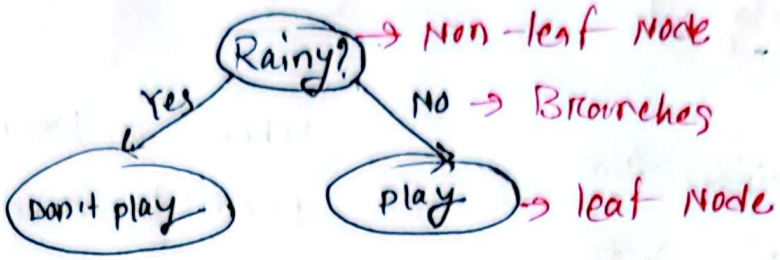
# Classification Tree

## Decision Tree –

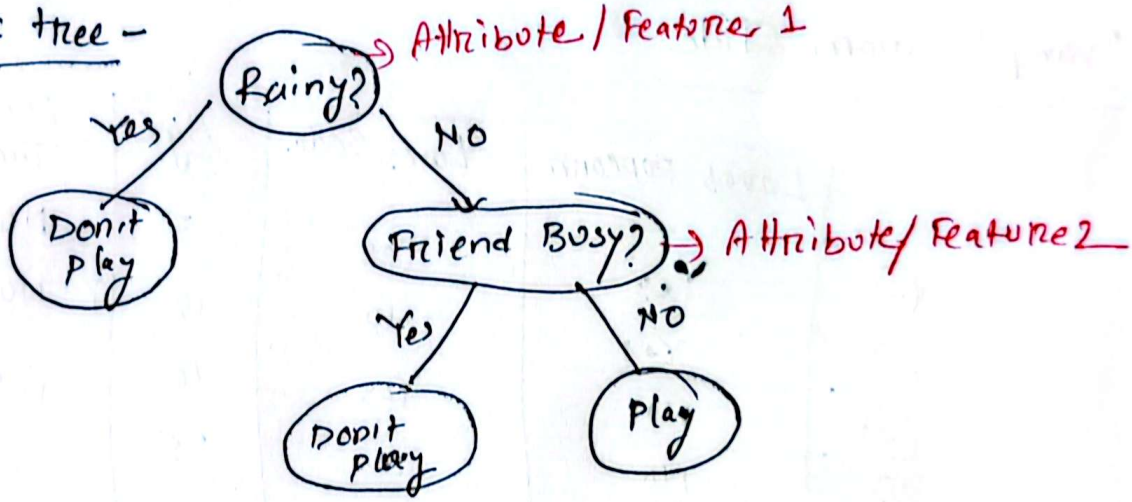



Non-leaf Nodes are Attributes / Features

leaf Nodes are Labels / classes

Branches are outcomes of an feature (can be more than 2)

More complex tree –

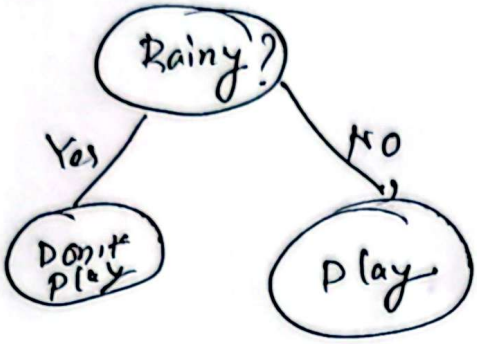


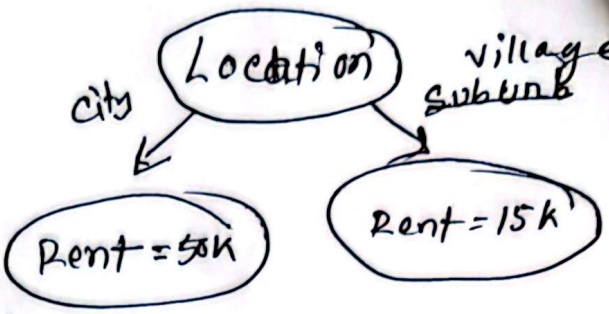

This tree can be represented as the following Data set –

| | Rainy? | Friend Busy? | Play? |
|---|---|---|---|
| sample 1 | Yes | Yes | NO/Don't play |
| sample 2 | Yes | NO | NO |
| sample 3 | NO | Yes | NO |
| sample 4 | NO | NO | Play/Yes |

Attributes / Features

We usually Derive model / tree from this type of Dataset.




When a Decision tree classifies things into categories — Classification Tree

When a Decision tree predicts numeric values — Regression Tree

The features can be Both Neumerical and Categorical.

