









DICE
ANALYTICS

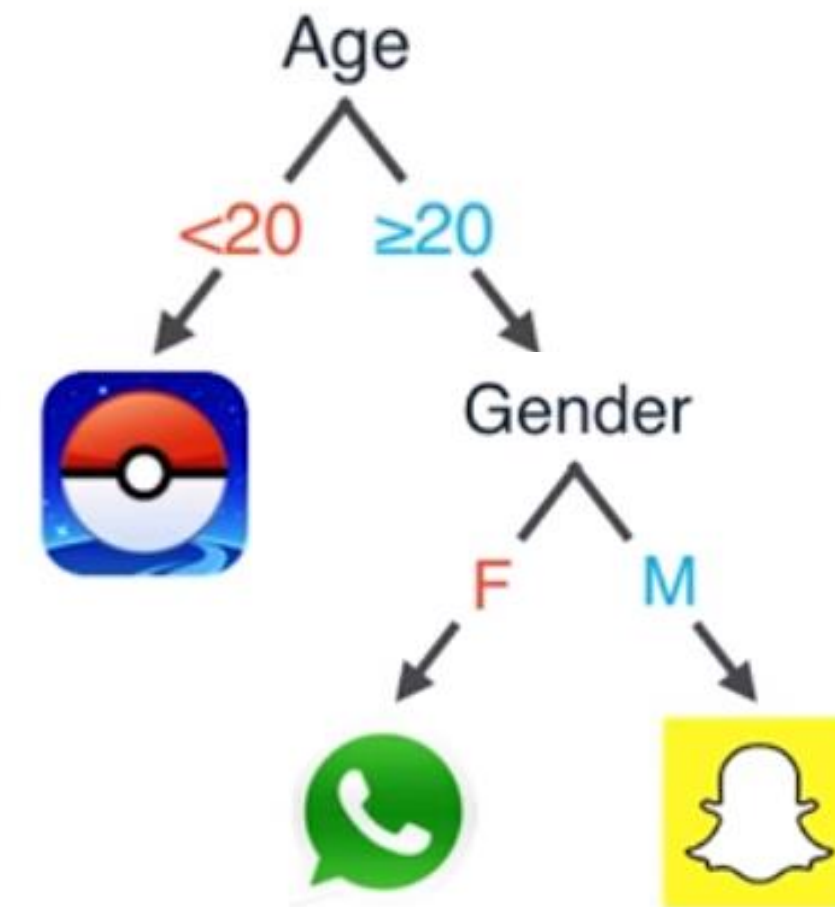
DATA SCIENCE & MACHINE LEARNING COURSE

<https://www.facebook.com/diceanalytics/>
<https://pk.linkedin.com/company/diceanalytics>

Let's Play a Game :D

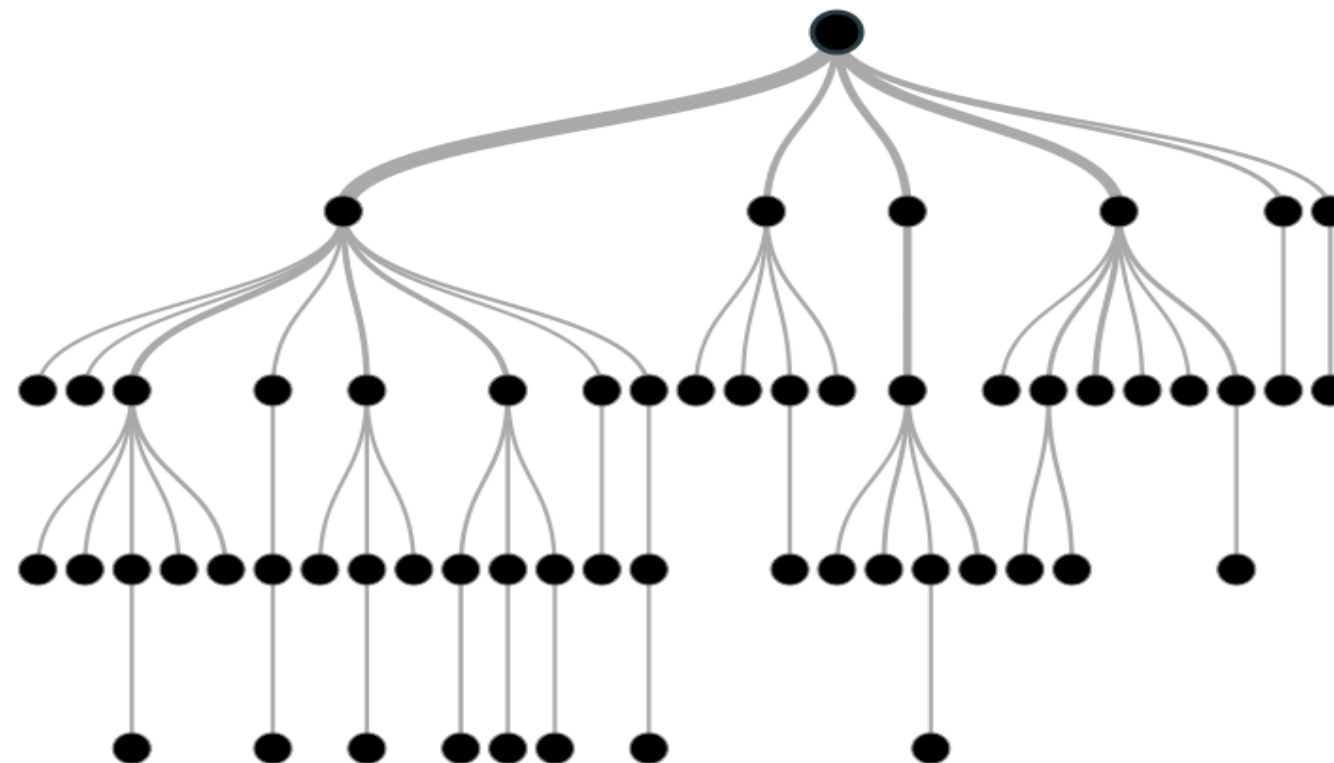
<https://en.akinator.com/>

Gender	Age	App
F	15	
F	25	
M	32	
F	40	
M	12	
M	14	



Decision Trees

- Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems.
- It works for both categorical and continuous input and output variables.
- In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.



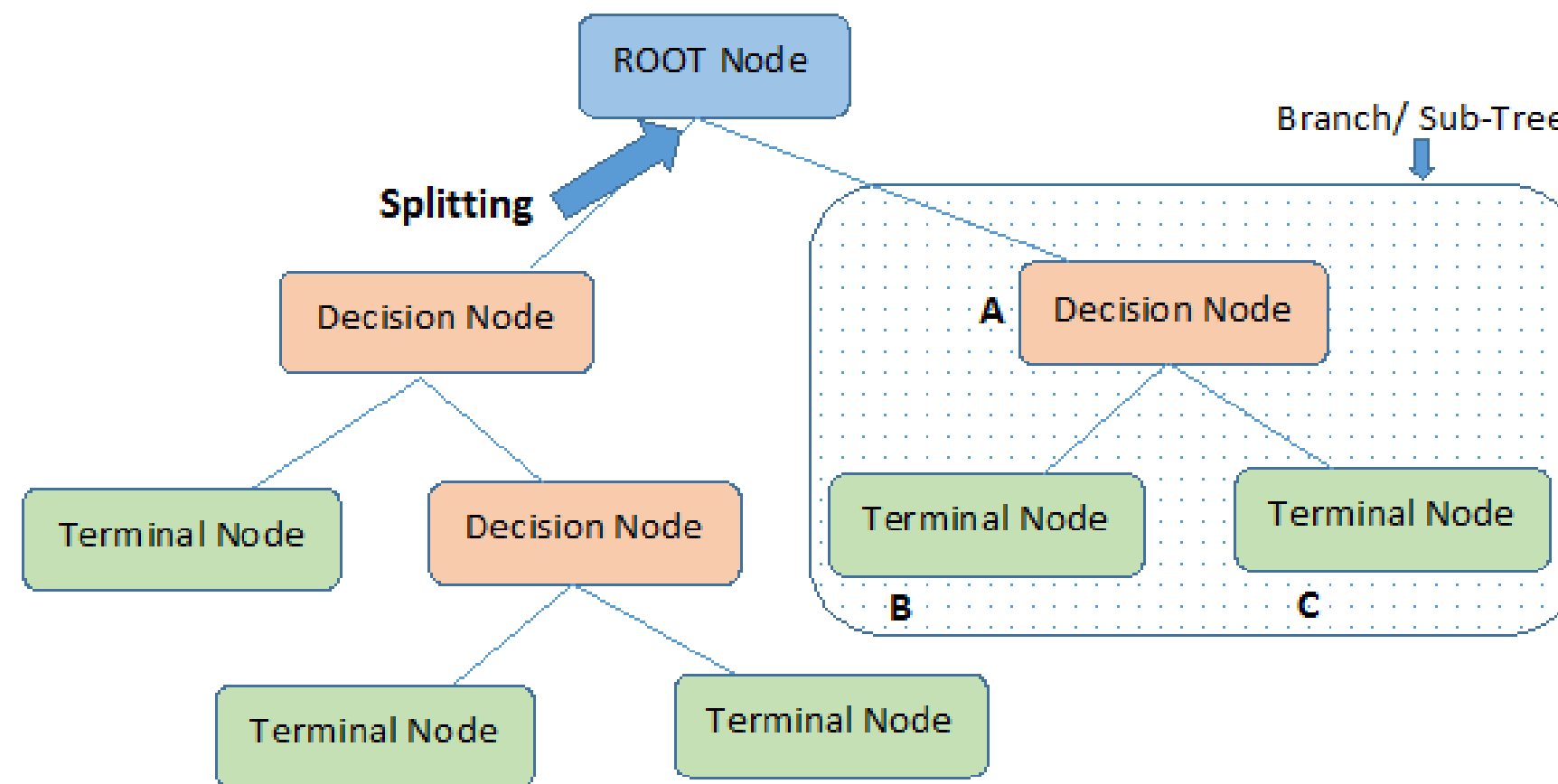
Important Terminology

Root Node: It represents entire population or sample and this further gets divided into two or more homogeneous sets.

