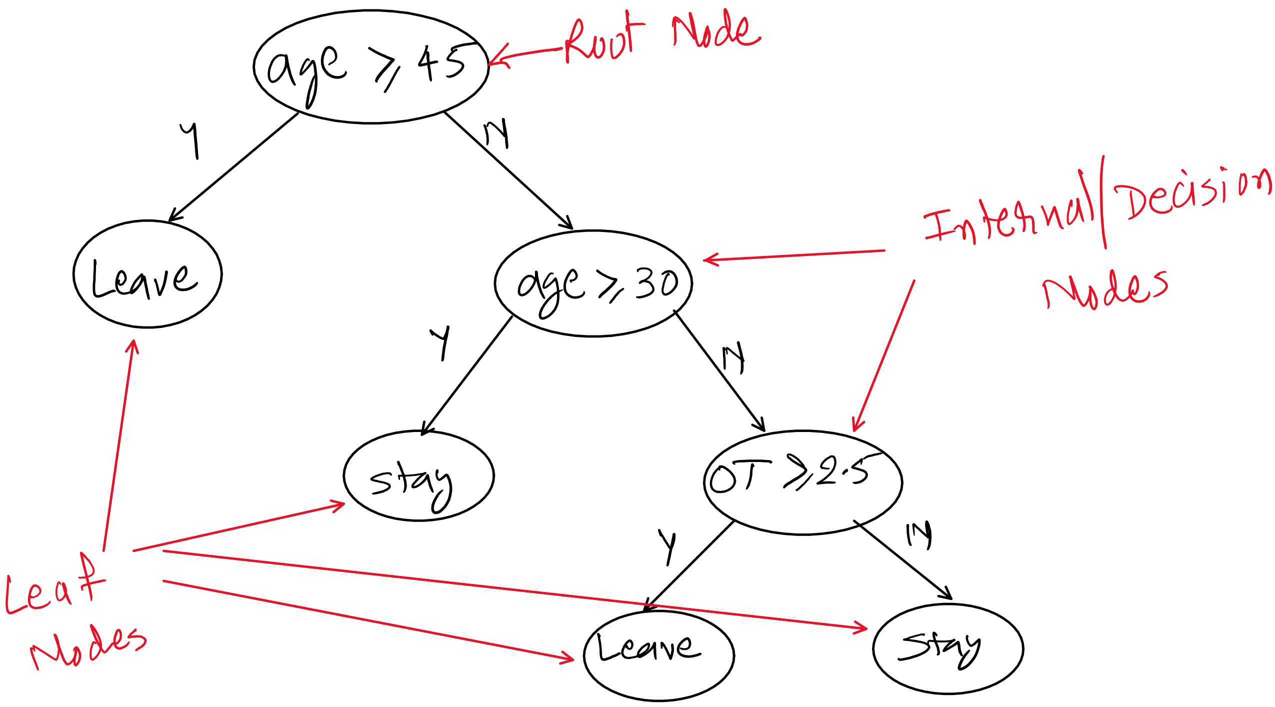
How can we implement this logic?

```
if age >= 45: leave
else:
    if 30 <= age < 45: stay
    else:
        if OT > 2.5: Leave
        else: stay
```

Axis Parallel Decision Boundaries

```
if OT > 2.5:
    if Age < 30: Leave
    else:
        if Age >= 45: Leave
        else: stay
else: if Age >= 45: Leave
    else: if Age < 30: stay
        else: stay
```

Can we write this logic in a different way as below?



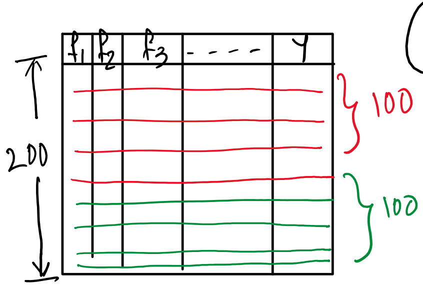
☆ Which questions should we start with | we ask to our data?

→ This is a v. imp point because the questions we decide will be asked to each & every datapoint and hence can become computationally very expensive if chosen in wrong sequence.