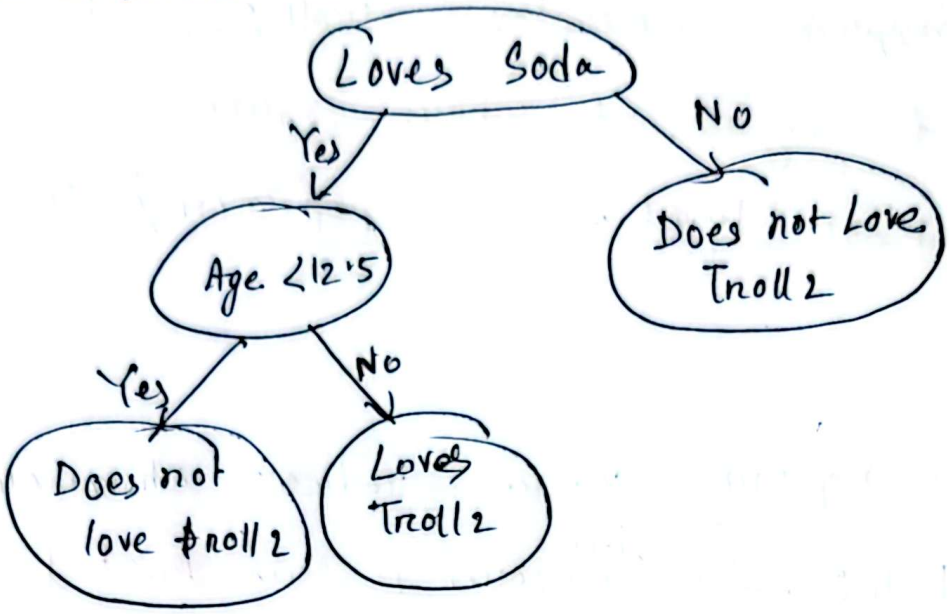
Example from Slide —

| | Loves Popcorn | Loves Soda | Age | Loves Troll2 |
|---|---|---|---|---|
| 1 | Yes | Yes | 7 | No |
| 2 | Yes | No | 12 | No |
| 3 | No | Yes | 18 | Yes |
| 4 | No | Yes | 35 | Yes |
| 5 | Yes | ~~No~~ Yes | 38 | Yes |
| 6 | Yes | No | 50 | No |
| 7 | No | No | 83 | No |

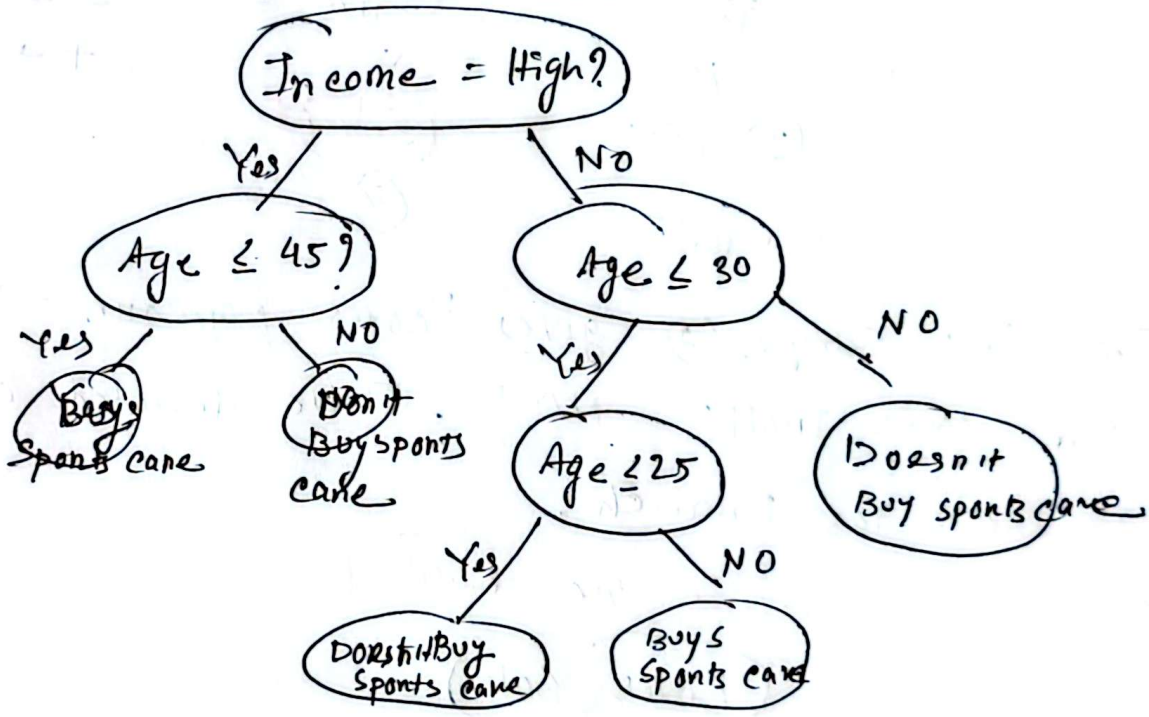
Categorical — Numerical

We want to predict if someone loves troll2 or not using Classification Tree.

