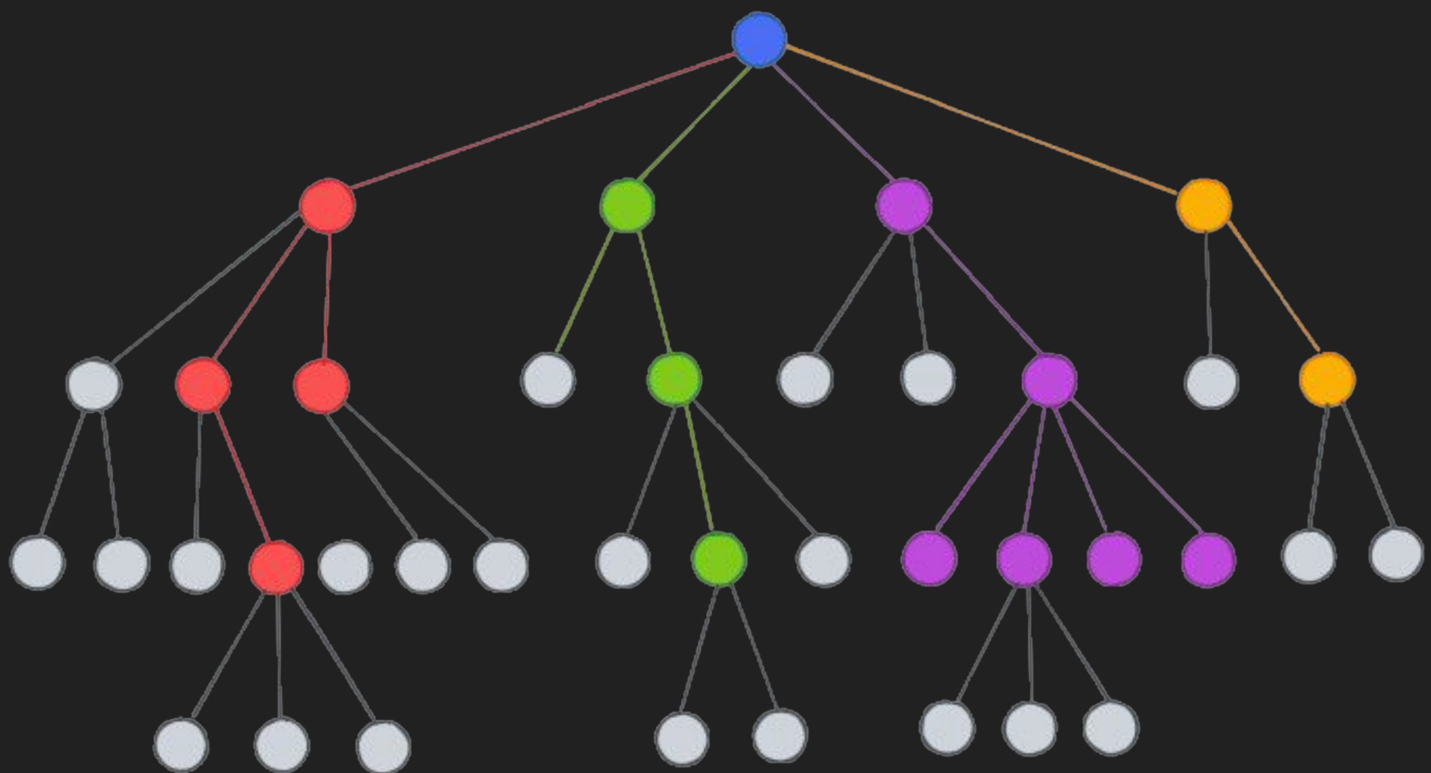
Decision Tree



Use Case : Employee Attrition

Launch of Jio led to rise in Employee Attrition in Airtel.

Airtel has appointed you as a Data Scientist to perform two tasks

TASK '2'

Identify the key indicators/factors leading to an employee leaving.

1. What all reasons can you think of contributing to attrition ?
2. Forcing employees to come to office daily
3. Unhealthy culture etc

Solved using - interpretability Model

TASK '1'

Identify the employees who may leave in future.

1. Targeted approaches can be undertaken to retain such employees.
2. These might include addressing their problems with the company and so on …

Solved using - Classification Model

Summary of EDA and Preprocessing

- Plotting Distributions
- Encode categorical features
  - Cardinality ≤ 2 : Binary Encode
  - Cardinality < 6 : OHE
  - Cardinality > 6 : Target Encode
- Check imbalance and rebalance

Male
Female
Other

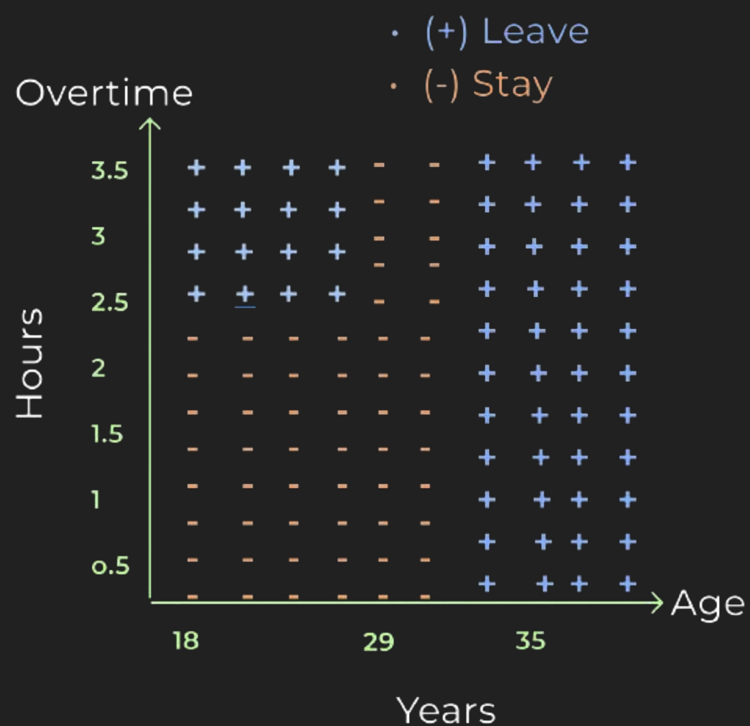
Male	Fem	Ot
1	0	0
0	1	0

**NOTE : Using OHE in feature cardinality > 6 will explode the number of features

Decision Tree Intuition

Supposedly, we have attrition data with two features :

