Decision Trees , Random Forests and Extra Trees

EDVANCER
EDUVENTURES

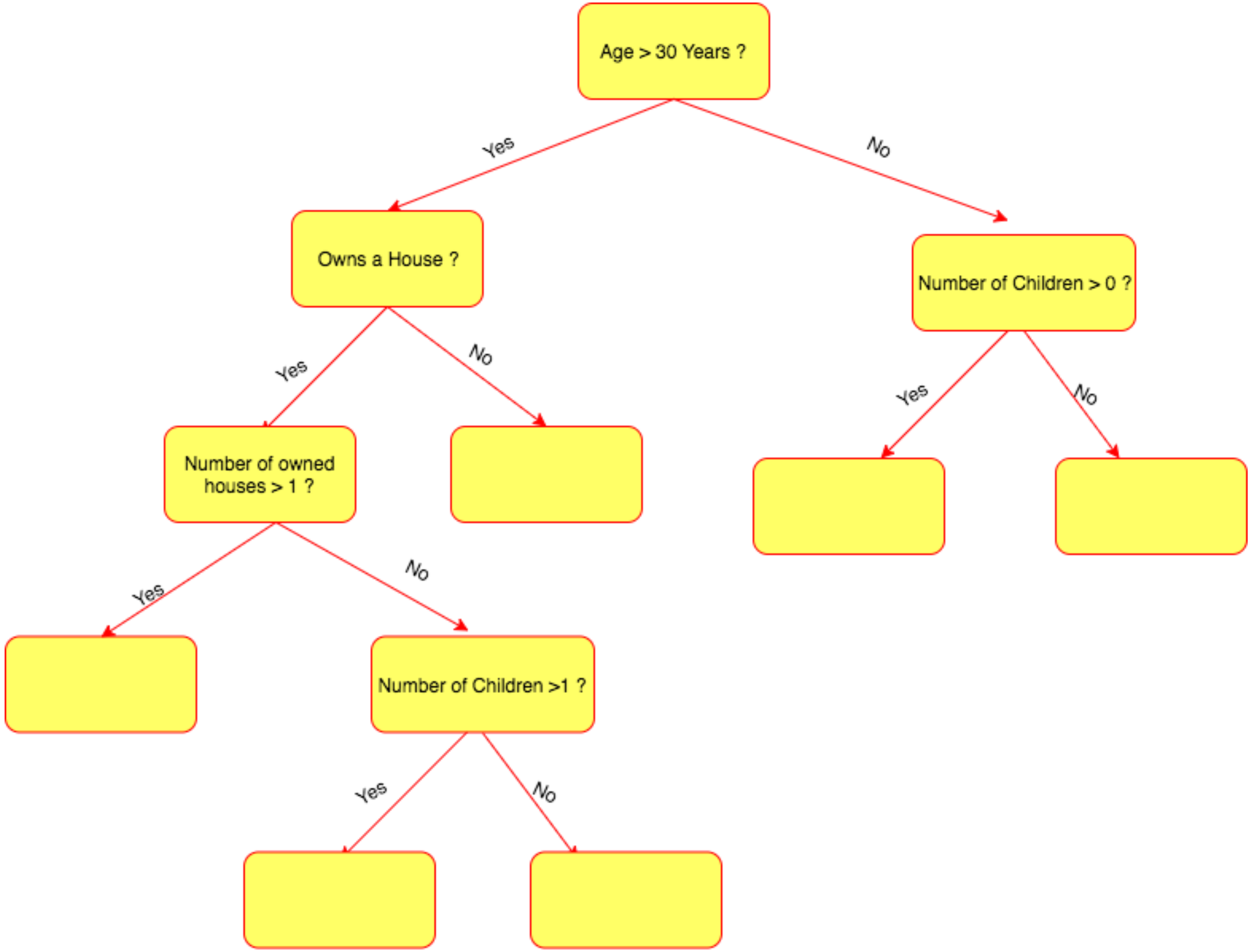
Agenda

Discussion Flow

- Basic Structure of a Decision Tree
- Building a classification/Regression Tree
- Interpretation in absence of parameters/coefficients
- Implementation in Python
- Overfitting issue with Decision Tree
- Random Forests
- Extra Trees
- Implementation in Python