The classification tree —




Another tree Example —

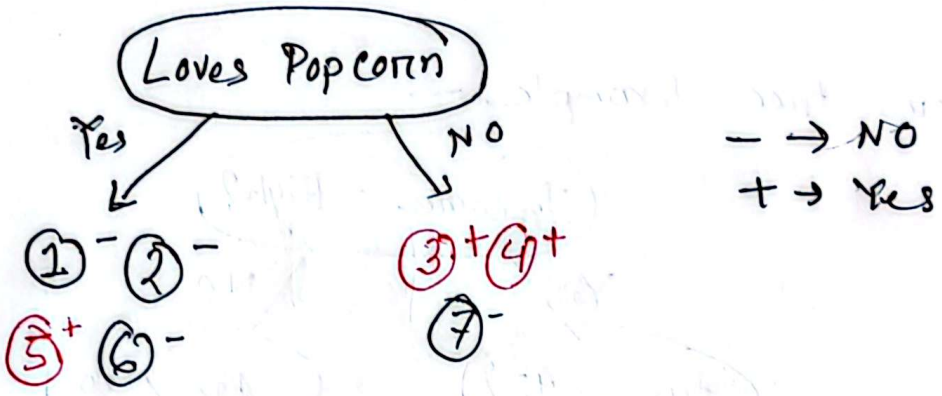



Numeric features threshold can be different for the same data at different levels of the tree.

From a classification tree we can predict a label / class for a new / unknown sample.

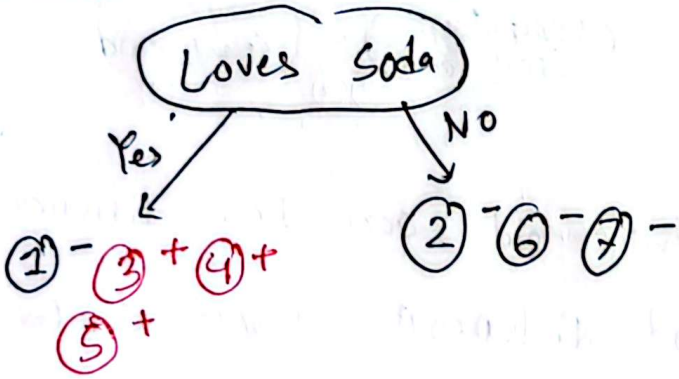
Now if we want to Build a tree for the Dataset to predict whether someone loves troll 2, we at first need to figure out which feature should be at the Root level — Loves popcorn / Loves soda / Age.

How well 'Loves Popcorn' feature predicts whether someone likes troll 2 — [if 'Loves popcorn' is the Root]




The first sample (1) gives 'Loves popcorn = Yes' and 'Loves Troll2 = NO'. So we denoted as (1)⁻ on the Yes Branch.

[if 'Loves Soda' is the Root]

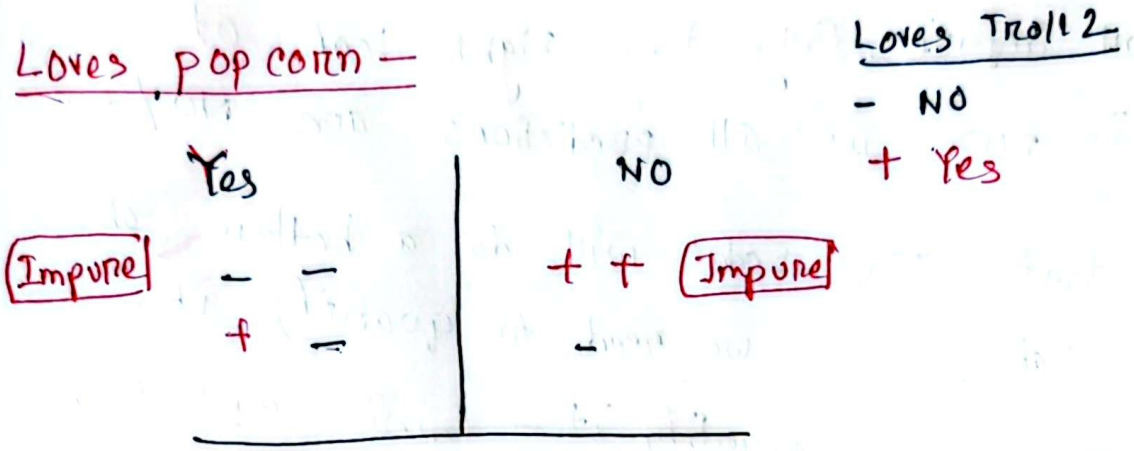



Neither of the categorical features does the perfect job to predict whether someone

