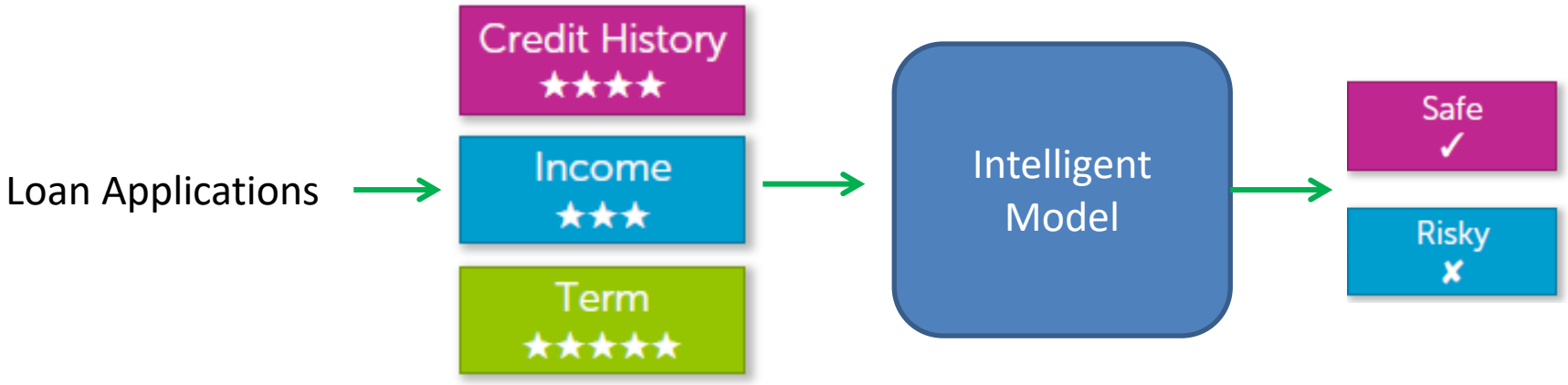
Decision Trees

Decision Trees

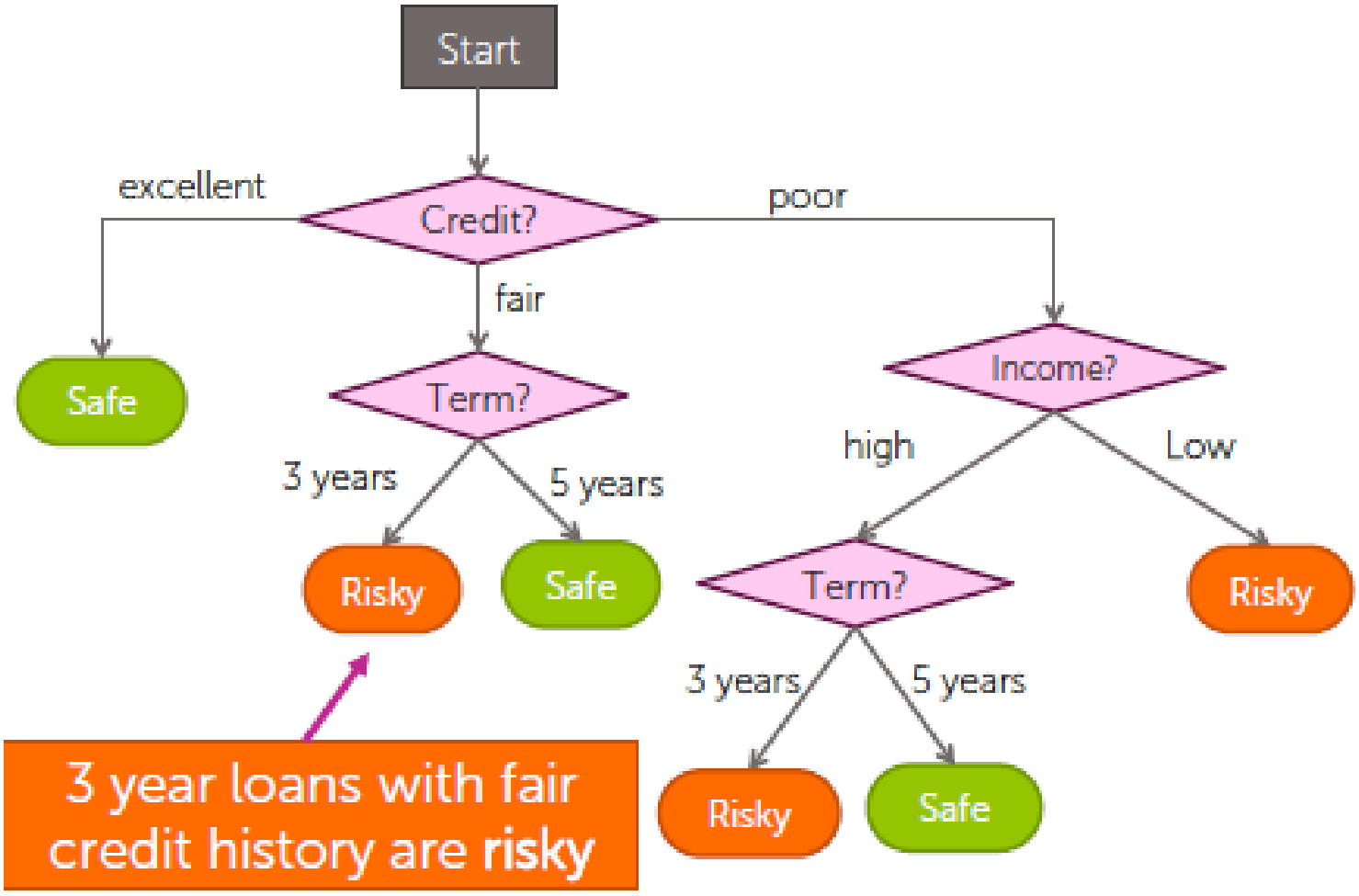
- Supervised learning algorithm
- flowchart-like tree structure
  - **Internal node** (*non leaf node*): denotes a test on an attribute
  - **Branch:** an outcome of the test
  - **Leaf node** (*terminal node*): holds a class label

Decision Tree Induction

learning of decision trees from class-labeled training tuples



Decision Tree Representation