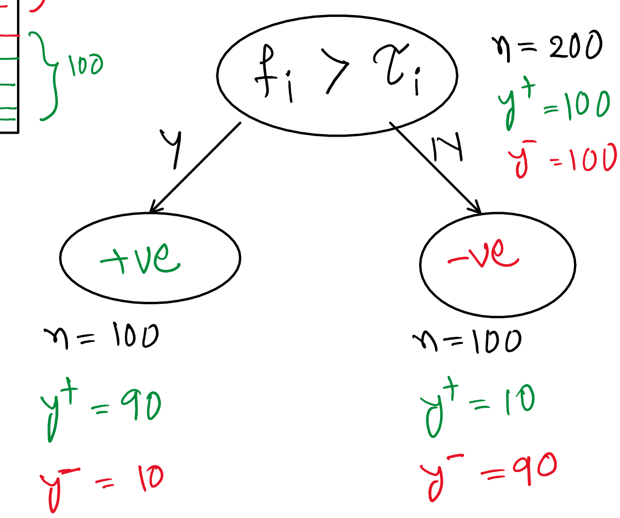
Let's understand this concept by an example.

→ Suppose we have 200 datapoints with two classes

y^+ : 100 datapoints y^- : 100 datapoints.

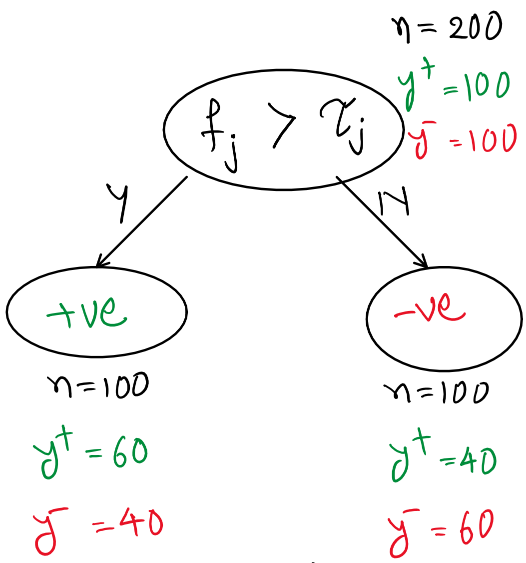


(I) I am picking feature 'i'



no. of missclassified points = 20

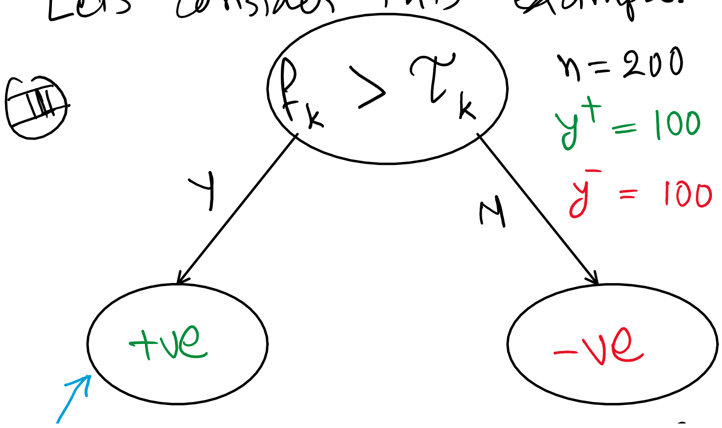
(II) I am taking jth feature first



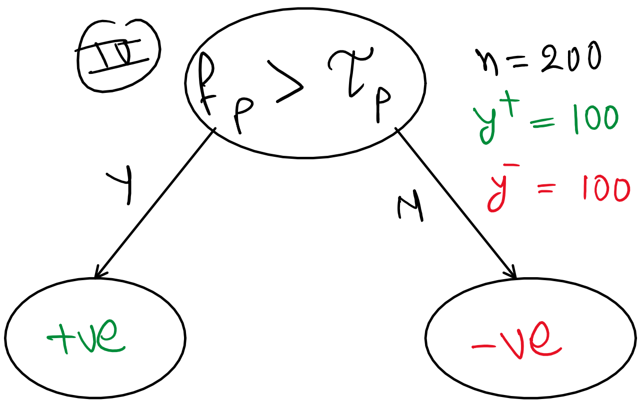
no. of missclassified points = 80

Let's consider this example:

(III)



(IV)



+ve	-ve	+ve	-ve
$n = 80$	$n = 120$	$n = 90$	$n = 110$
$y^+ = 80$	$y^+ = 20$	$y^+ = 85$	$y^+ = 15$
$y^- = 0$	$y^- = 100$	$y^- = 5$	$y^- = 95$

no. of misclassified points = 20

Pure Homogenous Region

Pure Homogenous Node

Slightly less homogenous

★ We need to quantify "homogenity" of a region/node

Ans - Entropy - measure of impurity | Heterogenity

Entropy high = high Heterogenity = Low Homogenity = Less Purity

Entropy of a node 'Y' is denoted by $H(Y)$ & given by:

$$H(Y) = -\sum_{i=1}^{m} P(y_i) \cdot \log_2 P(y_i)$$

In our example, our labels are : {+ve, -ve}

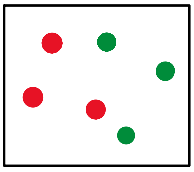
$$\boxed{\therefore P(-ve) = 1 - P(+ve)}$$

$$H(Y) = -\left[P(y^+) \cdot \log_2 P(y^+) + P(y^-) \cdot \log_2 P(y^-)\right]$$

$$H(Y) = -\left[P(y^+) \cdot \log_2 P(y^+) + (1 - P(y^+)) \cdot \log_2 (1 - P(y^+))\right]$$

↳ Recall Log loss ?