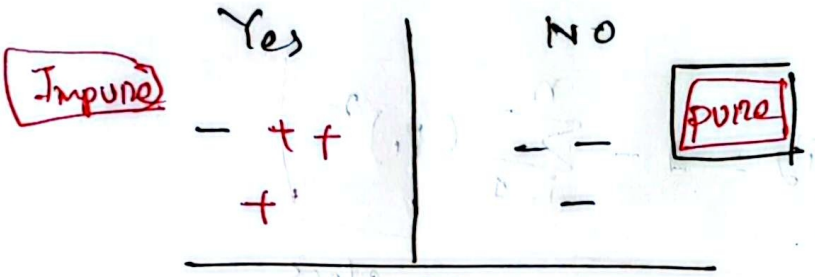
Loves Troll 2.

We can see/visualize this Graphically —

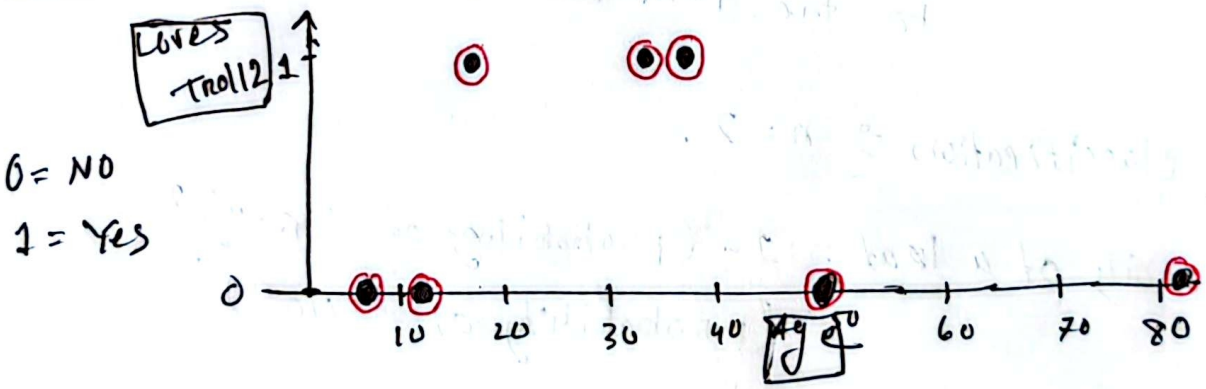
Loves popcorn —




Loves Soda —




Age —




We cannot fit a best fit line for these points to perfectly classify Loves Troll 2. So Age feature cannot perfectly predict as well.

As the leaves of 'Loves popcorn' and left leaf of 'Loves Soda' contains mixure of Yes, NO (+, -), They are call impure. Only the Right leaf of 'Loves Soda' is pure as all predictions are NO/--.

It seems that 'Loves Soda' will do a better job to predict the Labels. But we need to quantify it.

There are methods to quantify it - Gini impurity, Entropy, information Gain.

## Gini Impurity:

Formula → $$\text{Gini Impurity (of a leaf)} = 1 - \sum_{i=1}^{n} (P_i)^2$$

n = number of Label or class

$P_i$ = Probability of sample ~~being~~ belonging to the i-th class.

For binary classification → n = 2.

∴ Gini Impurity of a leaf = 1 - (probability of "Yes")$^2$ - (probability of "No")$^2$


① - ② - ⑤ + ⑥ -
Yes Yes NO Yes

Gini impurity of Leaf

$= 1 - (P(Yes))^2 - (P(NO))^2$

$= 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2$

$= 0.375$

③ + ④ + ⑦ -

Gini impurity of leaf

$= 1 - (P(Yes))^2 - (P(NO))^2$

$= 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2$

$= 0.444$

As there are different number of people at each leaf (4 at left leaf, 3 at right leaf), we will calculate Total Gini Impurity By taking weighted average of the Gini of each leaf.

Total Gini impurity = weighted average of Gini impurities for the leaves.