- Age
- Overtime



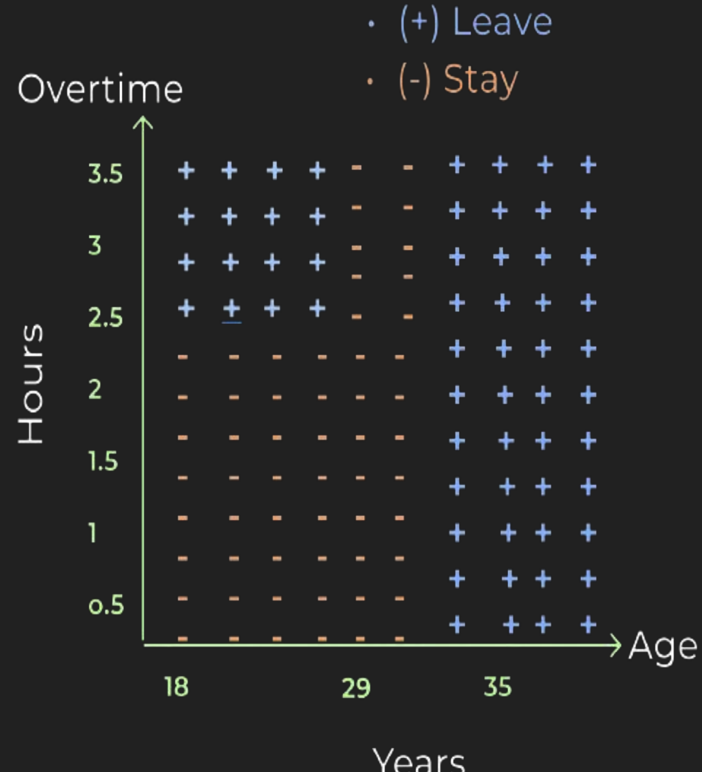
Can we use logistic regression to classify this data ?

- **No** as it is a **linear model** and we have **non linear data** with us.

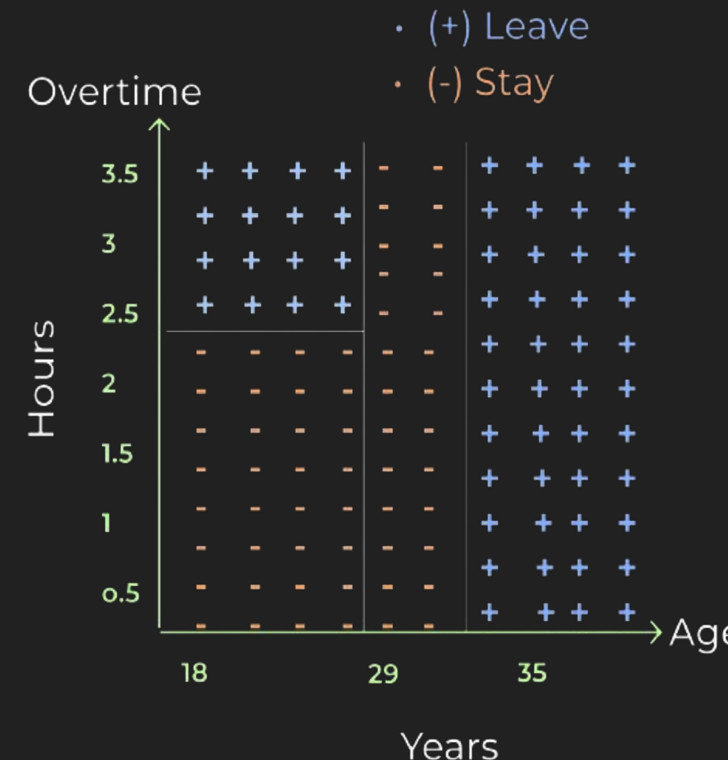
Can we use KNN to solve this problem ?

- **Yes** it will work well but its slow in test time.