Source: Emily Fox & Carlos Guestrin, Machine Learning Specialization

Decision Tree Learning

- Which attribute to split on
- Quality metric: classification error
  - Measures fraction of mistakes

$$Error = \frac{No.of\ incorrect\ predictions}{No.of\ examples}$$

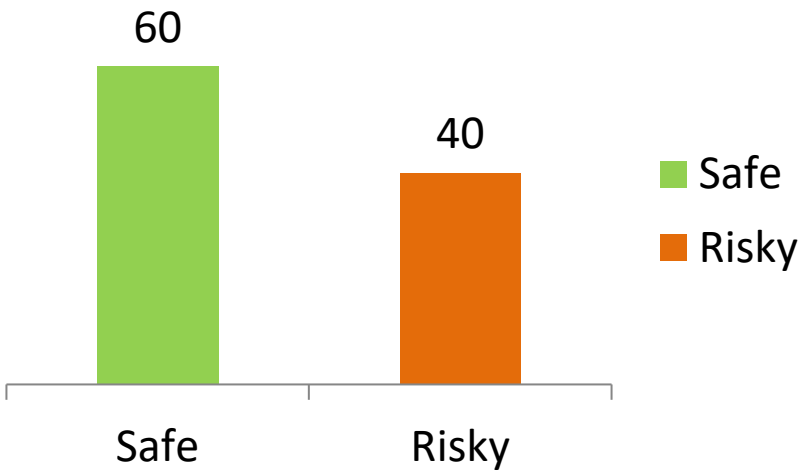
Best value= 0.0

Worst value = 1.0

- Finding the best tree : Optimize quality metric

Greedy Decision Tree Learning Algorithm

Start with all data (empty tree)