For binary classification

Total Gini impurity = P(left leaf) x Impurity of left leaf +
P(Right leaf) x impurity of right leaf.

∴ Total Gini impurity (Loves popcorn)

$= \frac{4}{7} \times 0.375 + \frac{3}{7} \times 0.44.$

$= 0.405$

Similarly, total Gini impurity (Loves Soda) = 0.214.

## Gini impurity of Numeric Data:

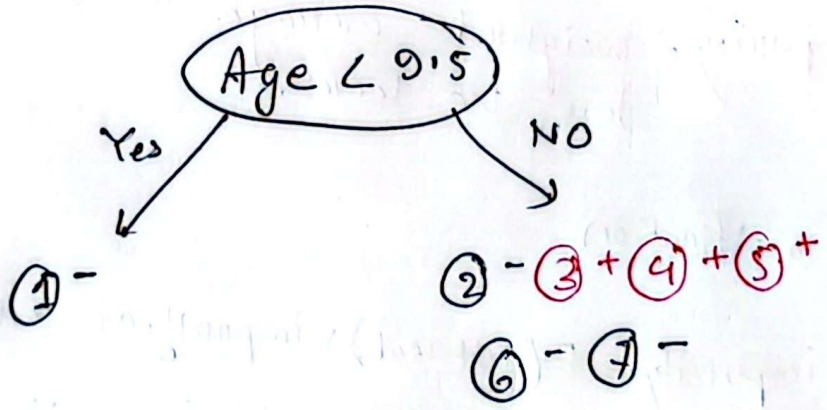
For the Age feature — [Sort the ~~colum~~ feature ~~Data set~~ by Age [Already sorted]. Then calculate average of all adjacent rows / Age of people]

7 12 18 35 38 50 83

↓ ↓ ↓ ↓ ↓ ↓

9.5 15 26.5 36.5 44 66.5

Now we have to calculate gini impurity ~~for~~ by setting each average Age as threshold.




Total Gini impurity (Age < 9.5) = $\frac{1}{7}\left\{1-\left(\frac{0}{1}\right)^2-\left(\frac{1}{1}\right)^2\right\}$

$+\frac{6}{7}\left\{1-\left(\frac{3}{6}\right)^2-\left(\frac{3}{6}\right)^2\right\}$

$= \frac{1}{7} \times 0 + \frac{6}{7} \times 0.5$

$= 0.429$

Similarly Gini impurity of all thresholds →

| 7 | 12 | 18 | 35 | 38 | 50 | 83 |
|---|---|---|---|---|---|---|
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| 9.5 | 15 | 26.5 | 36.5 | 44 | 66.5 | |
| Total Gini= 0.429 | **0.343** | 0.476 | 0.476 | **0.343** | 0.429 | |
| | Smallest | | | Smallest | | |

Total Gini for Age < 15 and Age < 50 ~~am~~ is 0.343 and this is the smallest.

Small impurity = Better feature to predict the class.

We can pick either ~~anyone~~.

