**Data Mining – Supervised**

**Decision Tree & Random Forest**

Machine Learning is ubiquitous

ML is ubiquitous

* Medical Diagnosis: Identify new disorders for observations.
* Loan Applications: Predict Risk of default
* Speech/Object Recognition:

From examples, generalize to others.

* Prediction: (climate, stocks, etc.)

Predict future from current and past data.

What is learning?

* More than just memorizing facts.
* Learning the underlying structure of the problem or data.

A fundamental aspect of learning is generalize:

* Give few examples, can you generalize to others?

What are Decision Trees?

A Decision Tree is a tree like structure in which internal node represent test on an attribute, each branch represents outcome of test and each leaf node represents class label (decision taken after computing all attributes).

A path from root to leaf node represents classification rules.

A decision tree consists of 3 types of nodes:

Types of nodes:

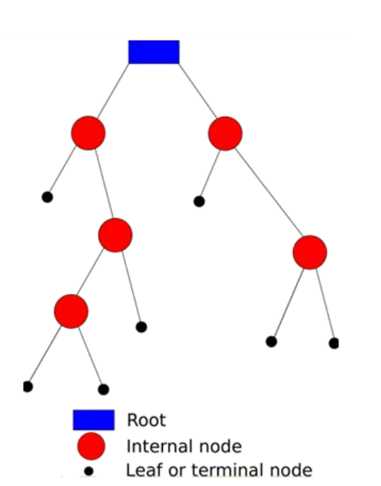
* Root node
* Branch node
* Leaf node

How to build a decision tree?

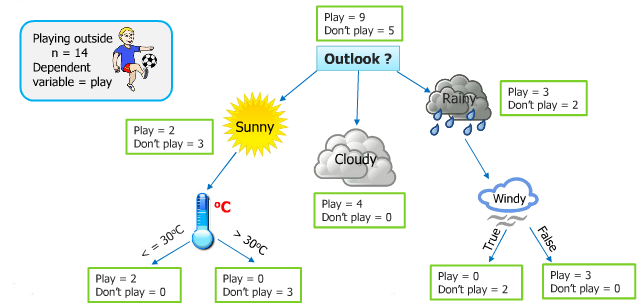
Uses training data to build model.

Tree generator determines:

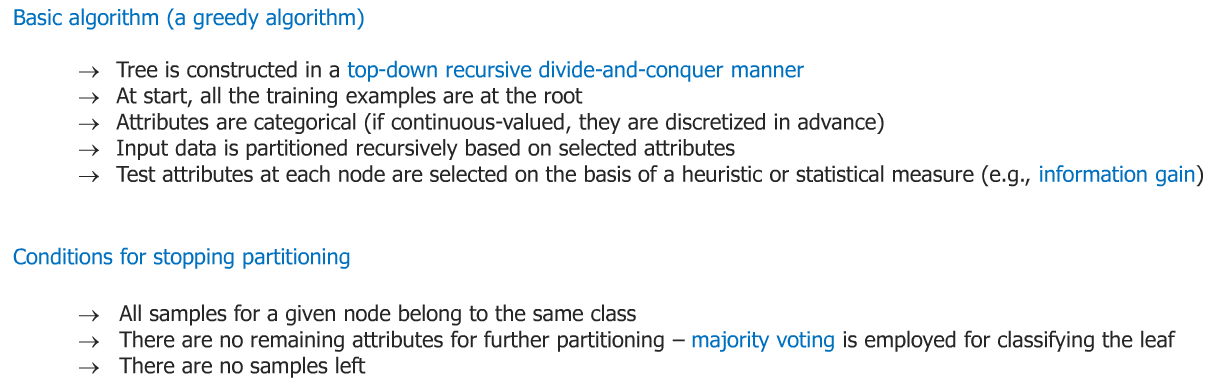
* Which variable to split at a node and what will be the value of the split.
* Decision to stop (make a terminal note) or split again has to be made.
* Assign terminal nodes to a label.



Decision trees – Example



Algorithm for Decision Tree Induction



Basic Algorithm (a greedy 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)
* Input data is partitioned recursively based on selected attributes.
* Test attributes at each node are selected on the basis of heuristic or statistical measure (e.g., information gain)

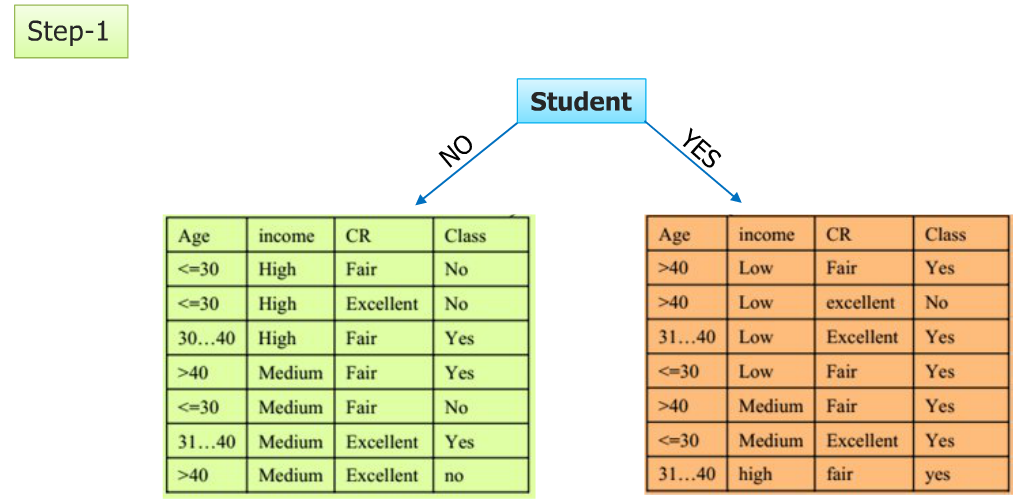
Conditions for stopping partitioning:

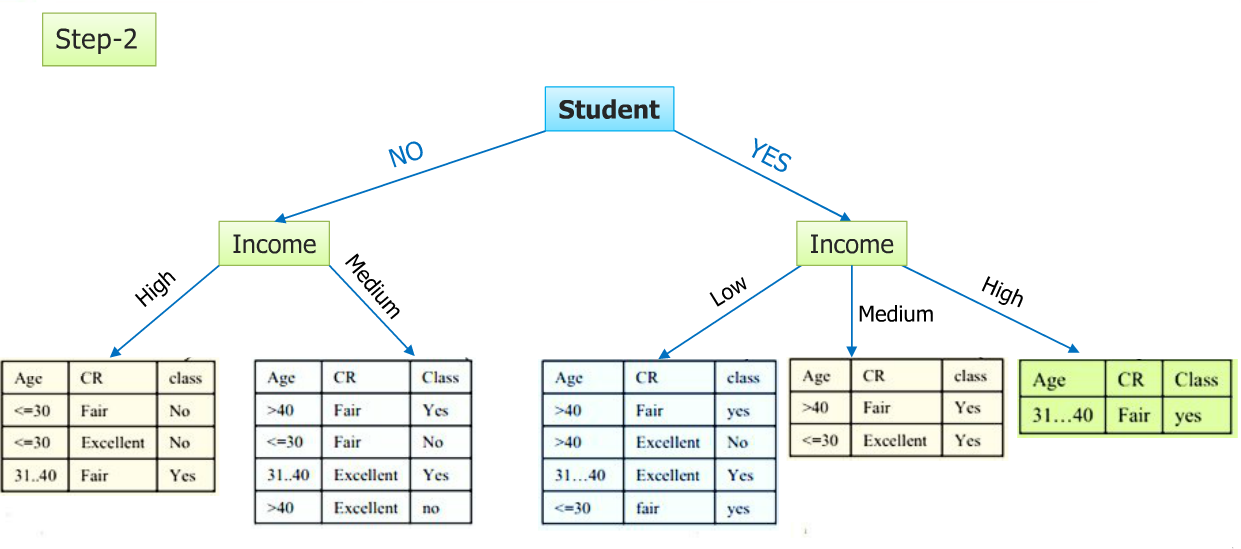
* All samples for a given node belong to the same class.
* There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
* There are no samples left.