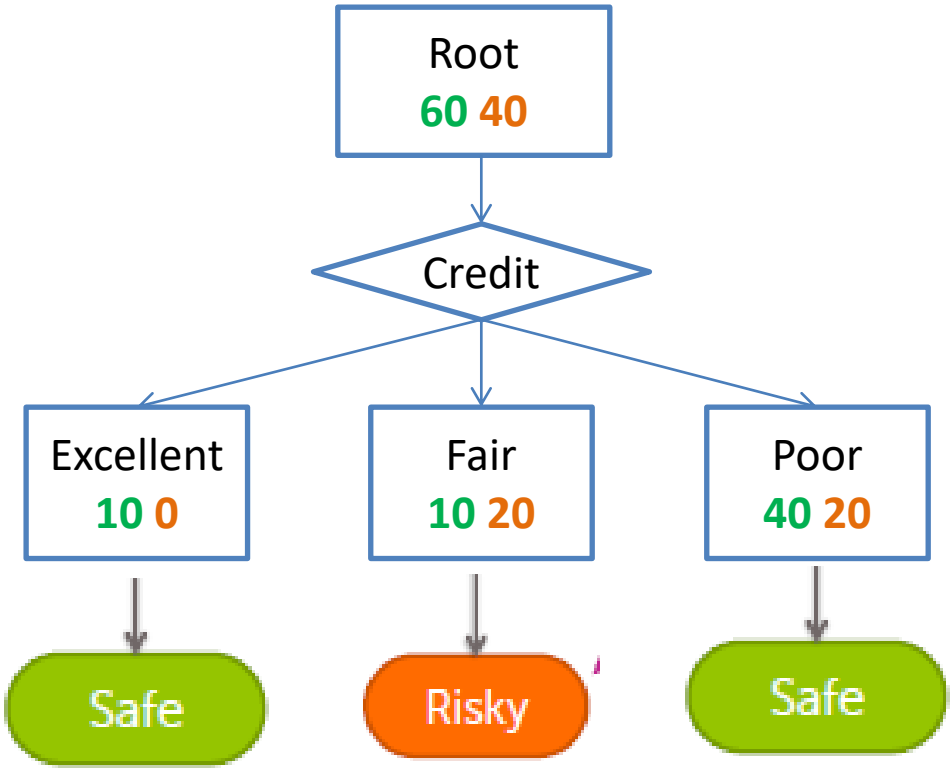


$$Error = \frac{40}{100} = 0.40$$

y‘= majority class = **Safe**

Greedy Decision Tree Learning Algorithm

Split on a feature (choice 1: credit history)



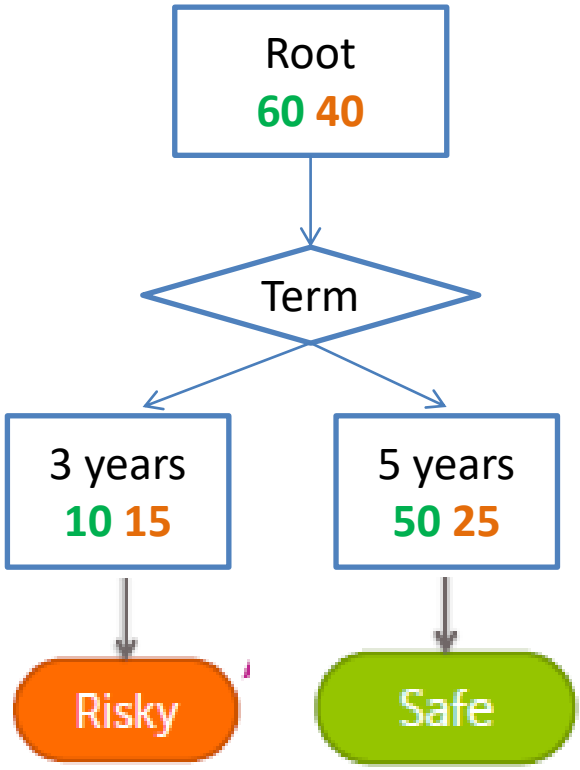
$$Error = \frac{0 + 10 + 20}{100} = 0.30$$

