Supervised Learning Algorithms

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian classification
- Rule-based classification
- Classification by back propagation
- Support Vector Machines (SVM)
- Model selection
- Summary

Objectives

- Learn basic techniques for data classification and prediction.
- Realize the difference between the following classifications of data:
  - supervised classification
  - prediction
  - unsupervised classification

What is Classification?

- The goal of data classification is to organize and categorize data in distinct classes.
  - A model is first created.
  - The model is then used to classify new data.
  - Given the model, a class can be predicted for new data.
- Classification = prediction for discrete and nominal values

What is Prediction?

- The goal of prediction is to forecast or deduce the value of an attribute based on values of other attributes.
  - A model is first created based on the data distribution.
  - The model is then used to predict future or unknown values

- **In Machine Learning**
  - If forecasting discrete value → **Classification**
  - If forecasting continuous value → **Prediction**

Classification Example

- Example training database
  - Two predictor attributes: Age and Car-type (**S**port, **M**inivan and **T**ruck)
  - Age is numeric, Car-type is categorical attribute
  - Class label indicates whether person bought product
  - Dependent attribute is *categorical*

Age	Car	Class
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

Regression (Prediction) Example

- Example training database
  - Two predictor attributes: Age and Car-type (**S**port, **M**inivan and **T**ruck)
  - Spent indicates how much person spent during a recent visit to the web site
  - Dependent attribute is *numerical*

Age	Car	Spent
20	M	$200
30	M	$150
25	T	$300
30	S	$220
40	S	$400
20	T	$80
30	M	$100
25	M	$125
40	M	$500
20	S	$420

Supervised and Unsupervised

- Supervised Classification = Classification
  - We know the class labels and the number of classes
- Unsupervised Classification = Clustering
  - We do not know the class labels and may not know the number of classes

Preparing Data Before Classification

- **Data transformation**:
  - Discretization of continuous data
  - Normalization to [-1..1] or [0..1]
- **Data Cleaning**:
  - Smoothing to reduce noise
- **Relevance Analysis**:
  - Feature selection to eliminate irrelevant attributes

Application

- Credit approval
- Target marketing
- Medical diagnosis
- Defective parts identification in manufacturing
- Crime zoning
- Treatment effectiveness analysis

Classification is a 3-step process

- **1. Model construction (Learning):**
  - Each tuple is assumed to belong to a predefined class, as determined by one of the attributes, called the **class label**.
  - The set of all tuples used for construction of the model is called **training set**.
  - The model is represented in the following forms:
    - Classification rules, (IF-THEN statements),
    - Decision tree
    - Mathematical formulae

1. Classification Process (Learning)

Name	Income	Age	Credit rating
Samir	Low	<30	bad
Ahmed	Medium	[30...40]	good
Salah	High	<30	good
Ali	Medium	>40	good
Sami	Low	[30..40]	good
Emad	Medium	<30	bad

Training Data

class

Classification Method

Classification Model

IF Income = 'High'
OR Age > 30
THEN Class = 'Good

OR

Decision Tree

OR

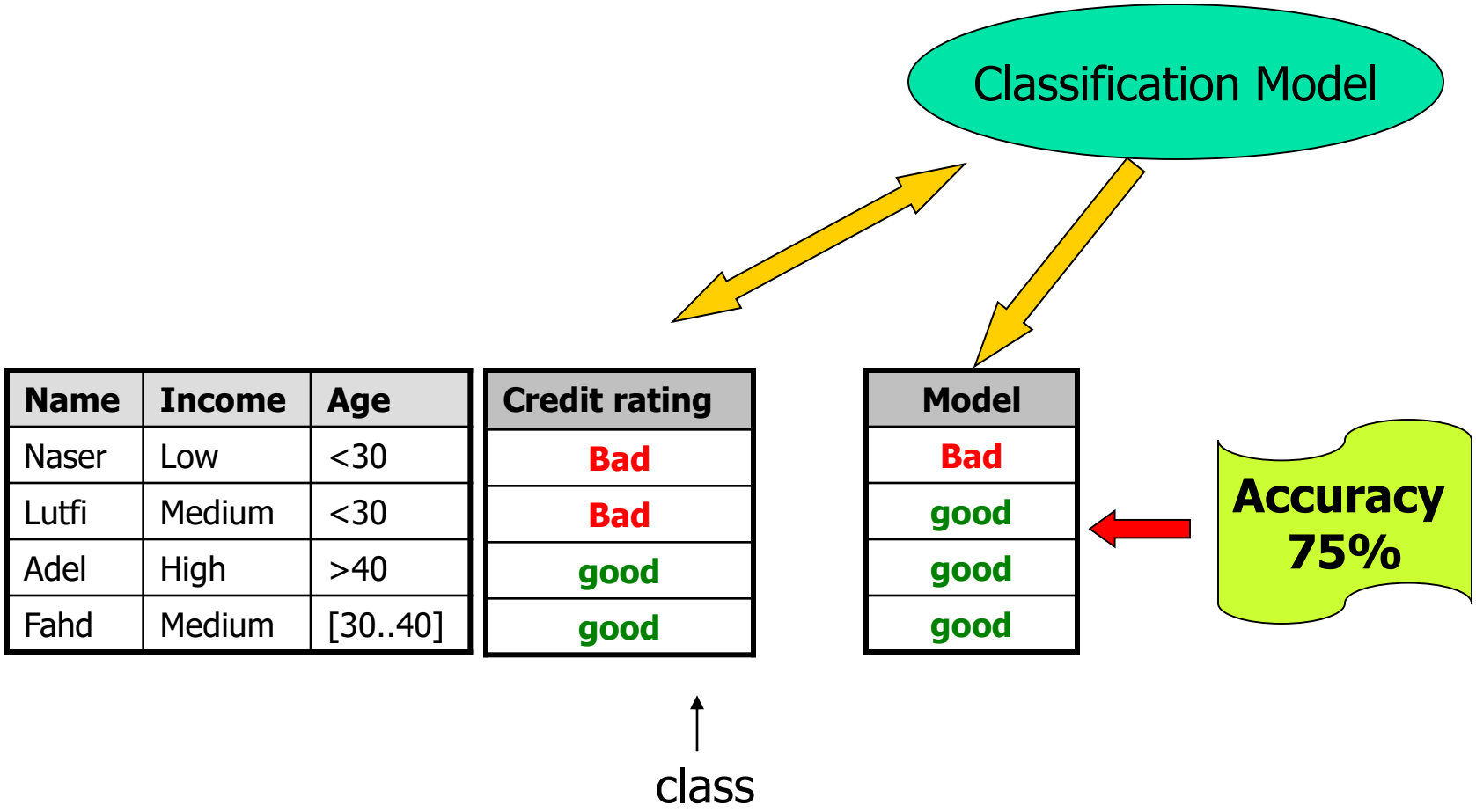
Mathematical For

Classification is a 3-step process

2. Model Evaluation (Accuracy):

- Estimate accuracy rate of the model based on a **test set**.
- The known label of test sample is compared with the classified result from the model.
- Accuracy rate is the percentage of test set samples that are correctly classified by the model.
- Test set is independent of training set otherwise over-fitting will occur

2. Classification Process (Accuracy Evaluation)

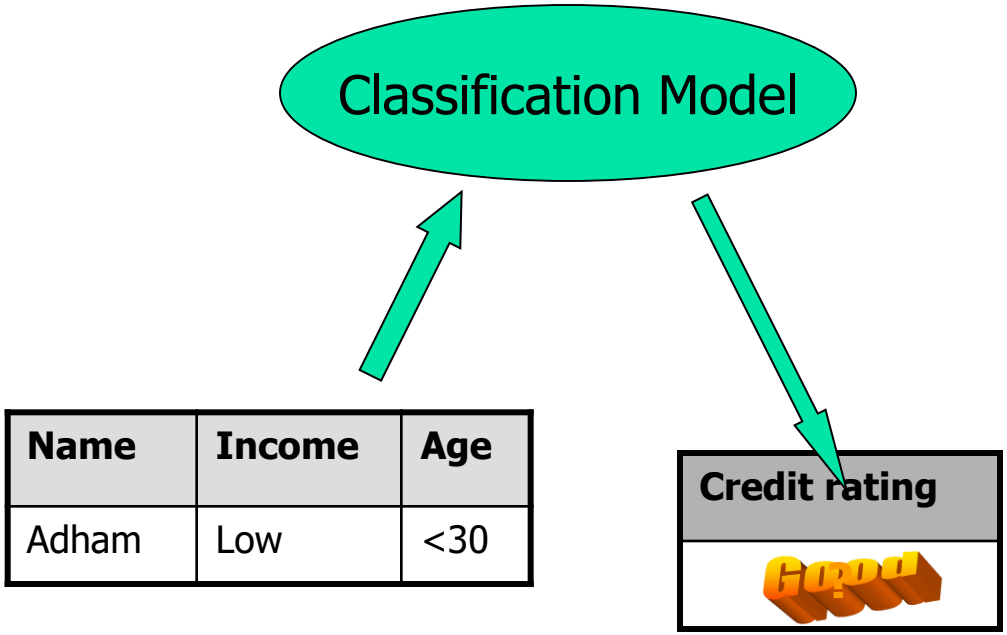


Classification is a three-step process

3. Model Use (Classification):

- The model is used to classify unseen objects.
  - Give a class label to a new tuple
  - Predict the value of an actual attribute <prediction>

3. Classification Process (Use)



Classification Methods

- Decision Tree Induction
- Neural Networks
- Bayesian Classification
- Association-Based Classification
- K-Nearest Neighbour
- Case-Based Reasoning
- Genetic Algorithms
- Rough Set Theory
- Fuzzy Sets
- Etc.

Comparing Classification and Prediction Methods

- Accuracy- *This is the ability of the model to correctly predict the class level of new or previously unseen data*.
  - classifier accuracy: predicting class label of new or previously unseen data.
  - predictor accuracy: guessing value of predicted attributes new or previously unseen data.
- Speed (computational cost)
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)

Comparing Classification and Prediction Methods

- Robustness: handling noise and missing values (ability of model to make correct predictions)
- Scalability: the ability to construct the model efficiently given large amounts of data.
- Interpretability:
  - This refers Level of understanding and insight provided by the model (classifier or predictor).
- Other measures, e.g., goodness of rules, such as decision tree size.

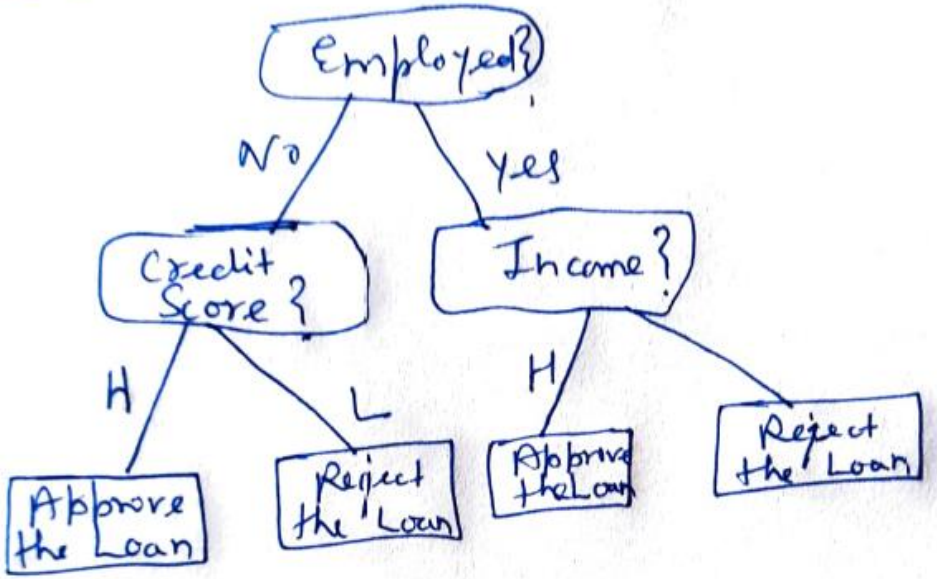
Decision Tree

Decision Tree

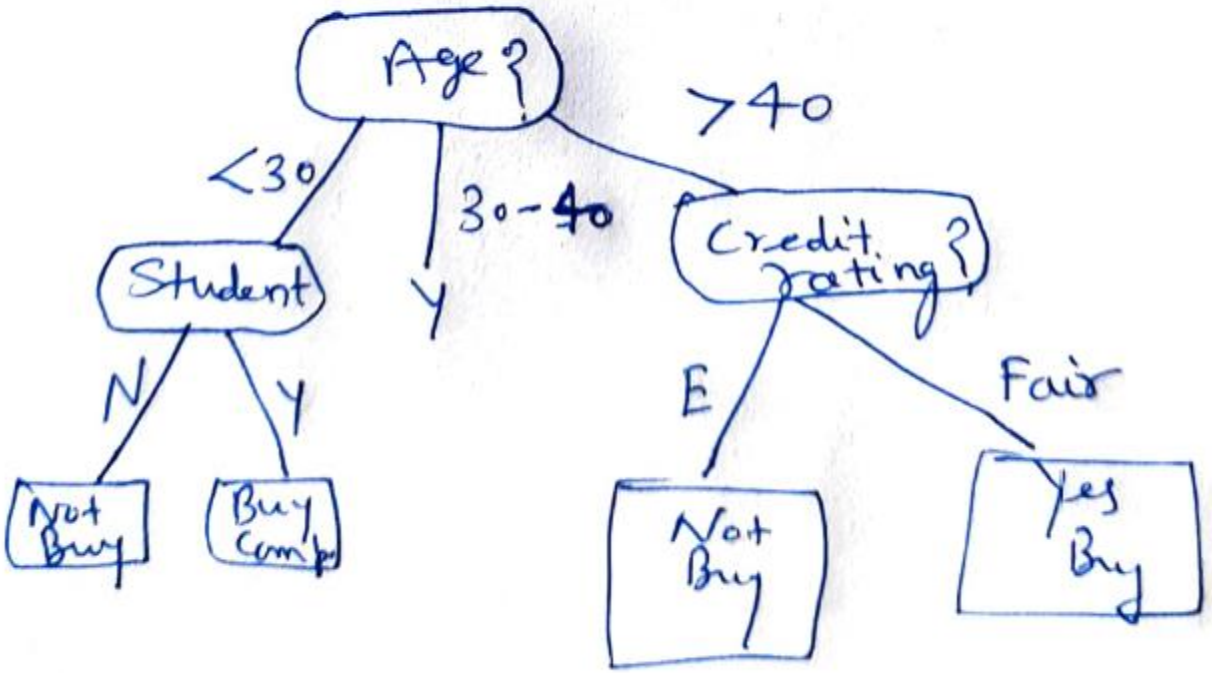
We have:

1) Decision Nodes / Test Nodes
2) Leaf Nodes
3) Edges outcome of test

Example: ① Approve the Loan OR Not Approve the Loan



② Person is likely to buy a computer ?