Let's try out splitting data using **Nested If Else Statements**

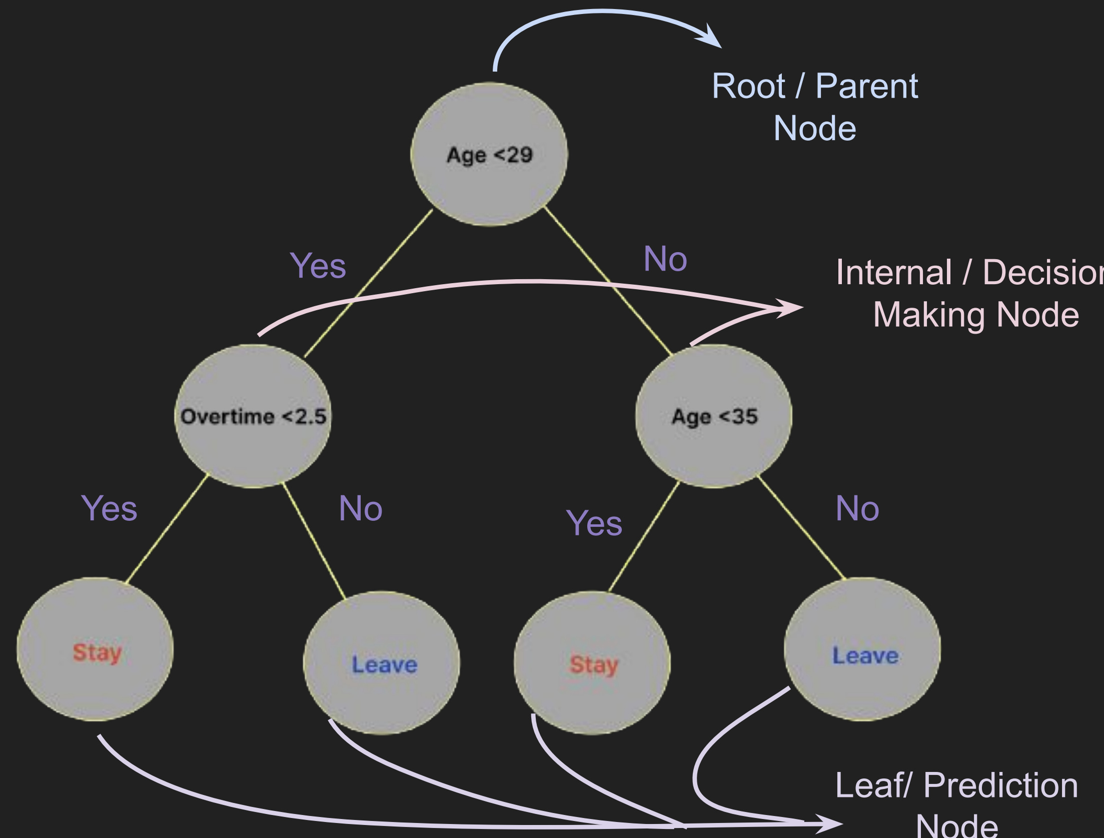



```
If age <29 :
    If overtime < 2.5:
        Employee will stay
else:
        Employee leave
else:
    If age < 35
        Employee stay
else:
        Employee leave
```



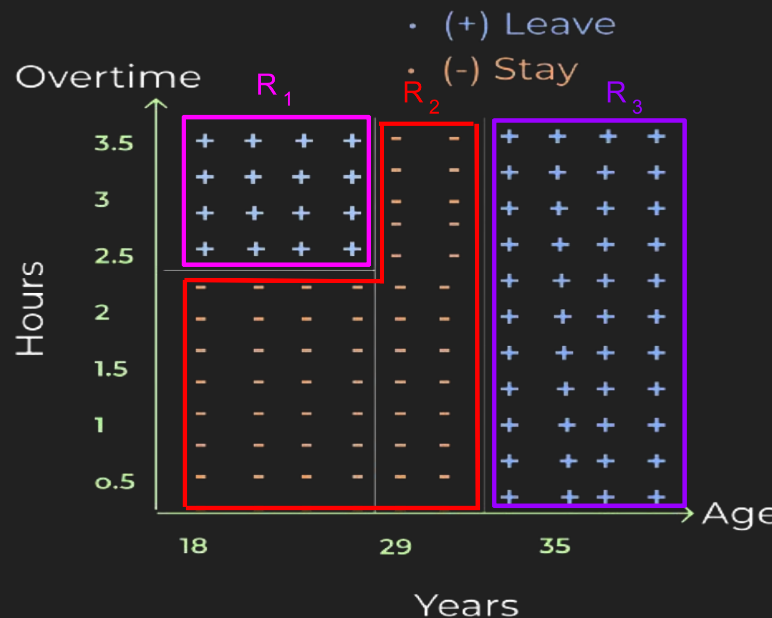
Now, Let’s visualize Nested If Else Statements

```
If age <29 :
    If overtime < 2.5:
      Employee will stay
else:
      Employee leave
else:
    If age < 35
      Employee stay
else:
      Employee leave
```



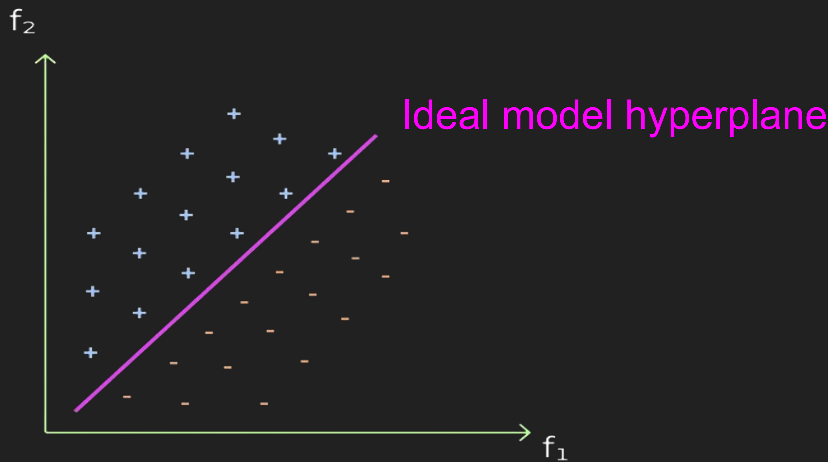
This tree like structure is known as Decision Tree

Decision Tree : Splits data into three homogeneous regions (R_1 , R_2 , R_3) using 3 axis hyperplanes (y = 2.5 , x = 29 , x = 35)