Splitting: It is a process of dividing a node into two or more sub-nodes.

Decision Node: When a sub-node splits into further sub-nodes, then it is called decision node.

Leaf/ Terminal Node: Nodes do not split is called Leaf or Terminal node.



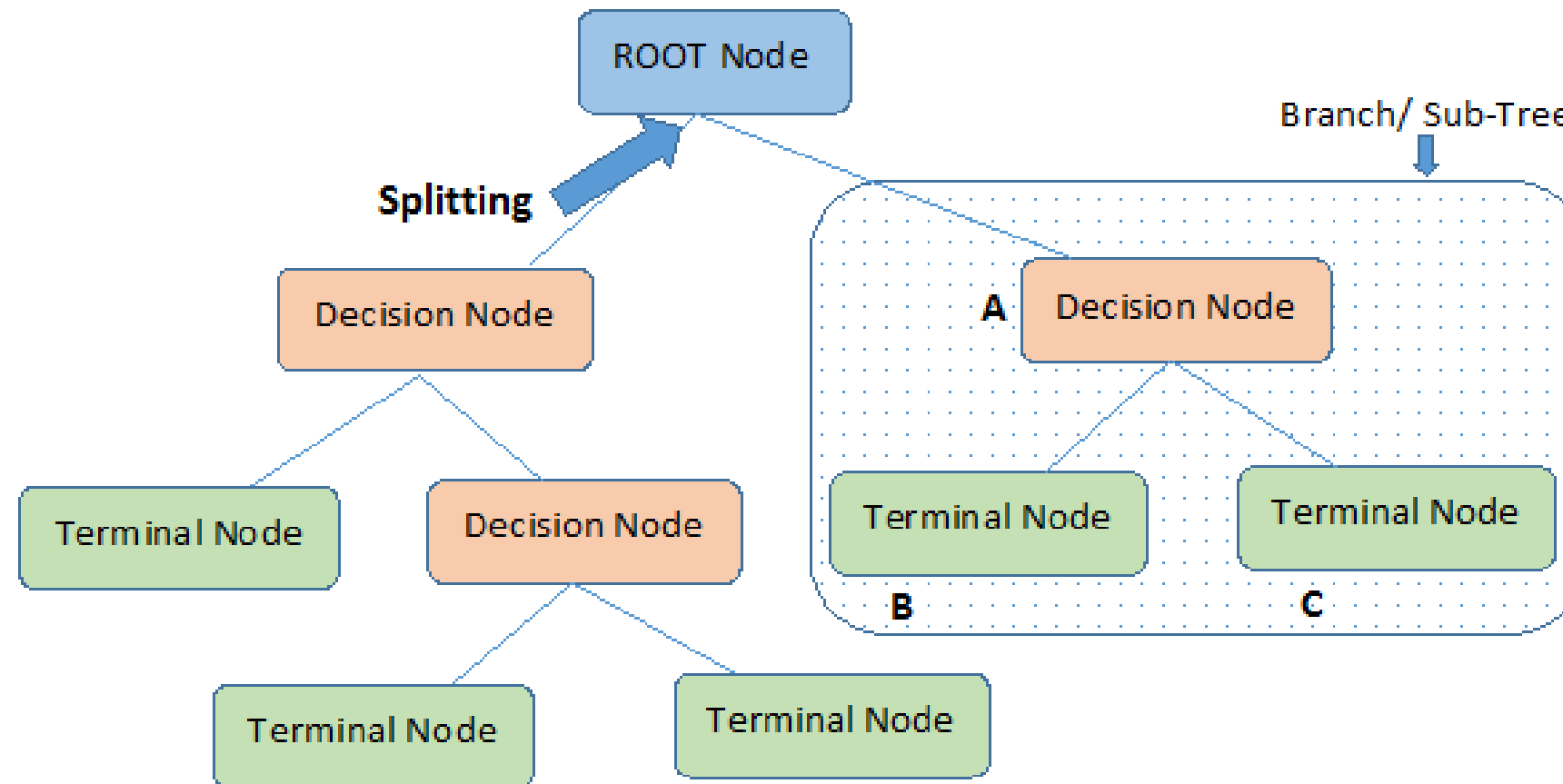
Note:- A is parent node of B and C.

Important Terminology

Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.

Branch / Sub-Tree: A sub section of entire tree is called branch or sub-tree.

Parent and Child Node: A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

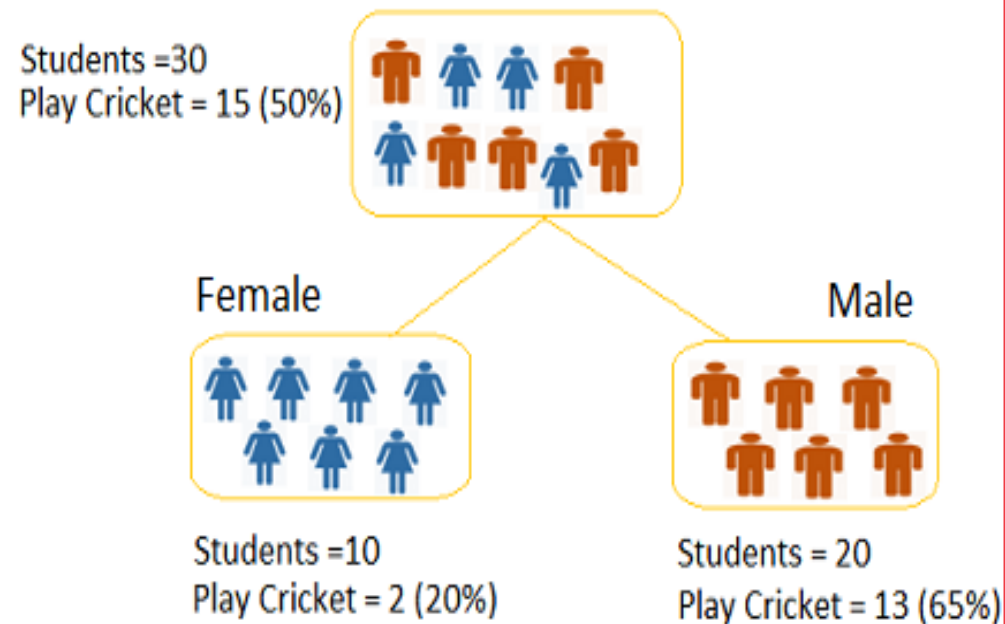


Note:- A is parent node of B and C.

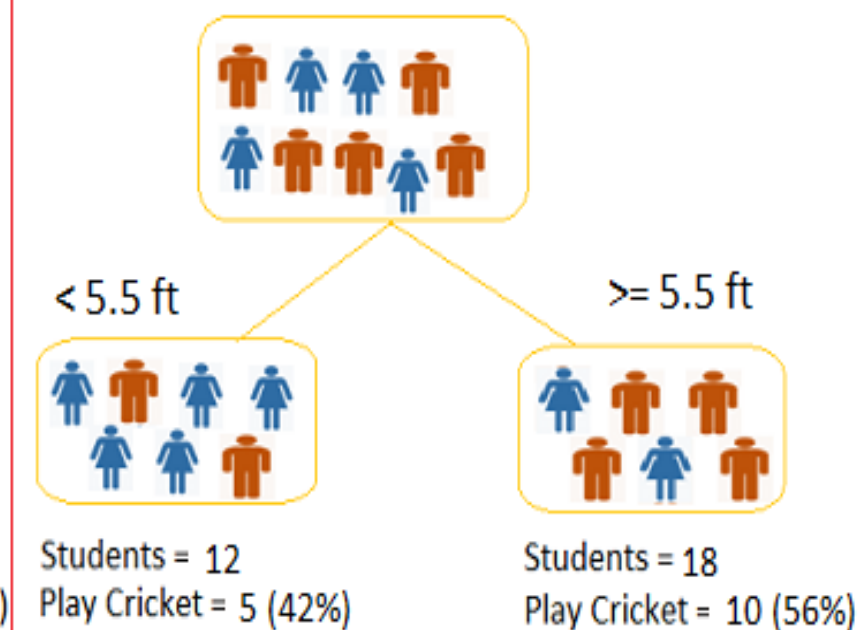
Decision Trees - Example

Let's say we have a sample of 30 students with three variables Gender (Boy/ Girl), Class(IX/ X) and Height (5 to 6 ft). 15 out of these 30 play cricket in leisure time. Now, I want to create a model to predict who will play cricket during leisure period? In this problem, we need to segregate students who play cricket in their leisure time based on highly significant input variable among all three.

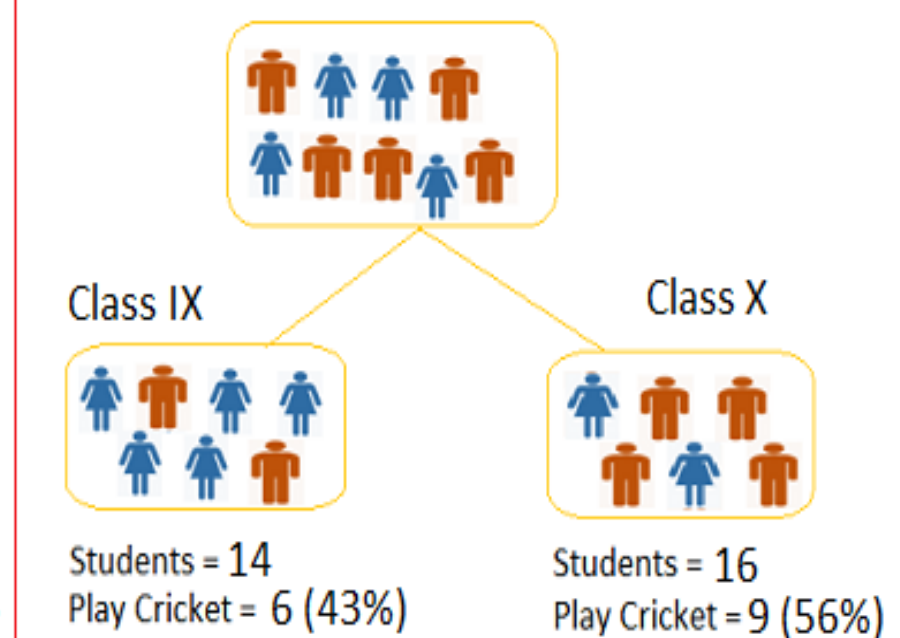
Split on Gender



Split on Height



Split on Class



How does a tree decide where to split?