Example 3

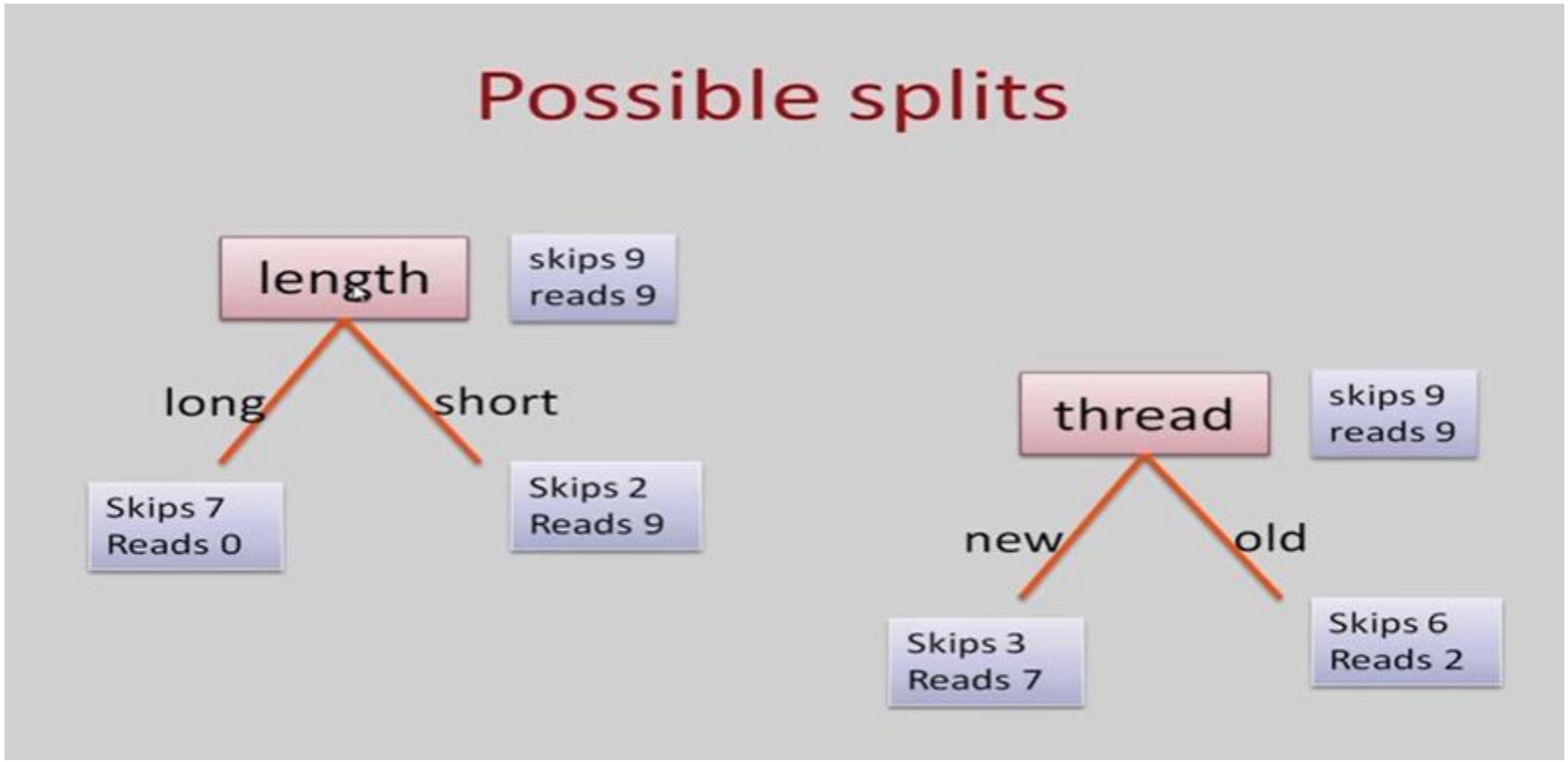
Example Data

Training Examples:

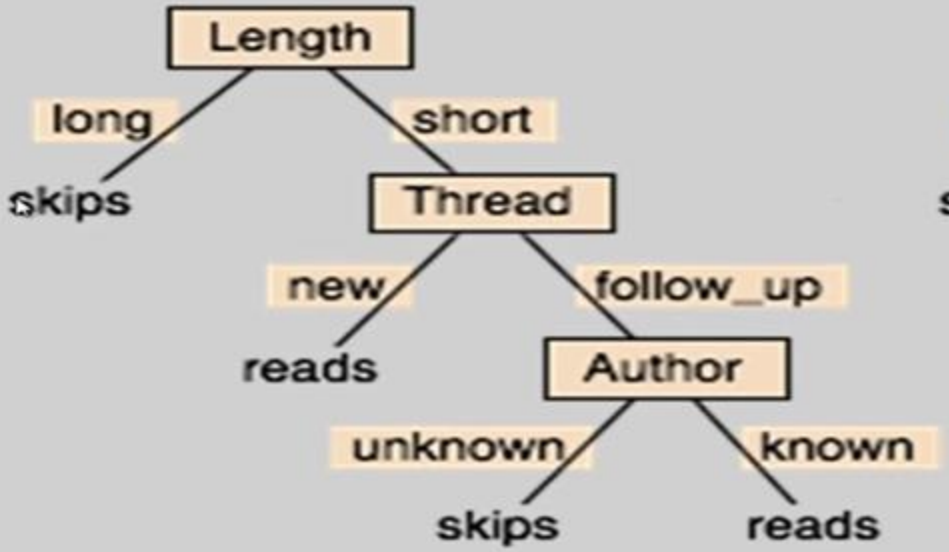
	Action	Author	Thread	Length	Where
e1	skips	known	new	long	Home
e2	reads	unknown	new	short	Work
e3	skips	unknown	old	long	Work
e4	skips	known	old	long	home
e5	reads	known	new	short	home
e6	skips	known	old	long	work

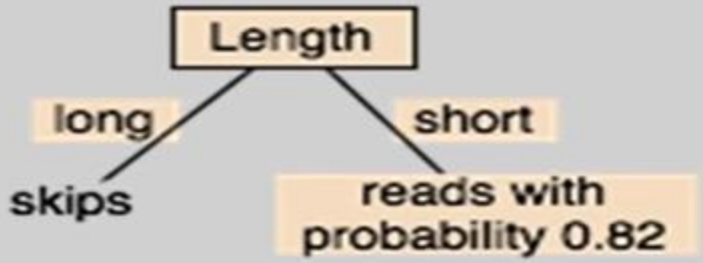
New Examples:

e7	???	known	new	short	work
e8	???	unknown	new	short	work


Possible splits
length
skips 9
reads 9
long
short
Skips 7
Reads 0
Skips 2
Reads 9
thread
skips 9
reads 9
new
old
Skips 3
Reads 7
Skips 6
Reads 2

Two Example DTs



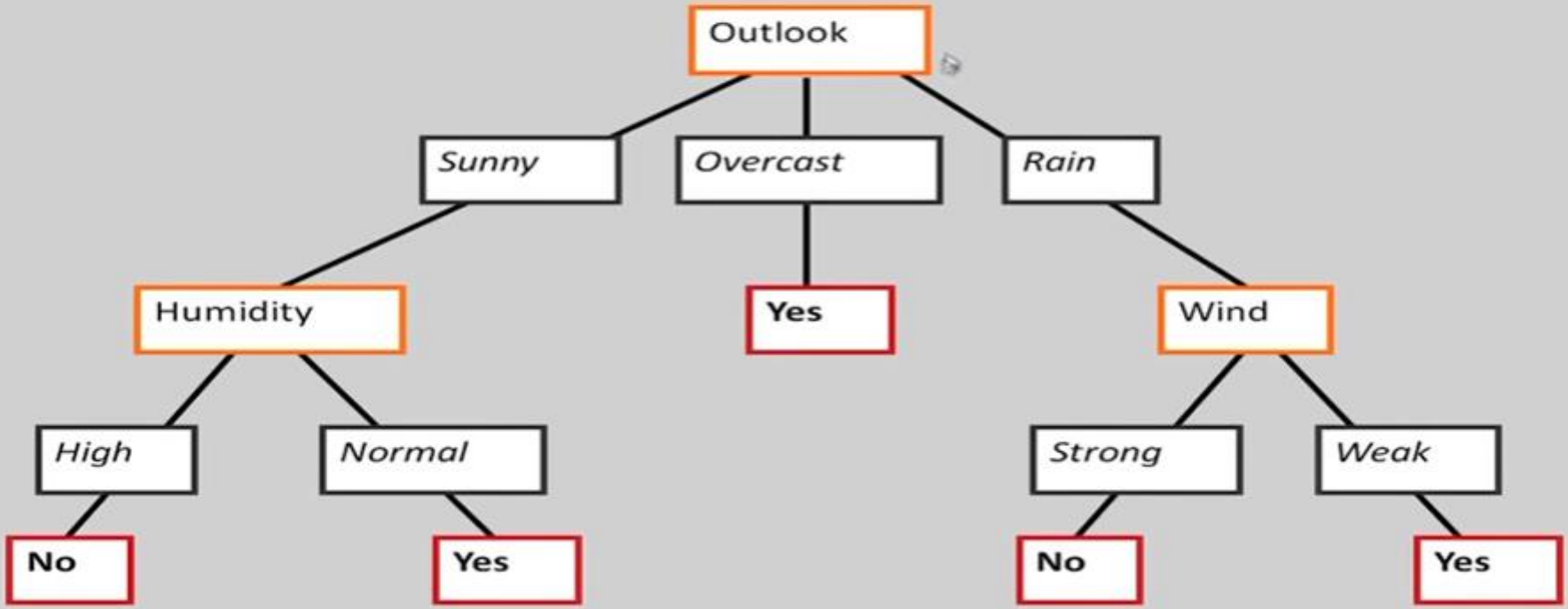


Example 4

Decision Tree for PlayTennis

- Attributes and their values:
  - Outlook: *Sunny, Overcast, Rain*
  - Humidity: *High, Normal*
  - Wind: *Strong, Weak*
  - Temperature: *Hot, Mild, Cool*
- Target concept - Play Tennis: *Yes, No*

Decision Tree for PlayTennis

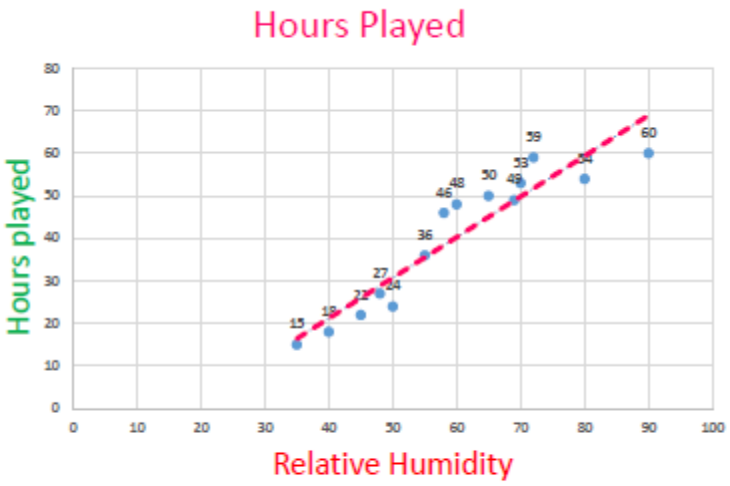


What is a Decision Tree?

- A decision tree is a flow-chart-like tree structure.
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf node represents class label

A quick recap of Linear Regression –Linear models

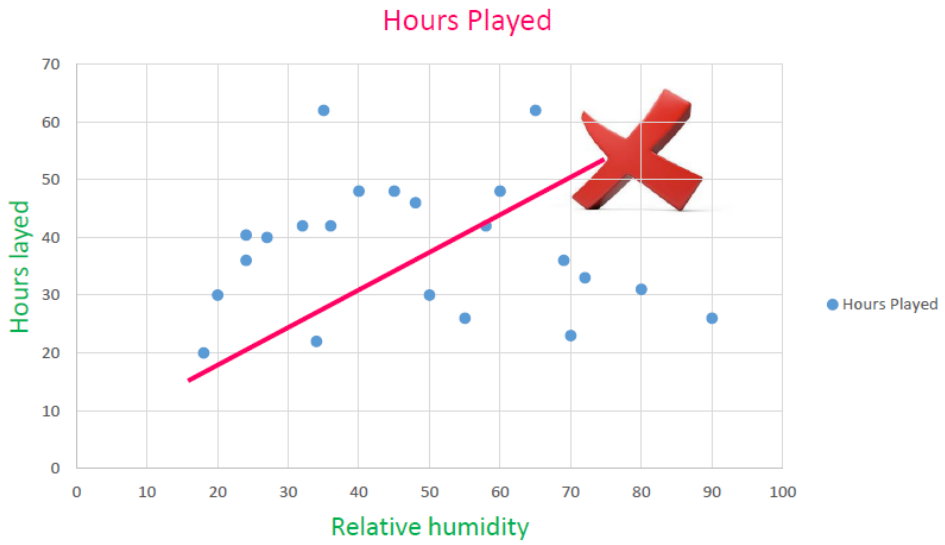
Relative Humidity	Hours Played
55	36
50	24
60	48
58	46
65	50
70	53
48	27
69	49
72	59
45	22
40	18
35	15
80	54
90	60



Linear Regression can handle this type of data quite well

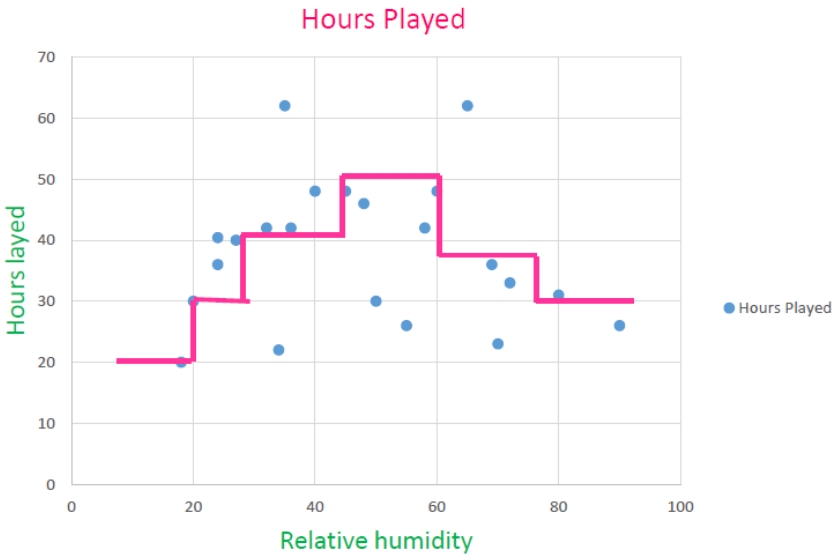
Can Linear Regression help us in this scenario?

Relative Humidity	Hours Played
55	26
50	30
60	48
58	42
65	62
70	23
48	46
69	36
72	33
45	48
40	48
35	62
80	31
90	26
27	40
36	42
24	40.4
32	42
24	36
34	22
18	20
20	30



Data are relatively scattered

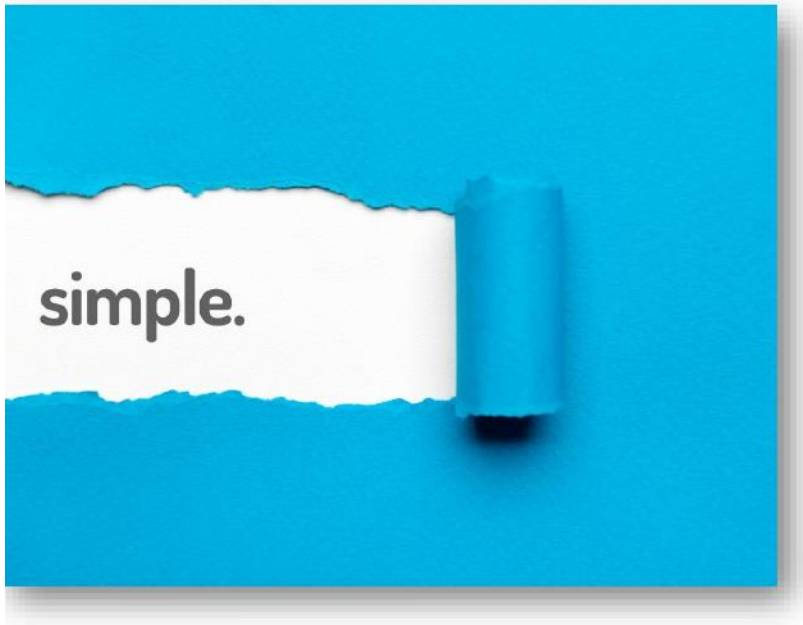
How does Decision Tree come to the rescue?

