An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Tree Based Methods

Conducted by:
iHUB Divya Sampark IIT Roorkee
and
Ritvij Bharat Private Limited (RBPL)

Presented by:
Shreyas Shukla

Tree Based Methods

Three main methods:

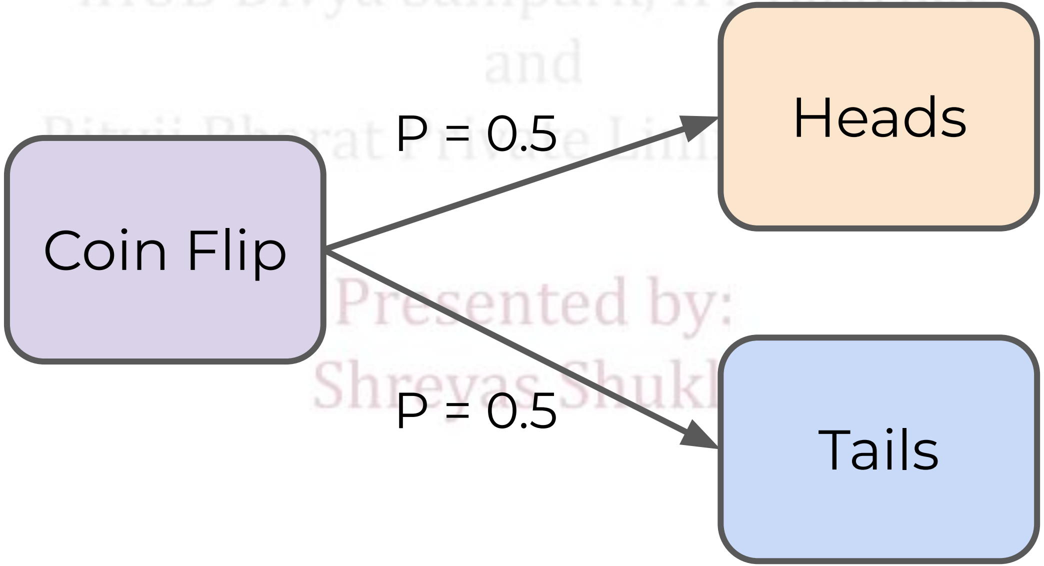
- Decision Trees
- Random Forests
- Boosted Trees

Decision Trees

Theory and Intuition

- While the use of basic decision trees for modeling choices and outcomes have been around for a very long time, statistical decision trees are a more recent development.
- **Note the difference here**!

The general term “decision tree” can refer to a flowchart mapping out outcomes.



Decision Tree Learning refers to the statistical modeling that uses a form of decision trees, where node splits are decided based on an information metric.

Decision trees methods is basically the ability to split data based on information from features.

We need a mathematical definition of information and the ability to measure it.

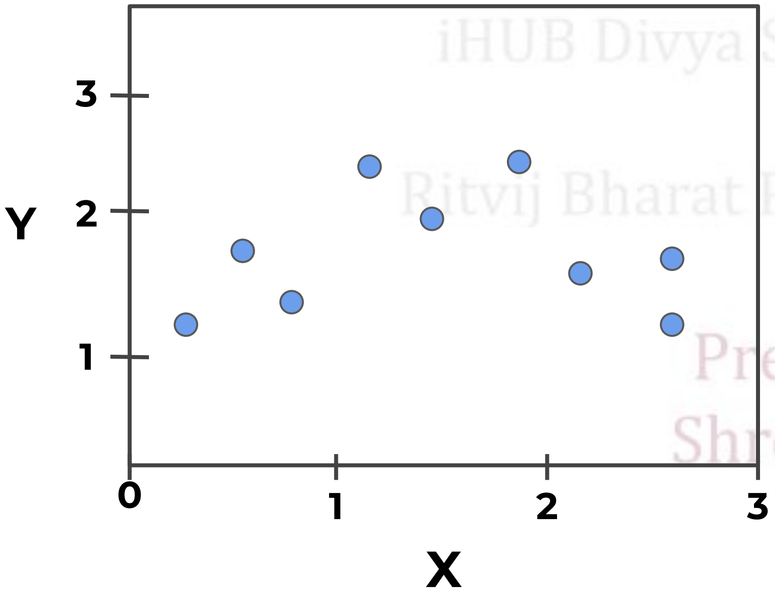
The ability to measure and define information will become more important as we learn the mathematics of tree based methods.

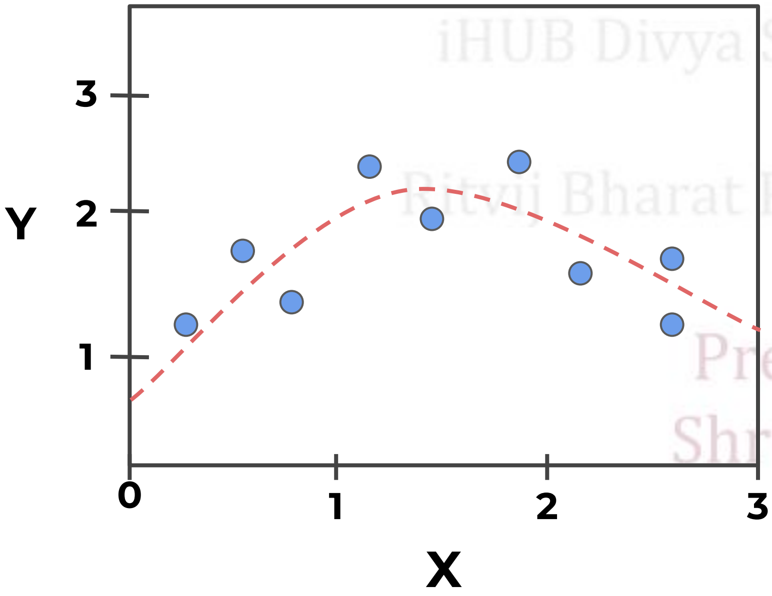
Let’s talk about the development of decision trees.

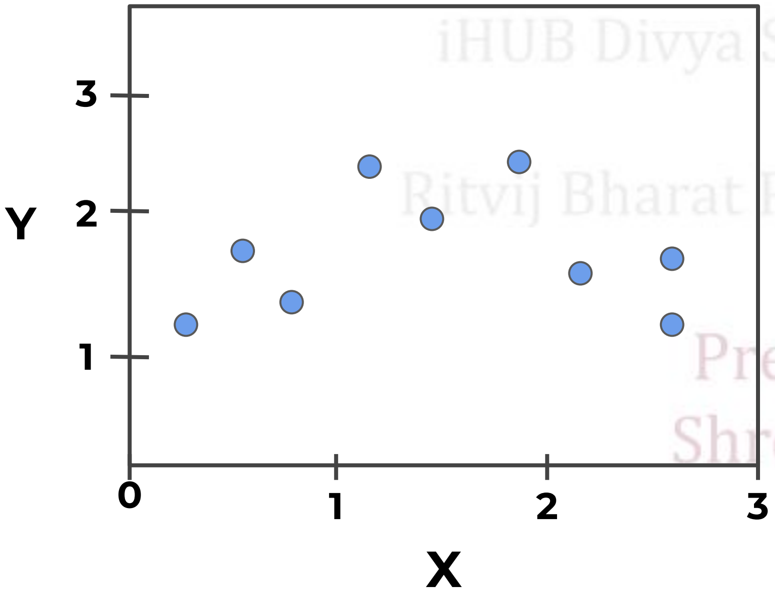
1963: First publication of regression tree algorithm by Morgan and Sonquist

1963: Morgan and Sonquist created piecewise-constant model with splits.

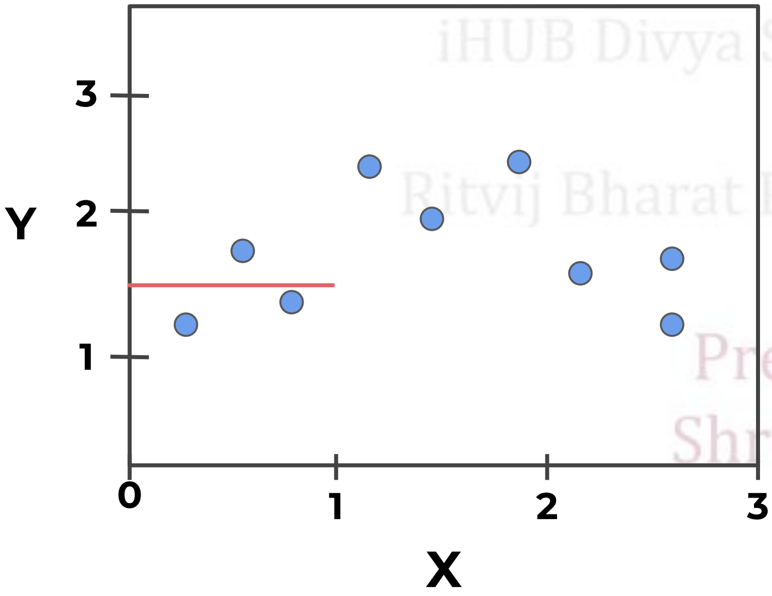
1963: Piecewise-constant regression tree