Tree	Classification Error
Empty (root)	0.40
Split on credit	0.30

y‘= majority class

Greedy Decision Tree Learning Algorithm

- Split on a feature (choice 2: term)



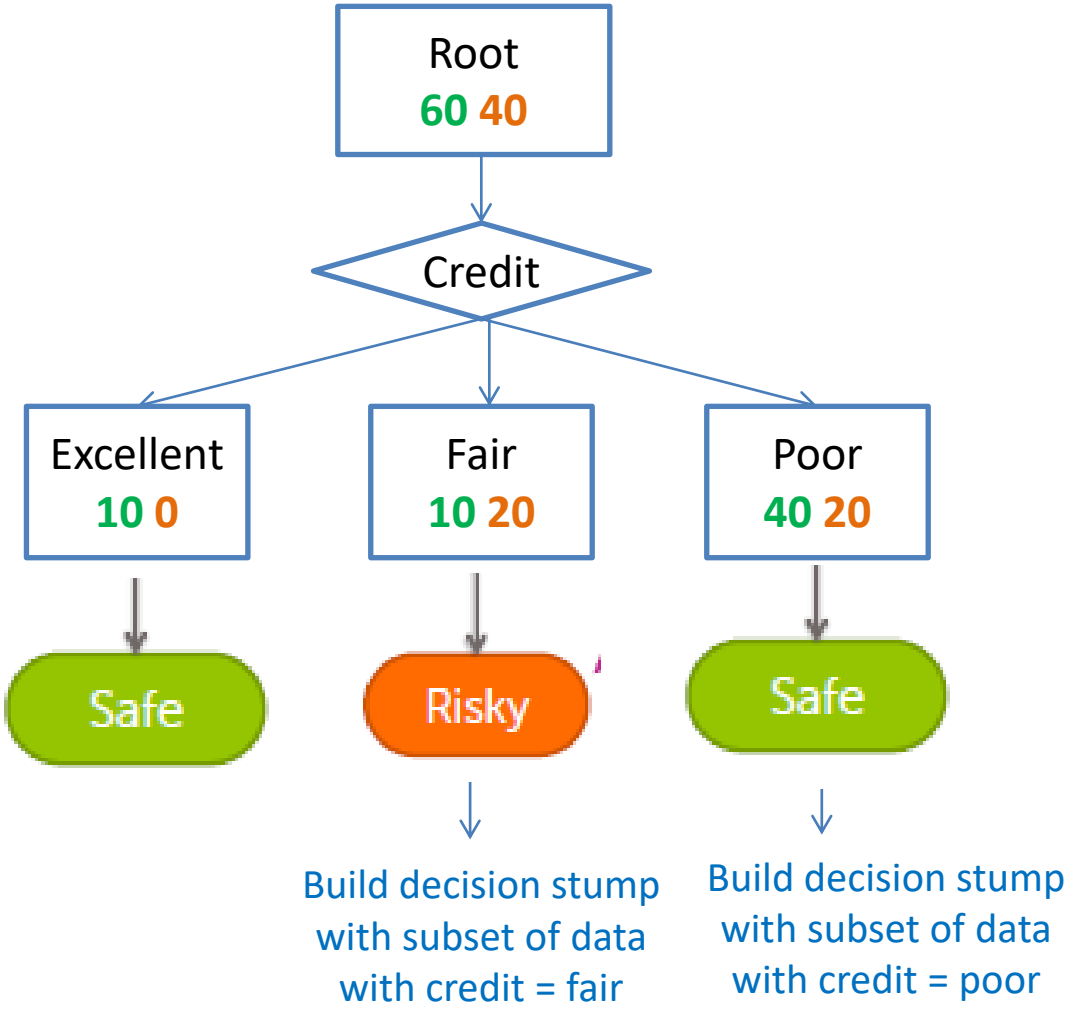
y'= majority class

$$Error = \frac{10 + 25}{100} = 0.35$$

Tree	Classification Error
Empty (root)	0.40
Split on credit	**0.30**
Split on term	0.35