Decision Trees

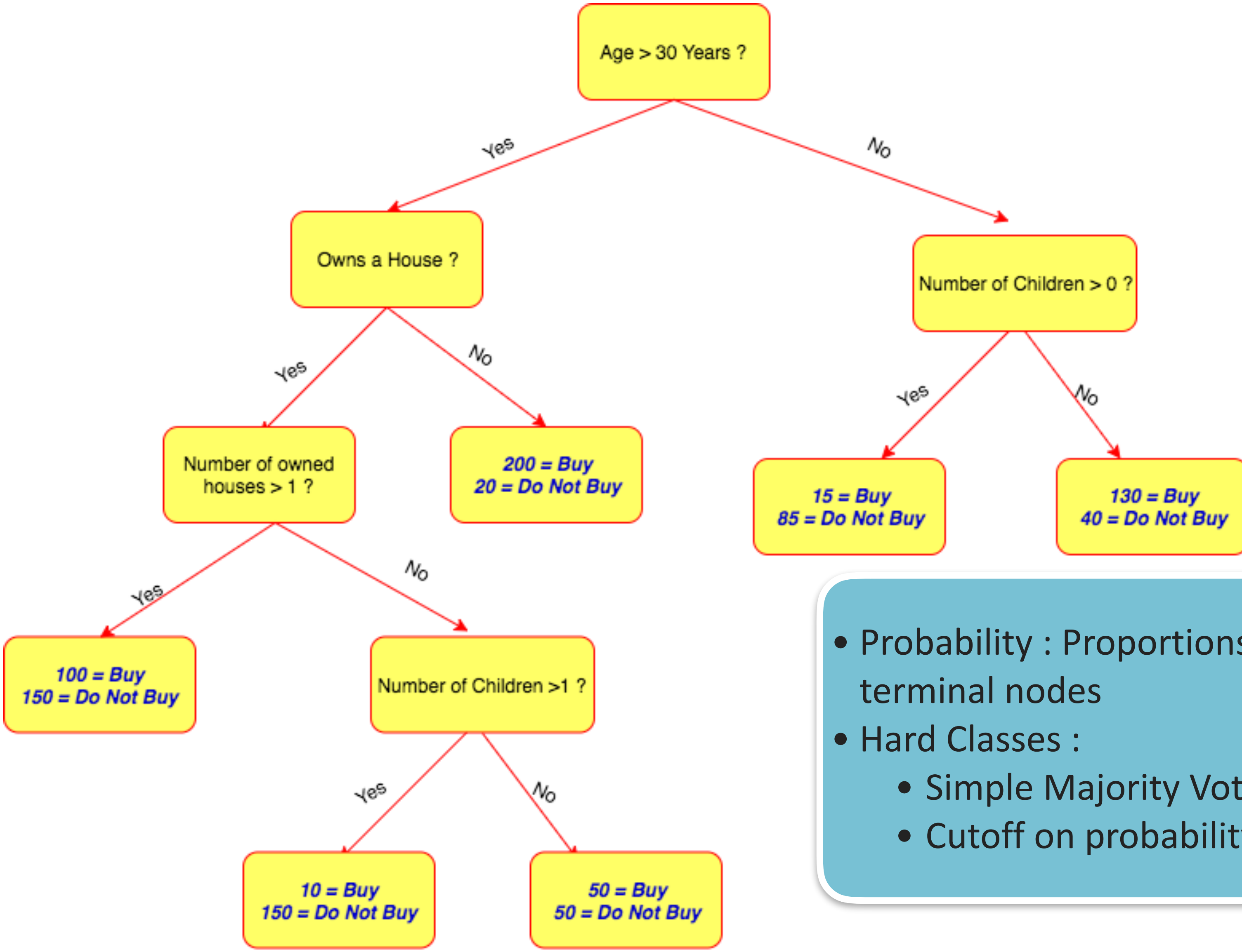
What does it look like?



Questions ? !!

- How do we take decisions ? (Classification/ Regression)
- Where do these rules come from ?
- How do we pick rules for splitting at each node?
- When do we stop splitting nodes ?