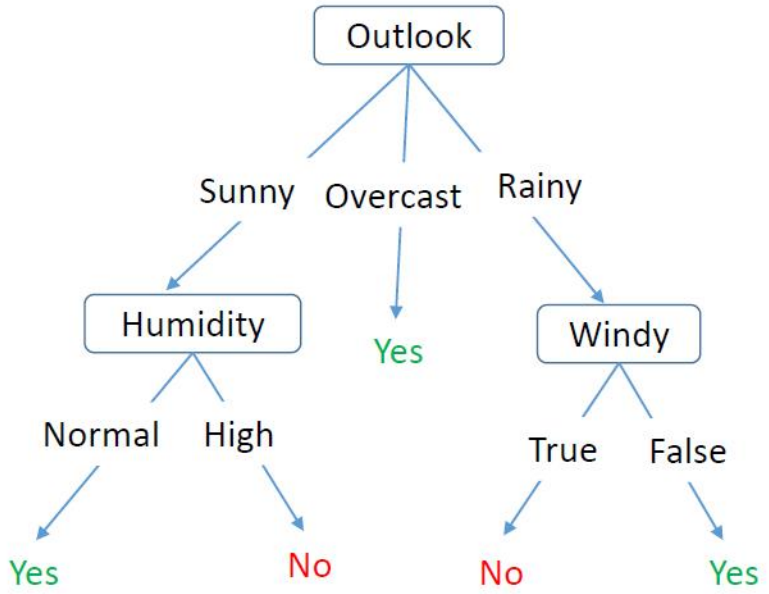


- Non linear data can well be handled by Decision Tree.
- It doesn't affect the performance of Decision Tree

Simplicity

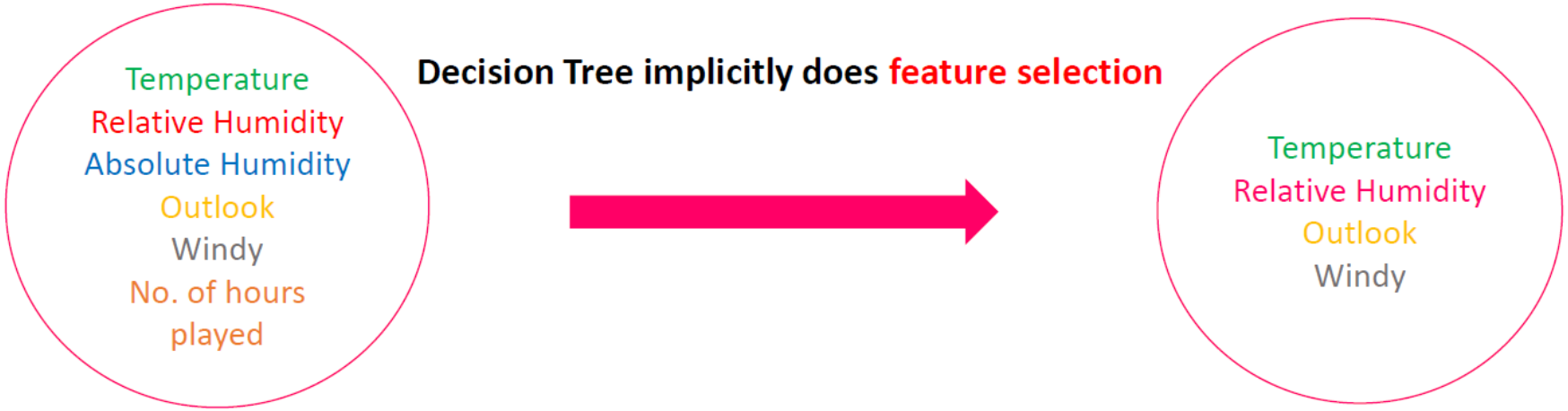
It’s simple to understand , interpret & visualize





Feature selection

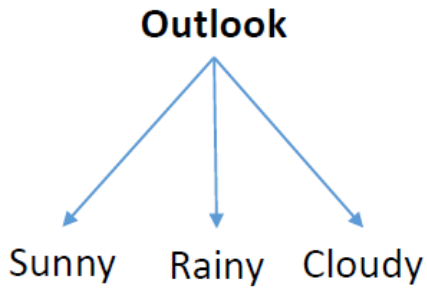
Decision Tree identifies and removes unnecessary, irrelevant & redundant attributes from data that do not contribute to accuracy of a predictive model



Handling different types of data

- Can handle both **categorical & numerical** data.
- So can be used both for **Regression & Classification**

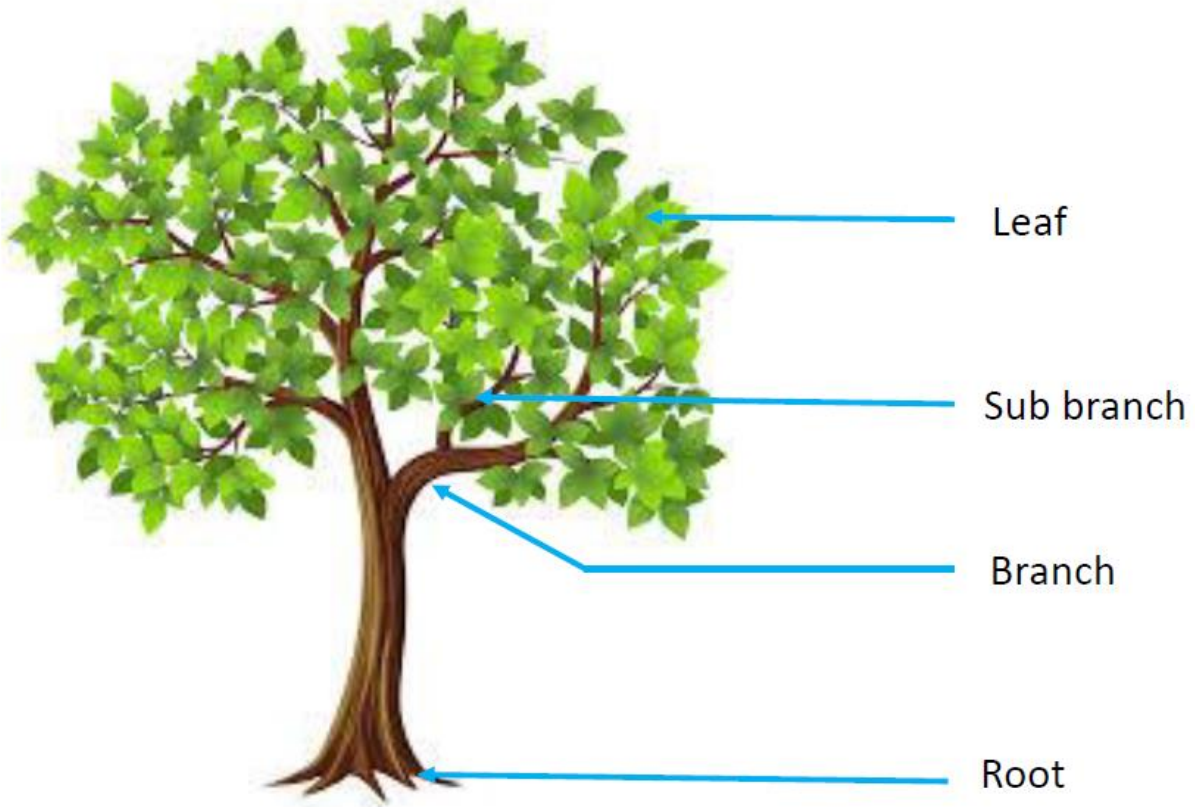
Categorical



Numeric

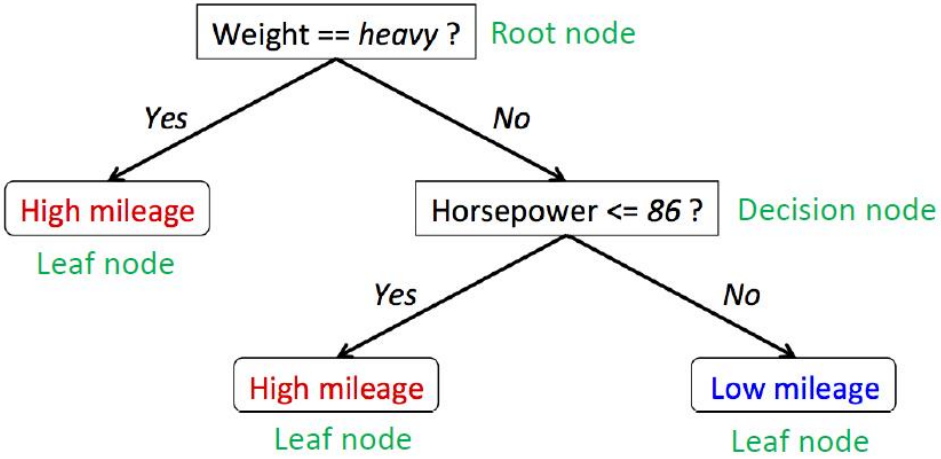
Hours Played
26
30
48
46
62
23
43
36

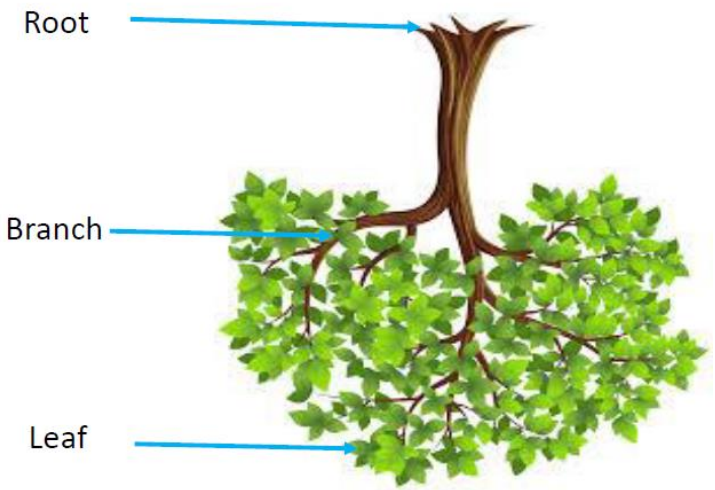
What is tree ?



What is Decision Tree(DT) ?

- A form of **supervised** learning used in ***classification*** & ***regression***,
- In which predicted values are obtained through a *inverted* **tree like structure**.





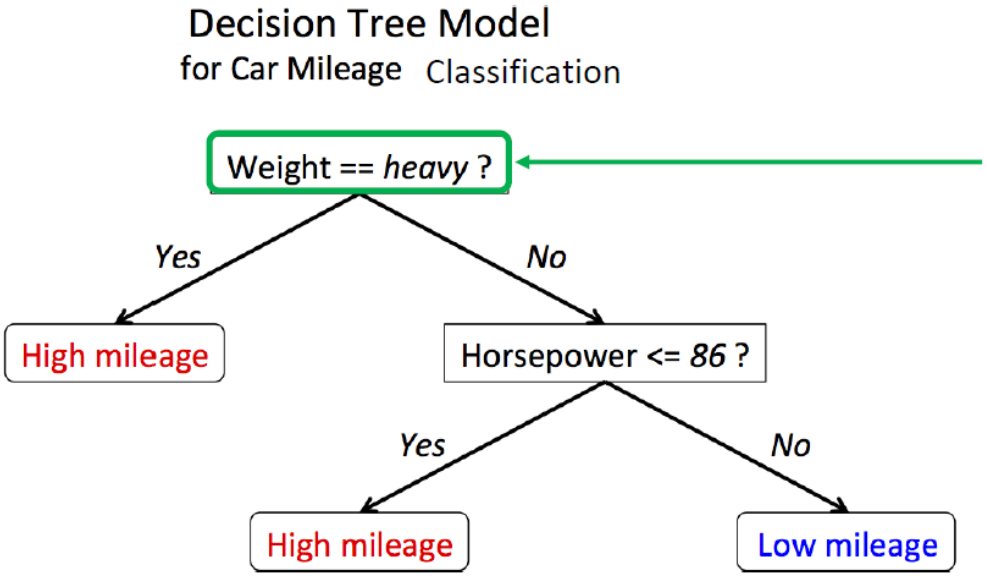
Use cases of Decision Tree

- Decision Tree is often called CART(*Classification & Regression Tree*).
- It is used in both *classification & regression* for predictive modelling.

Will London remain sunny tomorrow?

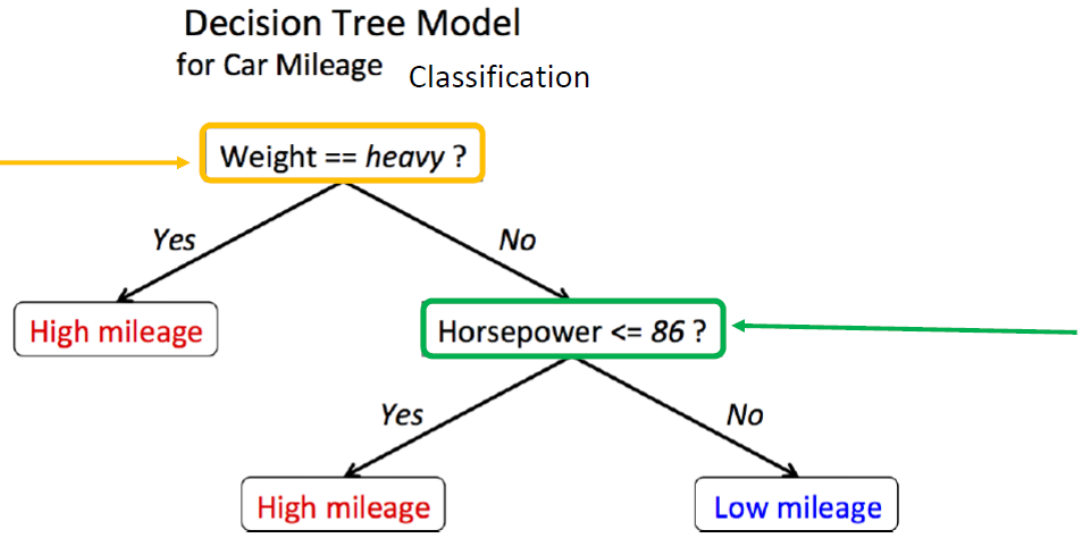
Predict Snowfall(in inch) in Berlin in January.

What is Root node of Decision Tree?



- Root node is first node of building a DT,
- In which only a single attribute is split.

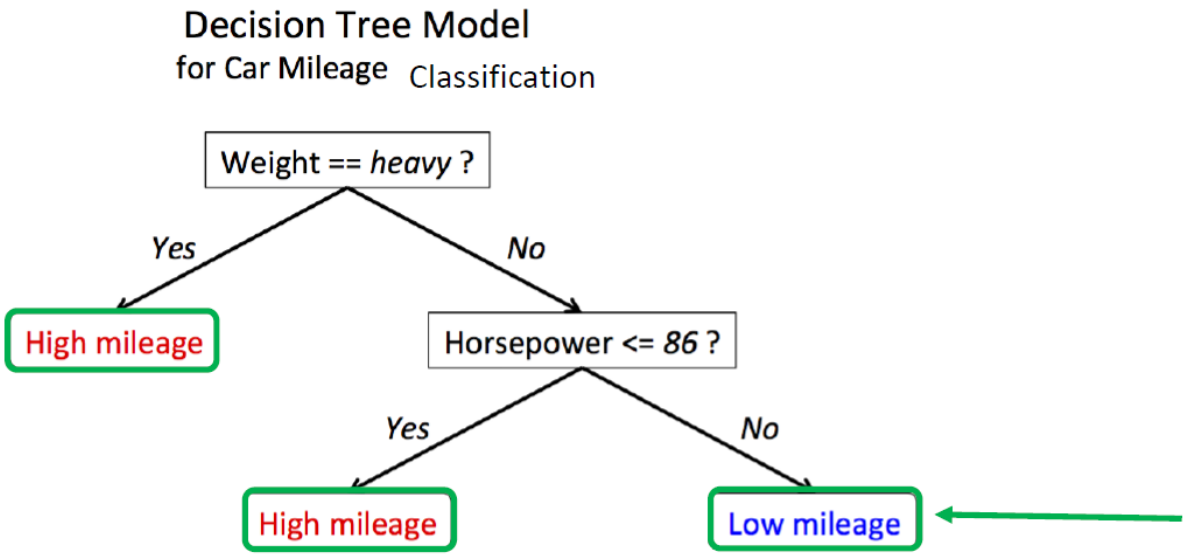
What is decision node of decision tree ?