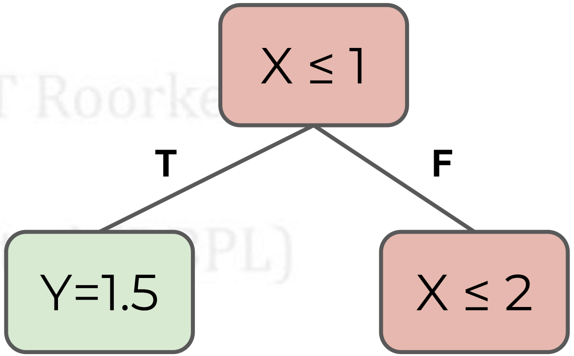

3
2
1
Y
0
1
2
3
X

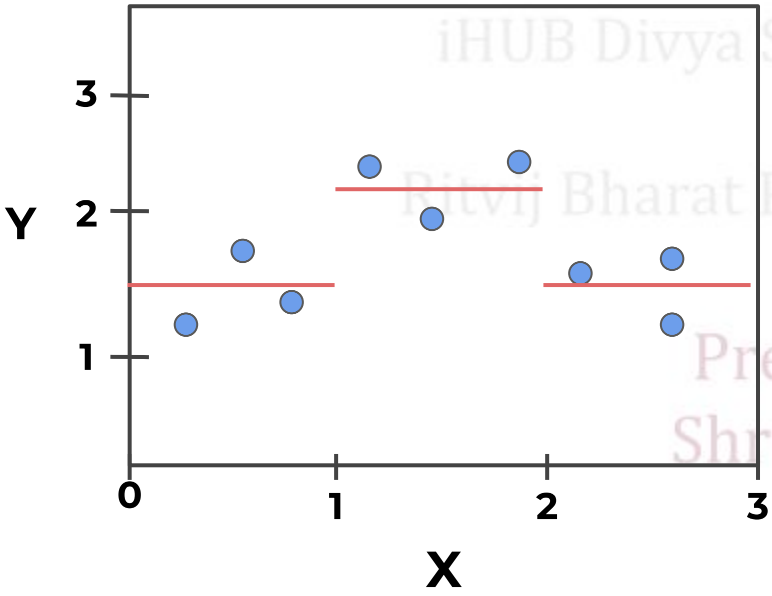

Y
3
2
1
0
1
2
3
X

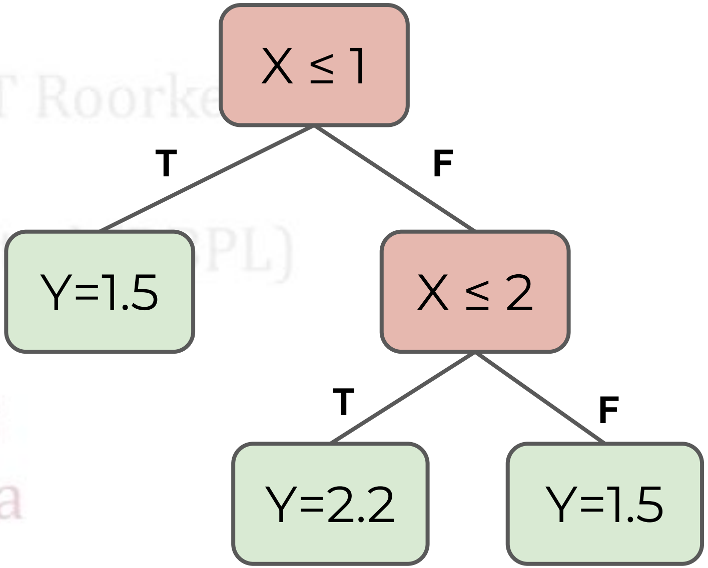


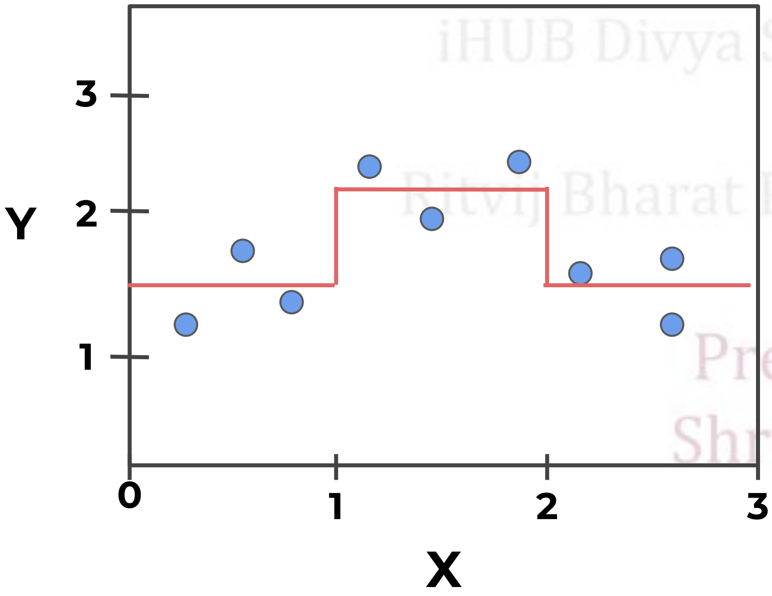
X < 1

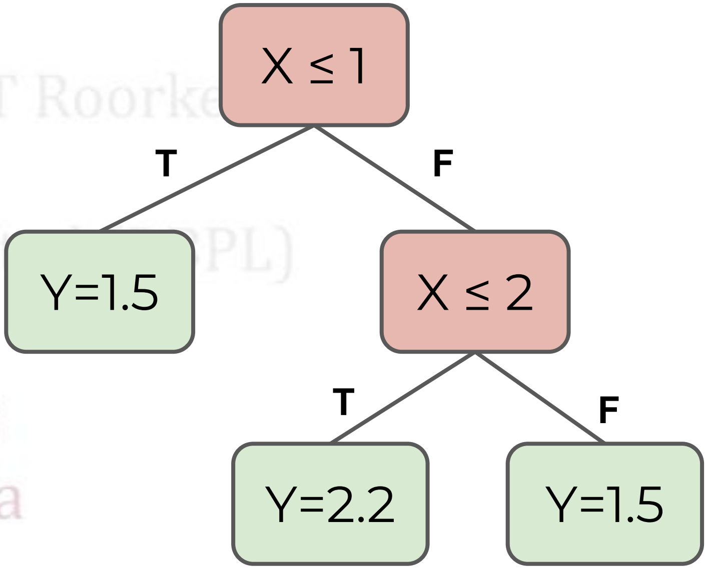

Y
3
2
1
0
1
2
3
X


X ≤ 1
T
F
Y=1.5
X ≤ 2


Y
3
2
1
0
1
2
3
X


X ≤ 1
T
F
Y=1.5
X ≤ 2
T
F
Y=2.2
Y=1.5


Y
3
2
1
0
1
2
3
X


X ≤ 1
T
F
Y=1.5
X ≤ 2
T
F
Y=2.2
Y=1.5

Node impurity

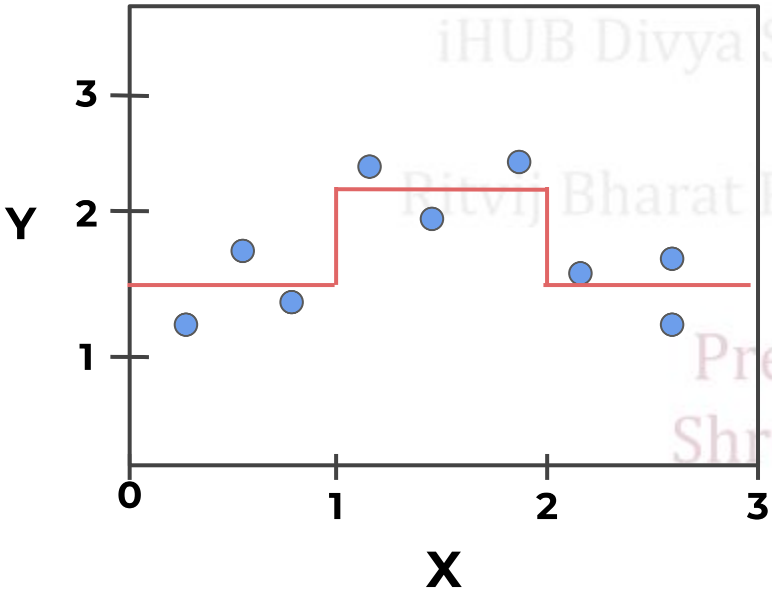
$$\phi(t) = \sum_{i \in t} (y_i - \bar{y})^2$$

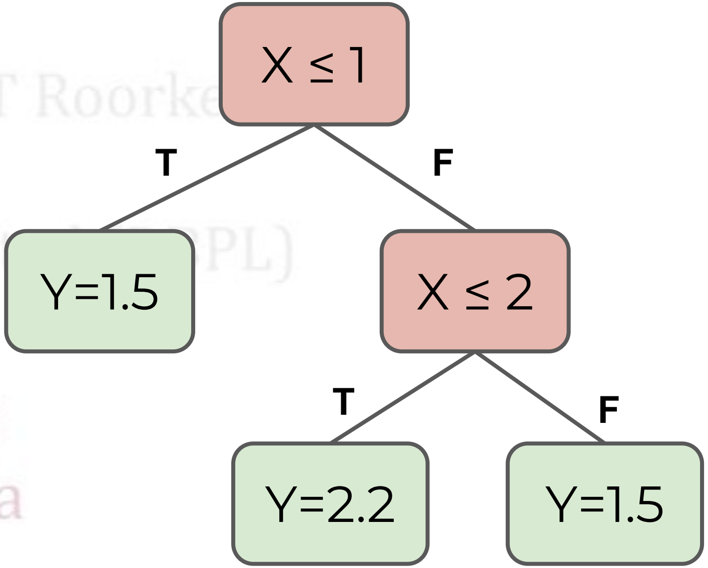
Decision Trees

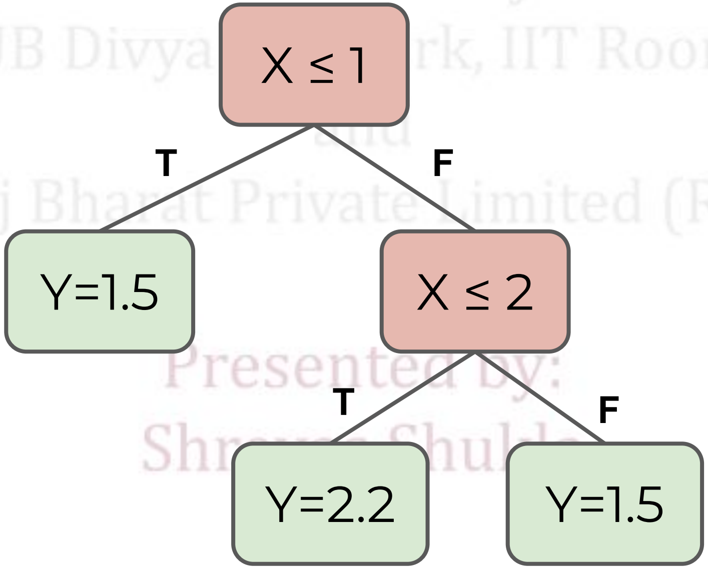
Decision Tree Basics

Let us understand some terminology about the decision tree components.