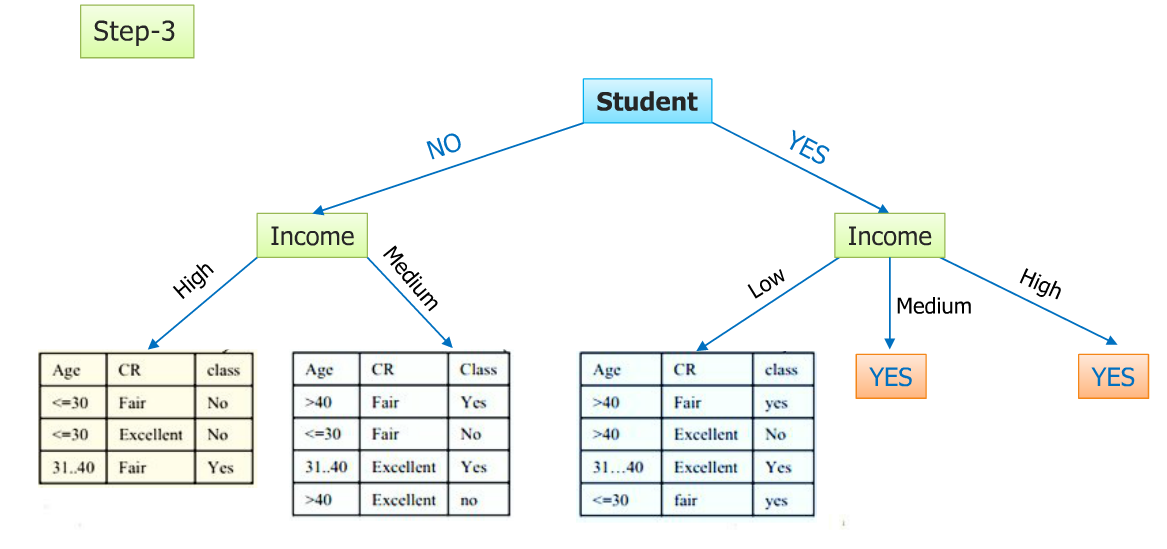
Decision Tree Examples

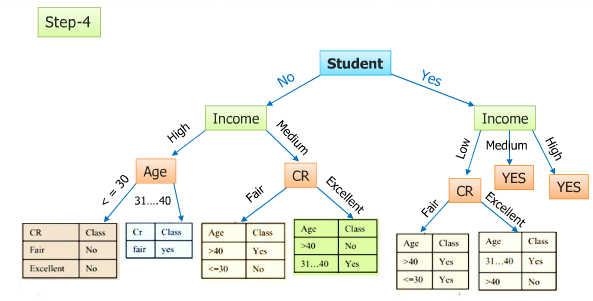


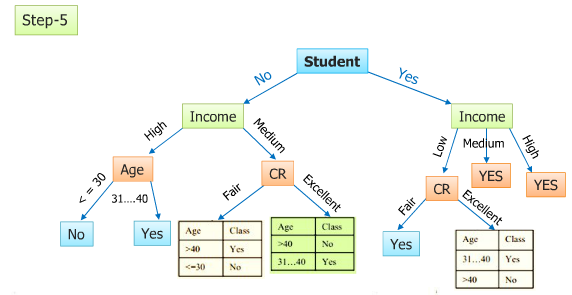
Decision Tree 1, Root: Students

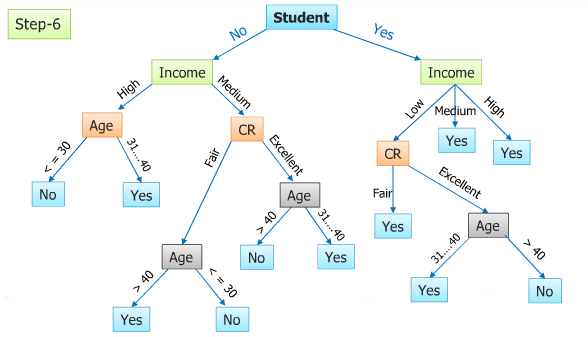




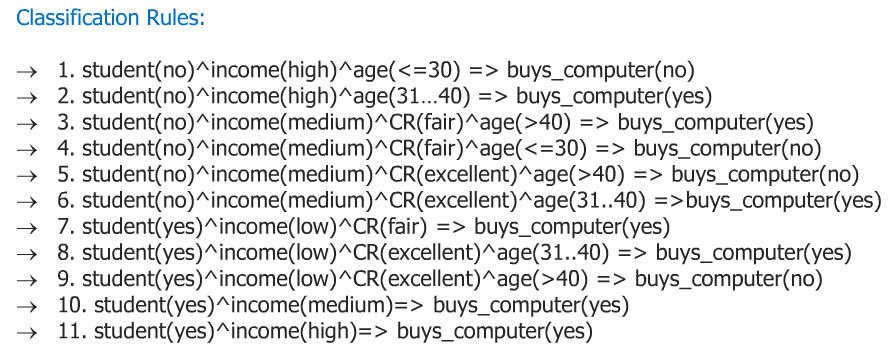




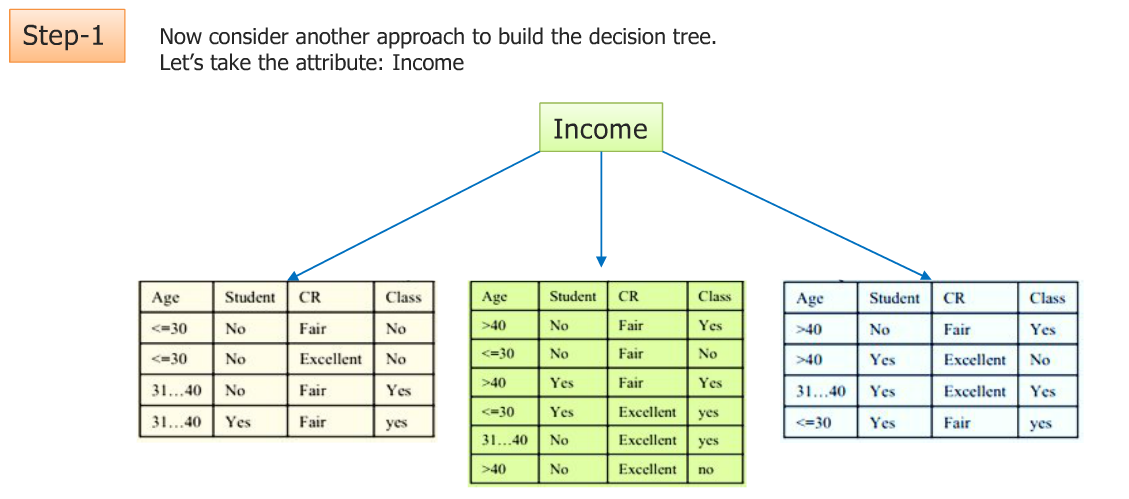




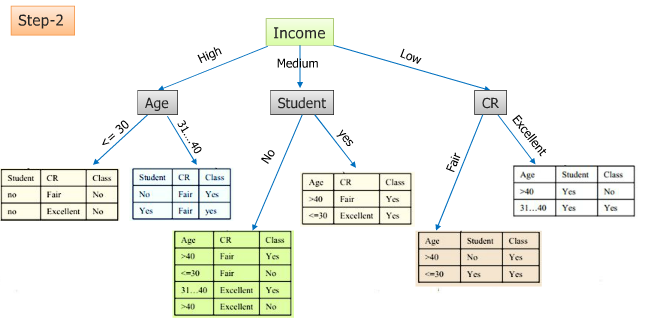
**Decision Tree 1, Root: Income**

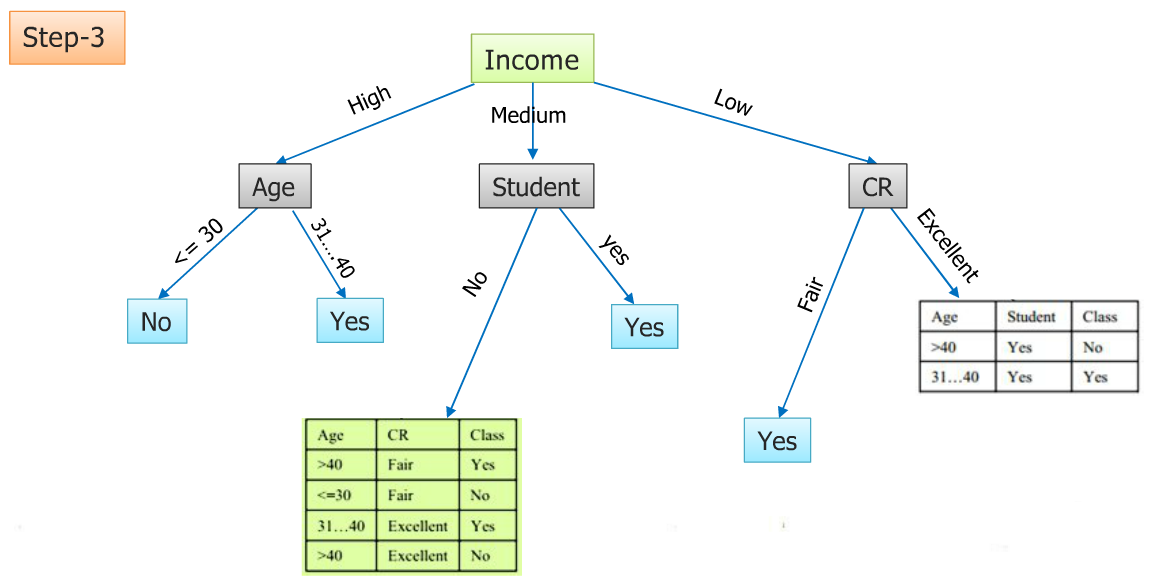


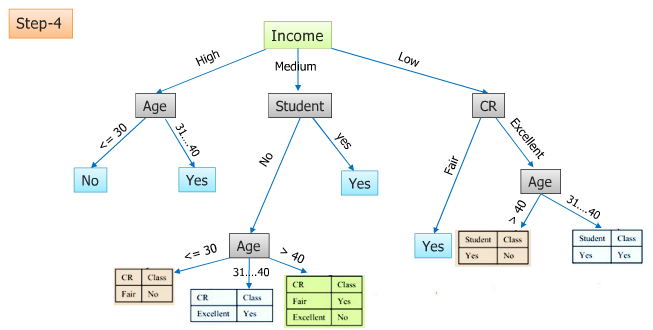
**Decision Tree 2, Root: Income**

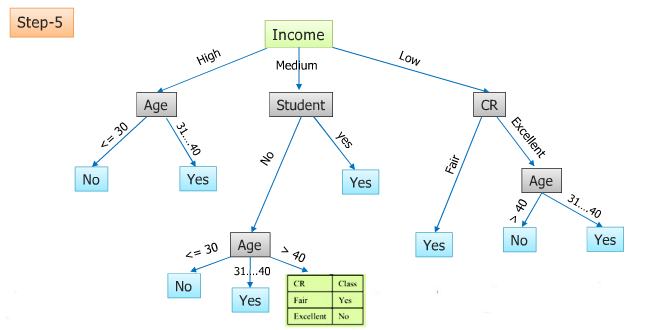


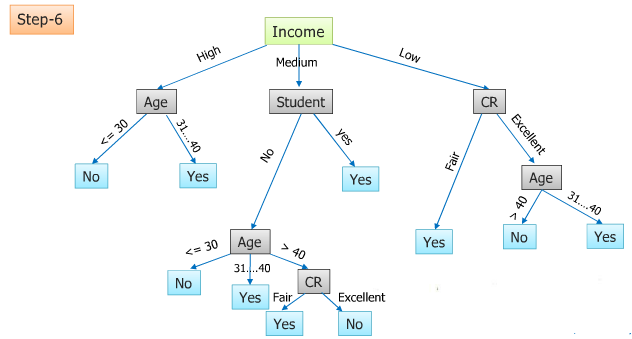
**Decision Tree, Root: Student**



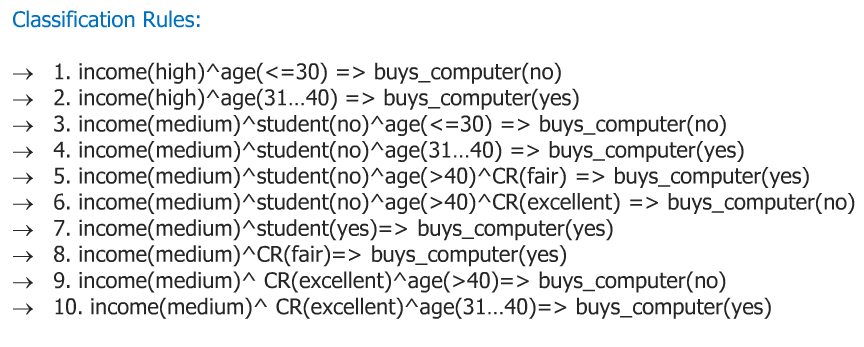