Decision node is that node of DT that is split into decisions.

N.B: Root node can be decision node if it's split into decisions .

What are leaf nodes of Decision Tree?



- Also called **terminal nodes**.
- Nodes have **predicted decisions** about *target* variable.

CART(Classification and Regression Tree)

- **Clear skies**, bright and shining sun are ideal conditions for playing golf.
- Rain, harsh winds, and a low temperature can all have negative effect on playing golf.

CART(Classification and Regression Tree)

Outlook	Temperature	Humidity	Windy	Hours Played	Plat Golf
Rainy	Hot	High	FALSE	26	No
Rainy	Hot	High	TRUE	30	No
Overcast	Hot	High	FALSE	48	Yes
Sunny	Mild	High	FALSE	46	Yes
Sunny	Cool	Normal	FALSE	62	Yes
Sunny	Cool	Normal	TRUE	23	No
Overcast	Cool	Normal	TRUE	43	Yes
Rainy	Mild	High	FALSE	36	No
Rainy	Cool	Normal	FALSE	38	Yes
Sunny	Mild	Normal	FALSE	48	Yes
Rainy	Mild	Normal	TRUE	48	Yes
Overcast	Mild	High	TRUE	62	Yes
Overcast	Hot	Normal	FALSE	44	Yes
Sunny	Mild	High	TRUE	30	No

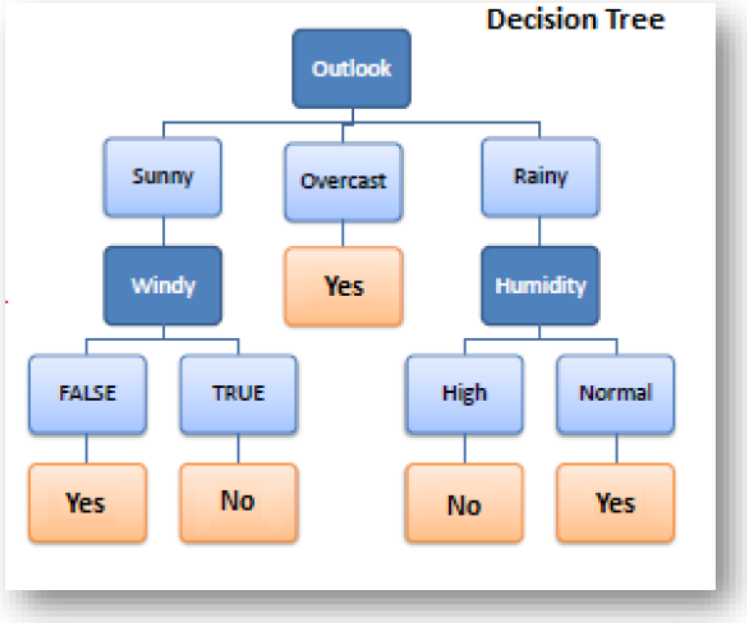
- From dataset we can do predictive analysis regarding whether Golf will be **played** or **not**(Classification problem).
- How many hours (Regression) will golf be played given weather conditions using Decision Tree algorithm.

CART: Classification Tree

Predictors

Target (**Categorical**)

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

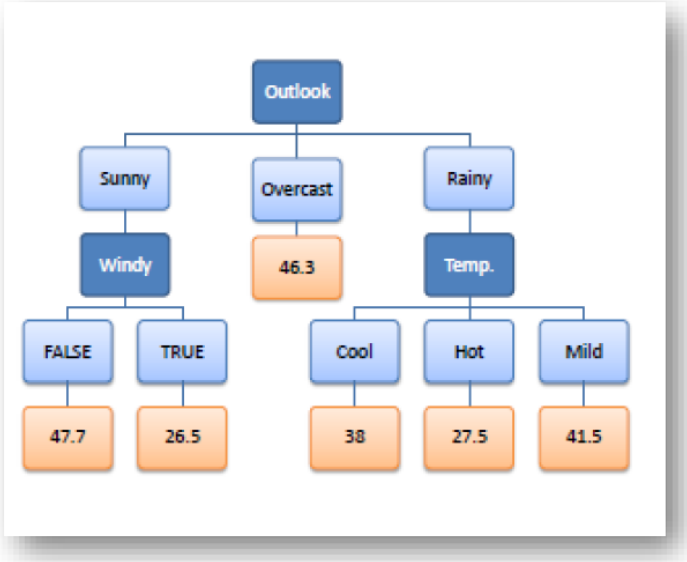


CART: Regression Tree

Predictors

Target (**Continuous**)

Outlook	Temperature	Humidity	Windy	Hours Played
Rainy	Hot	High	FALSE	26
Rainy	Hot	High	TRUE	30
Overcast	Hot	High	FALSE	48
Sunny	Mild	High	FALSE	46
Sunny	Cool	Normal	FALSE	62
Sunny	Cool	Normal	TRUE	23
Overcast	Cool	Normal	TRUE	43
Rainy	Mild	High	FALSE	36
Rainy	Cool	Normal	FALSE	38
Sunny	Mild	Normal	FALSE	48
Rainy	Mild	Normal	TRUE	48
Overcast	Mild	High	TRUE	62
Overcast	Hot	Normal	FALSE	44
Sunny	Mild	High	TRUE	30



How to build Decision Tree?

Let us build a Decision Tree using a very common algorithm namely:

- **ID3(Iterative Dichotomiser 3)**– uses entropy & information *gain* as metrics

ID3(Iterative Dichotomiser 3)

ID3 algorithm was invented by **Ross Quinlan**.

- The basic concept is to build a decision tree by employing **top-down , greedy search** through given data set to **test each attribute(variable) at each tree node.**

Which attribute should we start with?
And
How would we proceed once we get started?

What is Entropy?

Entropy: It is used to measure disorder in the system. If in a particular node, all examples are positive OR all examples are negative (i.e. all examples belong to the same class), then it is homogeneous set of examples and entropy is low.

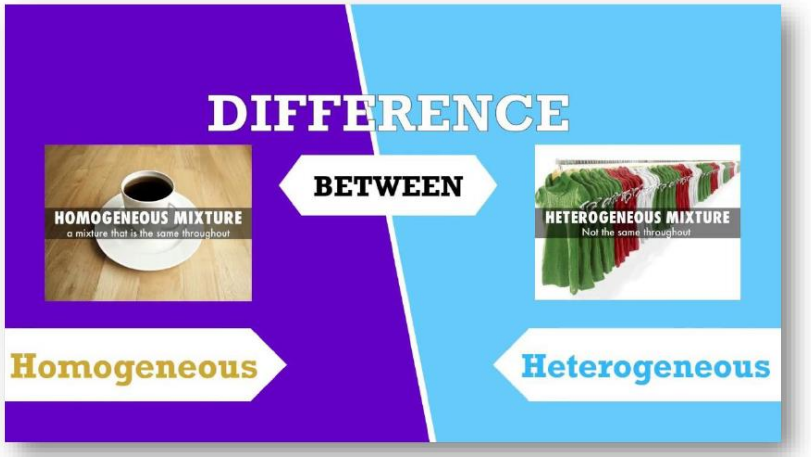
However if we have two classes and half of the examples belong to one class and half belong to another class, then entropy is high

$$Entropy(S) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- **Entropy(S)** is measure of **homogeneity** in data.

Important:

Entropy(S) = - p $\log_2$ p – q $\log_2$ q



Entropy of heterogeneous data

Homogeneous data

- Entropy is **0**.

$$Entropy(S) = - p \log_2 p - q \log_2 q$$

p=1, q= 0(it will be only heads)

$Entropy(S)= - 1 \log_2 (1) - 0 \log_2 0= 0$

Heterogeneous data

- Entropy is **1**.

$$Entropy(S) = - p \log_2 p - q \log_2 q$$

p=0.5, q= 0.5(it must be either heads or tails)

$Entropy(S)= - 0.5 \log_2 (0.5) - 0.5 \log_2(0.5)=1$

Information Gain(IG)

- Information Gain also called Kullback-Leibler divergence(KL divergence) indicates **relative change** in Entropy.
- IG is derived by **subtracting Entropy of a particular attribute (A) from Entropy of total samples(S)**.

Information Gain(S,A)= Entropy(S) – Entropy(S,A)

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Entropy(S)= $-\left(\frac{9}{14}\right) log_2\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) log_2\left(\frac{5}{14}\right) = 0.94$

Entropy(S ,Outlook)= $\frac{5}{14} * 0.971 + \frac{5}{14} * 0.971 + \frac{4}{14} * 0 = 0.693$

Don't worry we'll break up the equation in subsequent slides.

Information Gain = 0.94-0.693

= 0.247

$$Entropy(S, A) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

Calculate Entropy & Information Gain to build a Decision Tree

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

To build a Decision Tree:

- Step 1: Calculate Entropy of set of samples,
- Step 2: Calculate Entropy of each attribute,
- Step 3: Finally calculate information gain.

N.B: Attribute with highest information gain will be root node of Decision Tree

Step 1: Let's calculate Entropy for entire sample

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

The initial step is to calculate Entropy of total set(S).

There are 5 **No** & 9 **Yes** of playing golf

Yes	No	Total
9	5	14
p	q	

Entropy(S) = - p $\log_2$ p – q $\log_2$ q

$$\text{Entropy(S)} = -\left(\frac{9}{14}\right) log_2\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) log_2\left(\frac{5}{14}\right)$$

$$= \mathbf{0.94}$$

Step 2: Calculate Entropy for each column

$$Entropy(S,A)=\sum_{j=1}^{v}\frac{|D_j|}{|D|}\times I(D_j)$$

For demonstration we'll calculate for **Outlook** column

Outlook	Play Golf
Rainy	No
Rainy	No
Overcast	Yes
Sunny	Yes
Sunny	Yes
Sunny	No
Overcast	Yes
Rainy	No
Rainy	Yes
Sunny	Yes
Rainy	Yes
Overcast	Yes

Outlook={ Sunny, Rainy ,Overcast }

$Entropy(S_{Sunny})=-(\frac{3}{5})\log_2(\frac{3}{5})-(\frac{2}{5})\log_2(\frac{2}{5})=0.971$

$Entropy(S_{Rainy})=-(\frac{2}{5})\log_2(\frac{2}{5})-(\frac{3}{5})\log_2(\frac{3}{5})=0.971$

$Entropy(S_{Overcast})=-1\log_2 1-0\log_2 0=0$

$Entropy(S,Outlook)=\frac{5}{14}*0.971+\frac{5}{14}*0.971+\frac{4}{14}*0=0.693$

Step 3: Calculate Information Gain

Outlook	Play Golf
Rainy	No
Rainy	No
Overcast	Yes
Sunny	Yes
Sunny	Yes
Sunny	No
Overcast	Yes
Rainy	No
Rainy	Yes
Sunny	Yes
Rainy	Yes
Overcast	Yes

IG(S ,Outlook) = Entropy(s) - Entropy(S ,Outlook)

IG(S ,Outlook) = 0.94-0.693 = **0.247**

The same way can calculate information gain for other attributes.

Information Gain from all attributes

		Yes	No
Outlook	Sunny	3	2
	Rainy	2	3
	Overcast	4	0

Information Gain= **0.247**

		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

Information Gain= 0.029

		Yes	No
Humidity	High	3	4
	Normal	6	1

Information Gain= 0.152

		Yes	No
windy	FALSE	6	2
	TRUE	3	3

Information Gain= 0.048

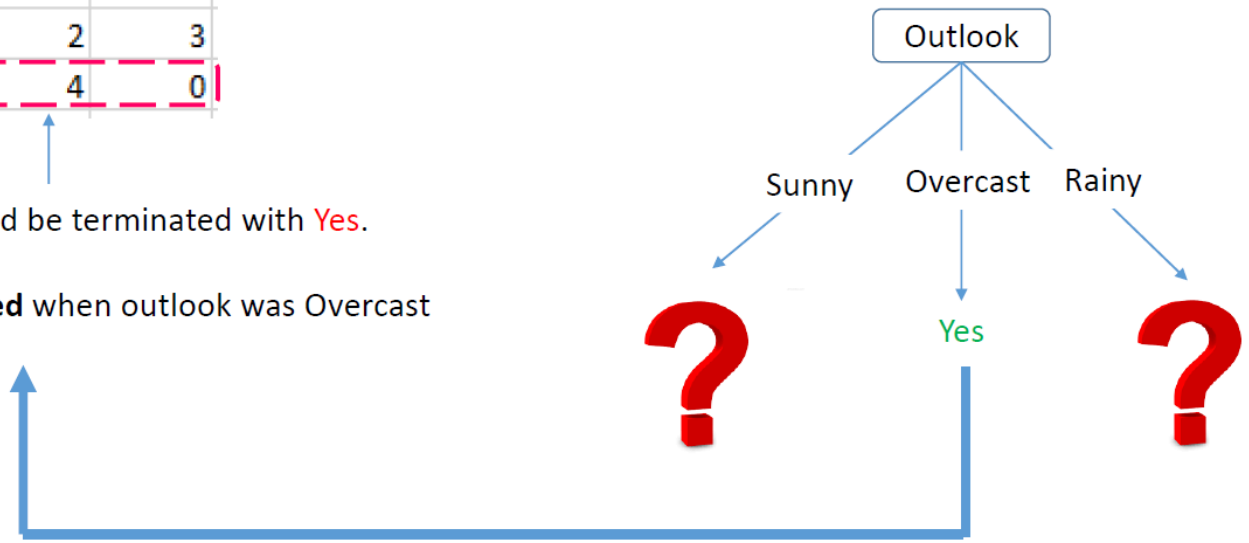
We'll use information gain to decide attribute node of tree

How does the tree look initially

As per information gain, Outlook will be root node because we've highest information gain from it.

		Yes	No
Outlook	Sunny	3	2
	Rainy	2	3
	Overcast	4	0

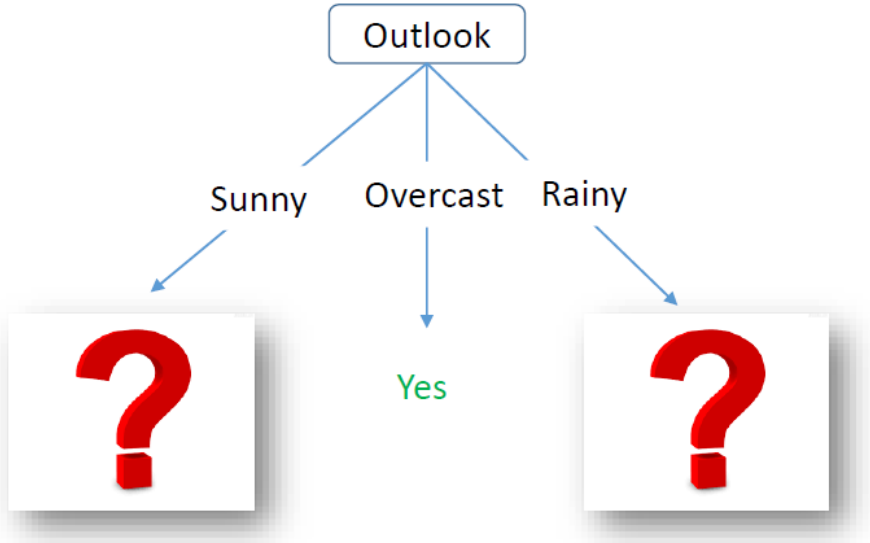
- **Overcast** outlook should be terminated with Yes.
- Because Golf was **played** when outlook was Overcast



Build Decision Tree –But what next?