So calculated Gini impurity values

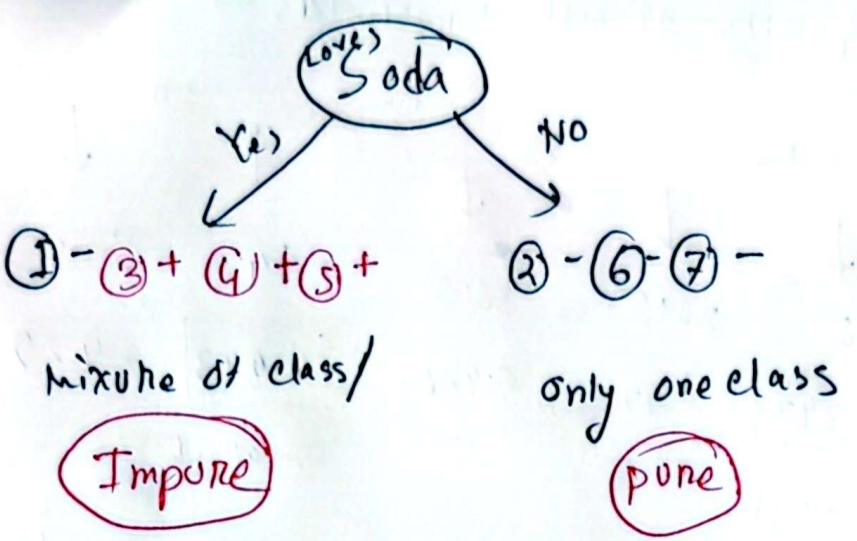
Total Gini (Loves popcorn) = 0.405

Total Gini (Loves Soda) = ~~0.343~~ 0.214

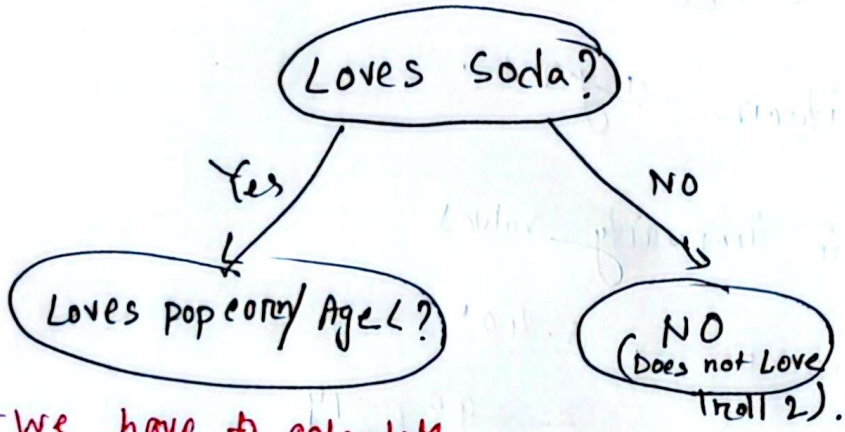
Total Gini (Age < 15) = 0.343.

smallest Gini impurity → Loves Soda. [It's leaves has lowest impurity]

So we put loves soda at the top / Root of the classification tree.




Now we have to feagure out what will be the left Node = Loves popcorn? Age < ?




* we have to calculate Gini impurity of Loves ~~Soda~~ popcorn and Age where Loves soda = Yes

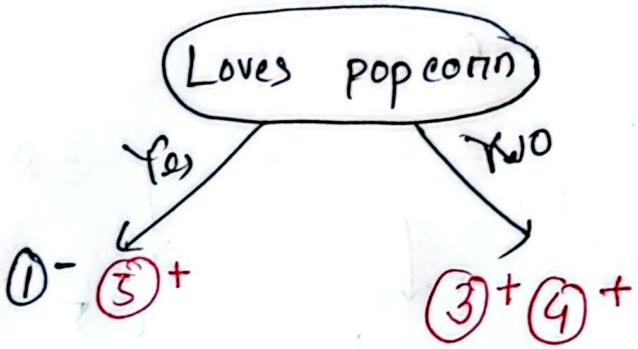
| | Loves Pop Corn | Loves Soda | Age | Loves Troll 2 |
|---|---|---|---|---|
| ① | Yes | Yes | 7 | NO |
| ② | Yes | NO | 12 | NO |
| ③ | NO | Yes | 18 | Yes |
| ④ | NO | Yes | 35 | Yes |
| ⑤ | Yes | Yes | 38 | Yes |
| ⑥ | Yes | NO | 50 | NO |
| ⑦ | NO | NO | 83 | NO |

**NOTE** We will skip the categorical features if it has already been used as node, but can use numerical feature with different threshold multiple times.

Data set with "Loves Soda = Yes" →

| | Loves popcorn | Loves soda | Age | Loves Troll2 |
|---|---|---|---|---|
| 1 | Yes | Yes | 7 | NO |
| 3 | NO | Yes | 18 | Yes |
| 4 | NO | Yes | 35 | Yes |
| 5 | Yes | Yes | 38 | Yes |

Gini Impurity of Loves popcorn –




$$\text{Gini(Loves popcorn)} = \frac{2}{4}\left\{1-\left(\frac{1}{2}\right)^2-\left(\frac{1}{2}\right)^2\right\}+\frac{2}{4}\left\{1-\left(\frac{2}{2}\right)^2-\left(\frac{0}{2}\right)^2\right\}$$

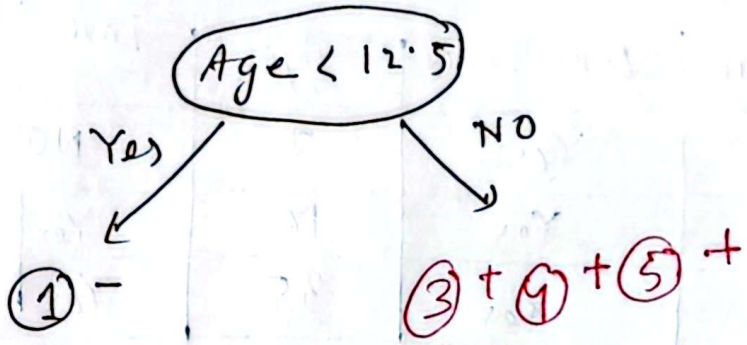
$$= 0.25$$

## Gini of Age -

7 18 35 38

↓ ↓ ↓

Average of Adjacent rows. → 12.5 26.5 36.5




$$\text{Gini}(\text{Age} < 12.5) = \frac{1}{4}\left\{1-\left(\frac{0}{1}\right)^2-\left(\frac{1}{1}\right)^2\right\} + \frac{3}{4}\left\{1-\left(\frac{3}{3}\right)^2-\left(\frac{0}{3}\right)^2\right\}$$