How do we take decisions ? (Classification)



- Probability : Proportions in the terminal nodes
- Hard Classes :
    - Simple Majority Vote
    - Cutoff on probability score

How do we take decisions ? (Regression)

- Much simpler than classification
- Simple Average of target at the terminal node becomes your predicted value for that terminal node

Where do these rules come from? (Numeric Vars)

Var
10
11
15
5
: :
: :
24

- Range of the numeric variable is broken into intervals
- Based on the values which are there in the data, breaks dont need to be equidistant

Var > 10 ?

Var > 15 ?

Var > 17.5 ?

Var > 20 ?

Where do these rules come from? (Categorical vars)

Var	*Var_Delhi*	*Var_NY*	*Var_Bejing*
Delhi	1	0	0
NY	0	1	0
Delhi	1	0	0
Bejing	0	0	1
: :	: :	: :	: :
: :			
NY	0	1	0

Var_Delhi > 0.5 ?

Var_NY > 0.5 ?

Var_Bejing > 0.5 ?

How to Select Rules for split? (Classification)

- Among all the rules available , the one which results in a split with most homogeneous system is selected

Measures of homogeneity

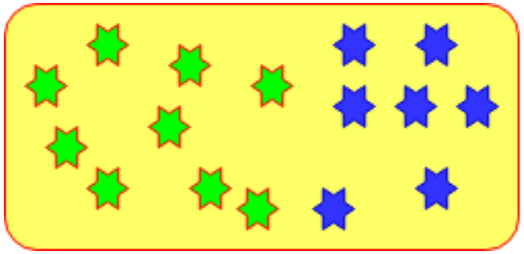
$$gini\ index = 1 - \sum_{i=1}^{k} p_i^2$$

$$entropy = -\sum_{i=1}^{k} p_i * log(p_i)$$

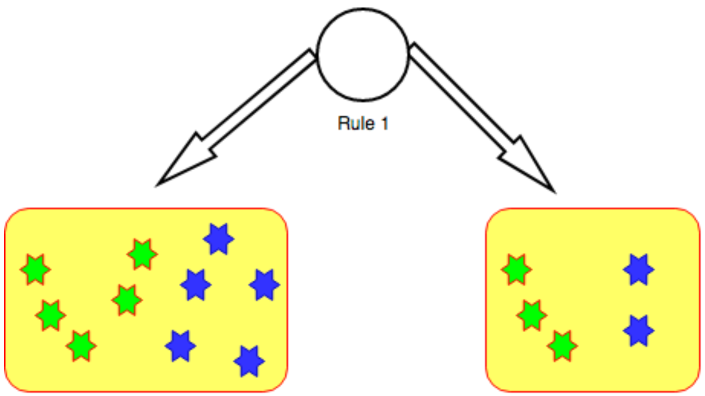
$$deviance = -\sum_{i=1}^{k} n_i * log(p_i)$$

Note : There is no theoretical favourite among them , its more of matter of convenience in implementation

Example: Using entropy for rule selection



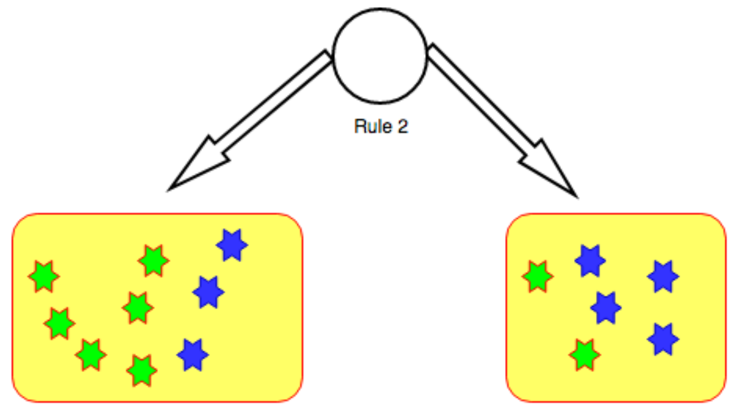
$$gini\_parent = 1 - \left(\left(\frac{8}{15}\right)^2 + \left(\frac{7}{15}\right)^2\right) = .498$$