- But question is which attribute should be tested at Sunny & Rainy branches.

Humidity, Temperature Or Windy?



Build Decision Tree –next is here

If we find Information Gain from Humidity, Temperature & Wind when outlook is Sunny.

Information Gain(S_{sunny}, Humidity) = 0.97- (3/5)*0.0 –(2/5)*0.0 = **0.97**

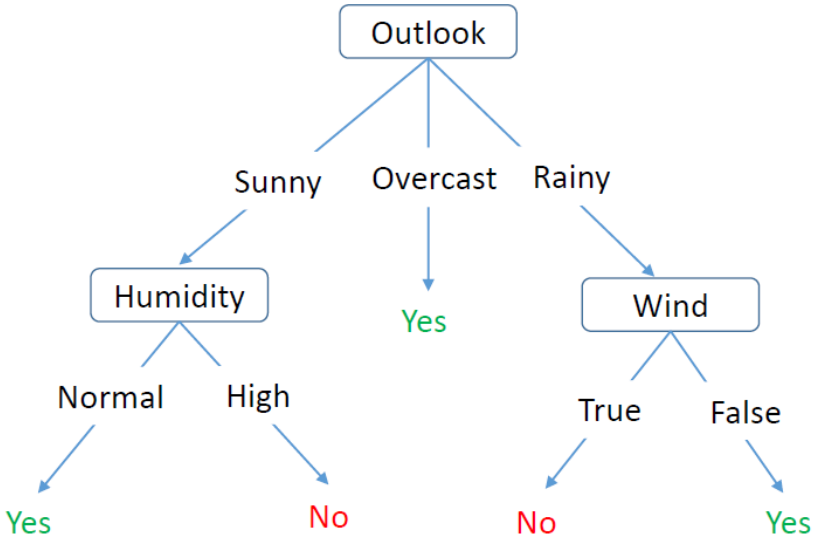
Information Gain(S_{sunny}, Temperature) = 0.97- (2/5)*0.0 - (2/5)*1.0 - (1/5)*0.0 = 0.57

Information Gain(S_{sunny}, Wind) = 0.97 - (2/5)*1.0 - (3/5)*0.918 = 0.019

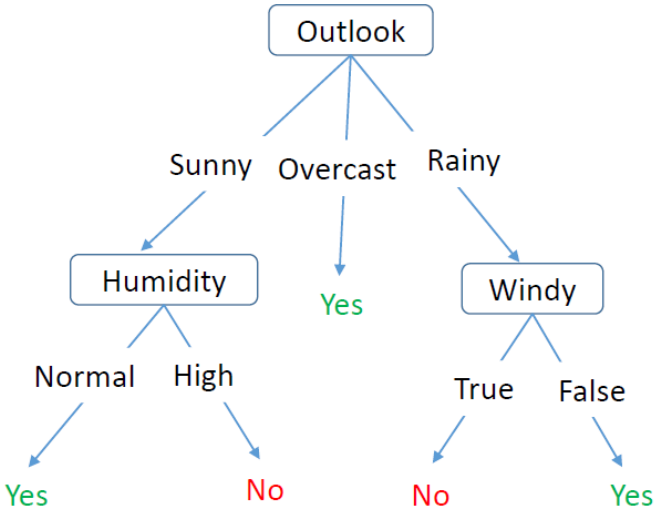
As per above information Gain, **Humidity** should be tested at Sunny branch.

How does the tree look finally?

- Proceeding with same way, we get Windy as immediate decision node for Rainy.
- The process of splitting continues until all data is classified .

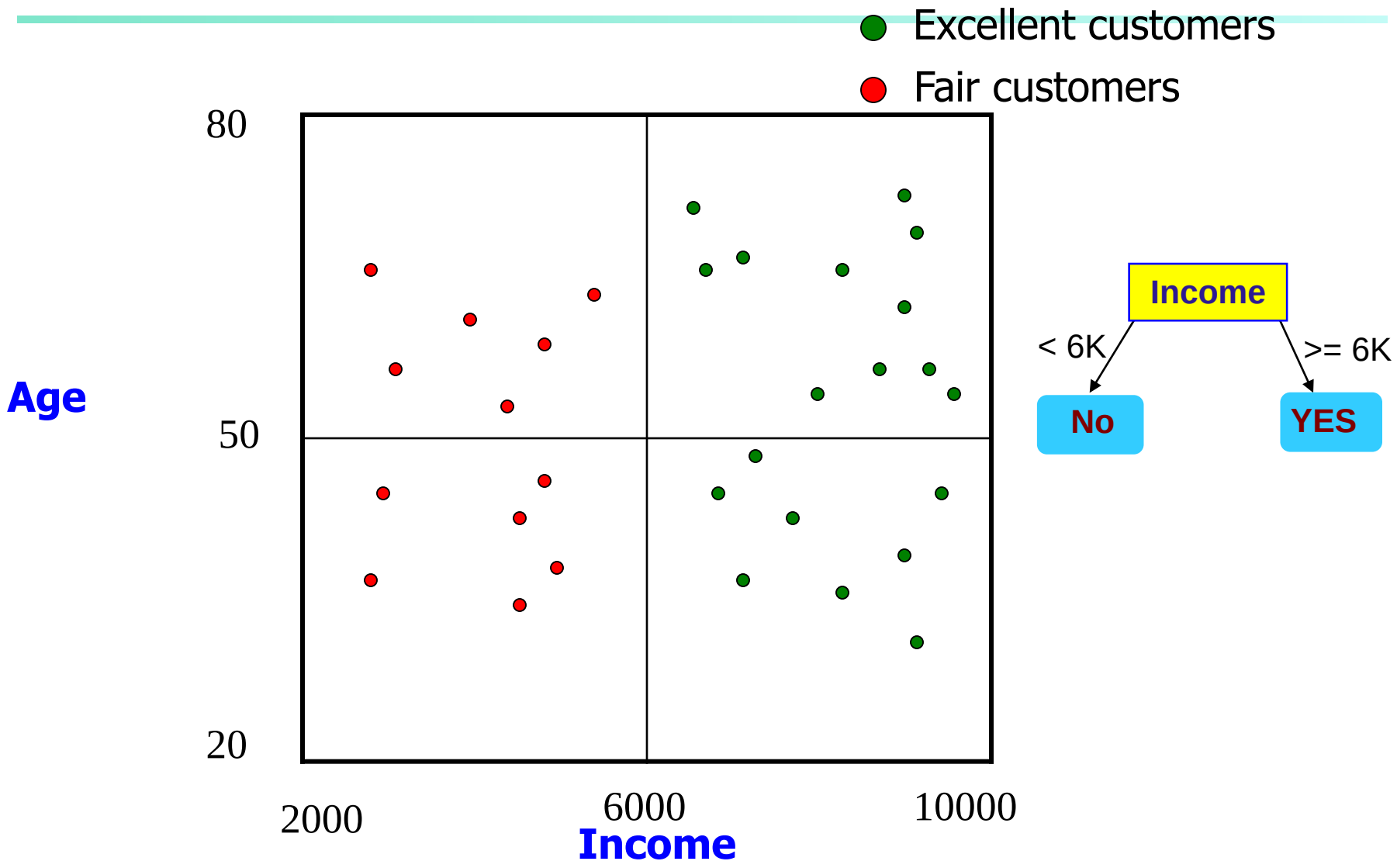


Decision rules

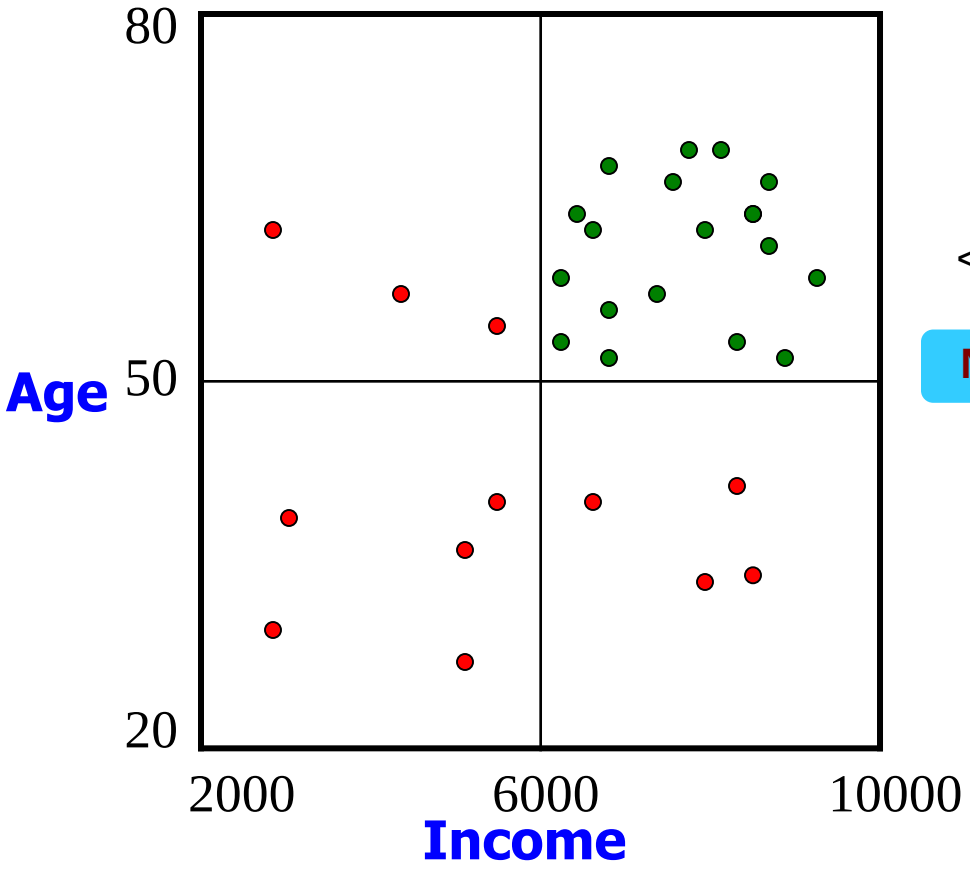


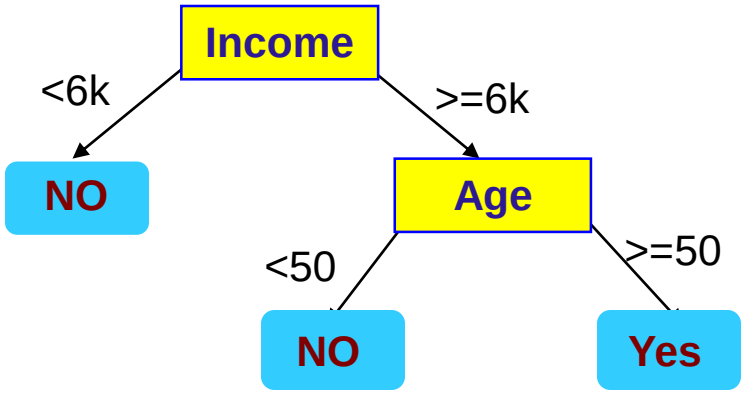
- IF outlook = sunny AND humidity = high THEN play golf = No
- IF outlook = sunny AND humidity = Normal THEN play golf =Yes
- IF outlook = rainy AND windy = True THEN play golf = No
- IF outlook = overcast THEN play golf = Yes
- IF outlook = rain AND wind = False THEN play golf = Yes

Sample Decision Tree



Sample Decision Tree





Decision-Tree Classification Methods

- The basic top-down decision tree generation approach usually consists of two phases:

1. **Tree construction**
   - At the start, all the training examples are at the root.
   - Partition examples are recursively based on selected attributes.
2. **Tree pruning**
   - Aiming at removing tree branches that may reflect noise in the training data and lead to errors when classifying test data → improve classification accuracy

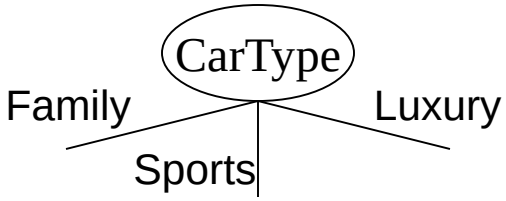
How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous

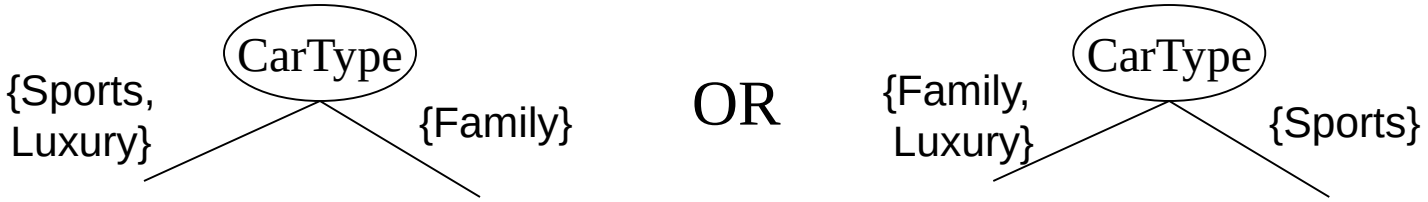
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

- Multi-way split: Use as many partitions as distinct values.

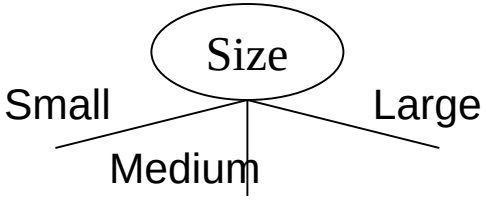


- Binary split: Divides values into two subsets.
  Need to find optimal partitioning.

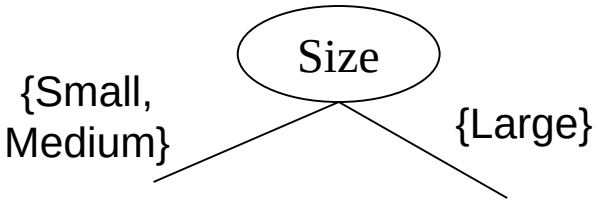


Splitting Based on Ordinal Attributes

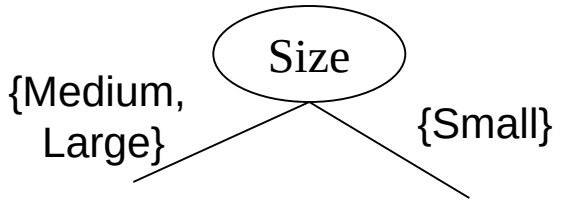
- Multi-way split: Use as many partitions as distinct values.

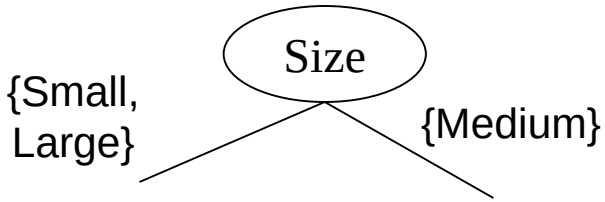


- Binary split: Divides values into two subsets.
  Need to find optimal partitioning.