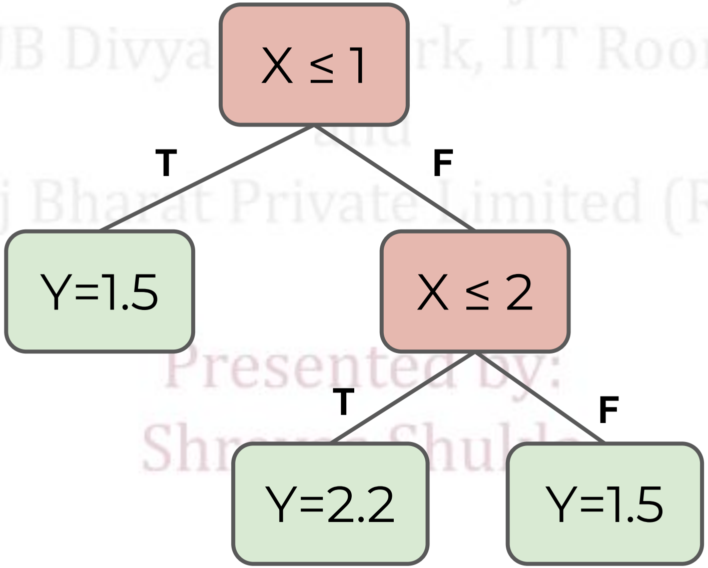
Recall our simple regression tree:



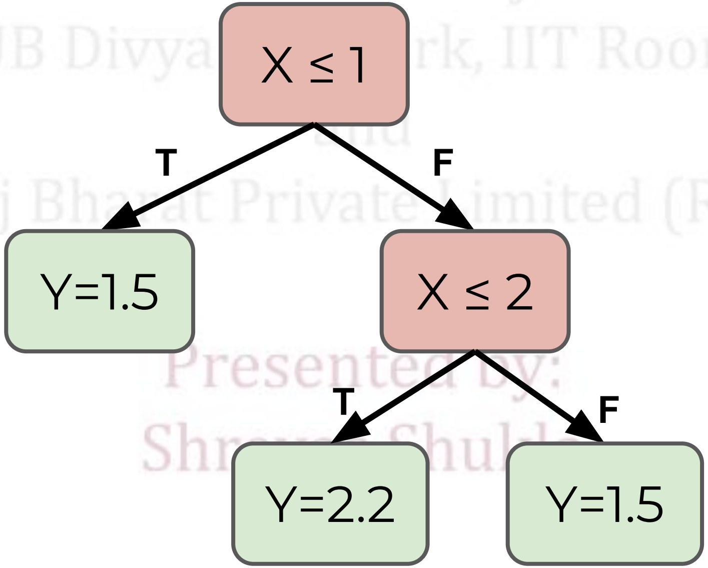



X ≤ 1
T
F
Y=1.5
X ≤ 2
T
F
Y=2.2
Y=1.5

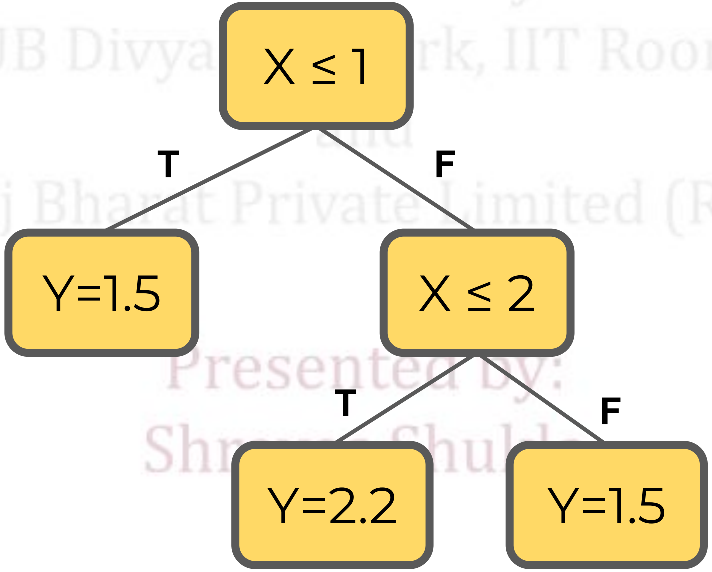
Splitting



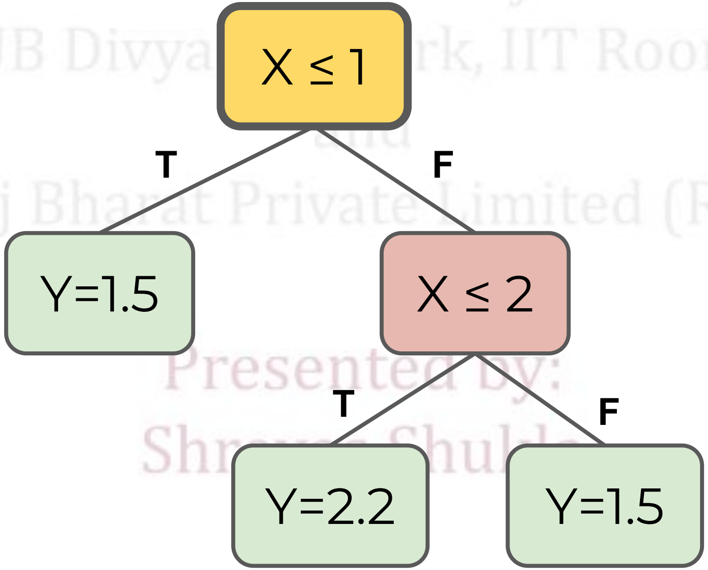
Splitting



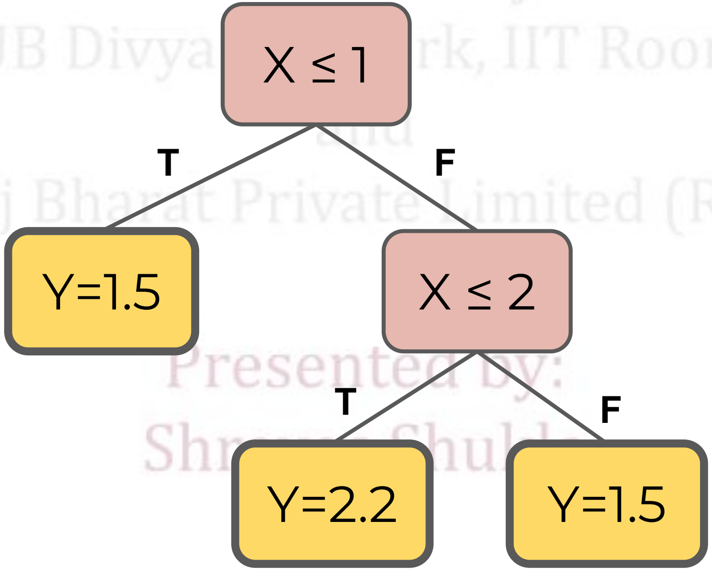
Nodes:



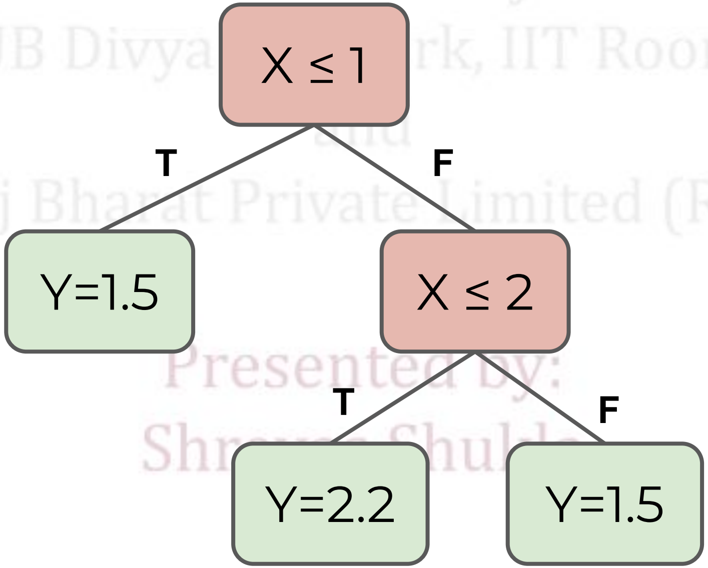
Root Node:



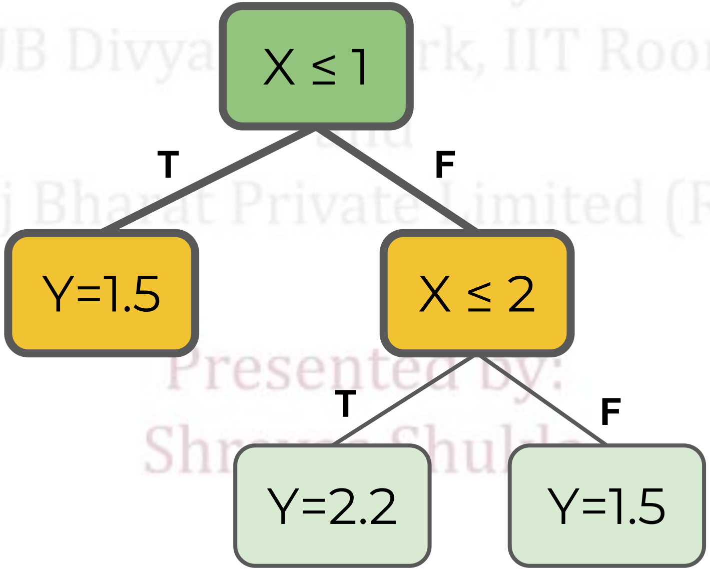
Leaf (Terminal) Nodes:



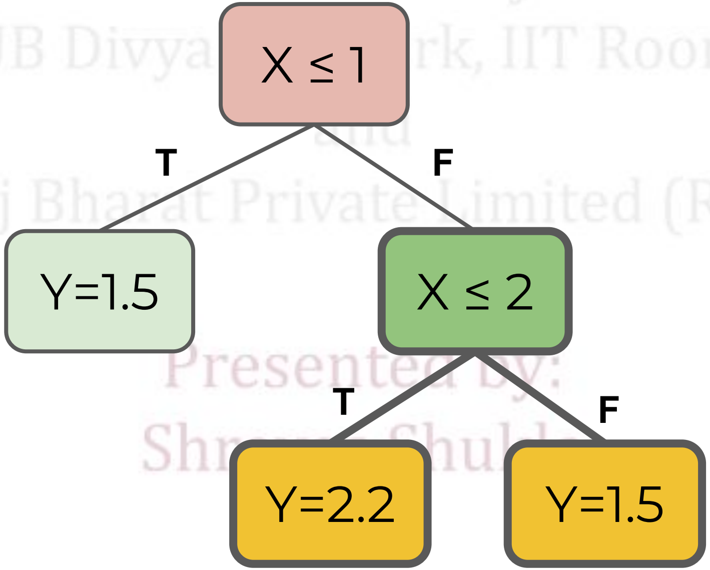
Parent and Children Nodes:



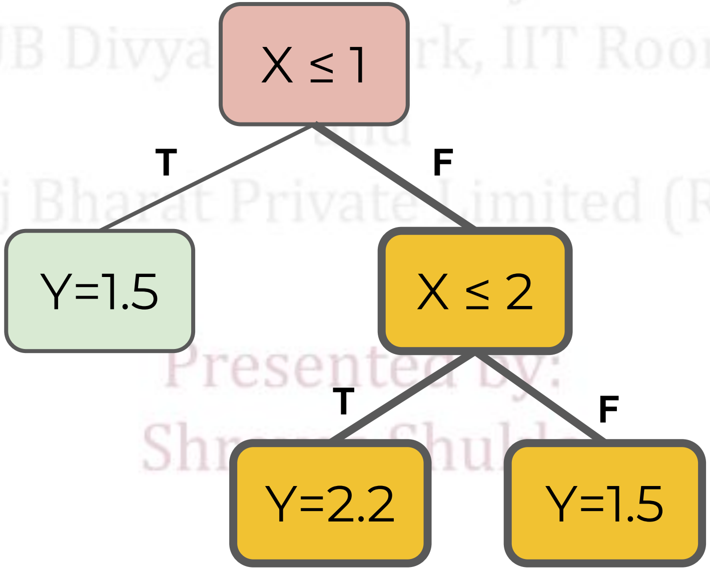
Parent and Children Nodes:



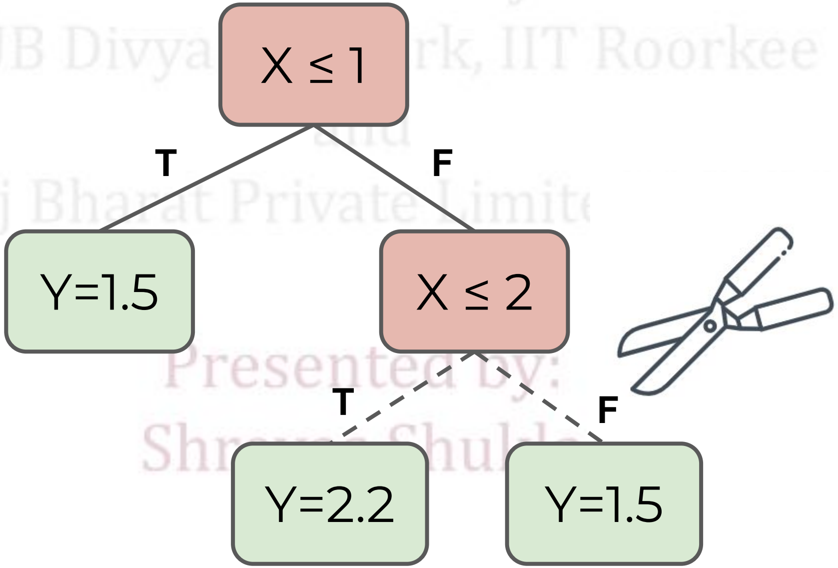
Parent and Children Nodes:



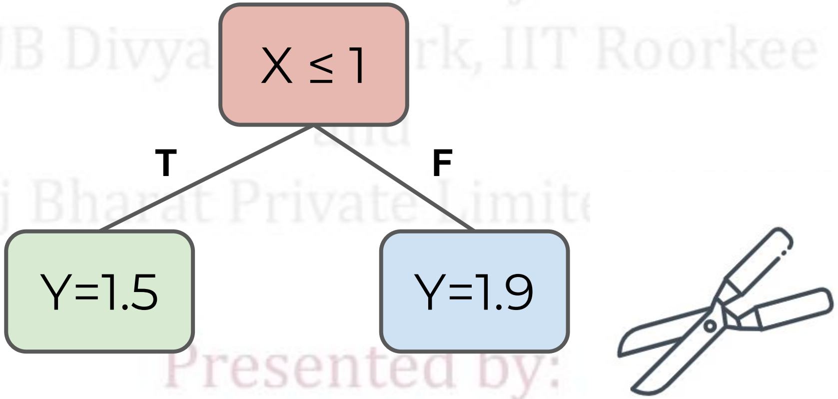
Tree Branches (Sub Trees):



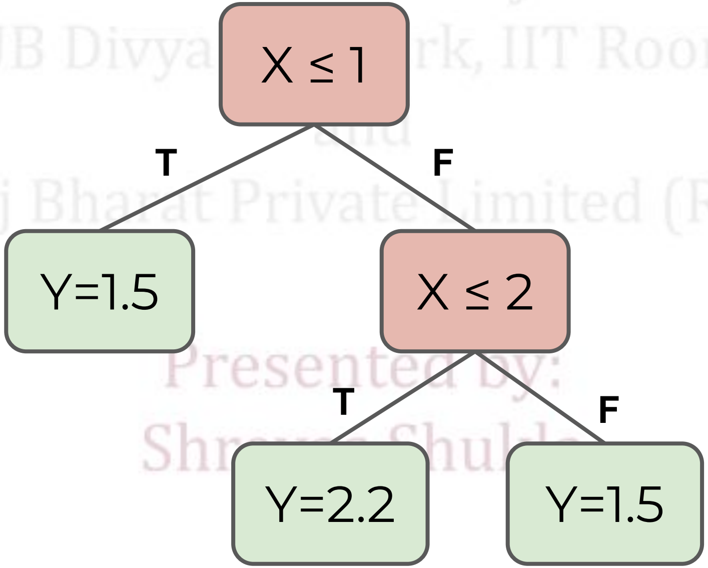
Pruning:



Pruning:



Let’s begin constructing a tree!



Decision Trees

Gini Impurity

Gini Impurity

A mathematical measurement of how “pure” the information in a data set is.

We can think of this as a measurement of class uniformity.

Gini Impurity for Classification:

- For a set of classes **C** for a given dataset **Q**:

$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

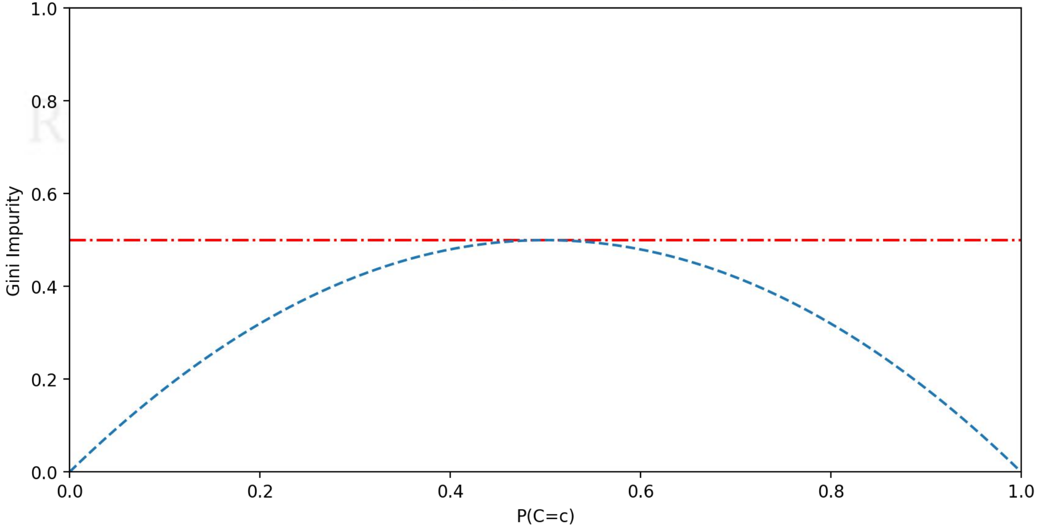
Gini Impurity for Classification:

- For a set of classes **C** for a given dataset **Q**, $\mathbf{p_c}$ is probability of class **c**.

$$p_c = \frac{1}{N_Q} \sum_{x \in Q} \mathbb{1}(y_{class} = c) \qquad G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

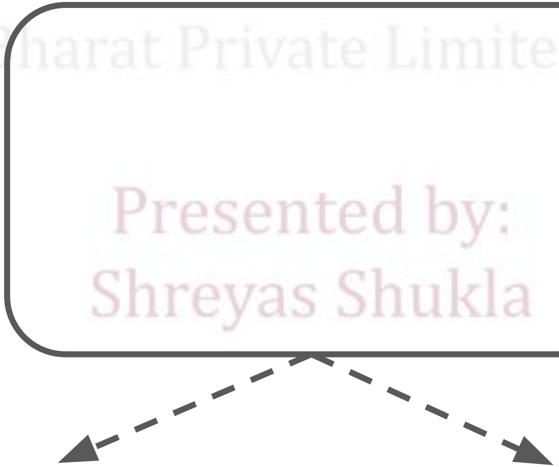
Gini Impurity for Classification:

$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$



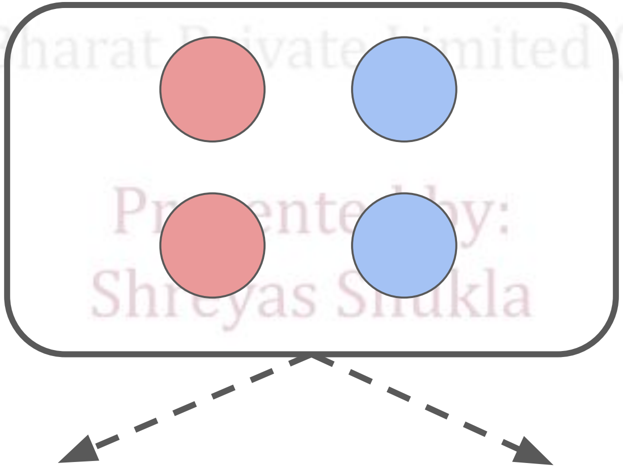
Gini Impurity for Classification:

$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$



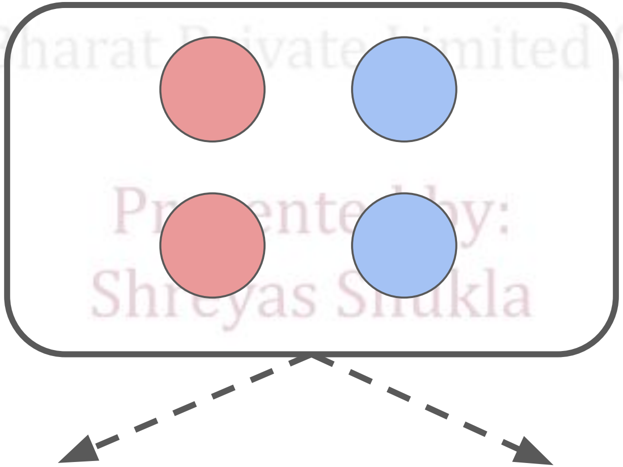
Gini Impurity for Classification:

$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

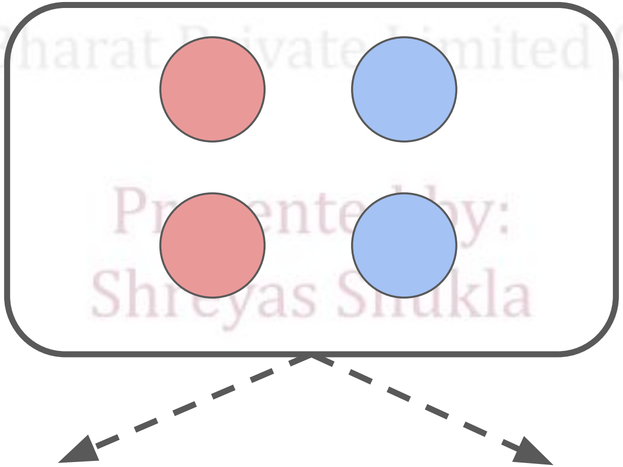


$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

Class Red
(2/4)(1 - 2/4) = 0.25



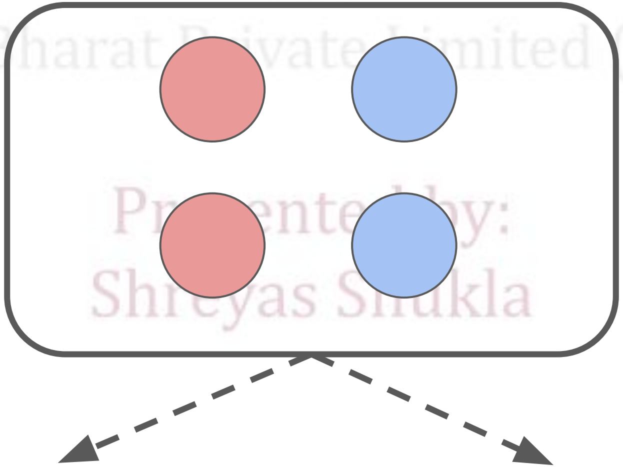
$$G(Q) = \sum_{c \in C} p_c (1 - p_c)$$



Class Red
(2/4)(1 - 2/4) = 0.25

Class Blue
(2/4)(1 - 2/4) = 0.25

$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$



Class Red
(2/4)(1 - 2/4) = 0.25

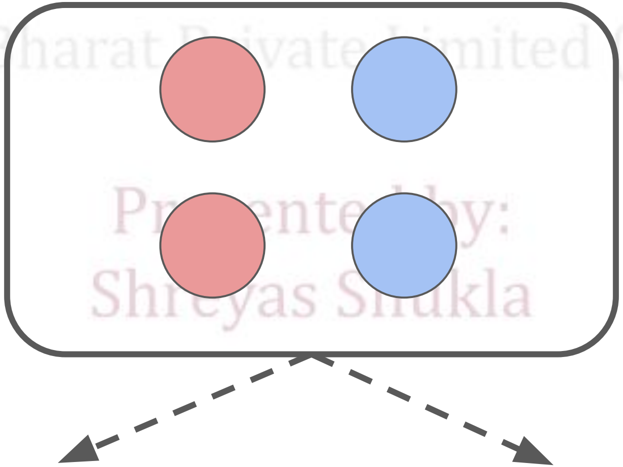
+

Class Blue
(2/4)(1 - 2/4) = 0.25

Gini Impurity
0.25 + 0.25 = 0.5

"Maximum" Impurity Possible

$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$



Class Red
(2/4)(1 - 2/4) = 0.25

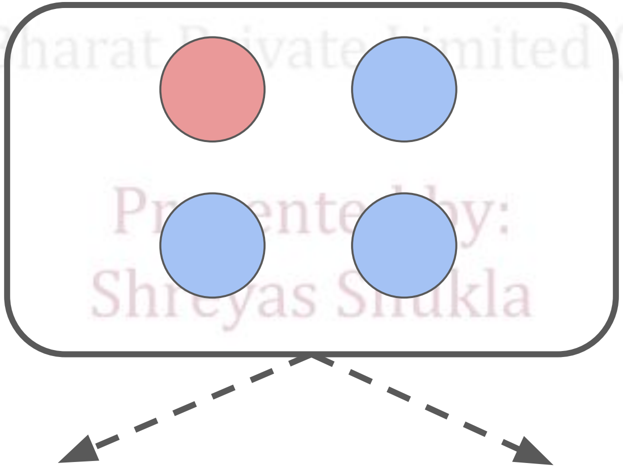
+

Class Blue
(2/4)(1 - 2/4) = 0.25

Gini Impurity
0.25 + 0.25 = 0.5

Data is more “pure” (less impurity)

$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$



Class Red
(1/4)(1 - 1/4) = 0.1875

+

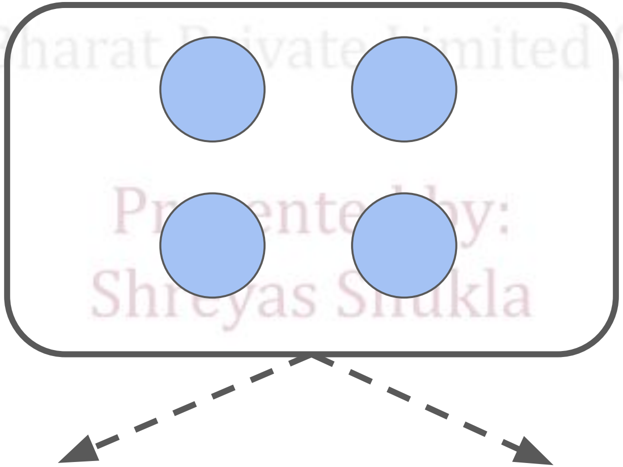
Class Blue (3/4)(1 - 3/4) =0.1875

⇩

Gini Impurity
0.1875+0.1875 = 0.375

Data is completely “pure” (no impurity)

$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$



Class Red
(0/4)(1 - 0/4) = 0

+

Class Blue
(4/4)(1 - 4/4) = 0

Gini Impurity
0 + 0 = 0

If the goal of a decision tree is to separate out classes, we can use gini impurity to decide on data split values.

We want to minimize the gini impurity at leaf nodes.

Minimized impurity at leaf nodes means we are separating classes effectively

Decision Trees

Gini Impurity in Trees

For constructing a tree, we have to decide what feature will be root node.

Use gini impurity to compare the information contained within features for the training data.

Gini Impurity for Classification:

- For a set of classes **C** for a given dataset **Q**, $\mathbf{p_c}$ is probability of class **c**.

$$p_c = \boxed{\frac{1}{N_Q} \sum_{x \in Q} \mathbb{1}(y_{class} = c)} \qquad G(Q) = \sum_{c \in C} \boxed{p_c}(1 - \boxed{p_c})$$

Create a decision tree to predict spam.

X - URL Link	Y-Spam
Yes	Yes
Yes	Yes
No	No
No	No
No	Yes
No	No
Yes	No

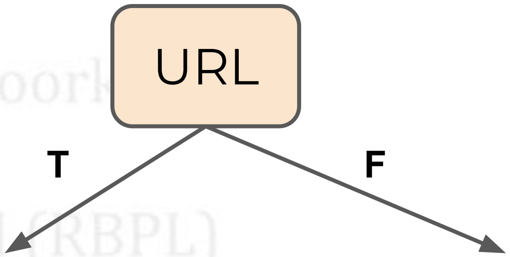
Only one X feature to use for a node.

X - URL Link	Y-Spam
Yes	Yes
Yes	Yes
No	No
No	No
No	Yes
No	No
Yes	No

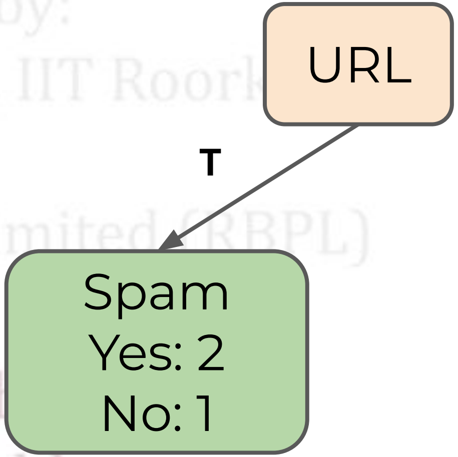
URL

Predict if email is spam if it contains a URL:

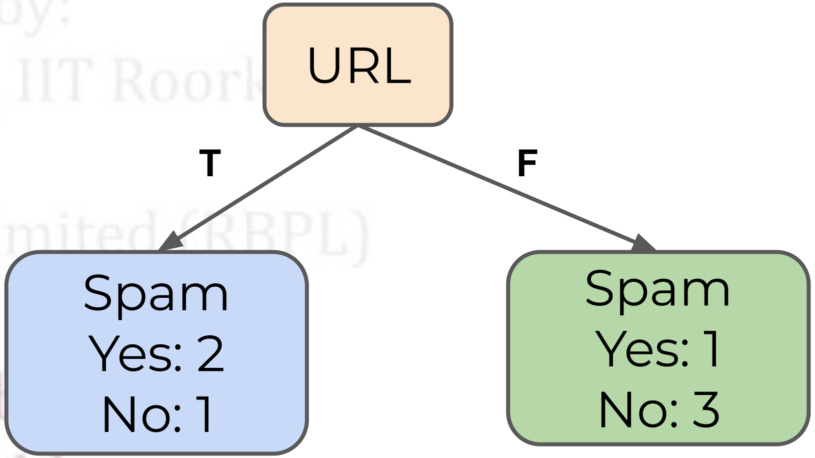
X - URL Link	Y-Spam
Yes	Yes
Yes	Yes
No	No
No	No
No	Yes
No	No
Yes	No