**Decision Tree, Root: Income**



**Greedy Approach and Entropy**

We want to use the attribute that does the “belt job” splitting up the training data, but can this be measured?

We use entropy and information again?

**Entropy:**

* Measure of disorder or impurity
* We will find entropy of the output values of a set of training instances.
* If output values split 50%-50% set is impure – 1
* If output is 0, set is pure -0
* If the output values are split 25-27% then entropy -0.811

**Information Gain:**

The information gain is based on the decrease in entropy after a dataset in split on attribute.

Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogenous branches).

**Entropy(Information theory)**

Measure of uncertainty(unpredictability ) of random variable

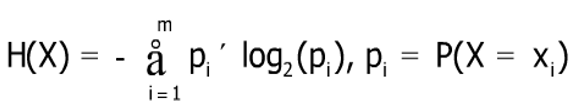
Measure of information content

Highly unpredictable = High information content = Large entropy

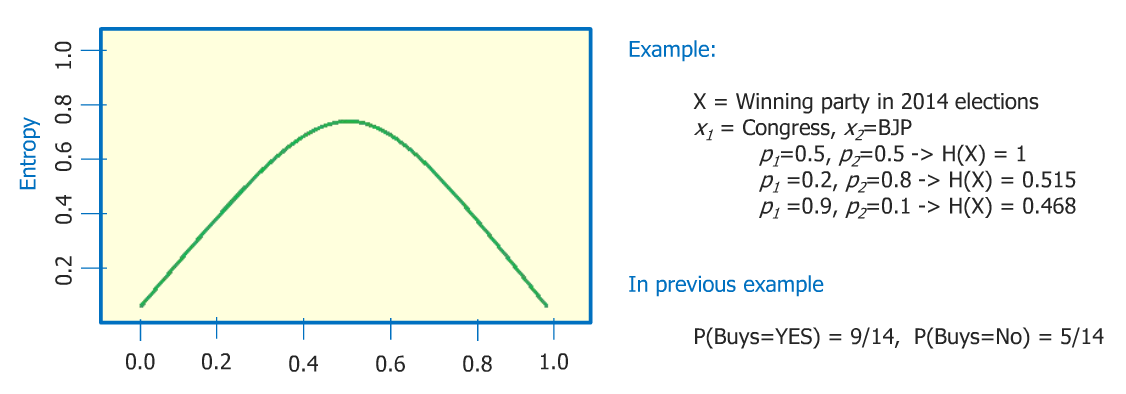
Less unpredictable = Low information content = Small entropy

Discrete random variable X taking m distinct values {x1, x2, ….. , xm}

**Mathematical Formula:**



**Entropy (Information Theory)**

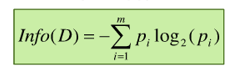
****

**Information Gain:** Attribute Selection Measure

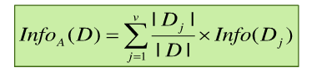
**Heuristic:** Select the attribute with the highest information gain i.e., attribute that results in most homogenous branches.

Let pi be the probability that an arbitrary tuple in D belongs to class Ci, estimated by |Ci, D|/|D|