OR

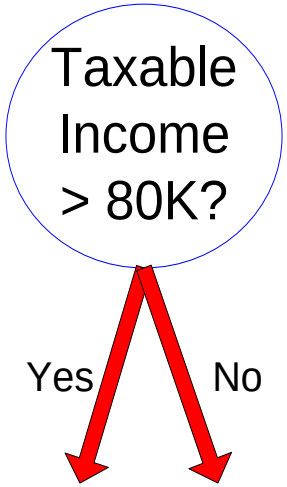




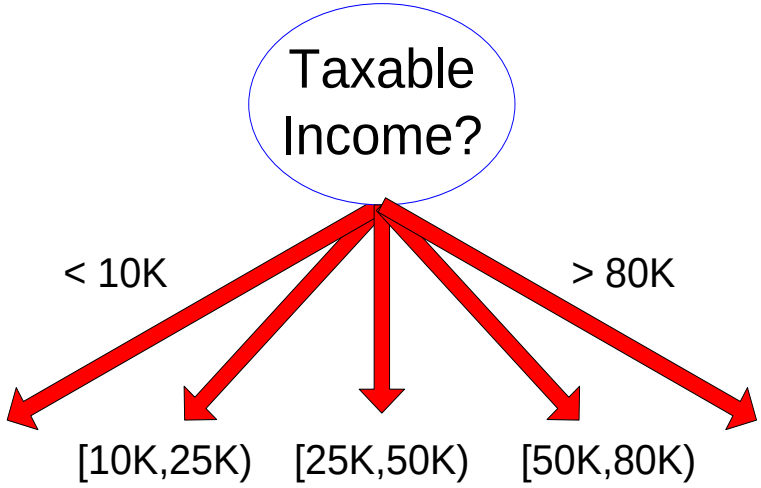
Splitting Based on Continuous Attributes

- Different ways of handling
  - Discretization to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
  - Binary Decision: $(A < v)$ or $(A \geq v)$
    - consider all possible splits and finds the best cut

Splitting Based on Continuous Attributes



(i) Binary split



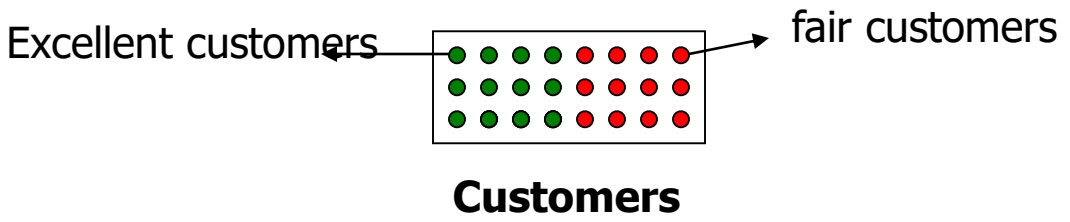
(ii) Multi-way split

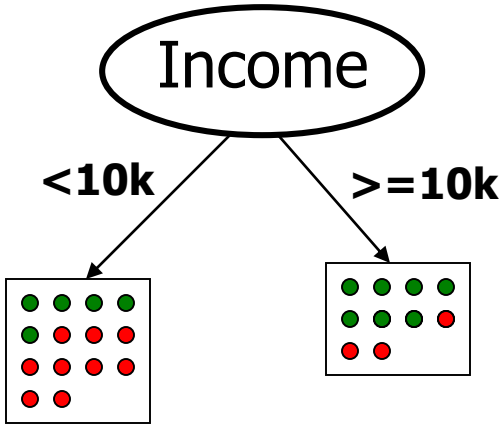
Tree Induction

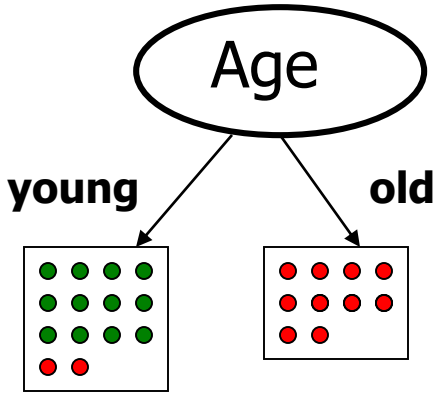
- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

How to determine the Best Split







Algorithm for Decision Tree Induction

- Basic algorithm
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - There are no remaining attributes for further partitioning
  - There are no samples left

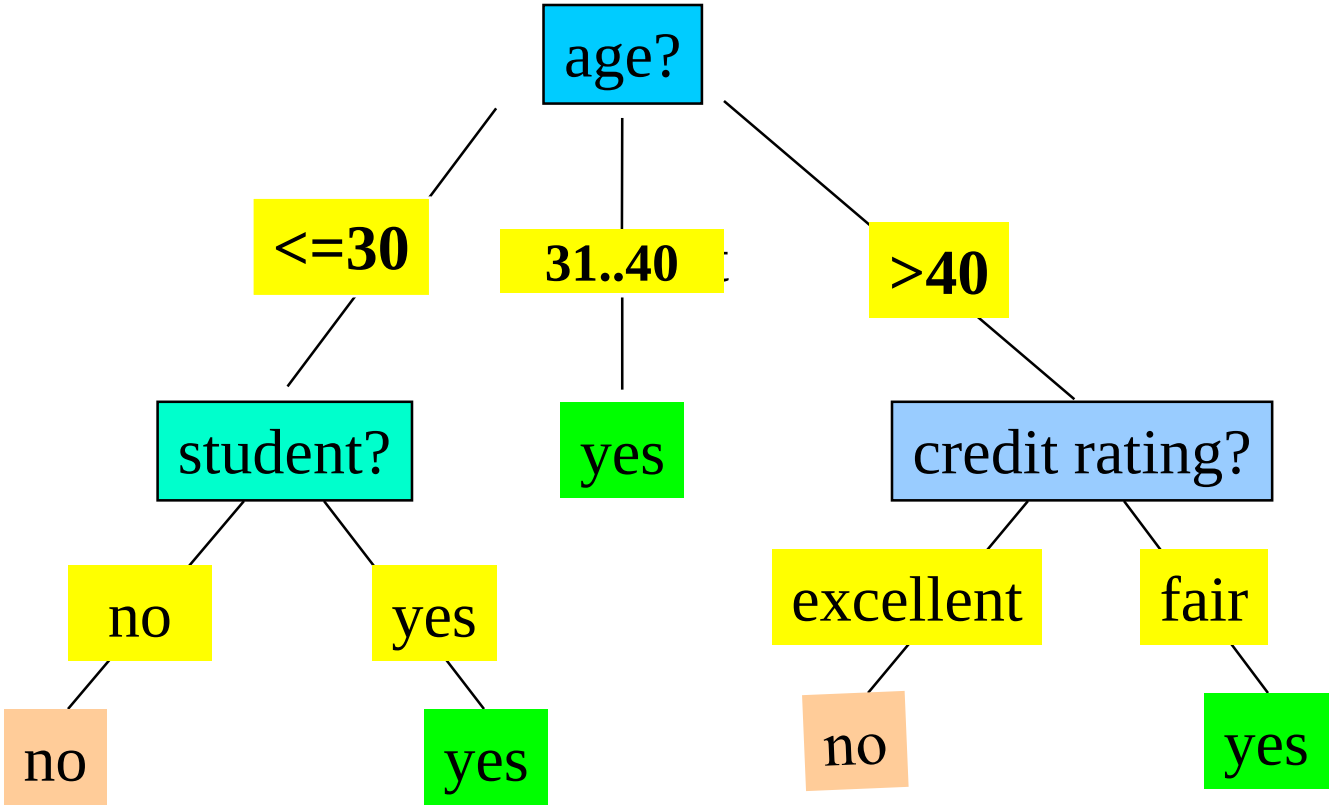
Classification Algorithms

- ID3
  - Uses information gain
- C4.5
  - Uses Gain Ratio

Decision Tree Induction: Training Dataset

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Output: A Decision Tree for "*buys_computer*"



ID3
Attribute Selection Measure: Information Gain

- Notations:
  - Let D, the data partition, be a training set of class-labeled tuples.
  - Suppose the class label attribute has m distinct values defining m distinct classes, Ci (for $i = 1, \ldots, m$).
  - Let $C_{i,D}$ be the set of tuples of class Ci in D. Let $|D|$ and $|C_{i,D}|$ denote the number of tuples in D and $C_{i,D}$, respectively.

Attribute Selection Measure: Information Gain

- Select the attribute with the highest information gain for current node
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information needed to classify a given tuple in D:
  (log function base 2 is used since the info. is encoded in bits.)

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- *Info (D)* is just the average amount of information needed to identify the class label of a tuple in *D*. *Info (D)* is also known as the entropy of *D*.
- Now, suppose we were to partition the tuples in *D* on some attribute *A* having *v* distinct values, (*a*1, *a*2, …. , *av*), as observed from the training data. If *A* is discrete-valued, then it gives the v outcomes of a test on *A*. Attribute *A* can be used to split *D* into *v* partitions or subsets, (*D*1, *D*2, ….., *Dv).*

Attribute Selection Measure: Information Gain

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

- The term $|Dj| / |D|$ acts as the weight of the *j* th partition. *Info A*(*D*) is the expected information required to classify a tuple from *D* based on the partitioning by *A*.
- Information gained by branching on attribute A

Information gain = entropy (parent) – [weightes average] * entropy (children)

$$Gain(A) = Info(D) - Info_A(D)$$

Attribute Selection: Information Gain

- In training data set, The class level attribute, *buys_computer,* has two distinct values (namely, {yes,no}); therefore, there are two distinct classes (m=2).

- Let class P correspond to *yes* and N correspond to *no.*
- there are 9 samples of class yes and 5 samples of class no.
- To compute the information gain of each attribute, we first use Equation 1, to compute the expected information needed to classify a given sample.

Attribute Selection: Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$\frac{5}{14} I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

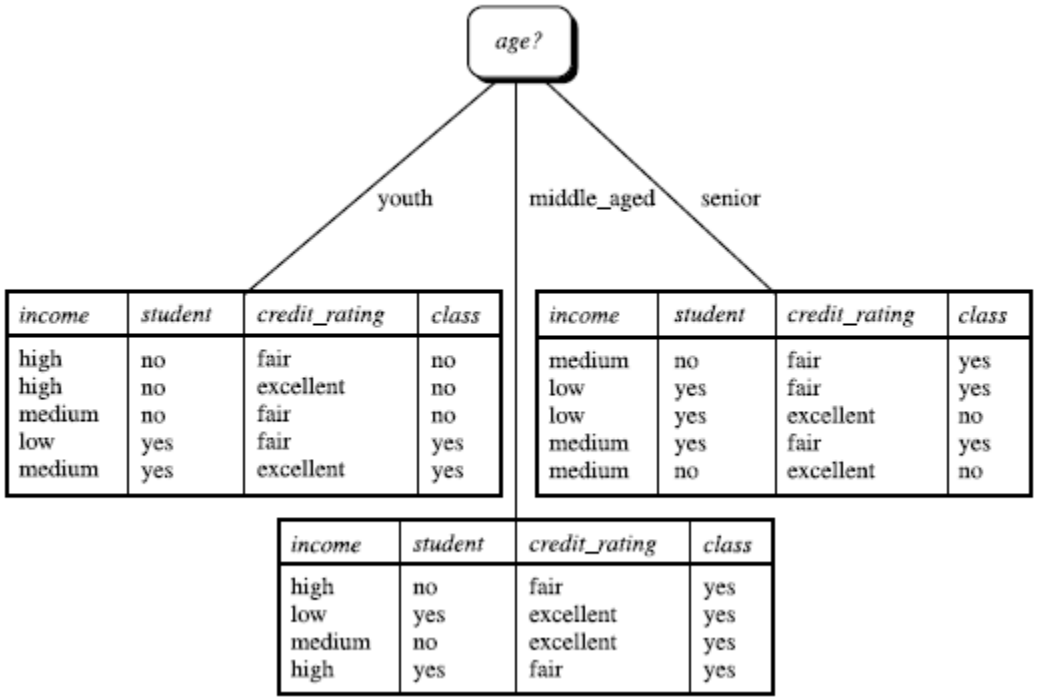
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

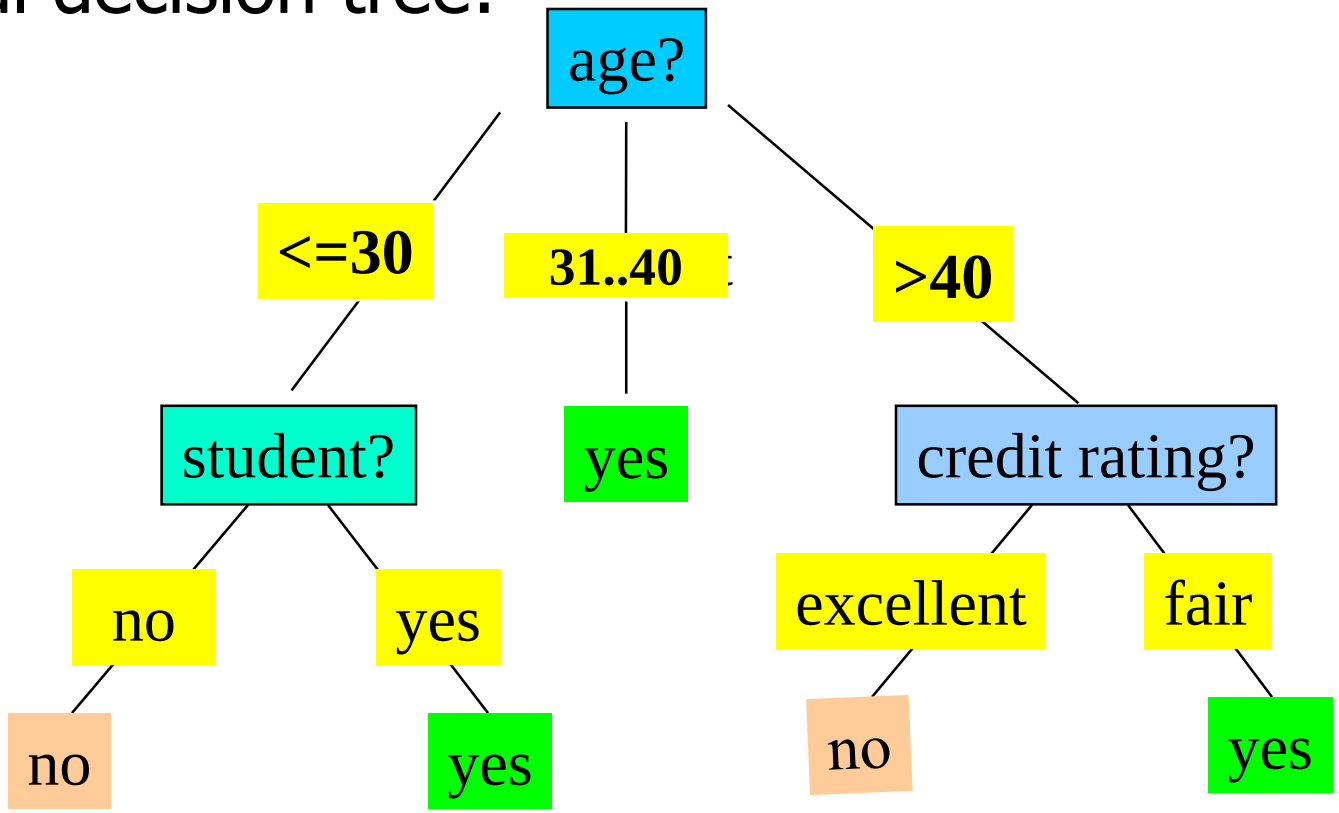
Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



- Final decision tree:



Why are decision tree classifiers so popular?

- The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for knowledge discovery.
- Decision trees can handle high dimensional data.
- Representation of acquired knowledge in tree form is generally easy to humans.
- The learning and classification steps of decision tree induction are simple and fast.
- In general, decision tree classifiers have good accuracy. However, successful use may depend on the data at hand.
- Decision tree induction algorithms have been used for classification in many application areas, such as medicine, manufacturing and production, financial analysis, and molecular biology.
- Decision trees are the basis of several commercial rule induction systems.

Gain Ratio for Attribute Selection

- The information gain measure is biased toward tests with many outcomes. That is, it prefers to select attributes having a large number of values.