- The decision of making strategic splits heavily affects a tree's accuracy.
- Decision trees use multiple algorithms to decide to split a node in two or more sub-nodes.
- The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that purity of the node increases with respect to the target variable.
- Decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.
- Commonly used algorithms for split:
 1. Gini Index
 2. Information Gain
 3. Reduction in Variance
 4. Chi-Square

Gini Index

- Gini index says, if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure.
 1. It works with categorical target variable “Success” or “Failure”.
 2. It performs only Binary splits
 3. Higher the value of Gini higher the homogeneity.
 4. CART (Classification and Regression Tree) uses Gini method to create binary splits.

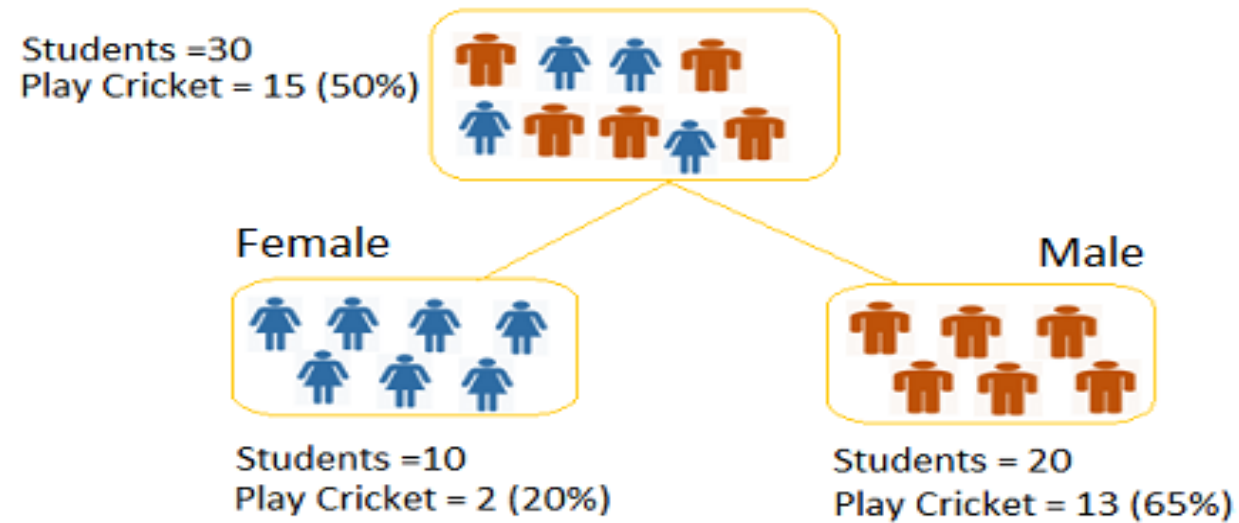
Steps to Calculate Gini for a split

1. Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure (p^2+q^2).
2. Calculate Gini for split using weighted Gini score of each node of that split

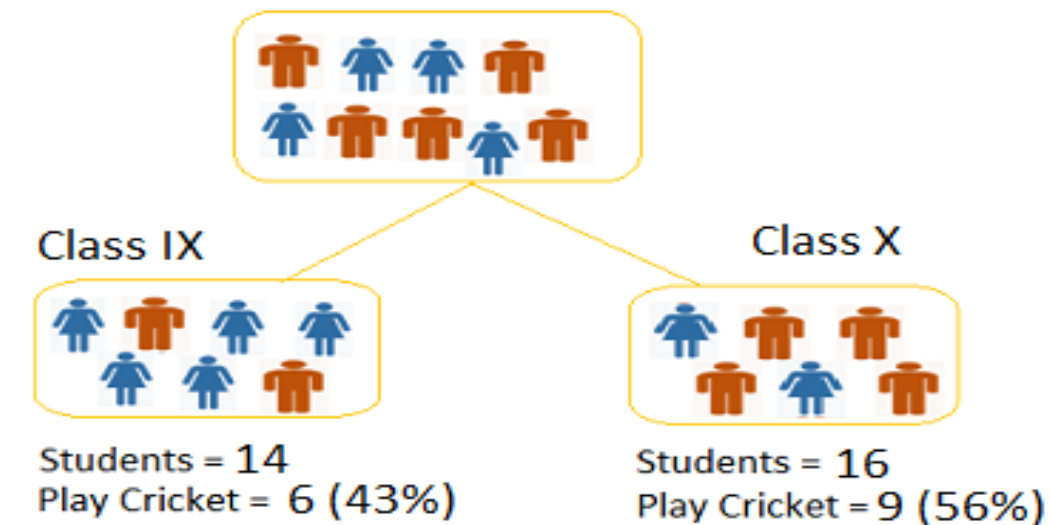
Gini Index - Example

In the snapshot below, we split the population using two input variables Gender and Class. Now, we want to identify which split is producing more homogeneous sub-nodes using Gini index.

Split on Gender



Split on Class



Split on Gender:

Calculate, Gini for sub-node Female = $(0.2) \times (0.2) + (0.8) \times (0.8) = 0.68$

Gini for sub-node Male = $(0.65) \times (0.65) + (0.35) \times (0.35) = 0.55$

Calculate weighted Gini for Split Gender = $(10/30) \times 0.68 + (20/30) \times 0.55 = \mathbf{0.59}$

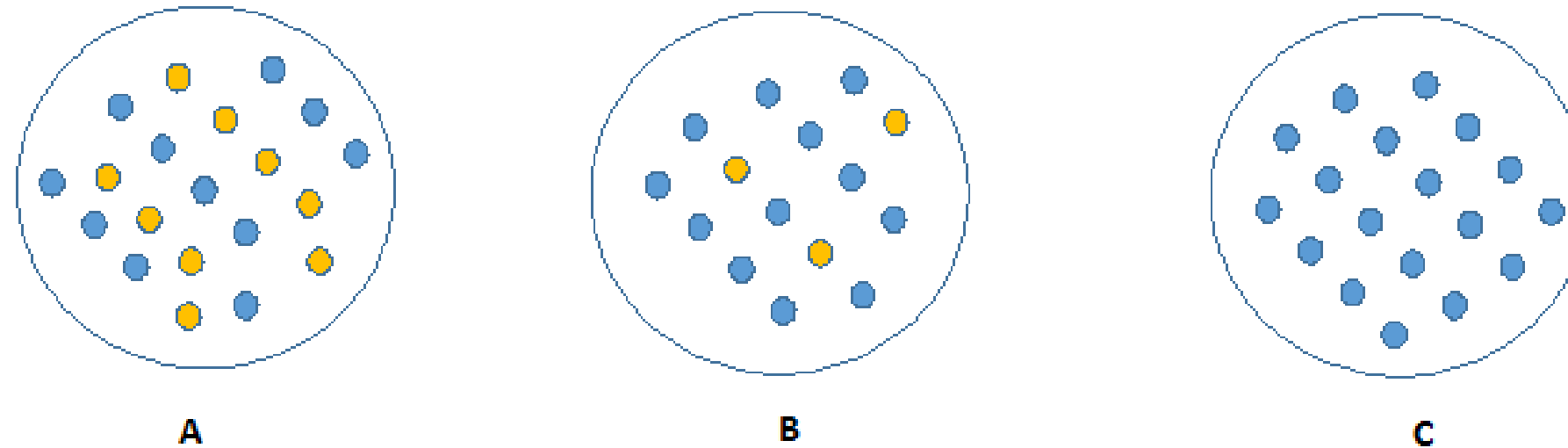
Similar for Split on Class:

Gini for sub-node Class IX = $(0.43) \times (0.43) + (0.57) \times (0.57) = 0.51$

Gini for sub-node Class X = $(0.56) \times (0.56) + (0.44) \times (0.44) = 0.51$

Calculate weighted Gini for Split Class = $(14/30) \times 0.51 + (16/30) \times 0.51 = \mathbf{0.51}$

Information Gain



Information theory is a measure to define degree of disorganization in a system known as Entropy. If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided (50% – 50%), it has entropy of one.