X - URL Link	Y-Spam
Yes	Yes
Yes	Yes
No	No
No	No
No	Yes
No	No
Yes	No

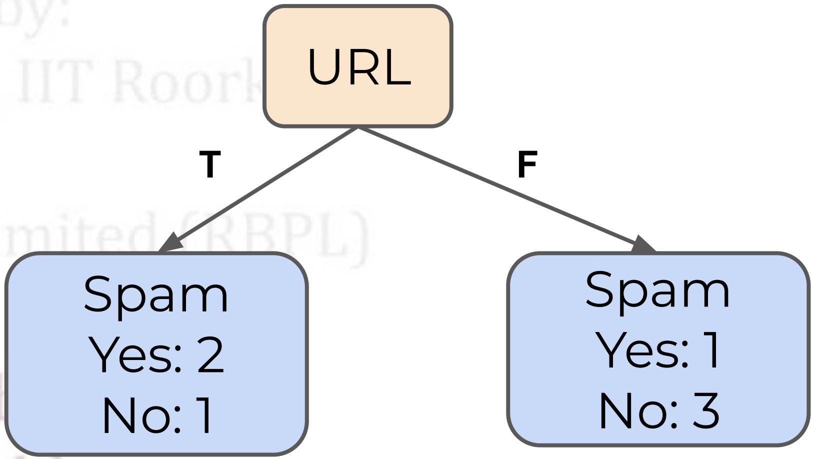


X - URL Link	Y-Spam
Yes	Yes
Yes	Yes
No	No
No	No
No	Yes
No	No
Yes	No



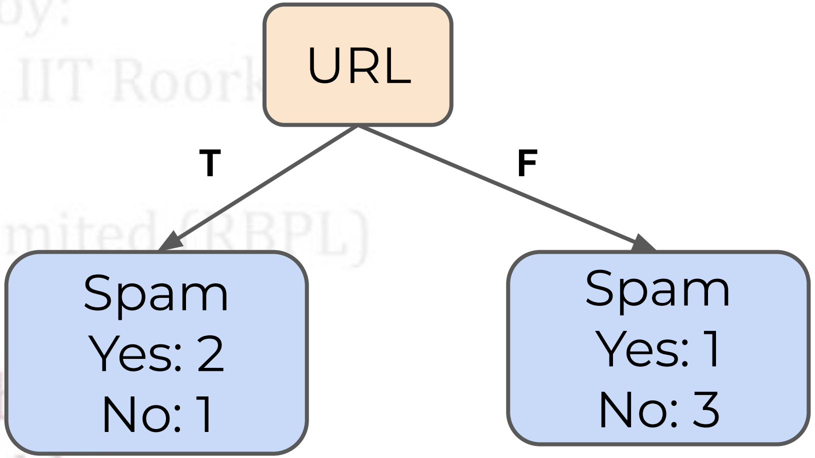
Predict if email is spam if it contains a URL:

X - URL Link	Y-Spam
Yes	Yes
Yes	Yes
No	No
No	No
No	Yes
No	No
Yes	No



Recall the gini impurity formula:

X - URL Link	Y-Spam
Yes	Yes
Yes	Yes
No	No
No	No
No	Yes
No	No
Yes	No

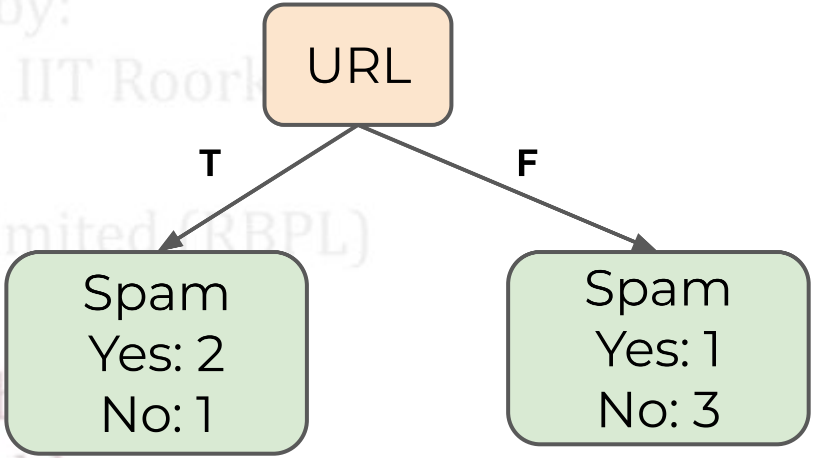


$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

Treat Yes Spam and No Spam as **c** classes:

Left Leaf Node:

$(2/3)(1-2/3) + (1/3)(1-1/3)$



$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

Treat Yes Spam and No Spam as **c** classes:

Left Leaf Node:

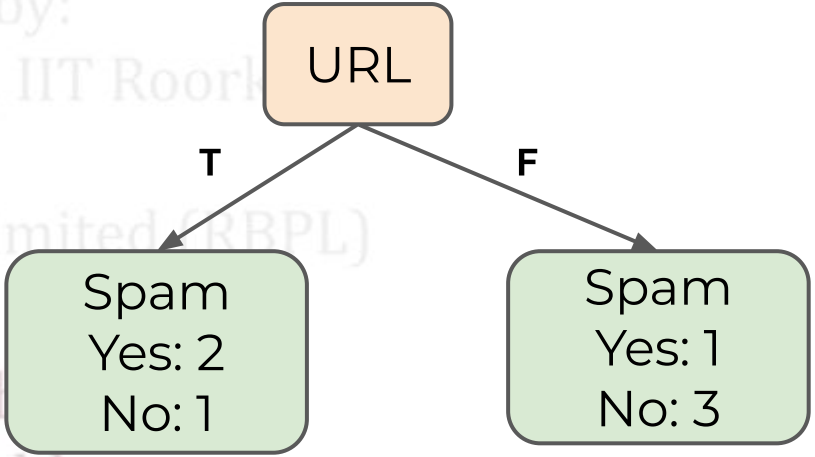
$(\frac{2}{3})(1-\frac{2}{3}) + (\frac{1}{3})(1-\frac{1}{3})$

Left Leaf Gini=0.44

Right Leaf Node:

$(\frac{1}{4})(1-\frac{1}{4}) + (\frac{3}{4})(1-\frac{3}{4})$

Right Leaf Gini=0.375

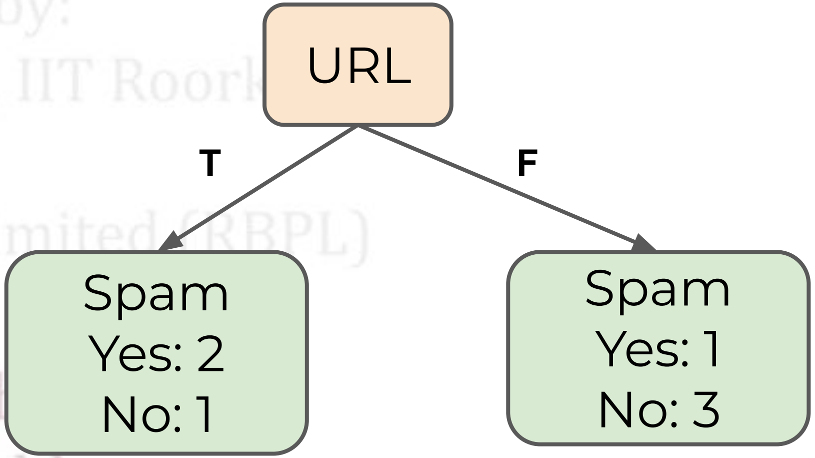


$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

Calculate gini impurity of URL feature.

Weighted Average of both:

- Left Leaf Gini=0.44
- Right Leaf Gini=0.375



$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

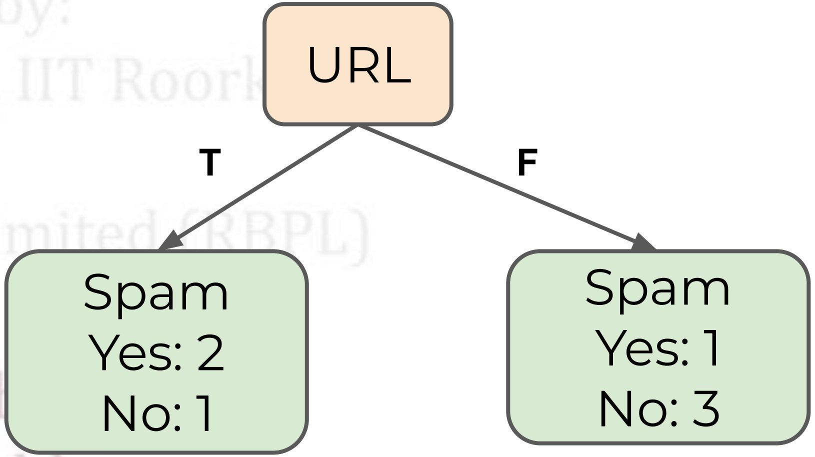
Total Emails: (2+1) + (1+3) = 7

Left Leaf Gini=0.44

Right Leaf Gini=0.375

Left Emails: 3

Right Emails: 4



$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

Total Emails: (2+1) + (1+3) = 7

Left Leaf Gini=0.44

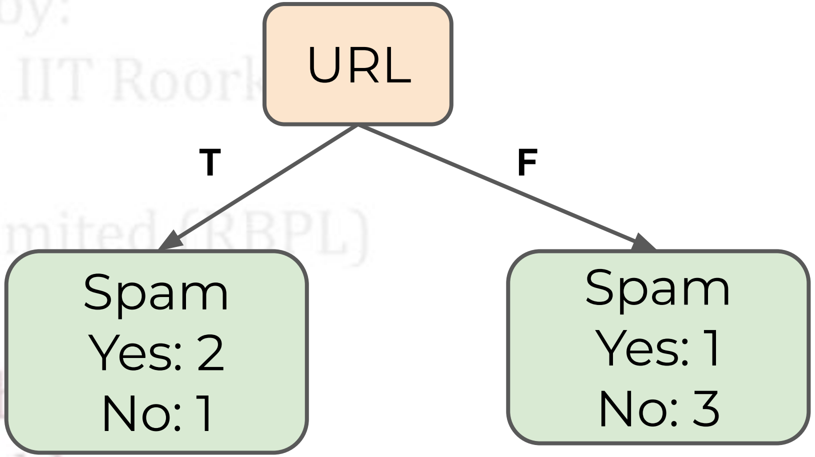
Right Leaf Gini=0.375

Left Emails: 3

Right Emails: 4

(3/7)*0.44 + (4/7)*0.375

Gini Impurity: 0.403



$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

More issues to consider:

- Multiple Features
- Continuous Features
- Multi-categorical Features

We use the gini impurity to each of these issues to solve for best root nodes and best split parameters for leaves.