$$= 0$$

| 7 | 18 | 35 | 38 |
|---|---|---|---|
| ↓ | | ↓ | ↓ |
| 12.5 | | 26.5 | 36.5 |
| ↓ | | ↓ | ↓ |
| Gini → 0 | | 0.25 | 0.33 |

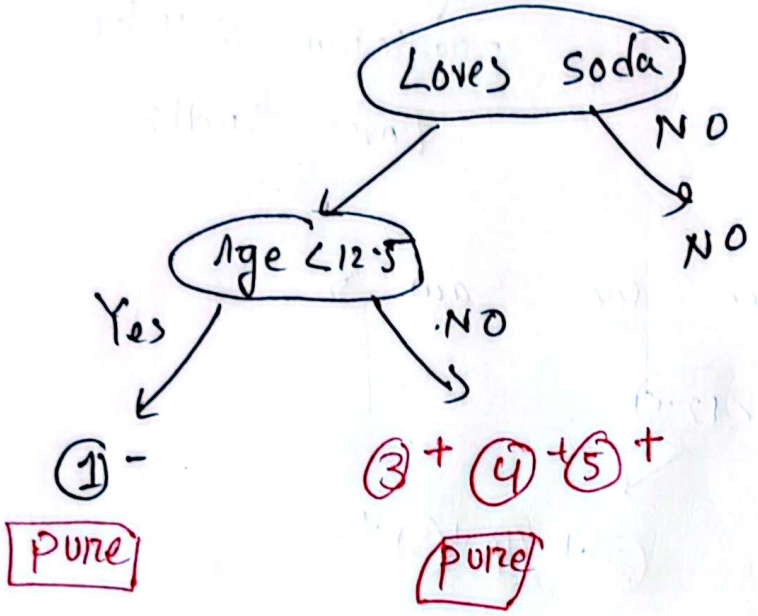
So smallest is Gini (Age < 12.5) = 0

Calculated Gini →

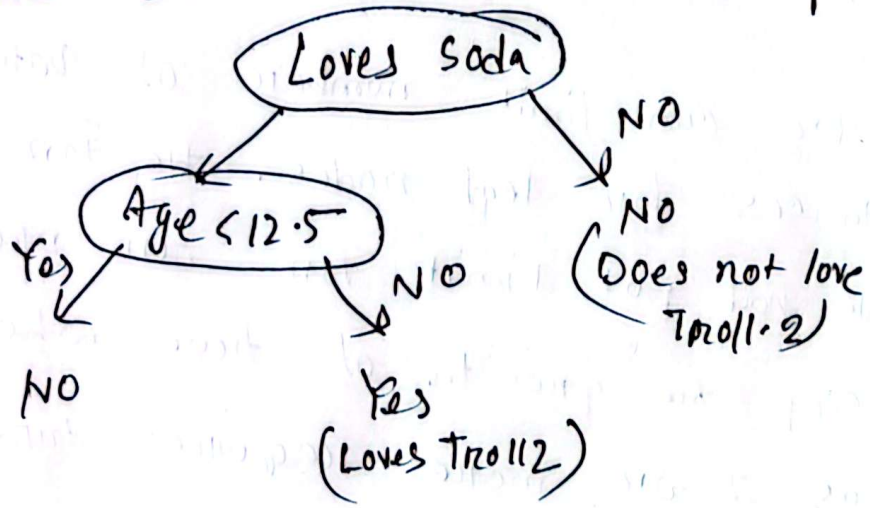
Gini (Popcorn) = 0.25

Gini (Age < 12.5) = 0

∴ Gini (Age < 12.5) is smallest




As we got pure leaves, we can stop building the tree now. So the Classification Tree for the Data set after assigning output values →