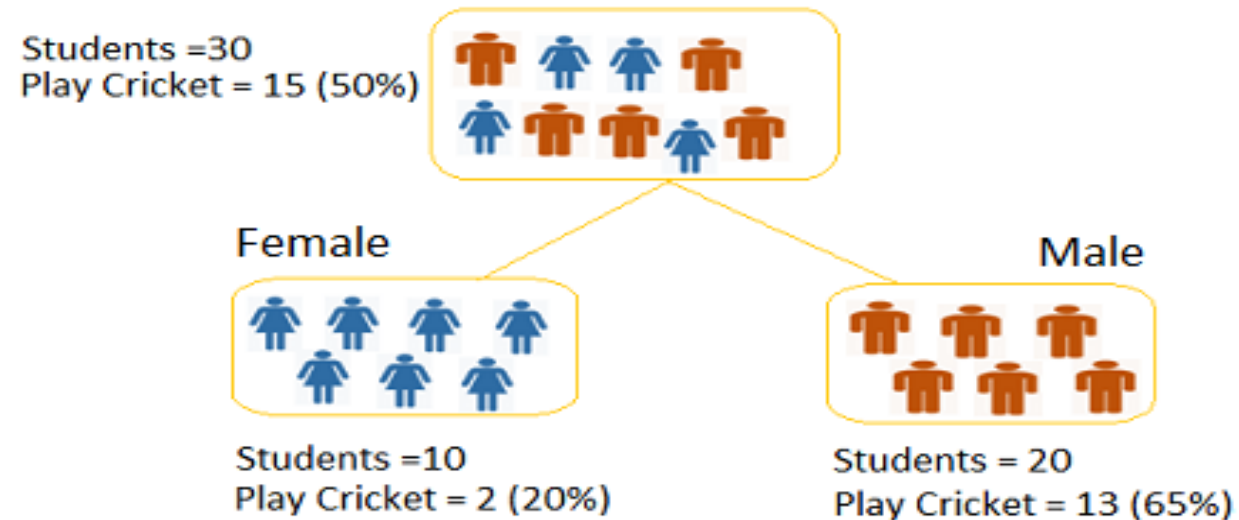
$$\text{Entropy} = -p \log_2 p - q \log_2 q$$

Steps to calculate entropy for a split:

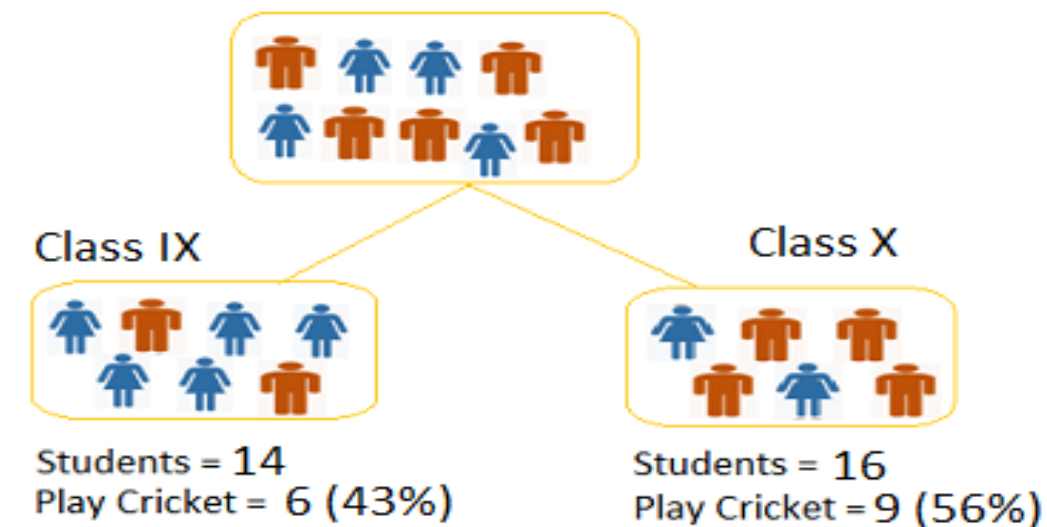
1. Calculate entropy of parent node
2. Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.
3. Information gain = Parent Entropy – Split Entropy

Information Gain - Example

Split on Gender



Split on Class



1. Entropy for parent node = $-(15/30) \log_2 (15/30) - (15/30) \log_2 (15/30) = 1$. Here 1 shows that it is a impure node.
2. Entropy for Female node = $-(2/10) \log_2 (2/10) - (8/10) \log_2 (8/10) = 0.72$ and for male node, $-(13/20) \log_2 (13/20) - (7/20) \log_2 (7/20) = \mathbf{0.93}$
3. Entropy for split Gender = Weighted entropy of sub-nodes = $(10/30)*0.72 + (20/30)*0.93 = \mathbf{0.86}$
4. Information gain Gender = $1.00 - 0.86 = 0.14$
5. Entropy for Class IX node, $-(6/14) \log_2 (6/14) - (8/14) \log_2 (8/14) = 0.99$ and for Class X node, $-(9/16) \log_2 (9/16) - (7/16) \log_2 (7/16) = 0.99$.
6. Entropy for split Class = $(14/30)*0.99 + (16/30)*0.99 = \mathbf{0.99}$
7. Information gain Class = $1.00 - 0.99 = 0.01$

Chi-Square

It is an algorithm to find out the statistical significance between the differences between sub-nodes and parent node. We measure it by sum of squares of standardized differences between observed and expected frequencies of target variable.

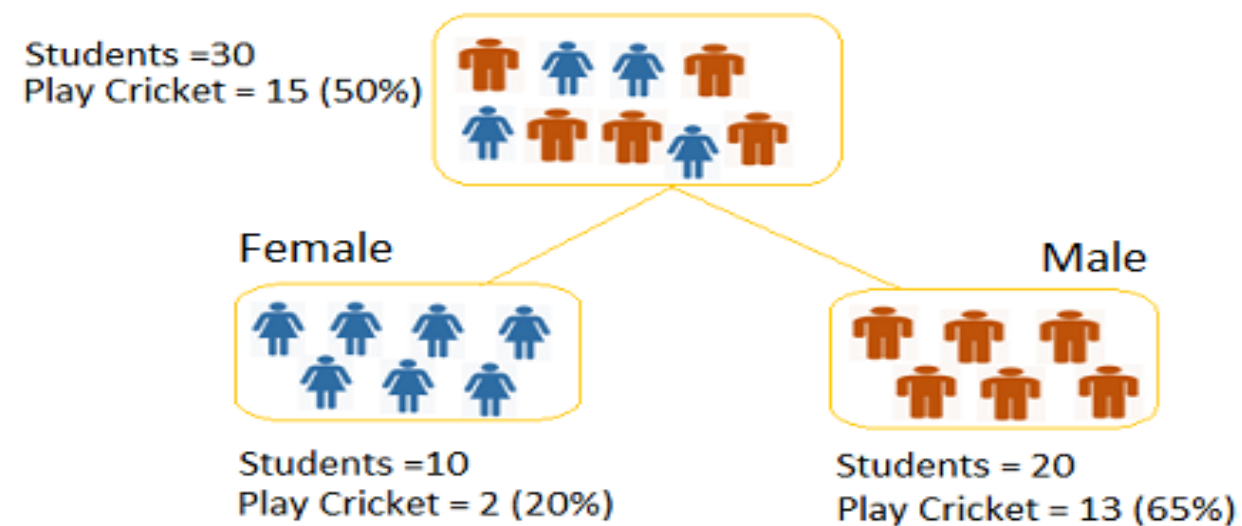
1. It works with categorical target variable “Success” or “Failure”.
2. It can perform two or more splits.
3. Higher the value of Chi-Square higher the statistical significance of differences between sub-node and Parent node.
4. Chi-Square of each node is calculated using formula,
5. $\text{Chi-square} = ((\text{Actual} - \text{Expected})^2 / \text{Expected})^{1/2}$
6. It generates tree called CHAID (Chi-square Automatic Interaction Detector)

Steps to Calculate Chi-square for a split:

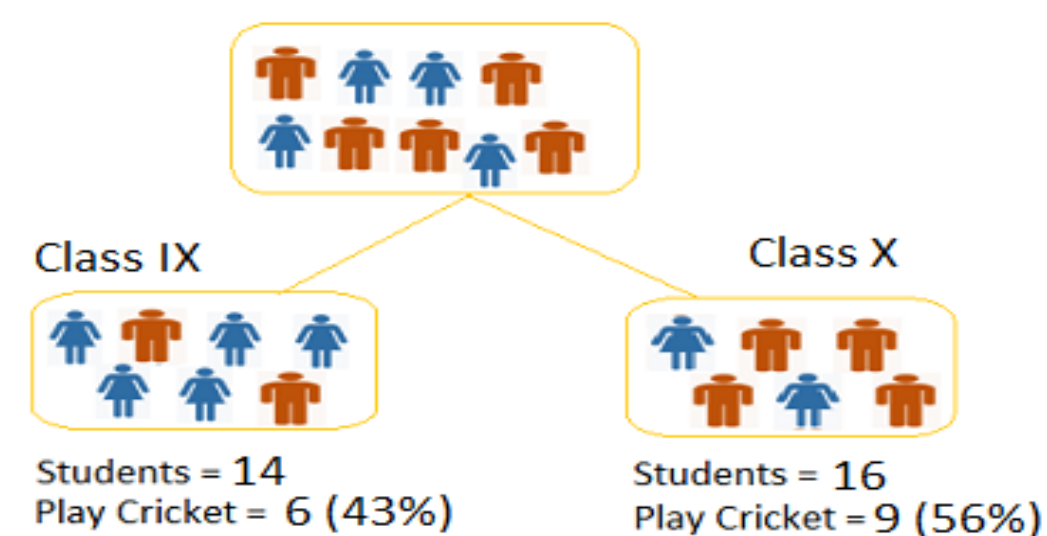
1. Calculate Chi-square for individual node by calculating the deviation for Success and Failure both
2. Calculated Chi-square of Split using Sum of all Chi-square of success and Failure of each node of the split