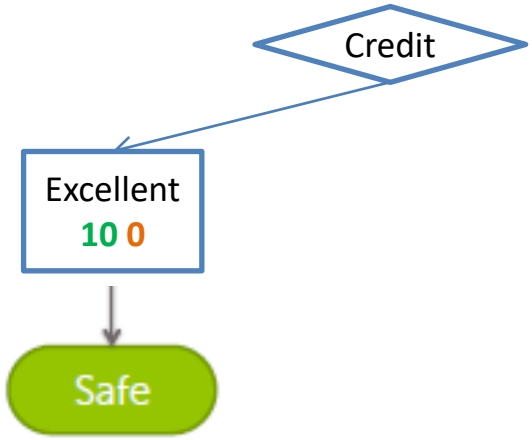
Choose feature with lowest classification error = **credit**

Recursion

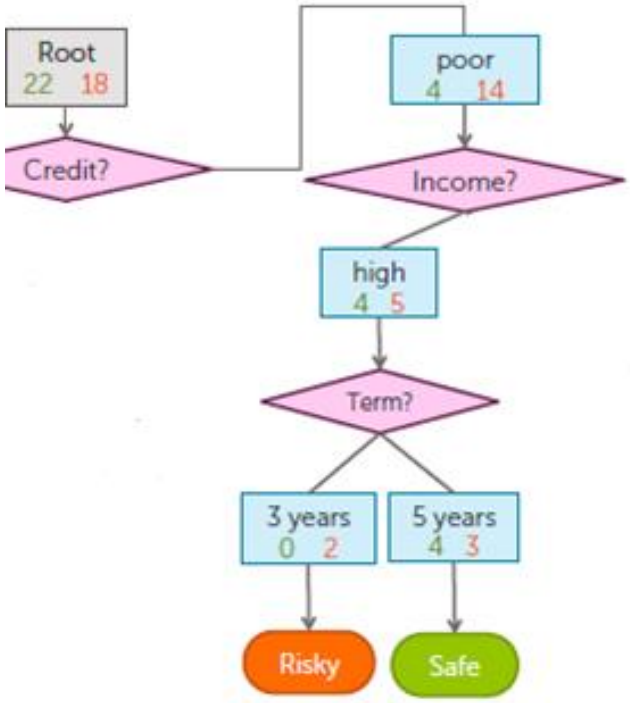

Root
60 40
Credit
Excellent
10 0
Fair
10 20
Poor
40 20
Safe
Risky
Safe
Build decision stump with subset of data with credit = fair
Build decision stump with subset of data with credit = poor