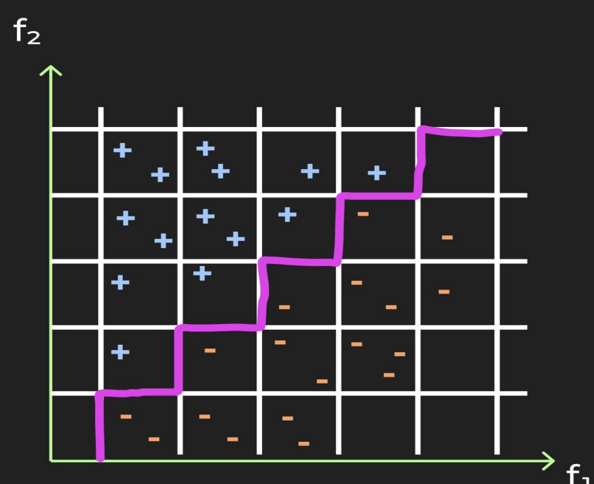


Advantage of using Decision Tree : Easy Interpretation

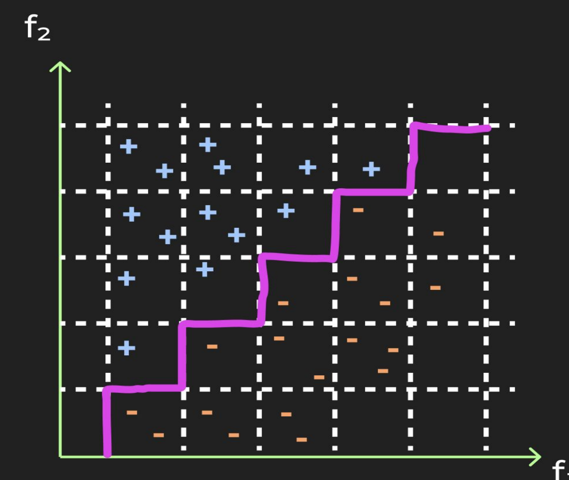
Suppose if we have this data, will DT only work when decision boundaries are axis parallel ?



Decision Tree's decision boundary is made through combination of axis parallel hyperplanes.



Multiple axis parallel hyperplane



Effective Decision Boundary

POINTS TO REMEMBER

- DT splits data into homogeneous regions using axis parallel hyperplanes.
- DT is easily interpretable.
- DT decision boundary are made as a combination of axis parallel hyperplanes.

What do you think, how to create decision tree?

Manually write if else statements?

Since data can be high dimensional and creating if else for each feature is impossible, we need to learn the rules to split data automatically.

How to split the nodes?

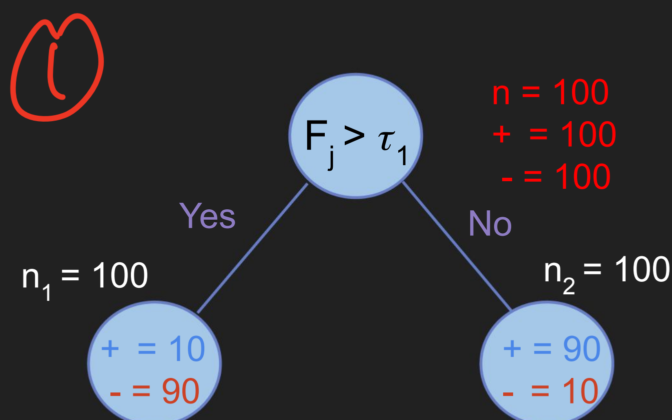
Suppose, there is

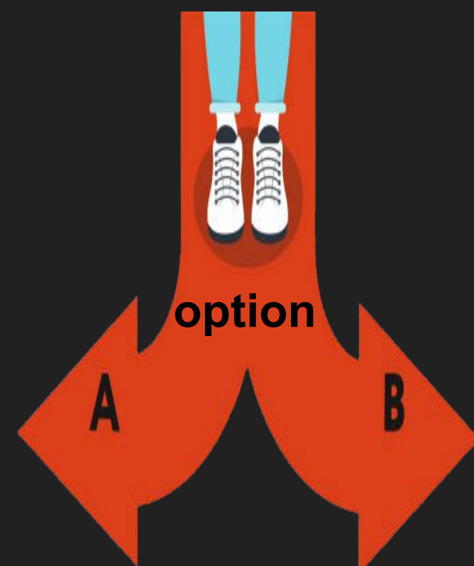
Data : (n = 200) → (+) 100, (-) 100

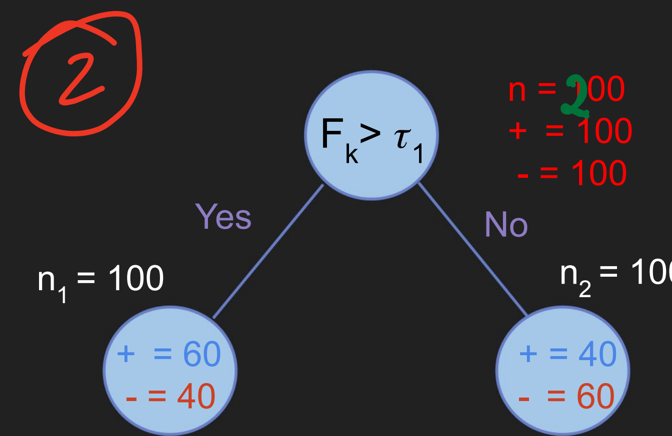
and

Two Features : → f_j, f_k

Which option will you choose?







n_1 = 90% confident for (-)
n_2 = 90% confident for (+)

20 samples misclassified

n_1 = 60% confident for (+)
n_2 = 60% confident for (-)

80 samples misclassified

- Clearly, Option A is better as model is more confident when the node has one class dominating the other, meaning when the node is homogenous/pure node.

Entropy → Impurity ↑

split is bad

How to measure if a node is pure (homogeneous) / impure (heterogenous) ?

Entropy is used to measure the impurity of nodes.

we don't want this → **Entropy ↑ Impurity ↑ Heterogeneity ↑ Homogeneity ↓**

we want this → **Entropy ↓ Impurity ↓ Heterogeneity ↓ Homogeneity ↑**

Entropy Formulation for K-class data : $y = y_1, y_2, y_3 \ldots\ldots\ldots\ldots y_k$

$$H(y) = -\sum_{i=1}^{k} p(y_i) log\, p(y_i)$$

What will be the entropy for our binary case classification problem ?