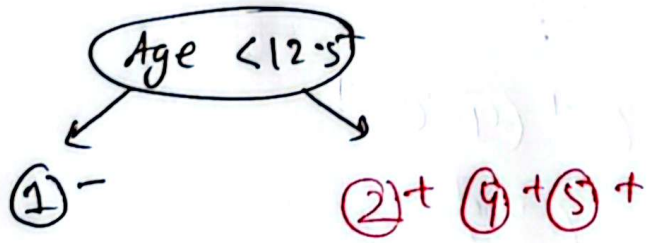


Predicting for unknown data —

| Loves Popcorn | Loves Soda | Age | Loves Troll2 |
|---|---|---|---|
| NO | Yes | 20 | ? |

using the tree, the prediction will be 'Loves Troll2"

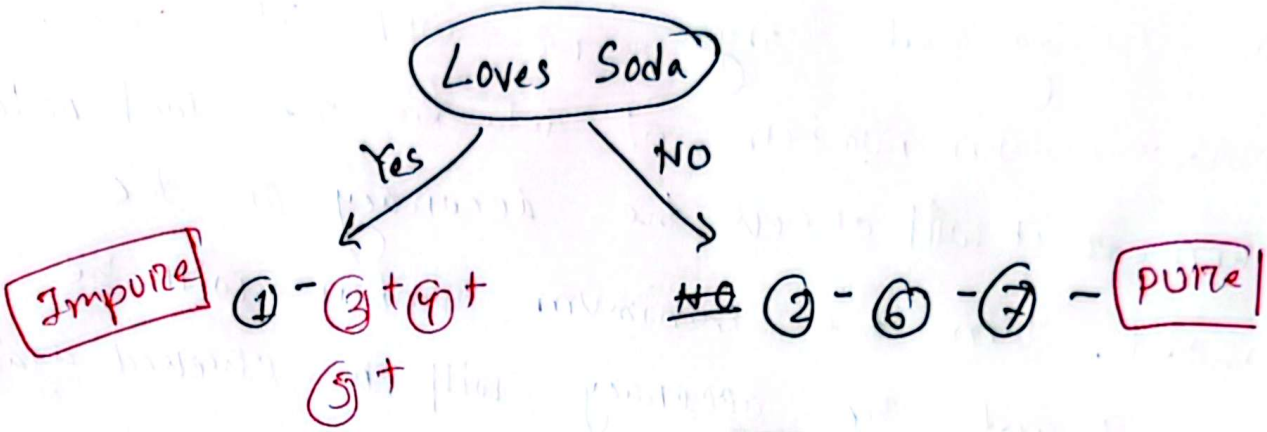
While building the tree we saw →




→ only one ~~person~~ / data could made it to this leaf. This could possibly cause Overfitting and might not work ~~good~~ ~~for~~ well for unknown data.

To solve this, we can limit number of Data point or instances for leaf nodes. ~~the~~ for example if we had put limit for our previous example as ~~Stop the growth of tree when~~ leaf node ~~has 3 or more~~ requires three or more data/instance, then the tree

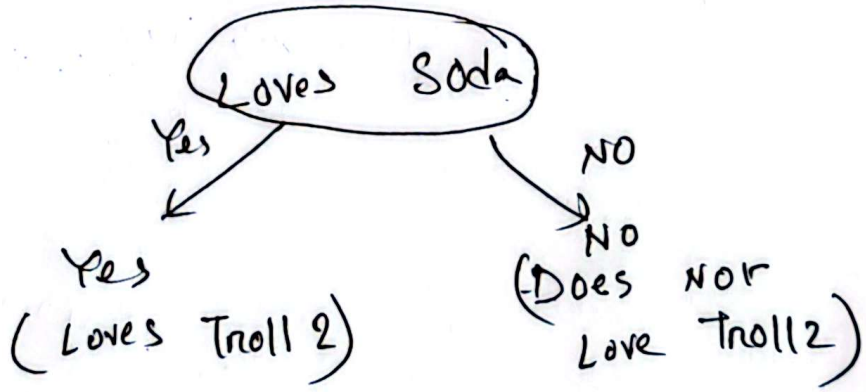
would stop at →




cause splitting these leaves ~~with~~ can ~~cause~~ leave us with leaves less than three data per leaves.

For pure leaves we can easily decide what will be the output value. But for impure leaves in this ~~case~~ case we will count which output / Label is maximum.

Here positive / Yes occurs 3 times in the impure ~~leaf~~ leaf, so we will put Yes.




The limit can be chosen by doing Cross Validation. We don't know which limit will give us better accuracy.

Cross validation — After splitting the data set for training and testing, It will set the minimum number of data for the leaf node. Then it will check the accuracy of the model. Then the minimum number will be reduced and the accuracy will be checked again We will continue doing it until we are getting high accuracy. If we get low accuracy we will stop.

For example we can start with limit = 10, calculate model accuracy then limit = 9, calculate accuracy and continue until the accuracy is decreasing.