Chi-Square – Example

Split on Gender



Split on Class

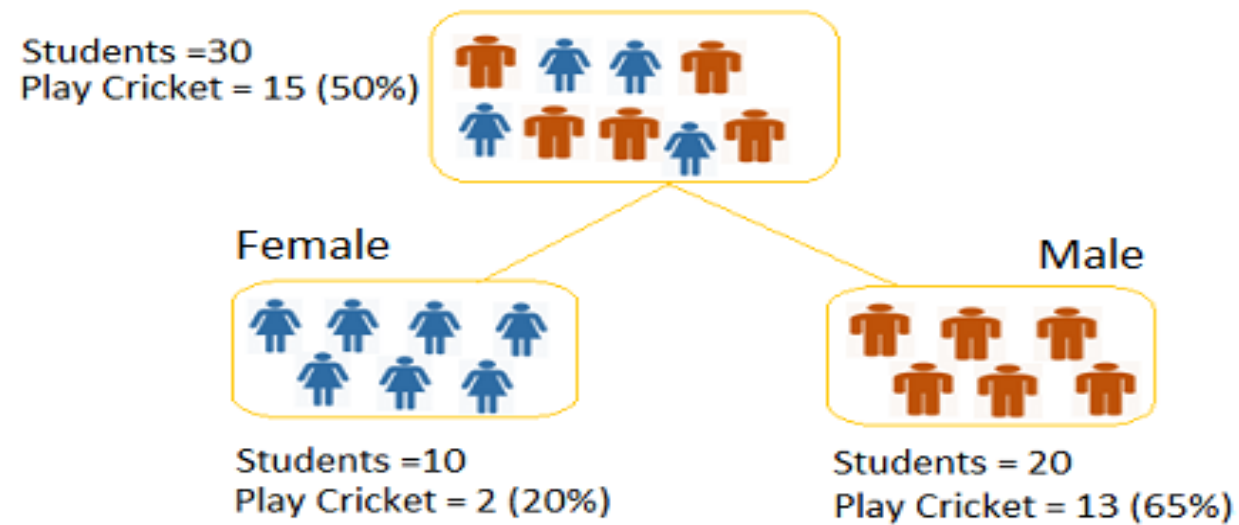


Split on Gender:

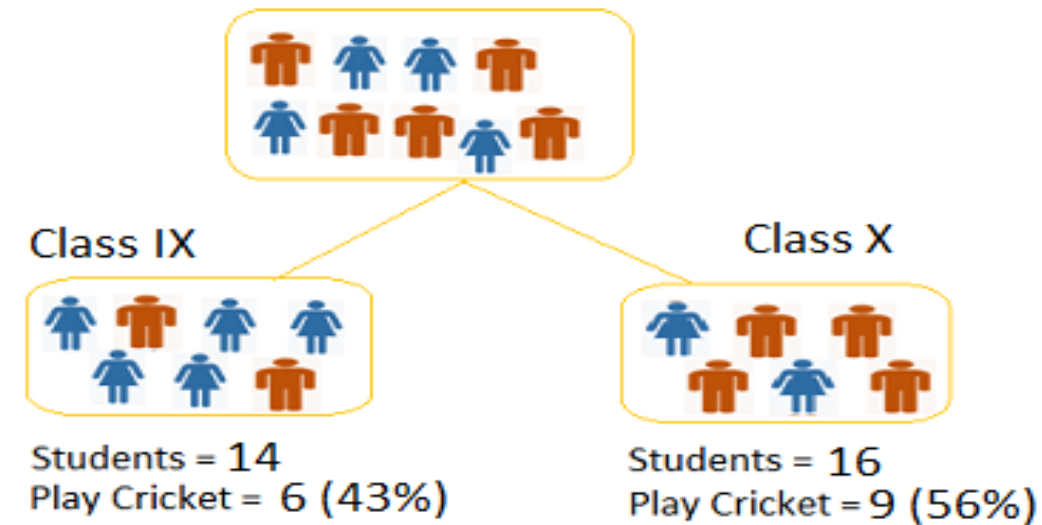
1. First we are populating for node Female, Populate the actual value for “**Play Cricket**” and “**Not Play Cricket**”, here these are 2 and 8 respectively.
2. Calculate expected value for “**Play Cricket**” and “**Not Play Cricket**”, here it would be 5 for both because parent node has probability of 50% and we have applied same probability on Female count(10).
3. Calculate deviations by using formula, Actual – Expected. It is for “**Play Cricket**” ($2 - 5 = -3$) and for “**Not play cricket**” ($8 - 5 = 3$).
4. Calculate Chi-square of node for “**Play Cricket**” and “**Not Play Cricket**” using formula with formula, $= ((\text{Actual} - \text{Expected})^2 / \text{Expected})^{1/2}$. You can refer below table for calculation.
5. Follow similar steps for calculating Chi-square value for Male node.
6. Now add all Chi-square values to calculate Chi-square for split Gender.

Chi-Square – Example

Split on Gender



Split on Class

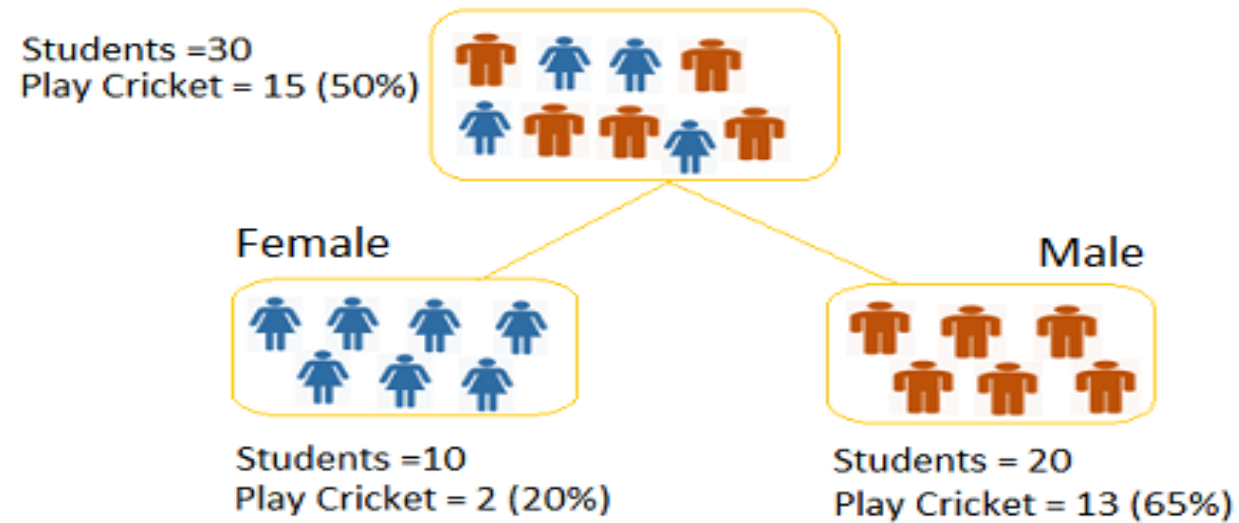


Split on Gender:

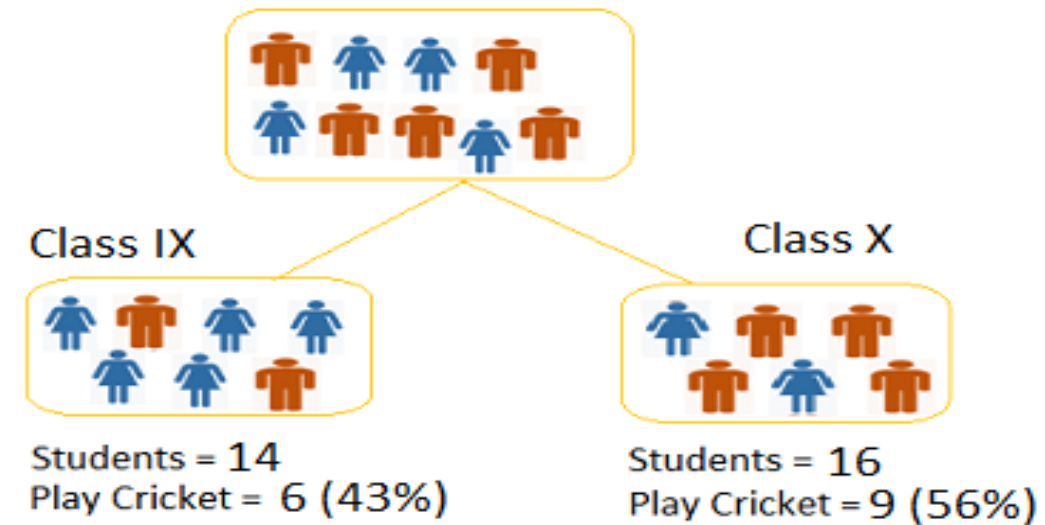
Node	Play Cricket	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
Female	2	8	10	5	5	-3	3	1.34	1.34
Male	13	7	20	10	10	3	-3	0.95	0.95
Total Chi-Square								4.58	