Stopping Conditions

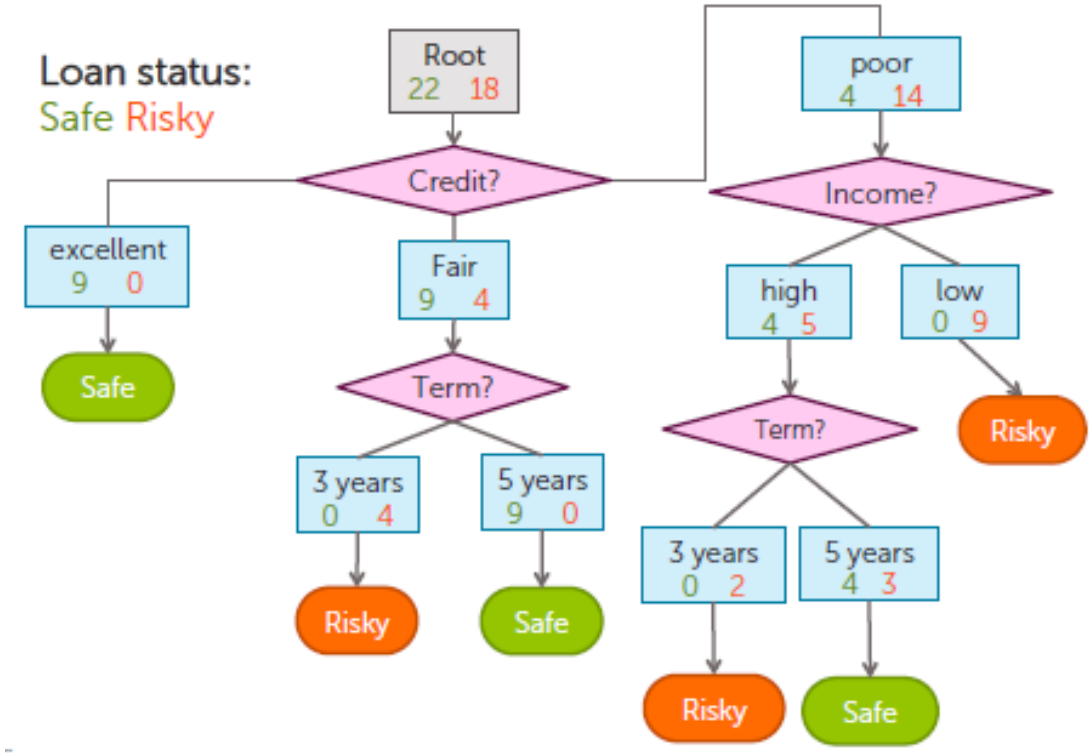
- Condition 1 : All data agrees on y



- Condition 2: Already split on all features



Decision Tree Built

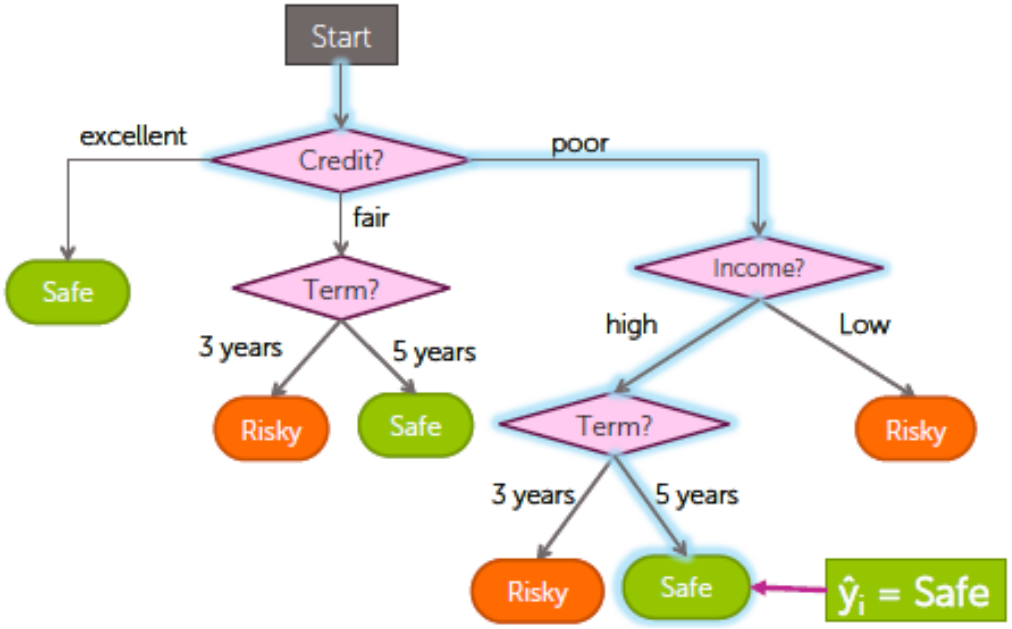


Source: Emily Fox & Carlos Guestrin, Machine Learning Specialization

Prediction

Traversing the built tree

Test point: x_i = (Credit = poor, Income = high, Term = 5 years)



Source: Emily Fox & Carlos Guestrin, Machine Learning Specialization

ID3 and Information Gain

- ID3
  - Greedy Decision Tree Induction Approach
  - Construct tree in a top-down recursive divide-and-conquer manner
- Attribute Selection Measure Used
  - Information Gain