$$gini\_1 = 1 - \left(\left(\frac{5}{10}\right)^2 + \left(\frac{5}{10}\right)^2\right) = .50$$

$$gini\_2 = 1 - \left(\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right) = .48$$

$$gini\_new = \left(\frac{10}{15}\right) * 0.50 + \left(\frac{5}{15}\right) * 0.48 = .493$$



$$gini\_1 = 1 - \left(\left(\frac{6}{9}\right)^2 + \left(\frac{3}{9}\right)^2\right) = .444$$

$$gini\_2 = 1 - \left(\left(\frac{2}{6}\right)^2 + \left(\frac{4}{6}\right)^2\right) = .444$$

$$gini\_new = \left(\frac{9}{15}\right) * 0.444 + \left(\frac{6}{15}\right) * 0.444 = .444$$

Rule 2 gets selected for higher decrease in gini

How to Select Rules for split and Make Prediction? (Regression)

- Average of the node is the prediction for the node
- Among all the rules available , the one which results in a split with least Sum of Square of Errors is selected

Measures of homogeneity

$$SSE = \sum_{i=1}^{n_k} (y_i - \overline{y_k})^2$$

Example: Using SSE for rule selection

Response	Error
5	-2.7
6	-1.7
12	4.3
11	3.3
4	-3.7
8	0.3
13	5.3
5	-2.7
6	-1.7
7	-0.7

Prediction
7.7

SSE Parent
92.1

Example Contd Rule 1

Response	Error
5	-2.6
6	-1.6
12	4.4
11	3.4
4	-3.6

Prediction
7.6

SSE 1
53.2

SSE new
92

Response	Error
8	0.2
13	5.2
5	-2.8
6	-1.8
7	-0.8

Prediction
7.8

SSE 2
38.8

Example Contd Rule 2

Response	Error
5	-0.2
6	0.8
4	-1.2
5	-0.2
6	0.8

Prediction
5.2

SSE 1
2.8

SSE new
29.6

Response	Error
12	1.8
11	0.8
8	-2.2
13	2.8
7	-3.2

Prediction
10.2

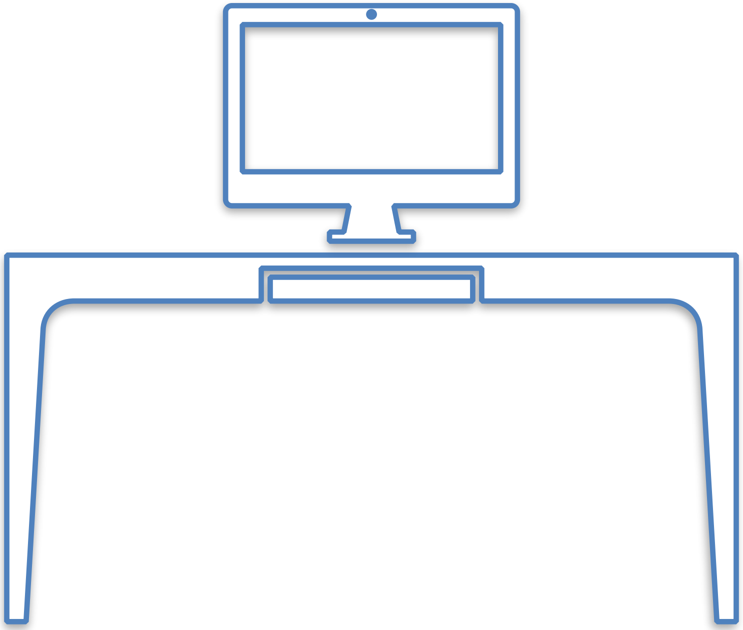
SSE 2
26.8

Rule 2 gets selected because of higher decrease in SSE

When do we stop splitting Nodes?