Chi-Square – Example

Split on Gender



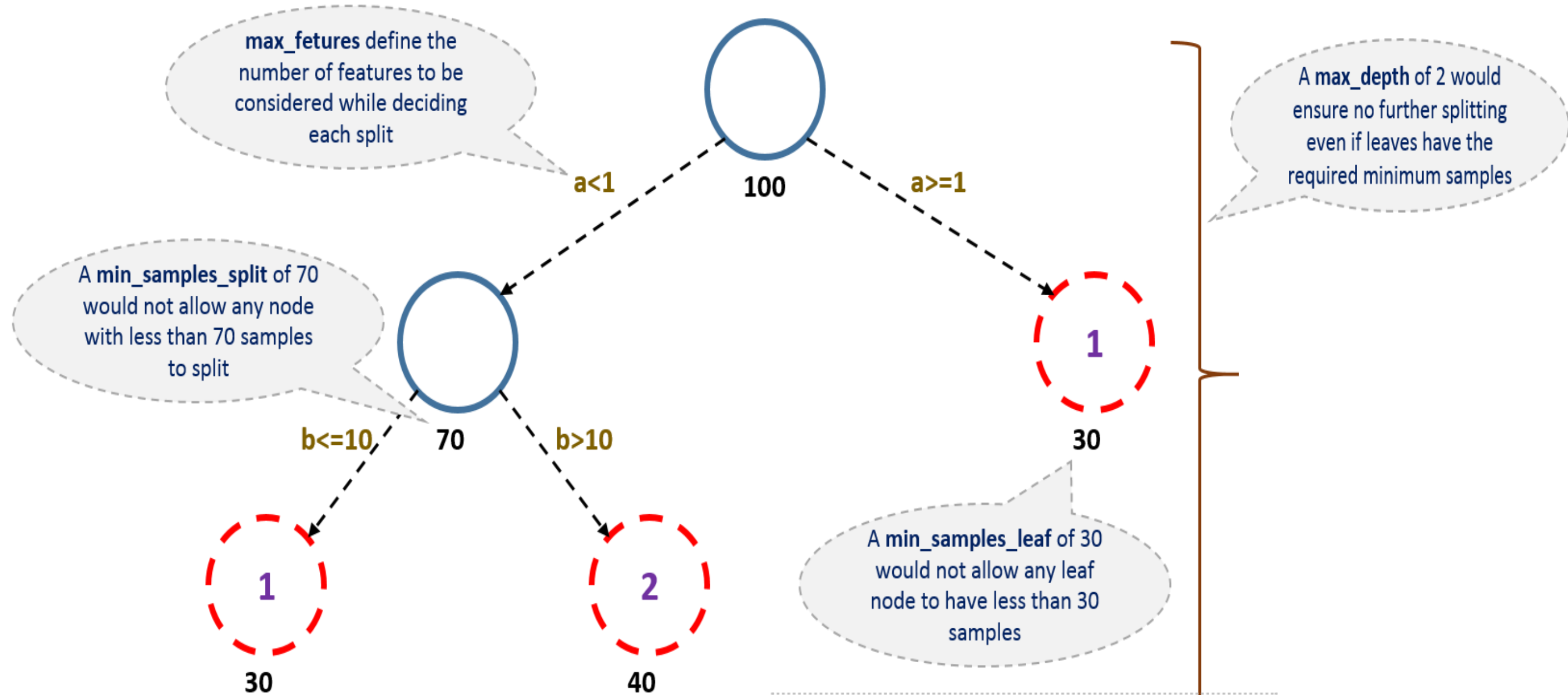
Split on Class



Split on Class:

Node	Play Cricket	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
IX	6	8	14	7	7	-1	1	0.38	0.38
X	9	7	16	8	8	1	-1	0.35	0.35
Total Chi-Square								1.46	

Key Hyper-parameters of Tree Modeling

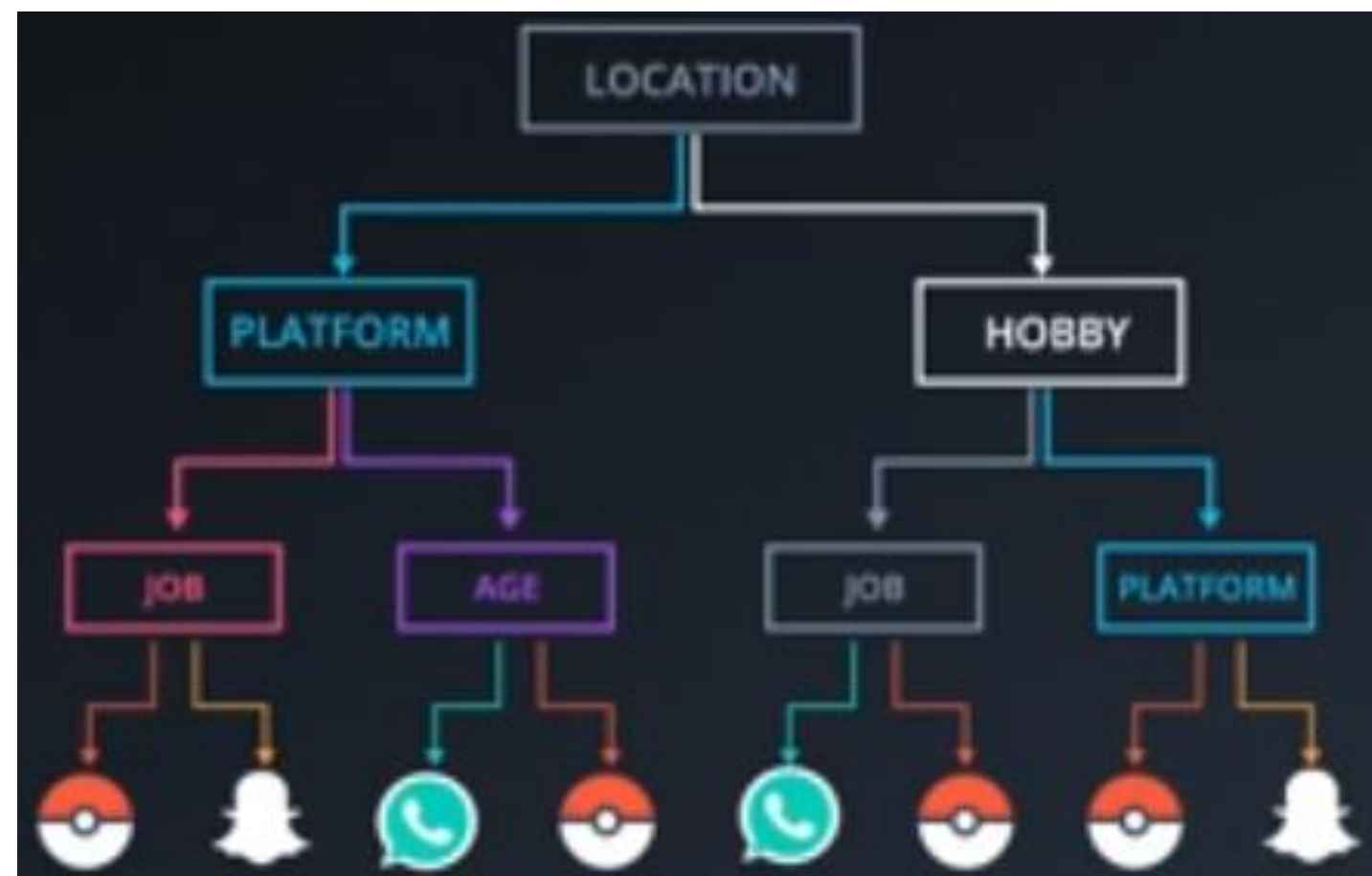


LEGEND

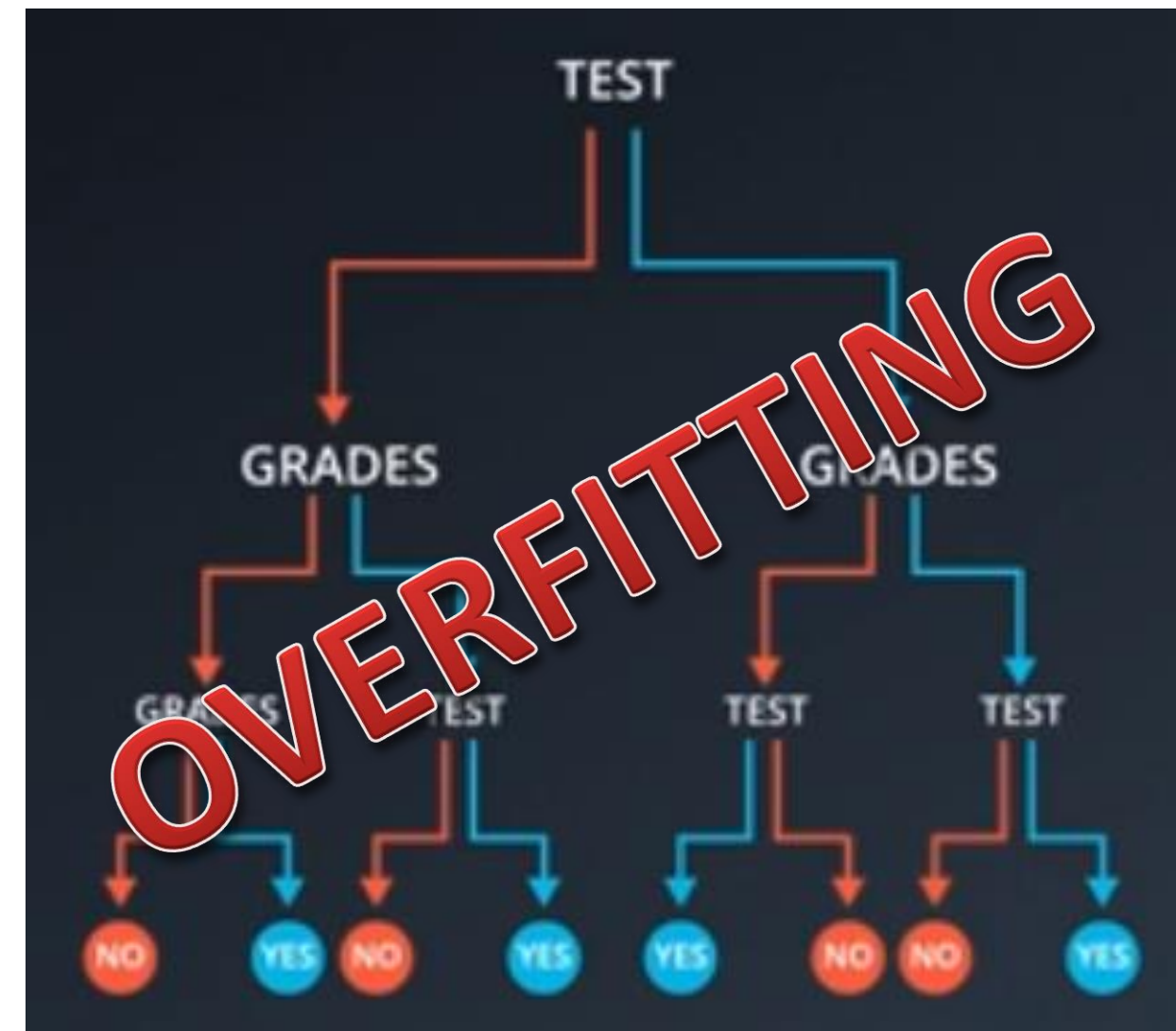
	Tree nodes except for terminal node
	Terminal nodes or leaves of tree
xx	Number of samples in a leaf
Y	The predicted value of a leaf node
abc	Node splitting criteria