Decision Trees

Gini Impurity Part Two

Let’s explore:

- Continuous numeric features
- Multi-categorical features (N>2)
- Choosing a root node feature

Imagine a continuous feature.
Calculate the feature gini impurity:

X - Words in Email	Y-Spam
10	Yes
40	No
20	Yes
50	No
30	No

Sort data:

X - Words in Email	Y-Spam
10	Yes
40	No
20	Yes
50	No
30	No

Calculate potential split values for node

X - Words in Email	Y-Spam
10	Yes
20	Yes
30	No
40	No
50	No

Words ≤ N

Use averages between rows as values:

X - Words in Email	Y-Spam
10	Yes
20	Yes
30	No
40	No
50	No

15

25

35

45

Words ≤ N

Perform all the potential split:

X - Words in Email	Y-Spam
10	Yes
20	Yes
30	No
40	No
50	No

15

25

35

45

Words ≤ 15

Calculate gini impurity for each split:

X - Words in Email	Y-Spam
10	Yes
20	Yes
30	No
40	No
50	No

15

Words ≤ 15

Calculate gini impurity for each split:

Words ≤ 15

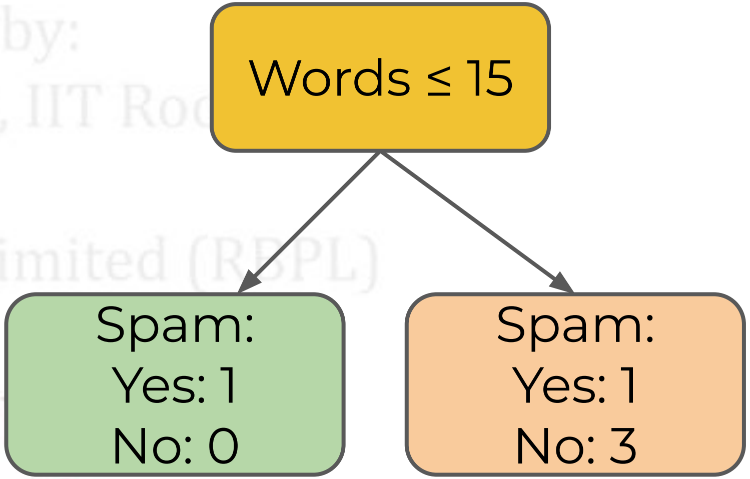
X - Words in Email	Y-Spam
10	Yes
20	Yes
30	No
40	No
50	No

15

Calculate gini impurity for each split:

X - Words in Email	Y-Spam
10	Yes
20	Yes
30	No
40	No
50	No

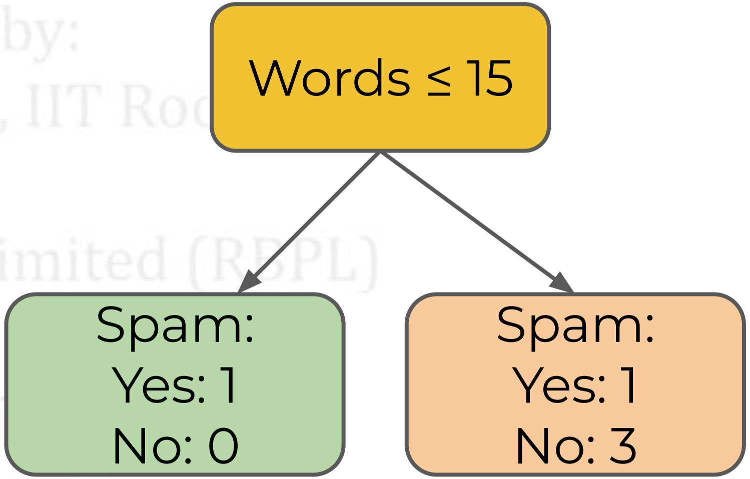
15



$$G(Q) = \sum_{c \in C} p_c(1 - p_c)$$

Calculate gini impurity for each split:

X - Words in Email	Y-Spam
10	Yes
20	Yes
30	No
40	No
50	No



$$G(Q) = (\tfrac{1}{5})(0+0) + (\tfrac{4}{5})((\tfrac{1}{4})(1-\tfrac{1}{4})+(\tfrac{3}{4})(1-\tfrac{3}{4})$$
$$= 0.3$$

Do it for all possible splits:

X - Words in Email	Y-Spam
10	Yes
20	Yes
30	No
40	No
50	No

15 → Gini=0.3

25 → Gini=0

35 → Gini=0.26

45 → Gini=0.4

Choose lowest impurity split value

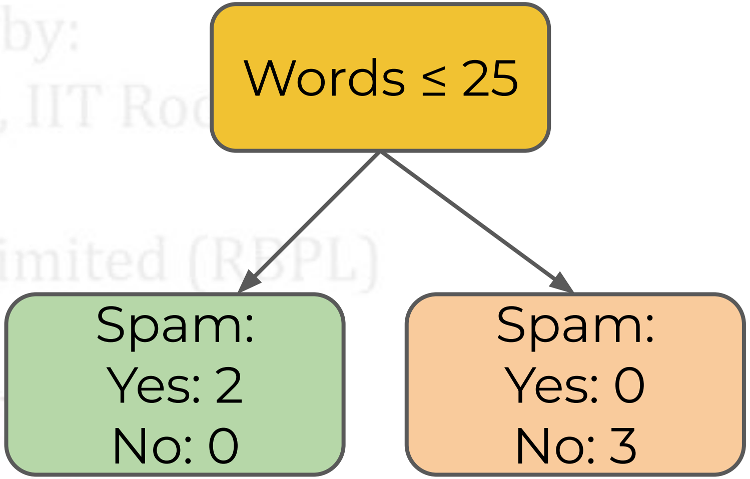
X - Words in Email	Y-Spam
10	Yes
20	Yes
30	No
40	No
50	No

25 → Gini=0

Choose this as split value for node

X - Words in Email	Y-Spam
10	Yes
20	Yes
30	No
40	No
50	No

25



$$G(Q) = 0$$

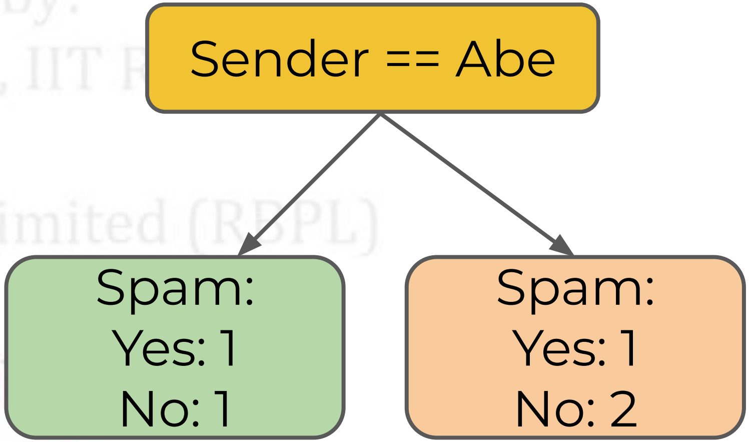
Multicategorical feature

Calculate gini impurity for all combinations:

X - Sender	Y-Spam
Abe	Yes
Bob	Yes
Claire	No
Abe	No
Bob	No

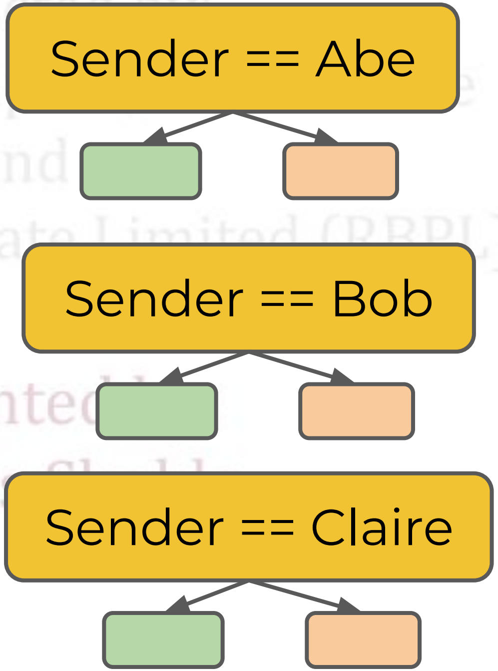
Calculate gini impurity for all combinations:

X - Sender	Y-Spam
Abe	Yes
Bob	Yes
Claire	No
Abe	No
Bob	No



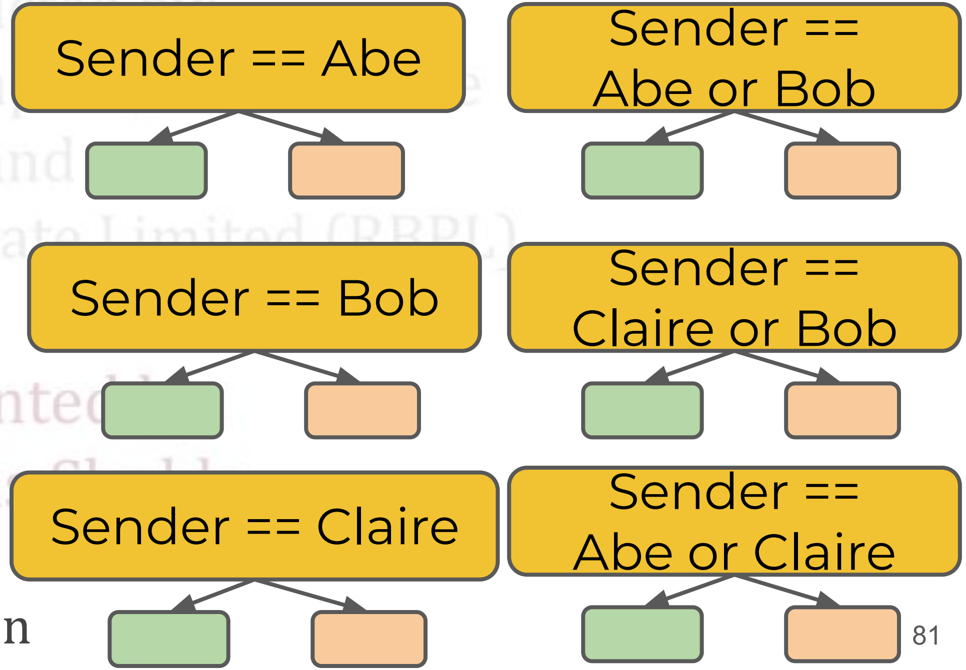
Calculate gini impurity for all combinations

X - Sender	Y-Spam
Abe	Yes
Bob	Yes
Claire	No
Abe	No
Bob	No



Calculate gini impurity for all combinations

X - Sender	Y-Spam
Abe	Yes
Bob	Yes
Claire	No
Abe	No
Bob	No



Choose lowest impurity split combination

Now we can split any type of feature.