For Binary Classification,

$y = \{ 0 , 1 \}$,

$$H(Y) = - [p(1) \log_2 p(1) + p(0) \log_2 p(0)]$$

Let $p(1) = p$, then $p(0) = 1 - p$

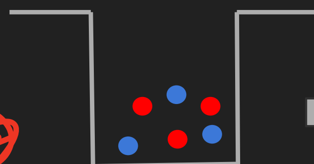
$$\boxed{H(Y) = - [p \log_2 p + (1-p) \log_2 (1-p)]}$$

- The formula is analogous to LogLoss

Understanding Entropy

Say, we have 3 jars containing 6 balls each

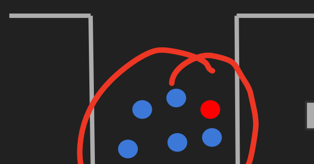
$$\text{Entropy } H(Y) = -\left[p(\text{blue}) \log_2 p(\text{blue}) + p(\text{red}) \log_2 (\text{red})\right]$$



C1

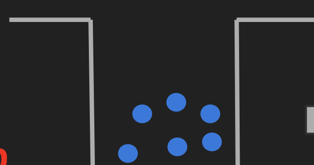
$$H(Y) = -\left[\tfrac{1}{2} \log_2 \tfrac{1}{2} + \tfrac{1}{2} \log_2 \tfrac{1}{2}\right] = 1$$

[0, 1]



C2

$$H(Y) = -\left[\tfrac{5}{6} \log_2 \tfrac{5}{6} + \tfrac{1}{6} \log_2 \tfrac{1}{6}\right] = 0.65$$



Pure

C3

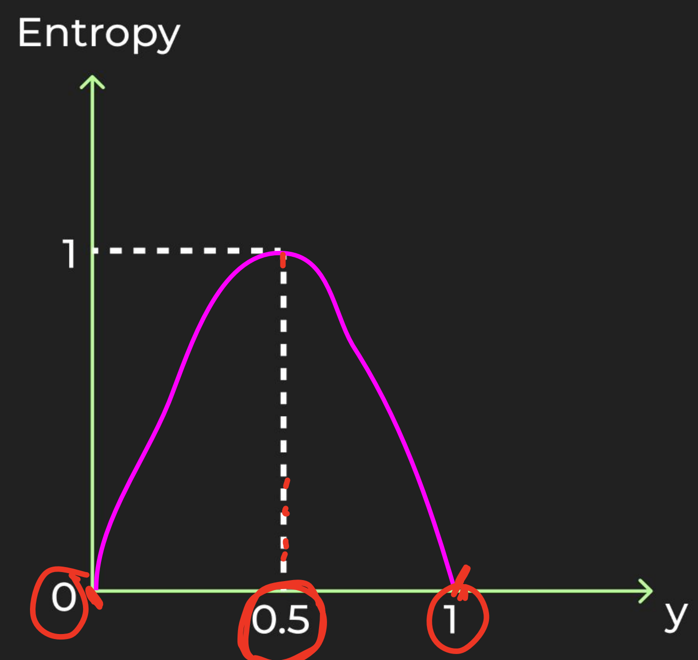
$$H(Y) = -\left[1 \log_2 1 + 0 \log_2 0\right] = 0$$

Understanding Entropy :

C1 → Highly Impure → Entropy ⬆

C3 → Highly Pure → Entropy ⬇

Plotting Entropy :



POINTS TO REMEMBER

- DT splits data into homogeneous regions using axis parallel hyperplanes.
- DT is easily interpretable.
- DT decision boundary are made as a combination of axis parallel hyperplanes.
- Entropy means Impurity.
- For an ideal DT, we want

Entropy ⬇ Impurity ⬇ Heterogeneity ⬇ Homogeneity ⬆

Building a DT intuition

Let data (n = 100) and (y = 0 , 1)

Y - = 60

Y + = 40

Stay

Leaves

We have two features

Gender

Age < 35 (categorical)

What will be the entropy of data (Parent Node) ?

1

Gender

M

F

E = 0.97

E = 0.8

n = 100

y - = 60

y + = 40

$$H(Parent) = -\sum_{i=1}^{k} p(y_i) log\, p(y_i)$$

0.97 - 0.8

$$H(Parent) = - [\, 0.6 \log_2 0.6 + 0.4 \log_2 0.4 \,]$$

0.17

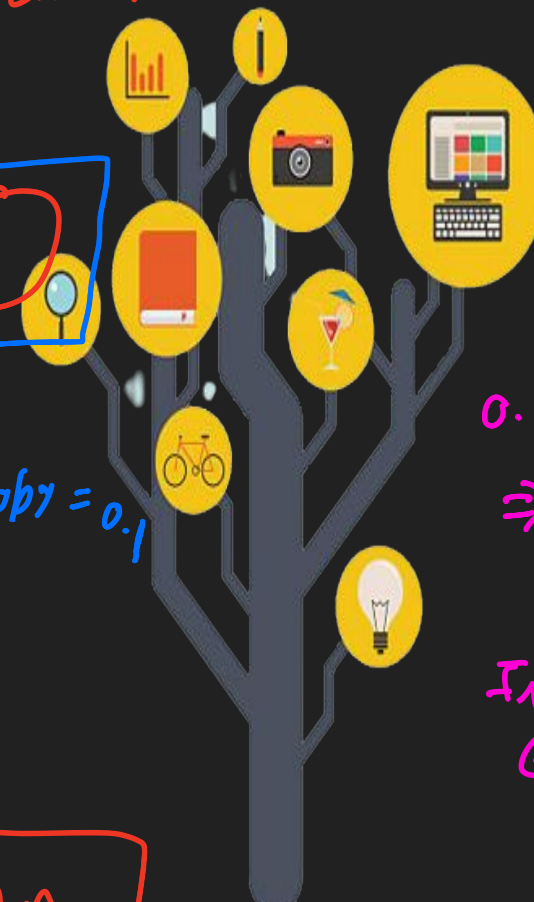
H(Parent) = 0.97 (Entropy very high)