Let's Practice

Gender	Age	Location	Platform	Job	Hobby	App
F	15	US	iOS	School	Videogames	
F	25	France	Android	Work	Tennis	
M	32	Chile	iOS	Temp	Tennis	
F	40	China	iOS	Retired	Chess	
M	12	US	Android	School	Tennis	
M	14	Australia	Android	School	Videogames	



If client is male, between 15 and 25, in the US on android, in school, likes tennis, pizza, but does not like to go walks on the beach, then they are likely to download Pokemon Go

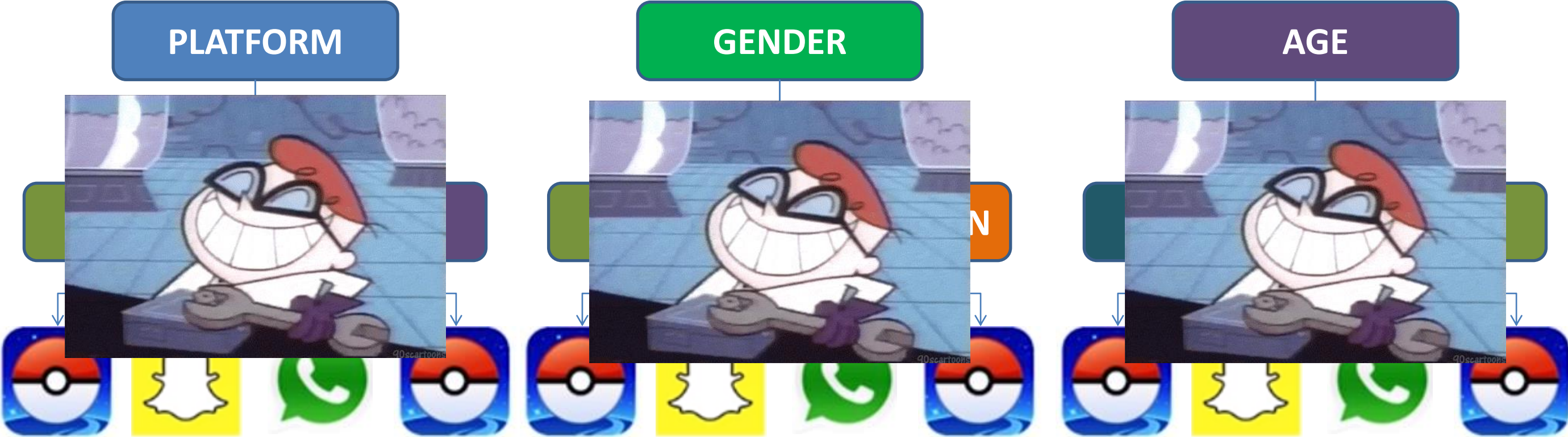


Random Forest

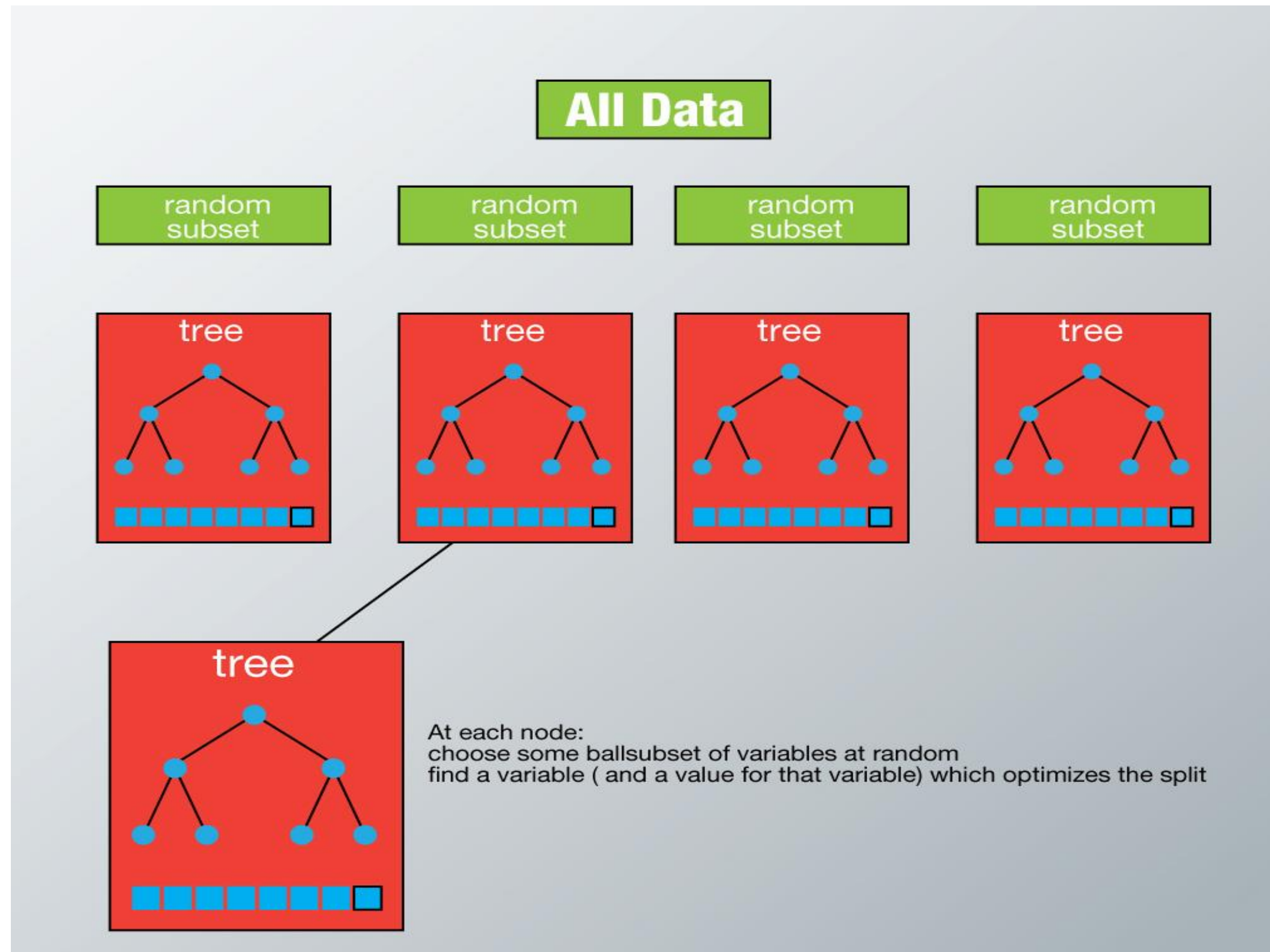
In Random Forest, we grow multiple trees as opposed to a single tree in CART model d. To classify a new object based on attributes, each tree gives a classification and we say the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees.



Gender	Age	Location	Platform	Job	Hobby	App
F	15	US	iOS	School	Videogames	
F	25	France	Android	Work	Tennis	
M	32	Chile	iOS	Temp	Tennis	
F	40	China	iOS	Retired	Chess	
M	12	US	Android	School	Tennis	
M	14	Australia	Android	School	Videogames	



How Random Forest Works?



Random Forest Hyperparameters

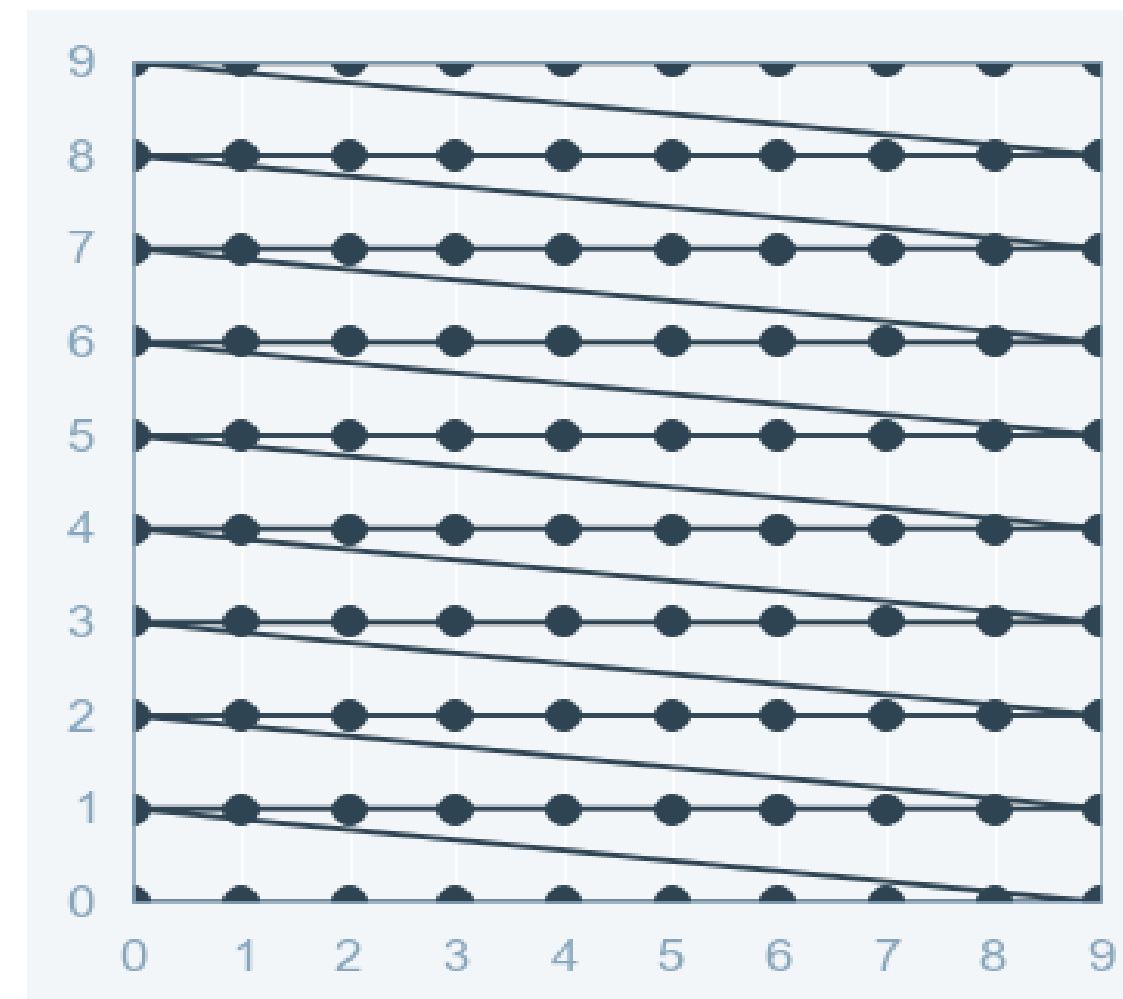
There are 4 hyperparameters required for a Random Forest classifier;

- The number of trees in the forest (`n_estimators`).
- The number of features to consider at each split. By default: square root of total number of features (`max_features`).
- The maximum depth of a tree i.e. number of nodes (`max_depth`).
- The minimum number of samples required to be at a leaf node / bottom of a tree (`min_samples_leaf`).