- For example, consider an attribute that acts as a unique identifier, such as *product ID*.

- A split on *product ID* would result in a large number of partitions (as many as there are values), each one containing just one tuple.

- Because each partition is pure, the information required to classify data set *D* based on this partitioning would be $Info_{product\ ID}(D) = 0$.

- Therefore, the information gained by partitioning on this attribute is maximal. Clearly, such a partitioning is useless for classification.

Gain Ratio for Attribute Selection

- C4.5, a successor of ID3, uses an extension to information gain known as *gain ratio.*
- (normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

  - GainRatio(A) = Gain(A)/SplitInfo(A)

(A test on income splits the data into three partitions, namely low, medium & high containing four,six & four tuples)

- Ex.

$$SplitInfo_A(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 0.926$$

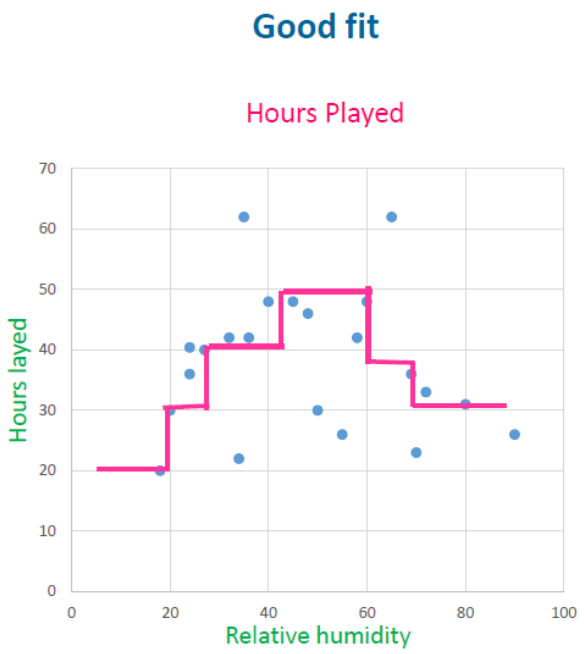
  - gain_ratio(income) = 0.029/0.926 = 0.031
- The attribute with the maximum gain ratio is selected as the splitting attribute

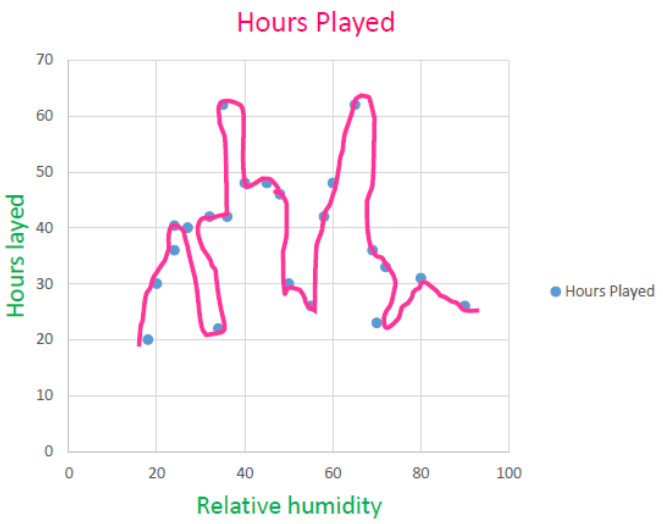
Comparing Attribute Selection Measures

- Information gain:
  - biased towards multi valued attributes
- Gain ratio:
  - tends to prefer unbalanced splits in which one partition is much smaller than the others

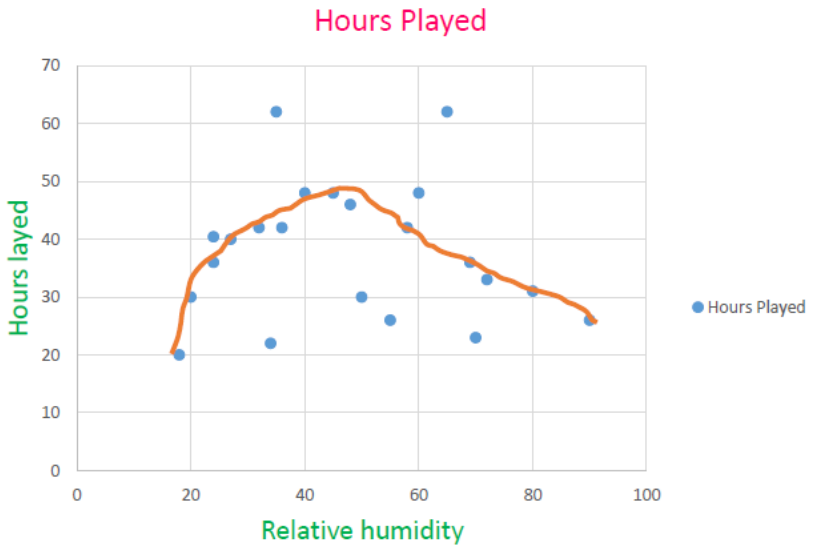
Challenge with Decision Tree models

Relative Humidity	Hours Played
55	26
50	30
60	48
58	42
65	62
70	23
48	46
69	36
72	33
45	48
40	48
35	62
80	31
90	26
27	40
36	42
24	40.4
32	42
24	36
34	22
18	20
20	30





Random Forest helps overcome this challenge



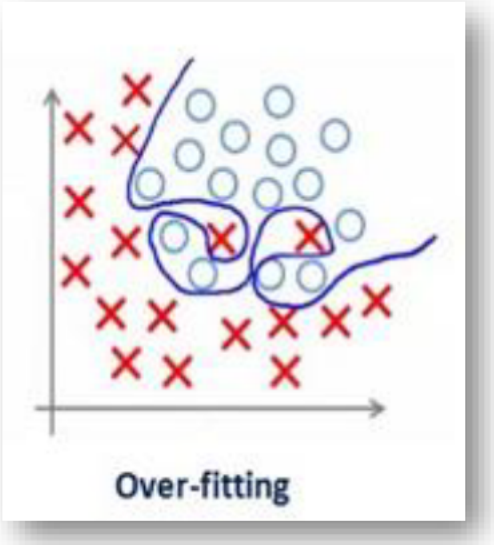
Random Forest algorithm can fit data points in such a smooth way that it neither overfits nor underfits model

Let's know what overfitting is How

How different will predictions of model be on unseen data.

Challenge with **over-fitting** model is that

- Model performs perfectly well(Overfitting) on **training** data
- But when same model is used for prediction on test data, it results in huge difference in prediction accuracy from that of training data.

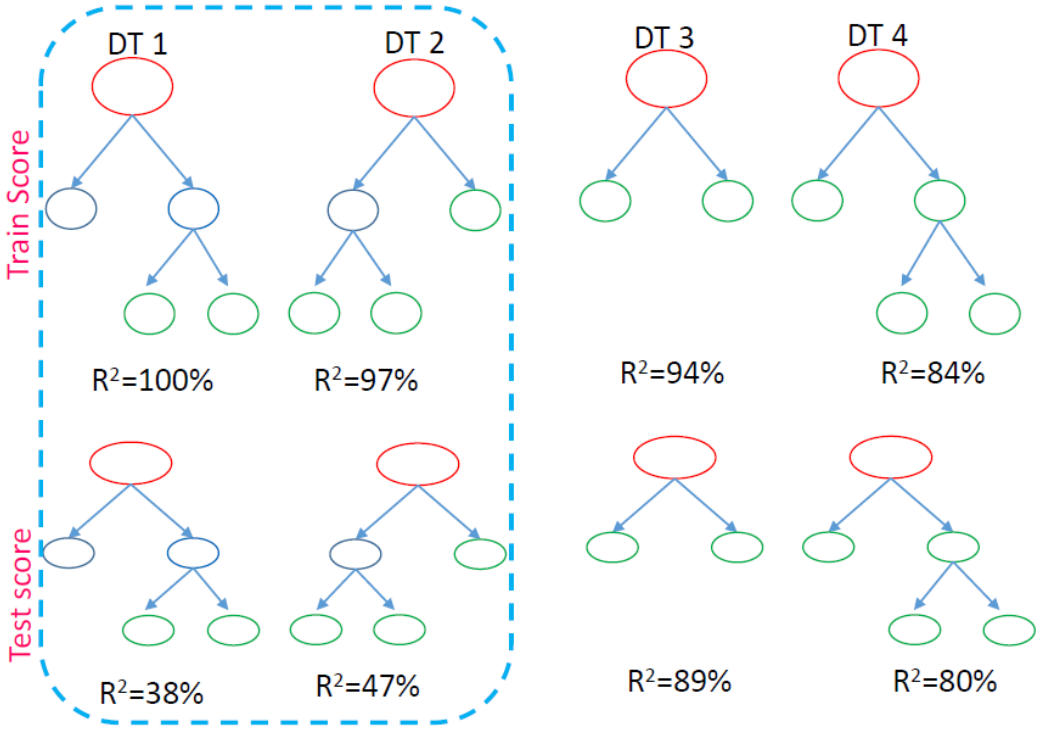


High variance

How overfitting causes challenge in Decision Tree

Random Forest to the rescue

Major advantage of Random Forest algorithm is to handle overfitting efficiently & outputs **overall better** result.



- A major drawback of *Decision tree* algorithm is
  - Decision Tree 1 & 2 have high variance(Overfitting)—performs relatively poor on test data.
- Random Forest aggregates all the trees and calculate **average score**.
- More like **averaging** out errors

R^2 is **a measure of the goodness of fit of a model**.

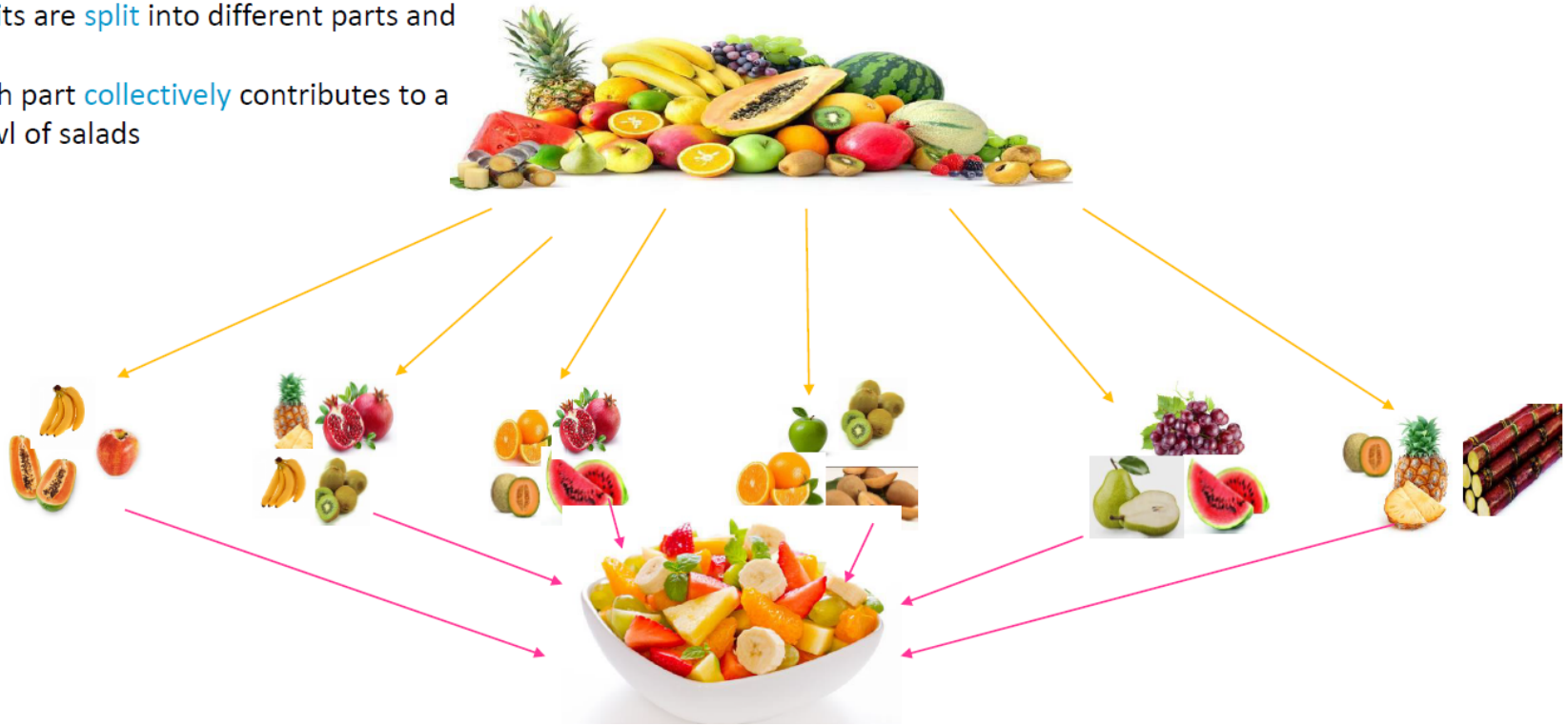
Random Forest

- Random forest is collection of many decision trees.
- Like forest is collection of many trees.

Random Forest algorithm is **example** of **bagging** technique.

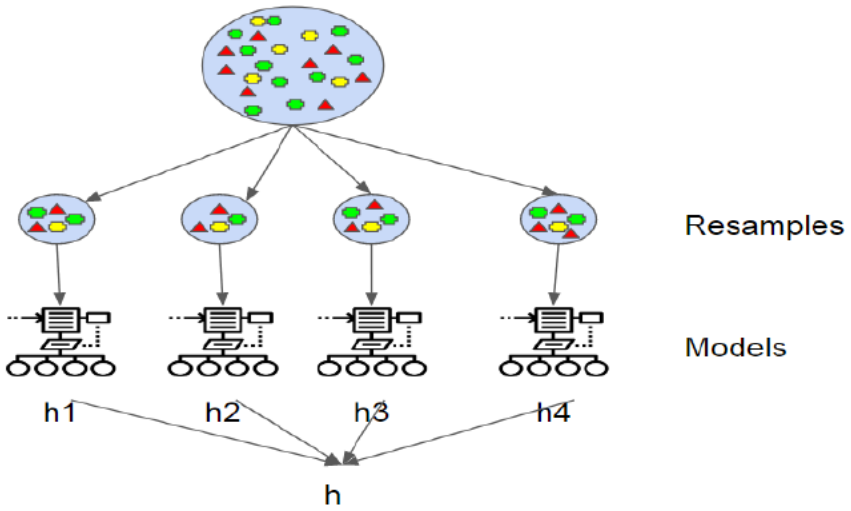
What is Bagging?