2

Age c3s

Entropy = 0.97

Entropy = 0.1



0.97 - 0.1

⇒ $\boxed{0.87}$

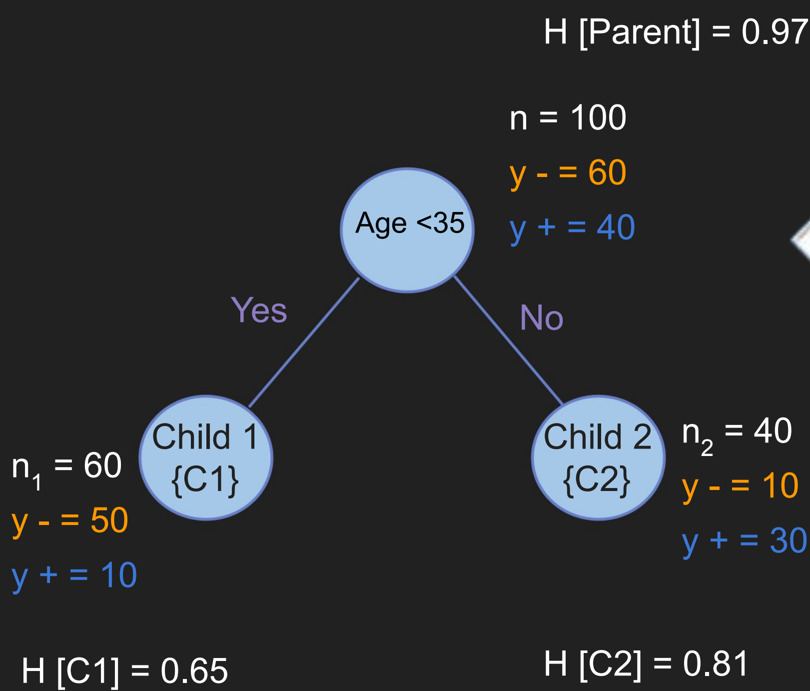
Information Gain

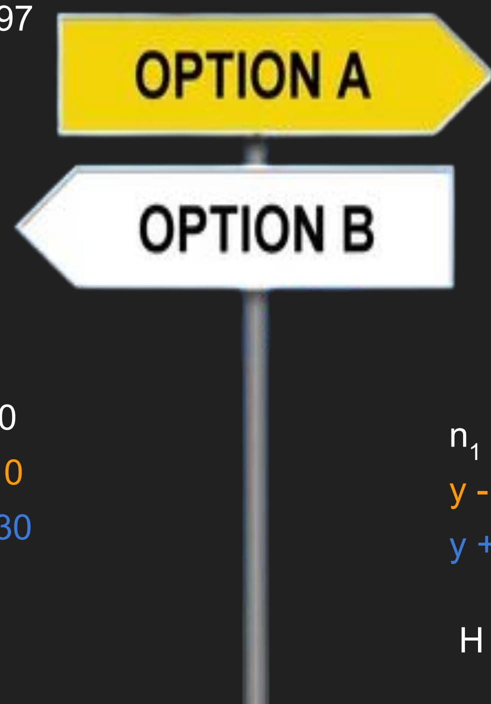
Break: 8:21 AM

Building Decision Tree using Entropy

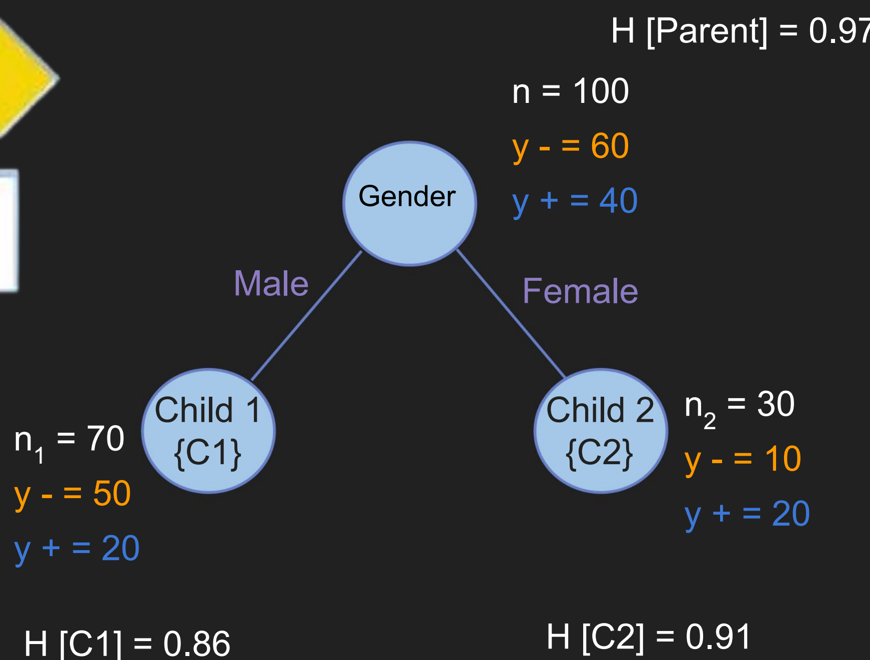
Which feature to use for root node?