Tuning by GridSearchCV

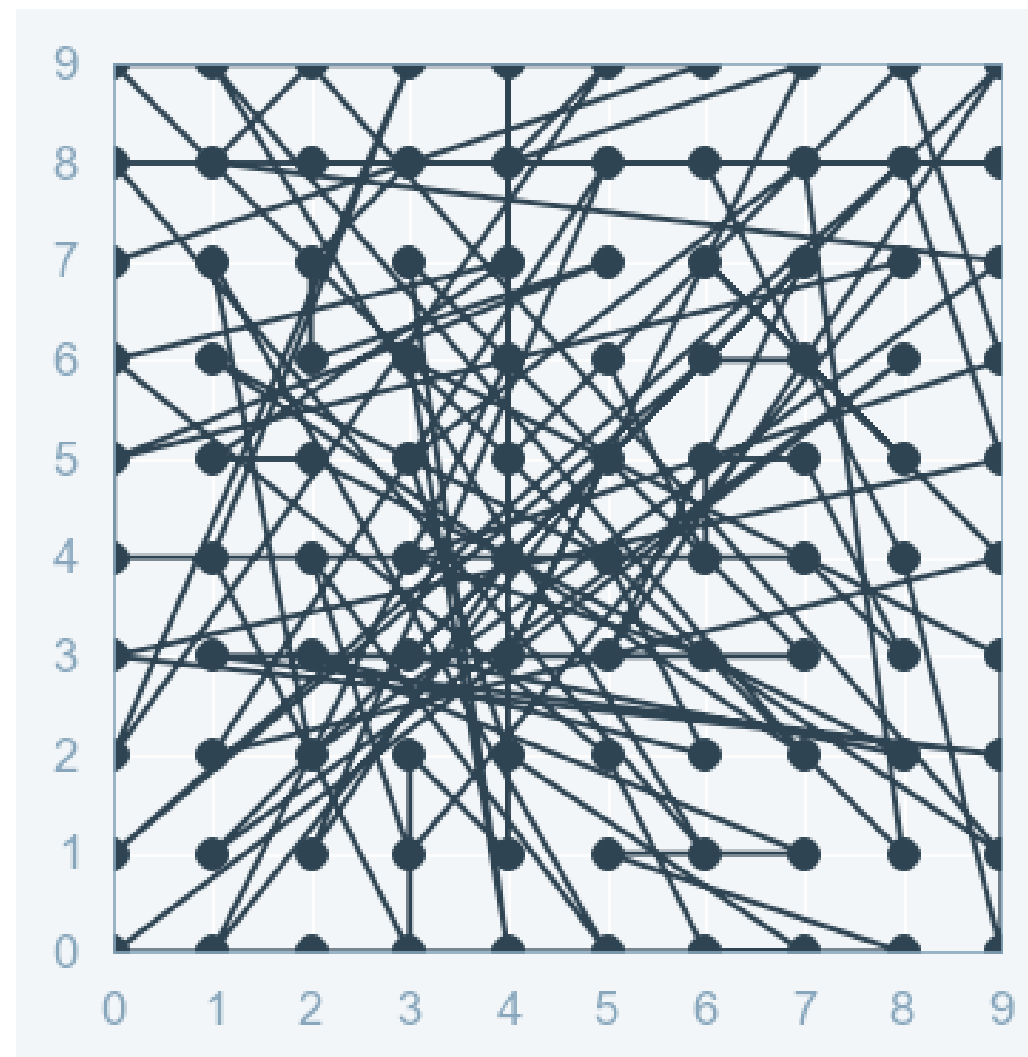
- We try every combination of a preset list of values of the hyper-parameters and evaluate the model for each combination.
- Pattern followed here is similar to the grid, where all the values are placed in the form of a matrix
- Once all the combinations are evaluated, the model with the set of parameters which give the top accuracy is considered to be the best.

max_depth	1	0.814	0.798	0.823	0.678	0.745
	5	0.669	0.668	0.767	0.667	0.667
	10	0.812	0.812	0.852	0.812	0.812
	15	0.706	0.746	0.796	0.756	0.746
	20	0.814	0.823	0.618	0.813	0.813
		1	2	4	6	8
		min_samples_leaf				



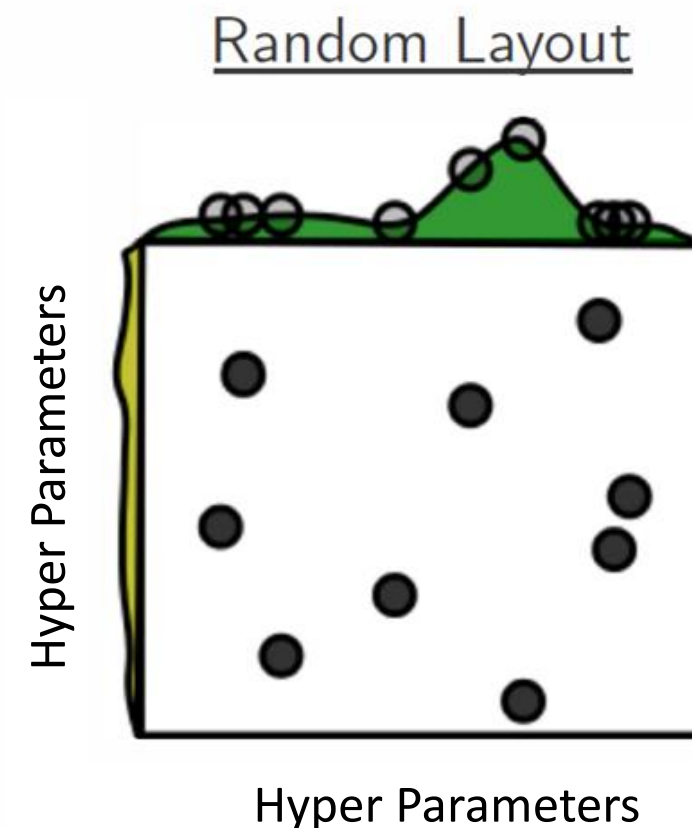
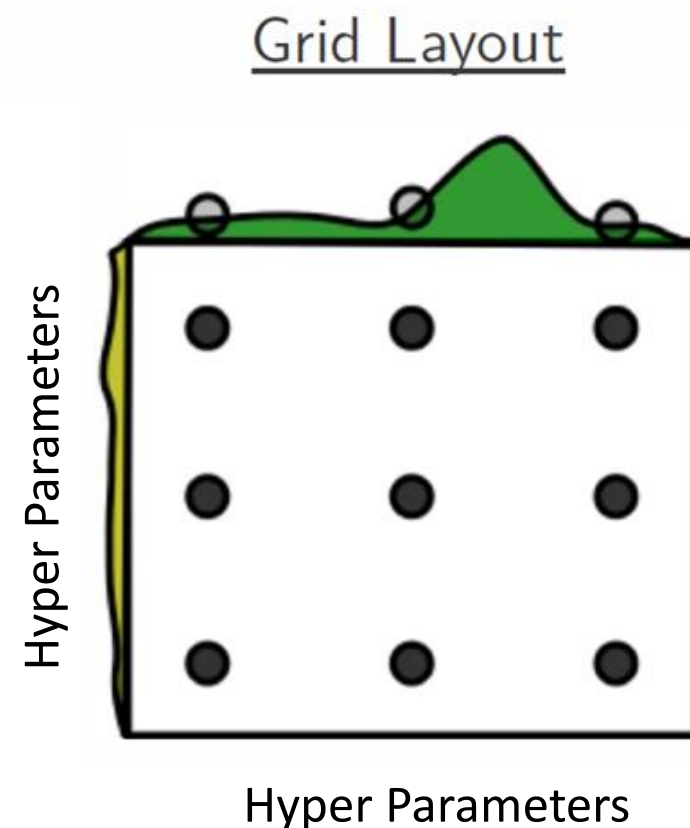
Tuning by RandomSearchCV

- Random search is a technique where random combinations of the hyper parameters are used to find the best solution for the built model
- It tries random combinations of a range of values
- To optimize with random search, the function is evaluated at some number of random configurations in the parameter space



GridSearch vs RandomSearch

- One of the major drawbacks of grid search is that when it comes to dimensionality, it suffers when the number of hyper parameters grows exponentially.
- The chances of finding the optimal parameter are comparatively higher in random search because of the random search pattern where the model might end up being trained on the optimized parameters without any aliasing number of random configurations in the parameter space



Let's Practice