Information Gain

- **Information Gain**
  - ID3 uses **information gain** as its attribute selection measure.
  - The attribute with the highest information gain is chosen as the splitting attribute for node N.
  - Minimizes the information needed to classify the tuples in the resulting partitions
  - Minimizes the degree of randomness or impurity (*entropy*) in partitions

Entropy

- For
  - a variable (event), X,
  - with n possible values (outcomes), $x_1, x_2 \ldots, x_n$
  - each outcome having probability, $p_1, p_2 \ldots, p_n$
  - the entropy of X , $H(X)$, is given by

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

- Entropy is between 0 and $\log_2 n$

It is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information.

Entropy Examples

- For a coin with probability p of heads and probability $q = 1 - p$ of tails

$$H = -p \log_2 p - q \log_2 q$$

  - For $p = 0.5$, $q = 0.5$ (fair coin) $H = 1$ → Entropy High -> Randomness
  - For $p = 1$ or $q = 1$, $H = 0$

↓

Entropy 0 -> Clarity

Entropy for Sample Data: Example

Hair Color	Count	p	$-p\log_2 p$
Black	75	0.75	0.3113
Brown	15	0.15	0.4105
Blond	5	0.05	0.2161
Red	0	0.00	0
Other	5	0.05	0.2161
Total	100	1.0	1.1540

Maximum entropy is $\log_2 5 = 2.3219$

Information Gain

The expected information needed to classify a tuple in *D*

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

p_i is the nonzero probability that a tuple in D belongs to class C_i

$$Gain\,(D, A) = Entropy(D) - \sum_{j=1}^{v} \frac{D_j}{D} Entropy(D_j)$$

$$= Entropy(D) - Info\,(A)$$

Attribute *A* having *v* distinct values (*a*1, *a*2, …. , *av*)

D_j contains those tuples in D that have outcome a_j of A

Information Gain

- These partitions would correspond to the branches grown from node *N*.
- Ideally, we would like this partitioning to produce an exact classification of the tuples. That is, we would like for each partition to be ***pure***.
- However, it is quite likely that the partitions will be **impure** (e.g., where a partition may contain a collection of tuples from different classes rather than from a single class).
- How much more information would we still need (after the partitioning) to arrive at an exact classification?
- This is calculated by **Info!**

ID	Age	Income	Student	Credit_Rating	Buys_computer
1	youth	high	no	fair	NO
2	youth	high	no	excellent	NO
3	middle	high	no	fair	YES
4	senior	medium	no	fair	YES
5	senior	low	yes	fair	YES
6	senior	low	yes	excellent	NO
7	middle	low	yes	excellent	YES
8	youth	medium	no	fair	NO
9	youth	low	yes	fair	YES
10	senior	medium	yes	fair	YES
11	youth	medium	yes	excellent	YES
12	middle	medium	no	excellent	YES
13	middle	high	yes	fair	YES
14	senior	medium	no	excellent	NO

Calculating Information Gain

- Calculate *Info(D)*

$$\text{Info}(D) = -9/14.\log_2(9/14) - 5/14.\log_2(5/14) = 0.940$$

- Calculate $Info_{Age}(D)$

$$\text{Info}_{Age}(D) = 5/14\text{x}(-2/5.\log_2(2/5)-3/5.\log_2(3/5)) + 4/14\text{x}(-4/4.\log_2(4/4)-0) + 5/14\text{x}(-3/5.\log_2(3/5)-2/5.\log_2(2/5)) = 0.694$$

youth (2+, 3-) middle (4+, 0-) senior (3+, 2-)