Age < 35





Using Gender



- Since just by individual children Entropy we cannot tell which option is better , we need to accommodate the children Entropy into a single formula

Final weighted average for { C1 , C2 } :

$$= -\left[\frac{n_1}{n}H(C_1) + \frac{n_2}{n}H(C_2)\right] \longrightarrow \text{H [Children]}$$

Gender

H [Children] = 0.88

Age < 35

H [Children] = 0.714

$\frac{60}{100} * .8 + \frac{40}{100} * .9$

E = 90%

100

60

40

E_{C_1} = 80%

E_{C_2} = $\boxed{70\%}$

$\frac{80\% + 70\%}{2}$

E = 90%

100

50

50

E_{C_1} = 80%

E_{C_2} = 70%

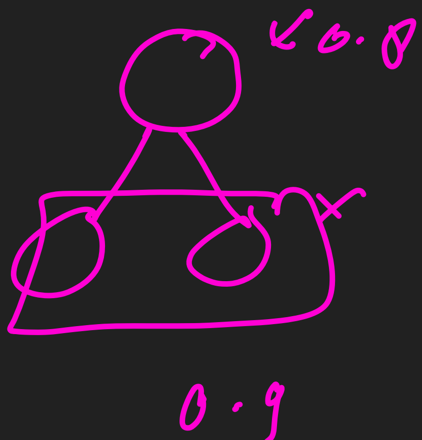
Reduction in Entropy

The **reduction in entropy** i.e. Parent - weight entropy of child is termed as **Information gain**

Reduction in entropy = H (Parent) - H (Children)

↓

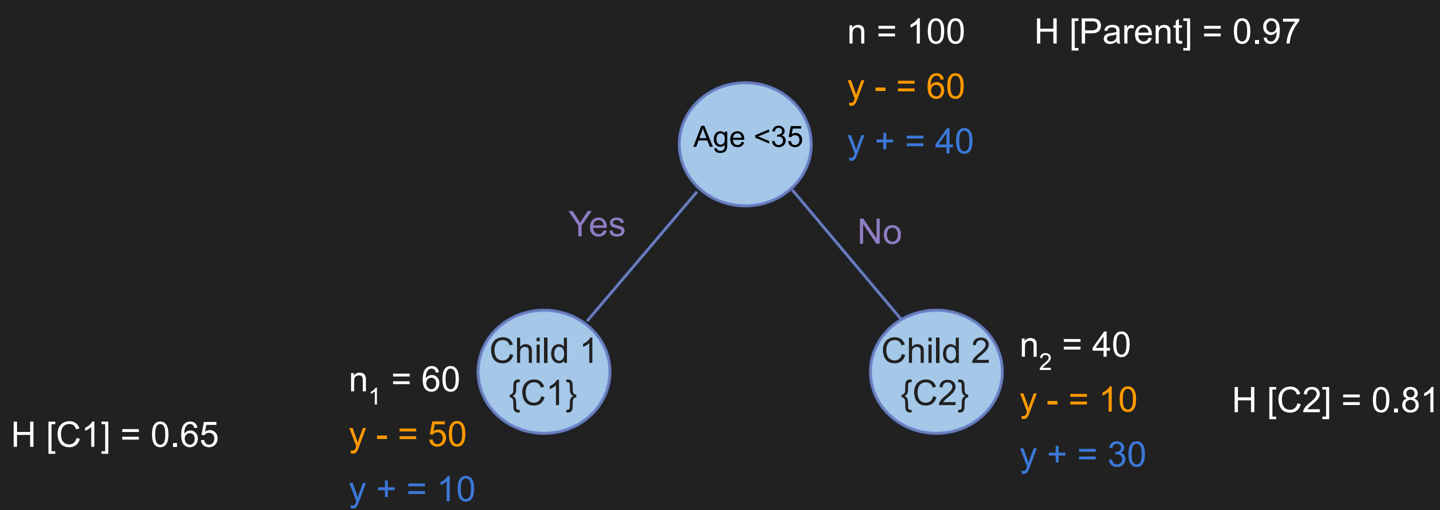
Information Gain = H (Parent) - H (Children)



Gender	Age < 35
IG = 0.97 - 0.88	IG = ~~0.57~~ - 0.719 − 0.57
= 0.09	= 0.257

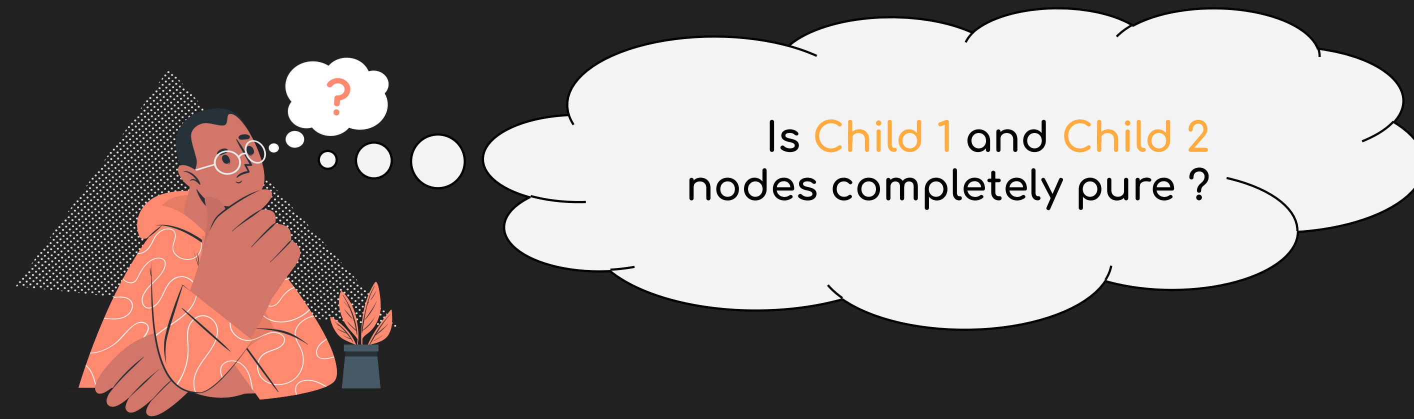
Information gain is more hence Age <35 is better than Gender.