**Expected information** (entropy) needed to classify a tuple in D:



**Information** needed (After using A to split D into v partitions) to classify D:

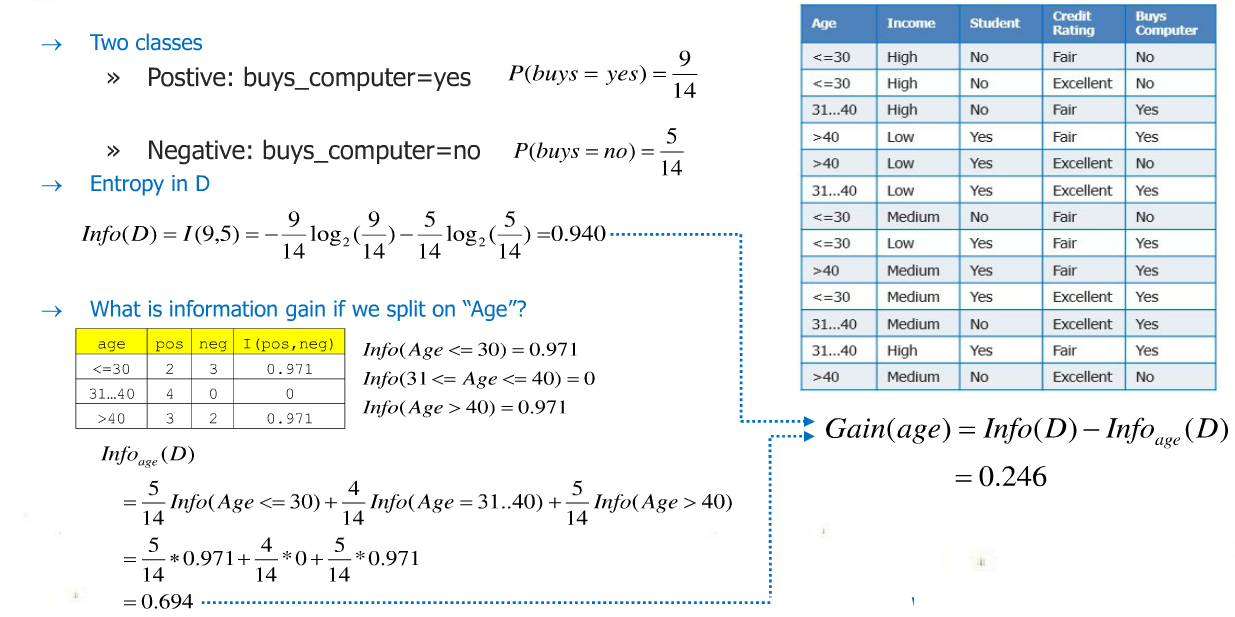


**Information gained** by branching on attribute A

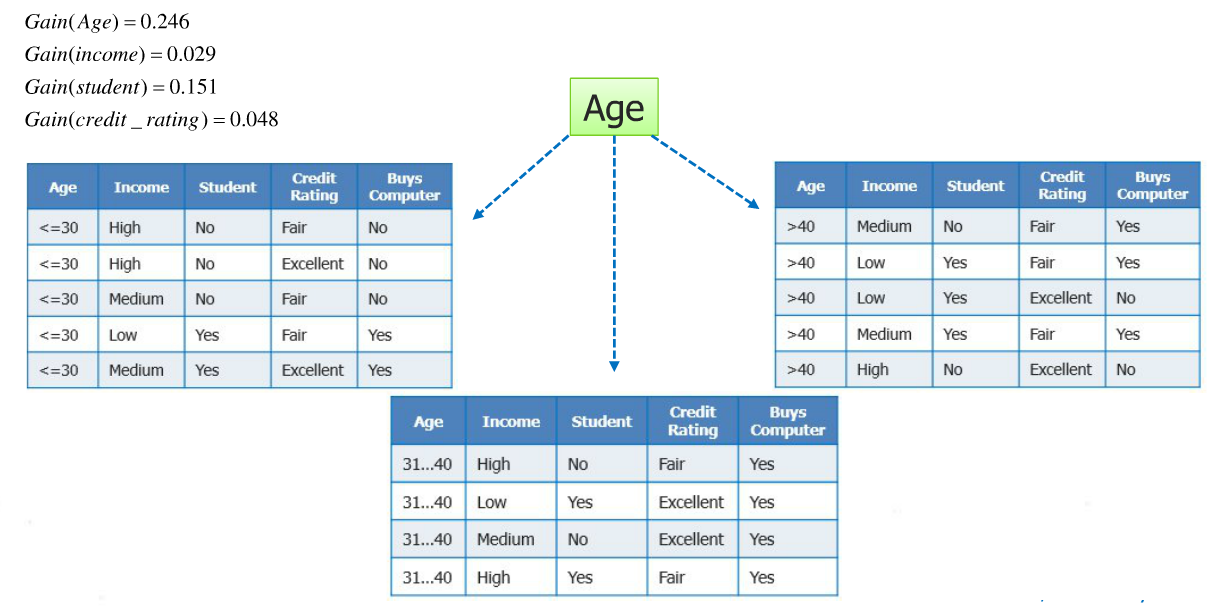


Creating a Perfect Decision Tree:

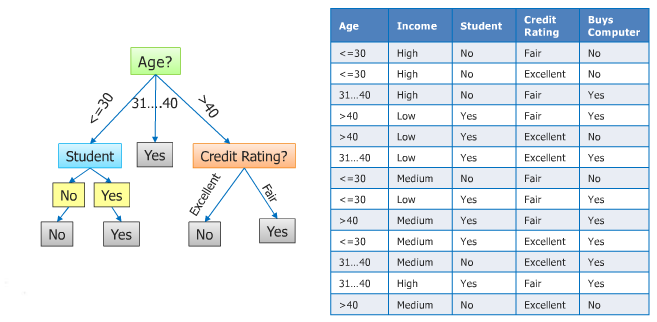
**Attribute Selection Example**

****

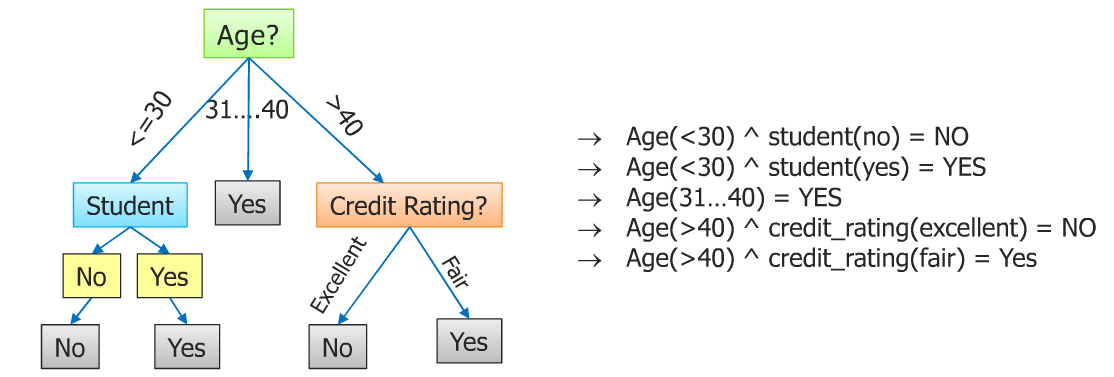
**Attribute Selection Example**

****

**Final Tree**

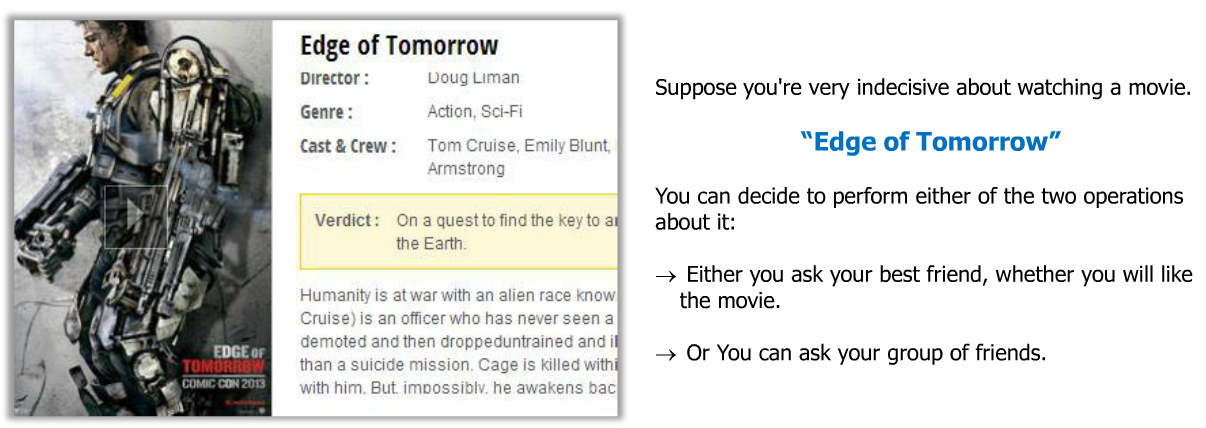
****

**Classification Tree**

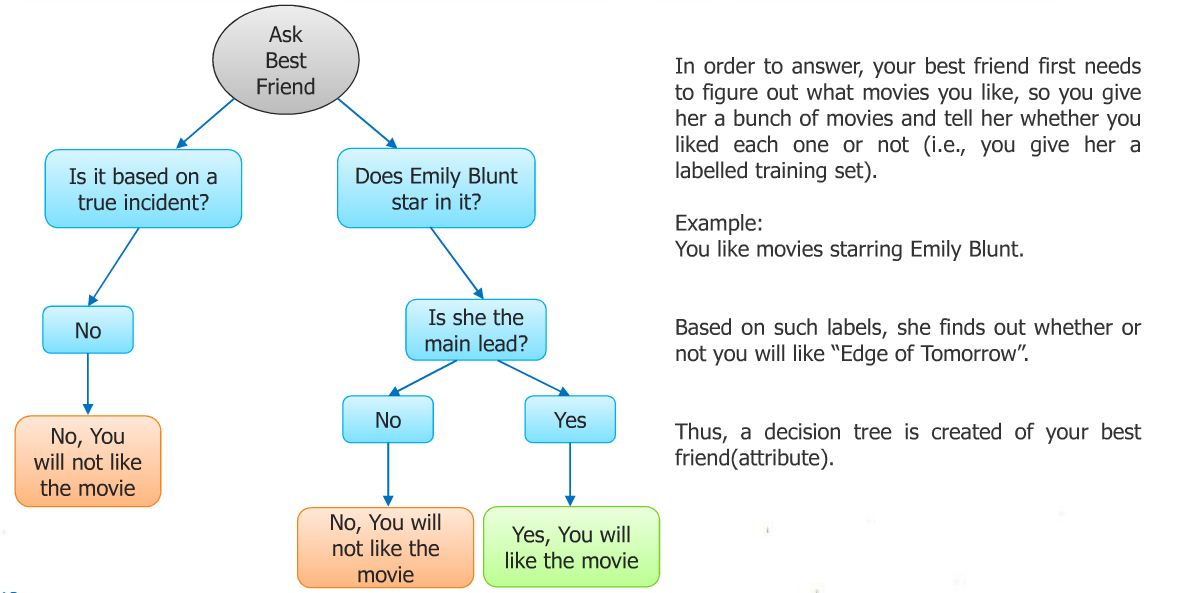
****

**Random Forest**

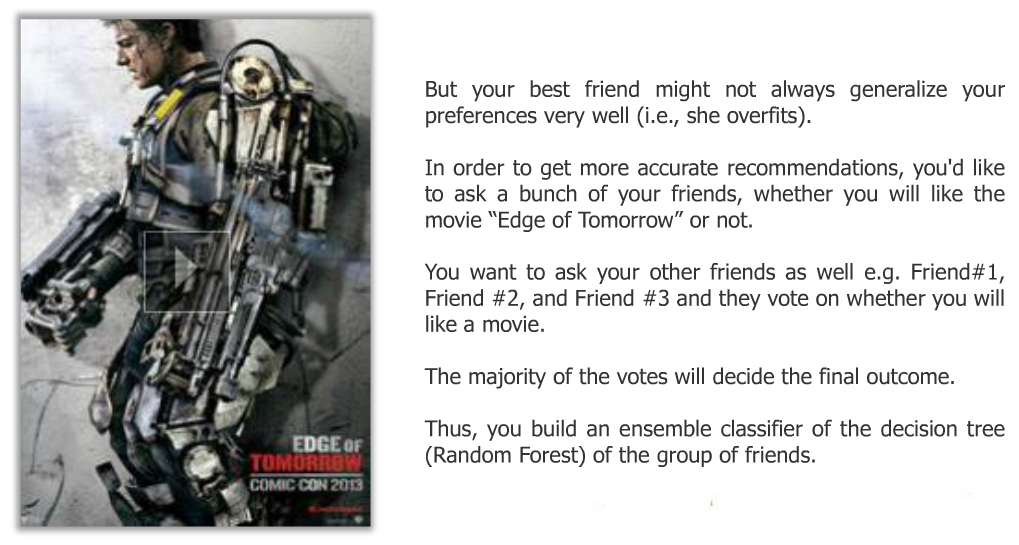
**Random Forest: An example**

****

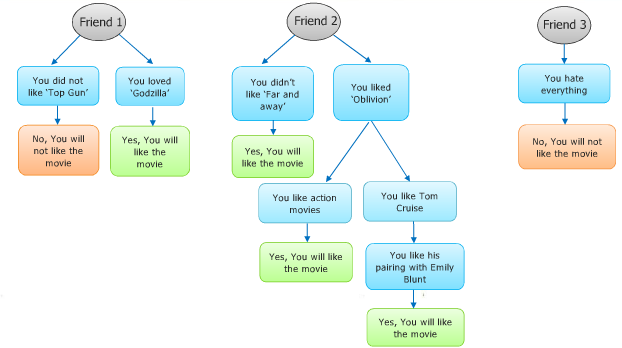
**Random Forest: An example**

****

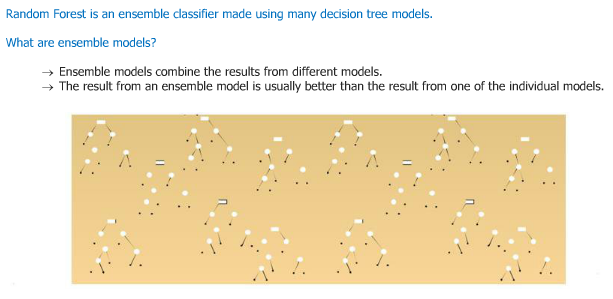
**Random Forest: An example**

****

**Random Forest: An example**

****

**What is Random Forest?**

****

**How Random Forest works?**

**Each Tree is grown as follow:**

* Let the number of training cases be N, and the number of variables in the classifier be M.
* From M input variables, a number m<<M is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the growth of the forest.
* Each tree is grown to the largest extent possible.
* The number of votes makes decision from all of the trees.
* A different subset of the training data are selected (~2/3), with replacement, to train each tree.
* Remaining training data is used to estimate error and variable importance.

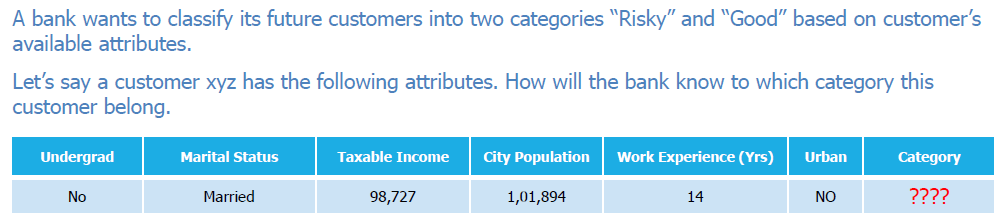
**Features of Random Forests:**

* It is unexcelled in accuracy among the current algorithms
* It runs efficiently on large databases.
* It can handle thousands of input variables without variable deletion.
* It gives estimates of what variables are important in the classification
* It generates an internal unbiased estimate of the generalization error as the forest building progresses.
* It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data is missing.
* Generated forests can be saved for future use on other data.
* Prototypes are computed, that gives information about the relation between the variables and the classification.

**Applications:**

**#** A manager has to decide whether he should hire more human resources or not in order to optimize the work load balance.

# An individual has to make a decision such as whether or not to undertake a capital project, or must choose between two competing ventures.

****

**Possible algorithms:**

Such type of problems comes under “classification”

It is the separation or ordering of objects into classes

* There are few techniques in classification method, like :
* Decision Tree
* Naïve Bayes
* K- nearest neighbors
* Support Vector machines etc..