Calculating Information Gain

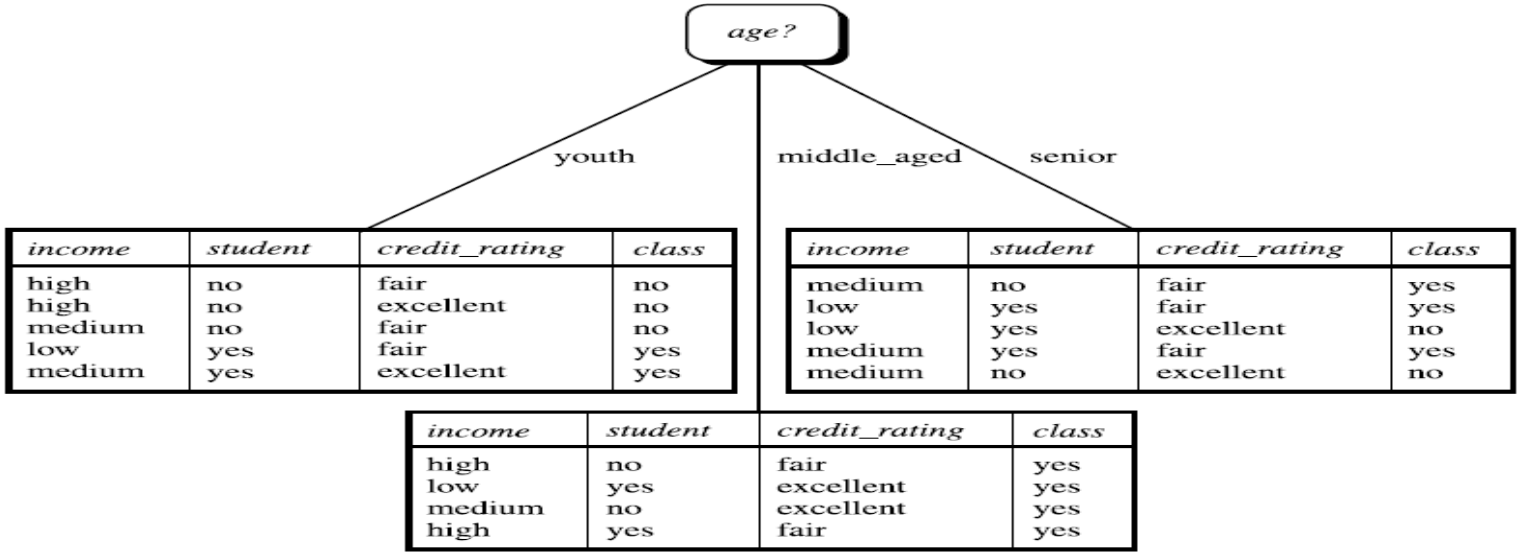
- Calculate *Gain(Age)*

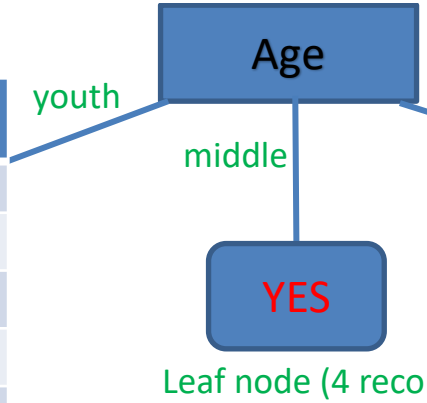
$$\text{Gain(Age)} = \text{Info(D)} - \text{Info}_{\text{Age}}\text{(D)} = 0.940 - 0.694 = 0.246$$

- Similarly,
  - Gain(Income) = 0.029
  - Gain(Student)= 0.151
  - Gain (Credit-rating)=0.048

Calculating Information Gain

- Since, *Age* has the highest *Info. Gain,* so *Age* will be the splitting attribute.





Income	Student	Credit-Rating	Buys-computer
High	No	Fair	No
High	No	Excellent	No
Medium	No	Fair	No
Low	Yes	Fair	Yes
Medium	Yes	Excellent	Yes

Income	Student	Credit-Rating	Buys-computer
Medium	No	Fair	Yes
Low	Yes	Fair	Yes
Low	Yes	Excellent	no
Medium	Yes	Fair	yes
Medium	No	Excellent	No