- Fruits are split into different parts and
- Each part collectively contributes to a bowl of salads

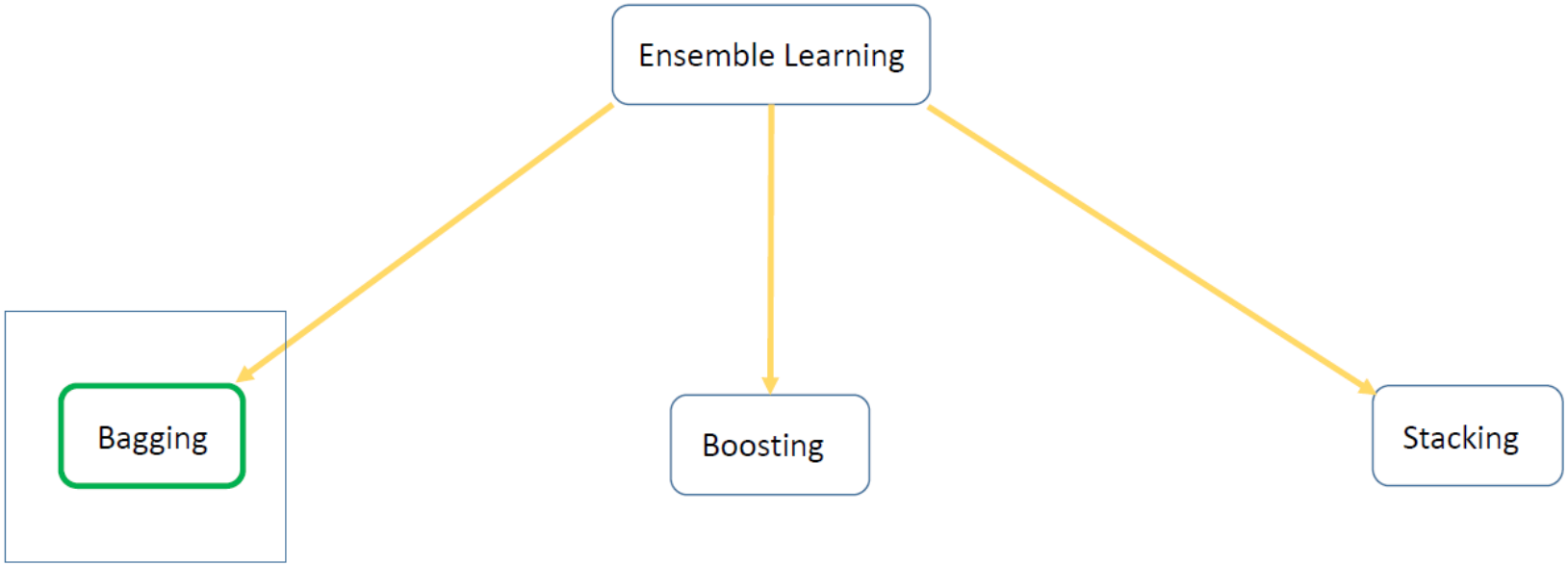


What is Bagging?

- Like fruits ,data are **split** into multiple **smaller** parts to make **multiple models**.
- Finally results from each model are **combined**.
  - **Majority** voting for classification
  - **Averaging** for regression

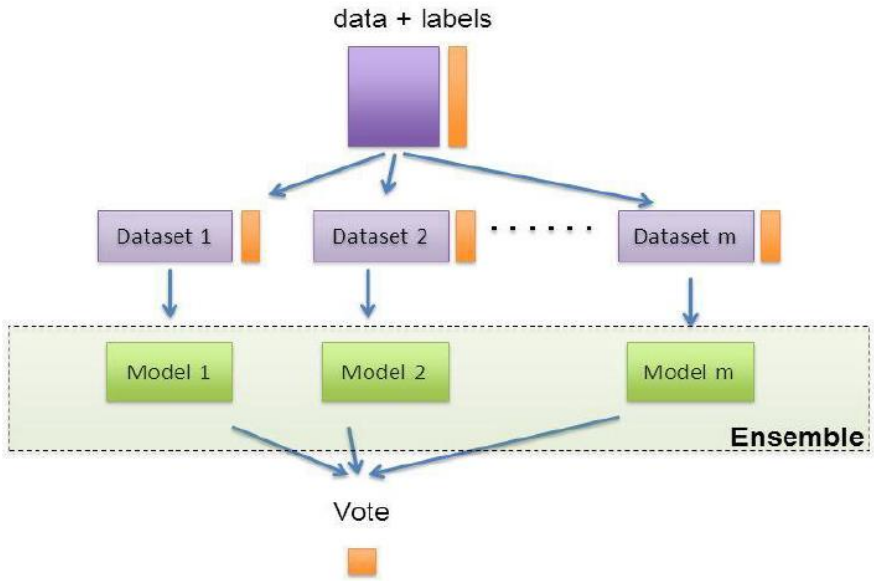


Types of Ensemble learning



What is ensemble learning?

Ensemble learning is technique that **creates** *multiple models* and then combines them to produce improved results



Ensemble

Group of individuals or items viewed as **whole** rather than individually.

Why Random Forest is called Random?

There are two types of randomness in RF algorithm—

1. Row level
2. Column level

Row level randomness in Random Forest

Let's say the data set has 1000 rows.

Row Level

- Each of decision trees will be made from **random sample**(For ex, 5%) of training data.
- It indicates that each decision tree will be made of **any 50 rows** of given data.

Passenger	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, M	male	22	1	0	A/5 2117	7.25		S
2	1	1	Cumings, I	female	38	1	0	PC 17599	71.28	C85	C
3	1	3	Heikkinen	female	26	0	0	STON/O2	7.925		S
4	1	1	Futrelle, M	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr.	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr	male		0	0	330877	8.458		Q
7	0	1	McCarthy,	male	54	0	0	17463	51.86	E46	S
8	0	3	Palsson, M	male	2	3	1	349909	21.08		S
9	1	3	Johnson, M	female	27	0	2	347742	11.13		S
10	1	2	Nasser, Mr	female	14	1	0	237736	30.07		C
11	1	3	Sandstrom	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, M	female	58	0	0	113783	26.55	C103	S
13	0	3	Saundercc	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Anderssor	male	39	1	5	347082	31.28		S
15	0	3	Vestrom, M	female	14	0	0	350406	7.854		S
16	1	2	Hewlett, M	female	55	0	0	248706	16		S
17	0	3	Rice, Mast	male	2	4	1	382652	29.13		Q
18	1	2	Williams,	male		0	0	244373	13		S
19	0	3	Vander Pl	female	31	1	0	345763	18		S

Column level randomness in Random Forest

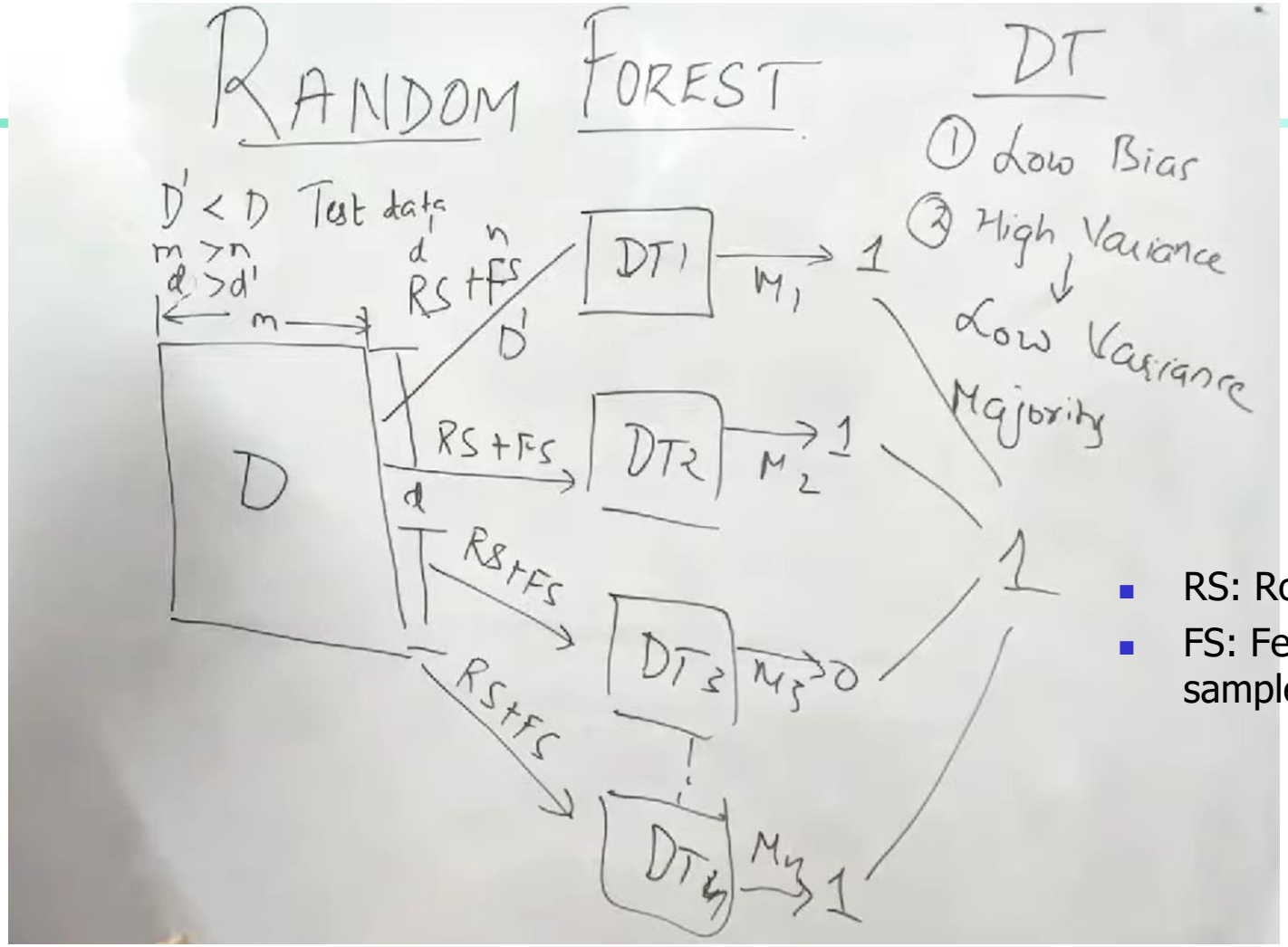
Let's say the data set has 50 columns.

Column Level

- At this level of randomness **not all the columns** are passed into training each decision tree,
- rather specified number(For ex, 10%) of columns i.e. 5 randomly chosen columns will be passed into each decision tree.

Passenger	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, M	male	22	1	0	A/5 2117	7.25		S
2	1	1	Cumings, I	female	38	1	0	PC 17599	71.28	C85	C
3	1	3	Heikkinen	female	26	0	0	STON/O2	7.925		S
4	1	1	Futrelle, M	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr.	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr	male		0	0	330877	8.458		Q
7	0	1	McCarthy,	male	54	0	0	17463	51.86	E46	S
8	0	3	Palsson, M	male	2	3	1	349909	21.08		S
9	1	3	Johnson, M	female	27	0	2	347742	11.13		S
10	1	2	Nasser, Mr	female	14	1	0	237736	30.07		C
11	1	3	Sandstrom	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, M	female	58	0	0	113783	26.55	C103	S
13	0	3	Saundercc	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Anderssor	male	39	1	5	347082	31.28		S
15	0	3	Vestrom, I	female	14	0	0	350406	7.854		S
16	1	2	Hewlett, M	female	55	0	0	248706	16		S
17	0	3	Rice, Mast	male	2	4	1	382652	29.13		Q
18	1	2	Williams,	male		0	0	244373	13		S
19	0	3	Vander Pl	female	31	1	0	345763	18		S

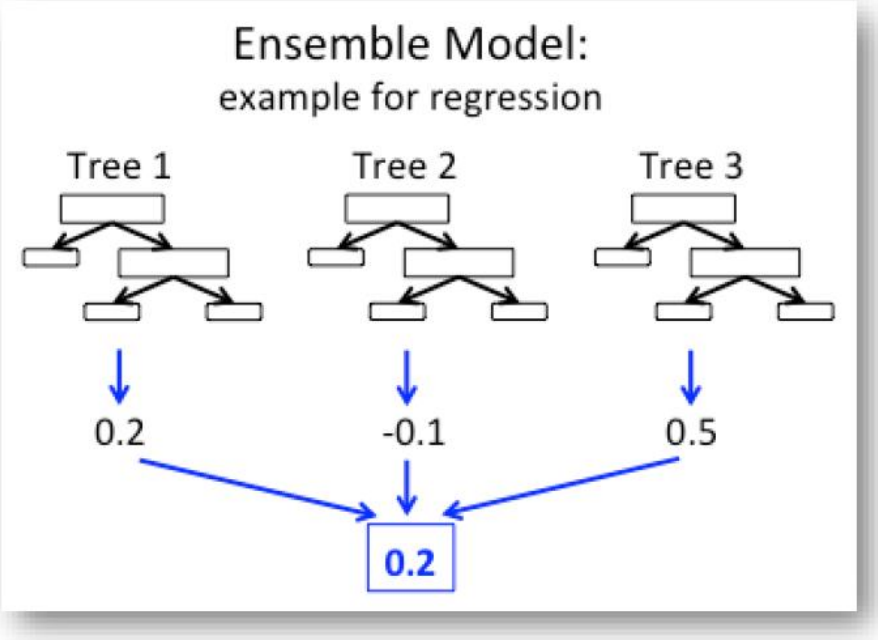
Example



- Low Bias: Basically it says that if I am creating my decision tree to its complete depth, then it will get properly trained for training dataset. So training error will be very less.
- High Variance: Whenever we get new test data, the decision tree is prone to give larger amount of error.

How does Random Forest work in Regression?

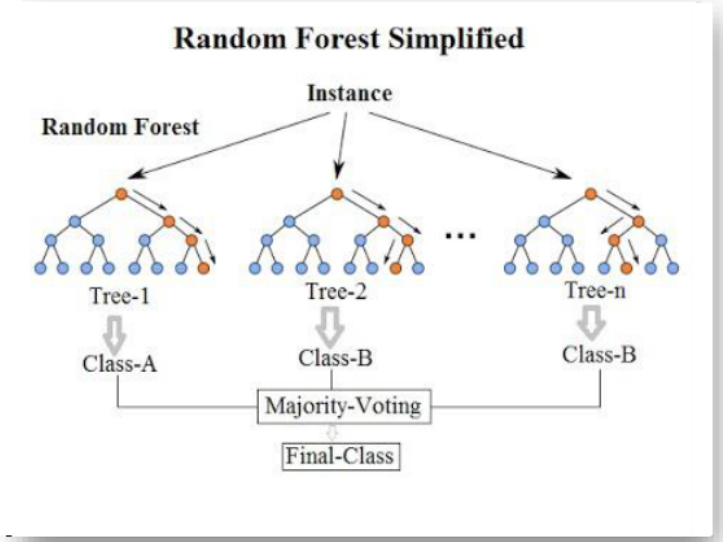
Random Forest Regression



- In Random Forest regression all values of decision trees are averaged.
- The average value is actual result for regression.

How does Random Forest work in Classification?

Random Forest classification



- To classify a new object based on attributes each tree gives classification, often termed as **tree votes** for that class.
- A class is chosen based on **majority** of voting.
- For ex, in picture left if major class is B then it'll be predicted as B.

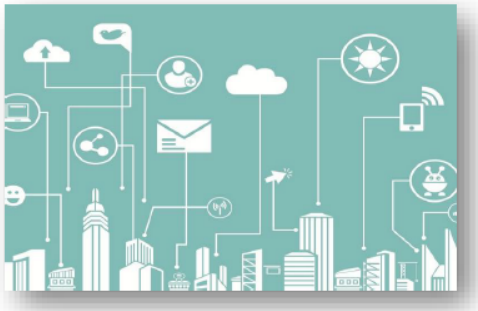
Benefits of Random Forest

Random Forest is one of the most powerful Machine Learning algorithms.

Accuracy

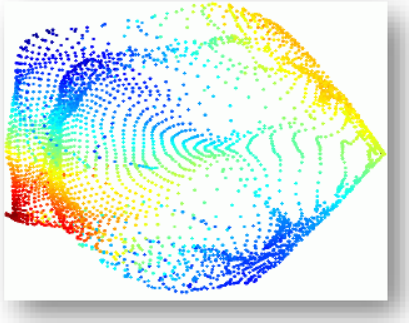
It provides excellent accuracy both for classification & regression.

Handling large scale data



Efficiently handles large scale data making it scalable based on increasing data size

Feature Selection



Takes only important features without deleting unwanted ones.

Use cases of Random Forest

Healthcare

- In healthcare domain it is used **to identify correct combination** of components in medicine
- Analyzes a patient's medical history to **identify** diseases

Finance

- In finance it **predicts prospect defaulters.**
- Used to detect **unusual transaction** in anomaly detection

Naïve Bayes classifier