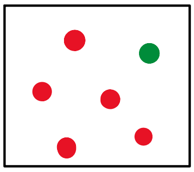
Example-



$P(-ve) = 1/2$

$P(+ve) = 1/2$

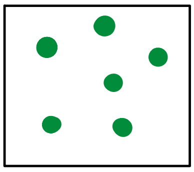
$H(Y) = 1$



$P(-ve) = 5/6$

$P(+ve) = 1/6$

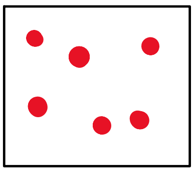
$H(Y) = 0.65$



$P(-ve) = 0$

$P(+ve) = 1$

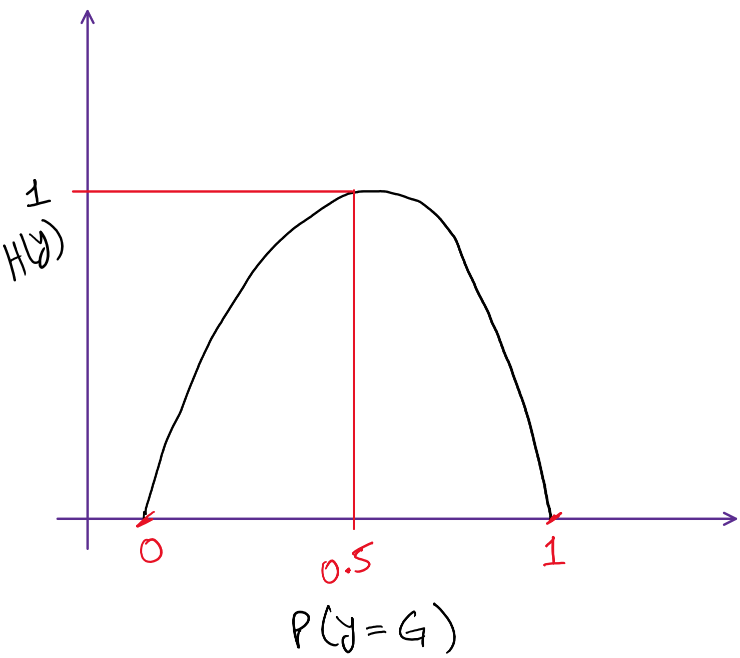
$H(Y) = 0$ → Purest Node



$P(-ve) = 1$

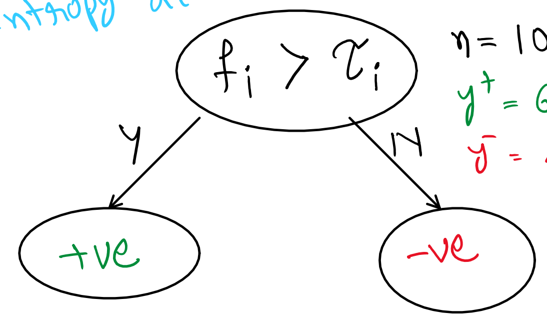
$P(+ve) = 0$

$H(Y) = 0$ → Purest Node



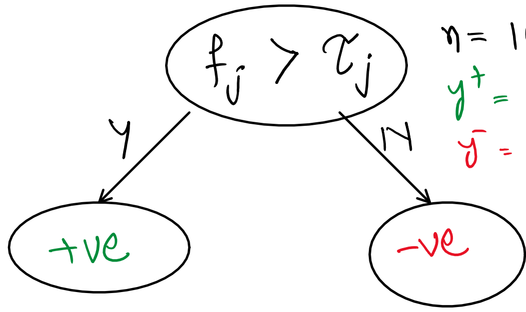
☆ Coming back to our question - Which feature should I consider first? / which question should I ask first

Entropy at Parent level = H_P



$n = 100$
$y^+ = 60$
$y^- = 40$

$n = 60$	$n = 40$
$y^+ = 50$	$y^+ = 10$
$y^- = 10$	$y^- = 30$



$n = 100$
$y^+ = 60$
$y^- = 40$

$n = 70$	$n = 30$
$y^+ = 50$	$y^+ = 10$
$y^- = 20$	$y^- = 20$

Entropy at children level = H_c

Drop in Entropy = $H_P - H_c$

1. Entropy at the parent level:

$P(G) = 6/10 \quad P(R) = 4/10$

$H(P) = 0.971$

Entropy at children level:

Left Question (Feature)

Left child : $H(y) \Rightarrow P\_g = 5/6$ &
$P\_r = 1/6$

$\therefore H(y) = 0.65$

Right Child : $H(y) \Rightarrow P\_g = 1/4$ &
$P\_r = 3/4$

$H(y) = 0.81$

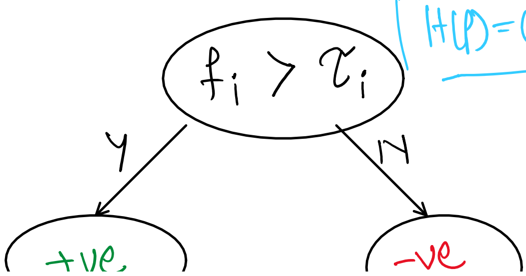
Right Question (Feature)

Left Child : $H(y) \Rightarrow P\_g = 5/7$ &
$P\_r = 2/7$

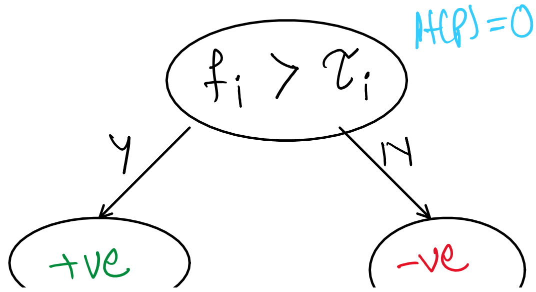
$\therefore H(y) = 0.86$

Right Child : $H(y) \Rightarrow P\_g = 1/3$ &
$P\_r = 2/3$

$\therefore H(y) = 0.92$



$\boxed{H(P) = 0.971}$



$H(P) = 0.971$

+ve   -ve

$H(y) = 0.65$ $H(y) = 0.81$

+ve   -ve

$H(y) = 0.86$ $H(y) = 0.92$

Need of combining these values

☆ Overall Entropy at the children level:

$$= \frac{n_1}{n} \cdot H_{C_1} + \frac{n_2}{n} \cdot H_{C_2}$$

$$H_{c_i} = \frac{60}{100} \times 0.65 + \frac{40}{100} \times 0.8$$

$$H_{c_i} = 0.71$$

Entropy drop with feature-i

$$= 0.971 - 0.71$$

$$= 0.261$$

$$H_{c_j} = \frac{70}{100} \times 0.86 + \frac{30}{100} \times 0.92$$

$$H_{c_j} = 0.88$$

Entropy drop with jth feature $= 0.971 - 0.88$

$$= 0.091$$

This drop in Entropy is also know as Information Gain