How does the decision tree decide on the root node of a multi-feature dataset?

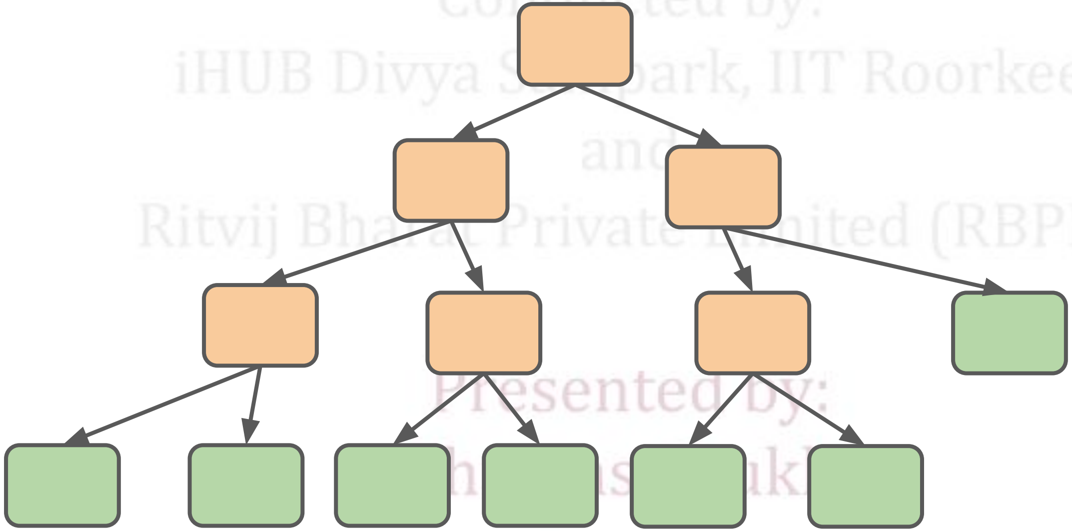
Calculate the gini impurity values of each feature and choose the lowest impurity value to split on first.

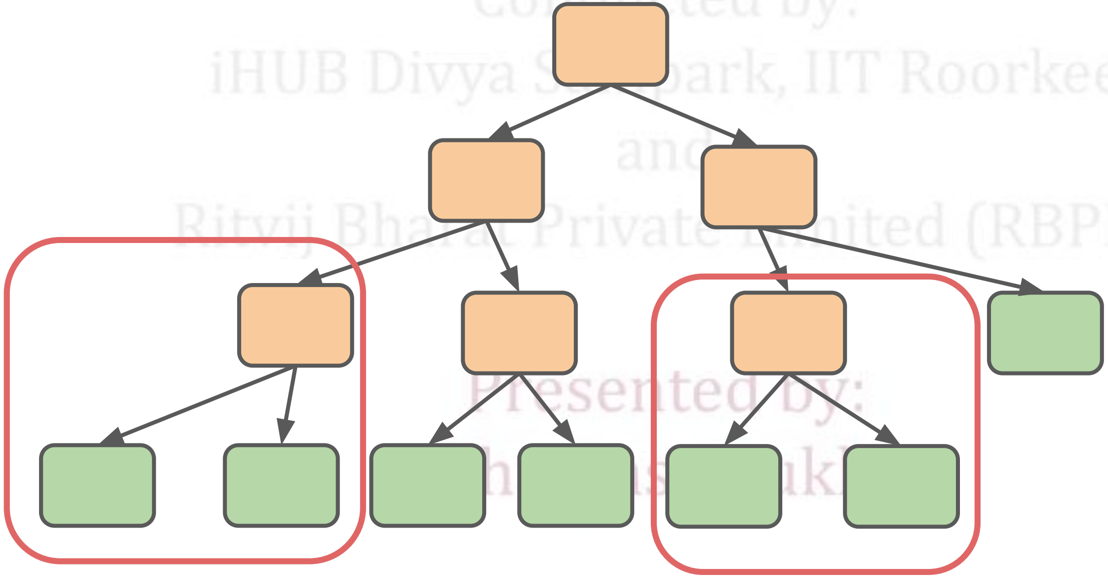
By choosing the feature with the lowest resulting gini impurity in its leaf nodes, we are choosing the feature that best splits the data into “pure” classes.

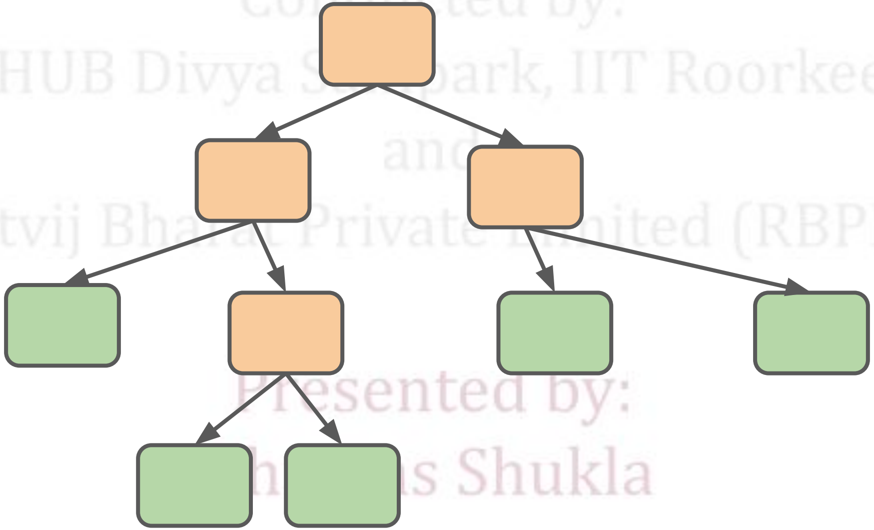
By using gini impurity as a measurement of the effectiveness of a node split, we can perform automatic feature selection by mandating an impurity threshold for an additional feature based split to occur.

A large overfitted tree.

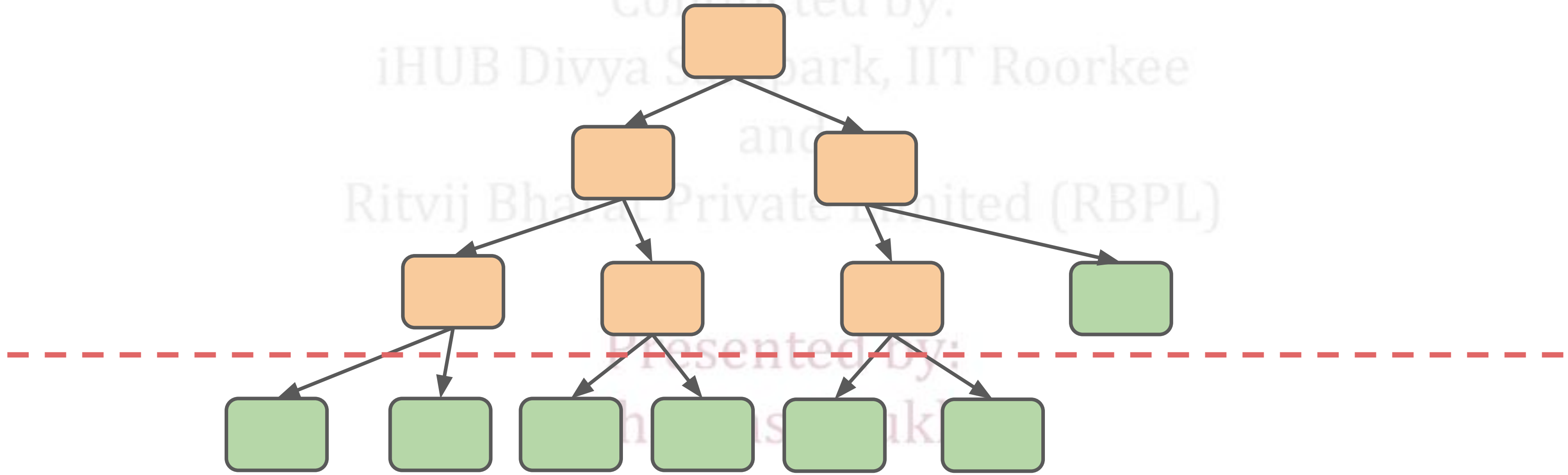
Add minimum gini impurity decrease

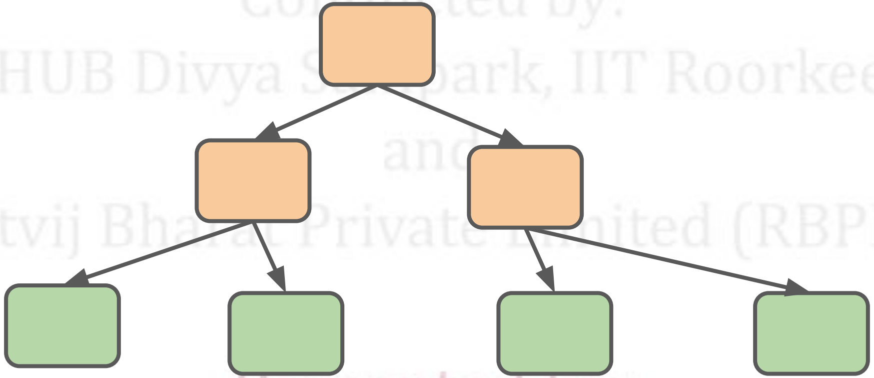






We can also mandate a max depth





An Introduction to Machine Learning with Python Programming
11 Sep 2023 - 20 Oct 2023

Conducted by:
iHUB Divya Sampark, IIT Roorkee
and
Ritvij Bharat Private Limited (RBPL)

Let’s code !!

Presented by:
Shreyas Shukla