- When does a node become a terminal node?
- When node is completely homogeneous
- When number of observation in the nodes are lower than the specified limit for split
- When all the rules result in a split such that one (or both) child node will have less observation than specified limit for child node
- When number of specified terminal node is reached

Lets see it in action in Python

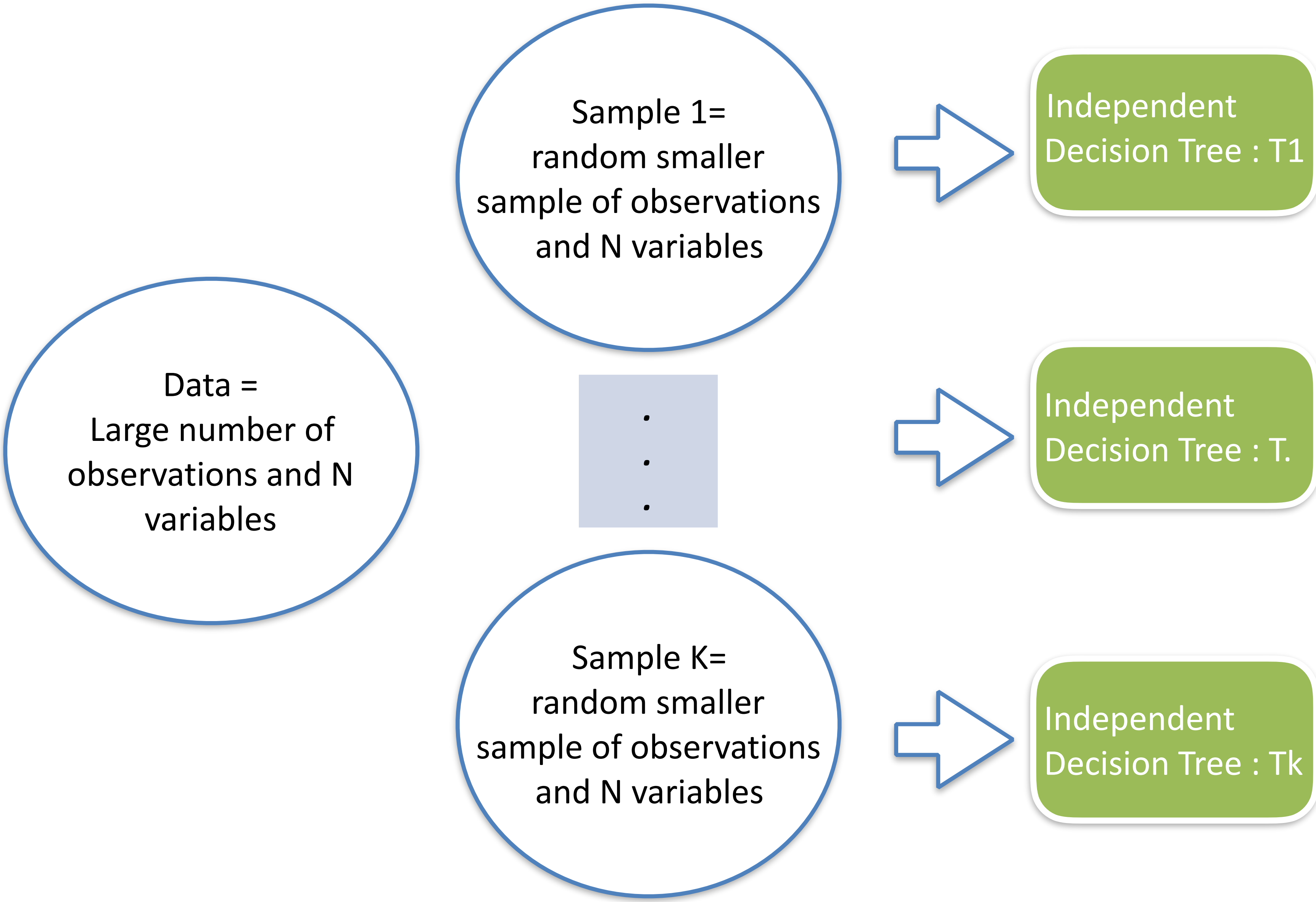


Issues with a single decision tree

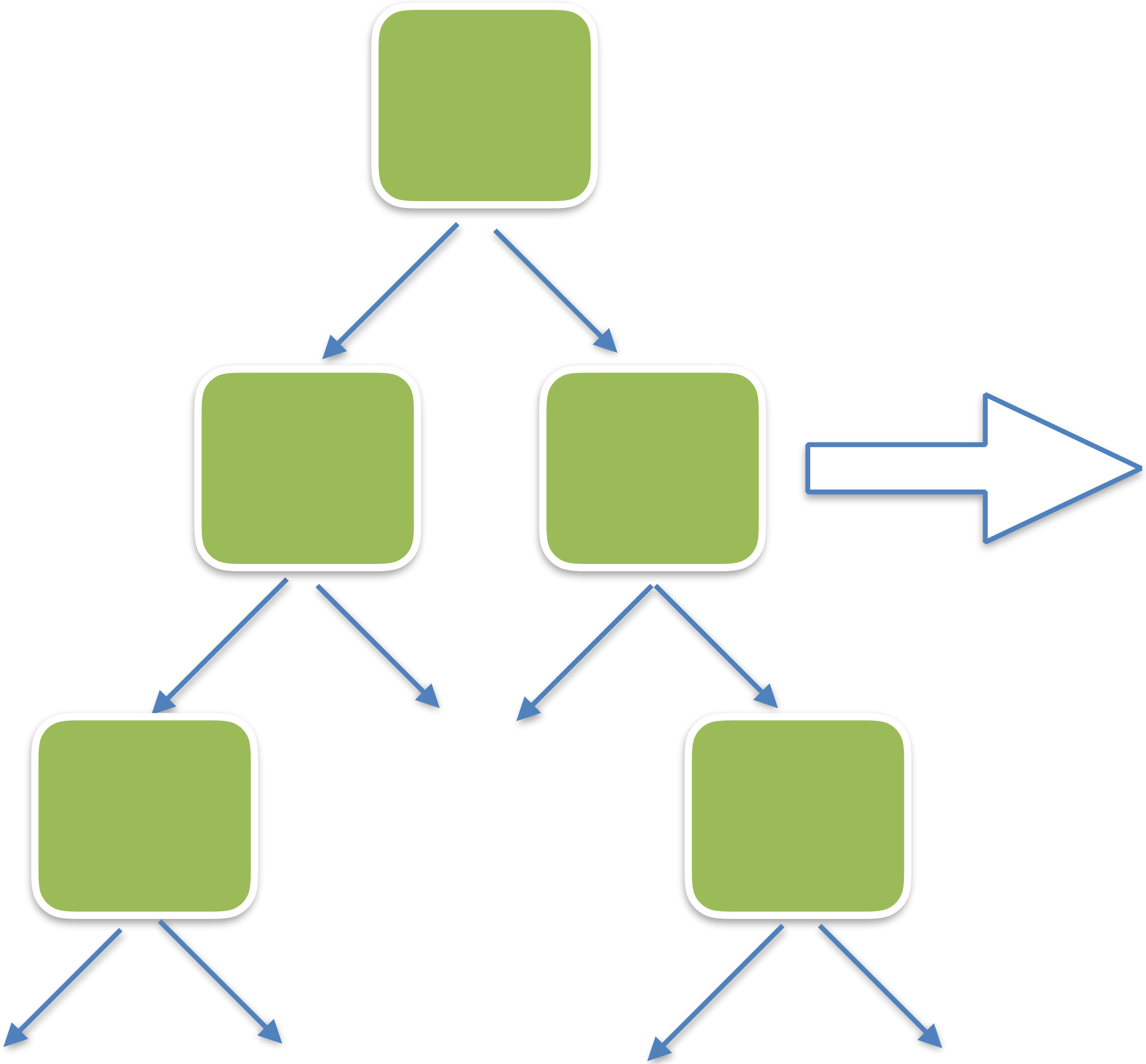
- Susceptible to noisy observations
- Susceptible to noisy variables
- In general overfits the training data

Random Forests

Introduction of random ness in the process



Contd… : for each tree



- At each node(Split) instead of all N variables , only a random subset of variables is considered for splitting
- This random Subset is different for each node across each tree

Extra Trees

Extra Trees

- Short for extremely randomised trees
- Extension of random forest
- In addition to what we do in random forest, extra trees randomly subset rules as well at each node before selecting the best rule for split
- Work well when there are less noisy features but with noisy ranges/categories

Lets see it in action in Python