Splitting using Age < 35 factor

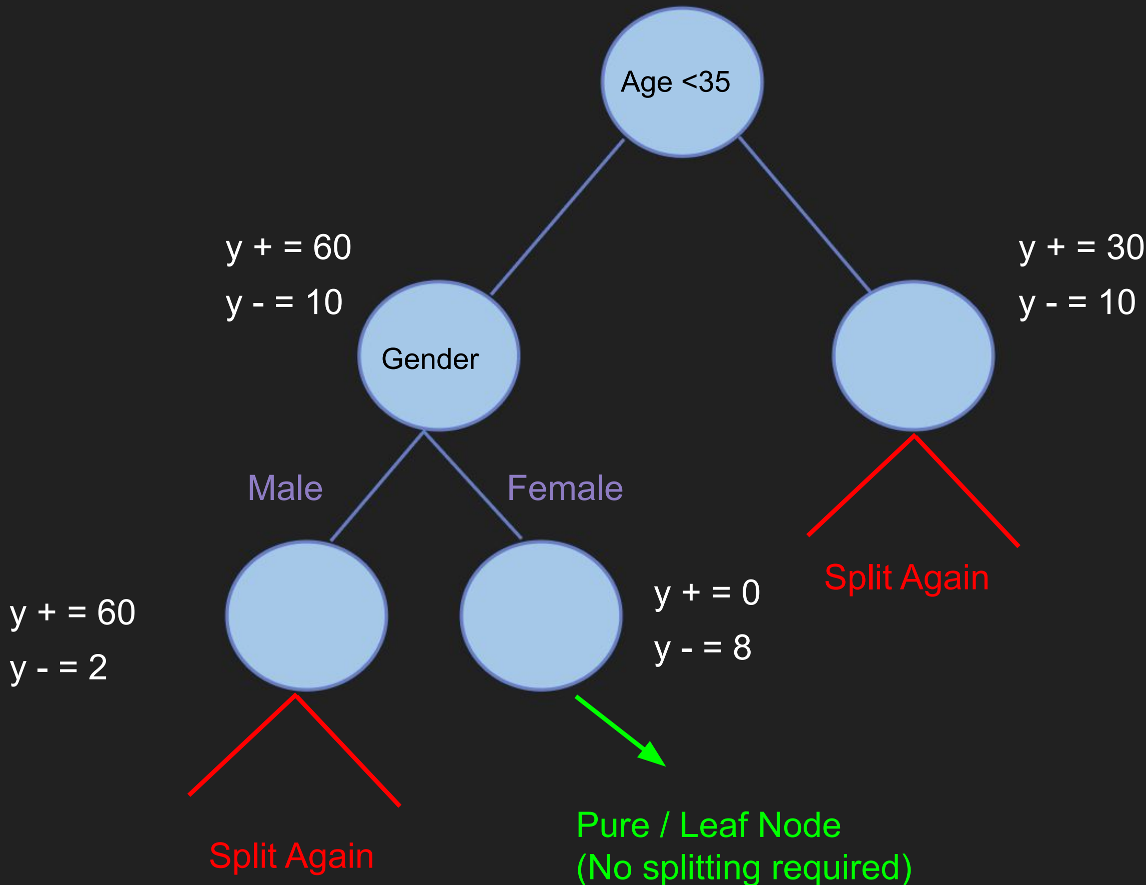


Weighted entropy of child = 0.714

Information Gain = 0.257



Splitting the Age < 35 nodes again



Step 1

For a node, calculate IG for all the features and choose the feature with the highest IG

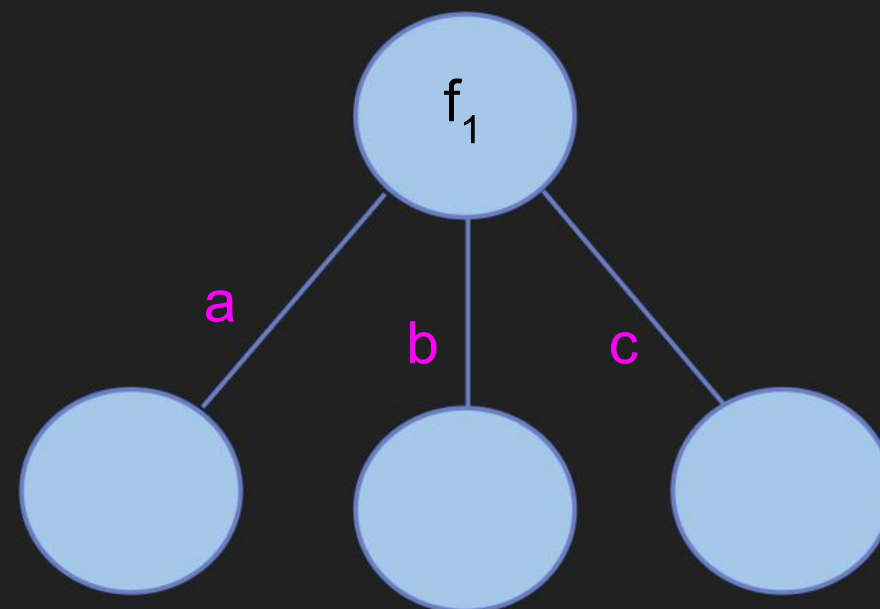
Step 2

Repeat Step 1 until we get the purer nodes

Splitting nodes with more than two feature categories

f_1	f_2	y
a	x_1	1
b	x_2	0
c	x_1	1

Easily split categorical features



Splitting using f_1

POINTS TO REMEMBER

- DT splits data into homogeneous regions using axis parallel hyperplanes.
- DT is easily interpretable.
- DT decision boundary are made as a combination of axis parallel hyperplanes.
- Entropy means Impurity.
- For an ideal DT, we want

Entropy ↓ Impurity ↓ Heterogeneity ↓ Homogeneity ↑

POINTS TO REMEMBER

- Information gain is the measure of how much information a feature provides to DT.
- Split the nodes until pure node is reached.
- Easily splits categorical data.