CS 7265 BIG DATA ANALYTICS

DECISION TREES

* Some contents are adapted from Dr. Hung Huang and Dr. Chengkai Li at UT Arlington

Mingon Kang, Ph.D.
Computer Science, Kennesaw State University

Terminology

- Features
  - An individual measurable property of a phenomenon being observed
  - The number of features or distinct traits that can be used to describe each item in a quantitative manner
  - May have implicit/explicit patterns to describe a phenomenon
- Samples
  - Items to process (classify or cluster)
  - Can be a document, a picture, a sound, a video, or a patient

Reference: http://www.slideshare.net/rahuldausa/introduction-to-machine-learning-38791937

Terminology

- Feature vector
  - An N-dimensional vector of numerical features that represent some objects
  - A sample consists of feature vectors
- Feature extraction (feature selection)
  - Preparation of feature vector
  - Transforms the data in the high-dimensional space to a space of fewer dimensions

Reference: http://www.slideshare.net/rahuldausa/introduction-to-machine-learning-38791937

Data in Machine Learning

- x_i: input vector, independent variable

$$\boldsymbol{x}_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,n} \end{bmatrix}, \quad x_{i,j} \in \mathbb{R}$$

- y: response variable, dependent variable
  - $y \in \{-1, 1\}$ or $\{0, 1\}$: binary classification
  - $y \in \mathbb{Z}$: multi-label classification
  - $y \in \mathbb{R}$: regression
  - Predict a label when having observed some new x

Types of Variable

- **Categorical variable:** discrete or qualitative variables
  - Nominal:
    - Have two or more categories, but which do not have an intrinsic order
  - Dichotomous
    - Nominal variable which have only two categories or levels.
  - Ordinal
    - Have two or more categories, which can be ordered or ranked.
- Continuous variable

Mathematical Notation

- Matrix: uppercase bold Roman letter, $\mathbf{X}$
- Vector: lower case bold Roman letter, $\mathbf{x}$
- Scalar: lowercase letter
- Transpose of a matrix or vector: superscript T or ‘
- E.g.
  - Row vector: $(x_1, x_2, \dots, x_N)$
  - Corresponding column vector: $\mathbf{x} = (x_1, x_2, \dots, x_N)^{\mathrm{T}}$
  - Matrix: $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_p}\}$

Supervised learning

Data: $D = \{d_1, d_2, \dots, d_n\}$ a set of *n* samples

where $d_i =< \boldsymbol{x_i}, y_i >$

$\boldsymbol{x_i}$ is a input vector and y_i is a desired output

Objective: learning the mapping $f: \boldsymbol{X} \rightarrow \boldsymbol{y}$

subject to $y_i \approx f(\boldsymbol{x_i})$ for all $i = 1,\dots,$n

Regression: $\boldsymbol{y}$ is continuous

Classification: $\boldsymbol{y}$ is discrete

Decision Tree

- A decision tree is a natural and simple way of inducing following kind of rules.

  If (Age is x) and (income is y) and (family size is z) and (credit card spending is p) then he will accept the loan

- It is powerful and perhaps most widely used modeling technique of all
- Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance

Ref: https://www.slideshare.net/INSOFE/decision-trees-36100490

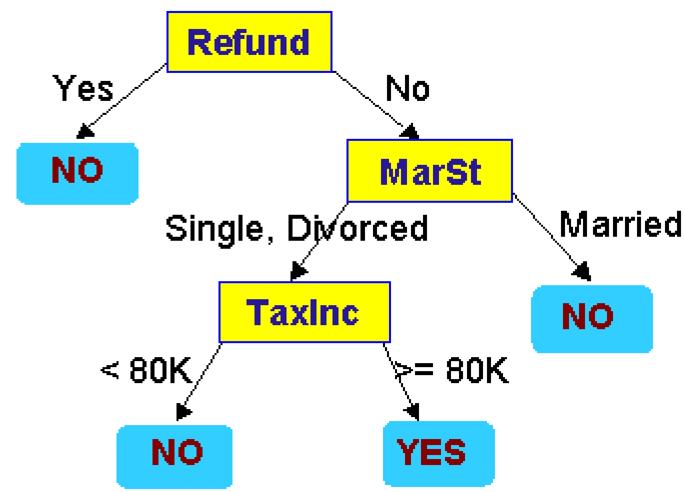
Example of a Decision Tree

Training data

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Decision Tree Induction

Decision Tree Model



Refund: Categorical

Marital Status: Categorical

Taxable Income: continuous

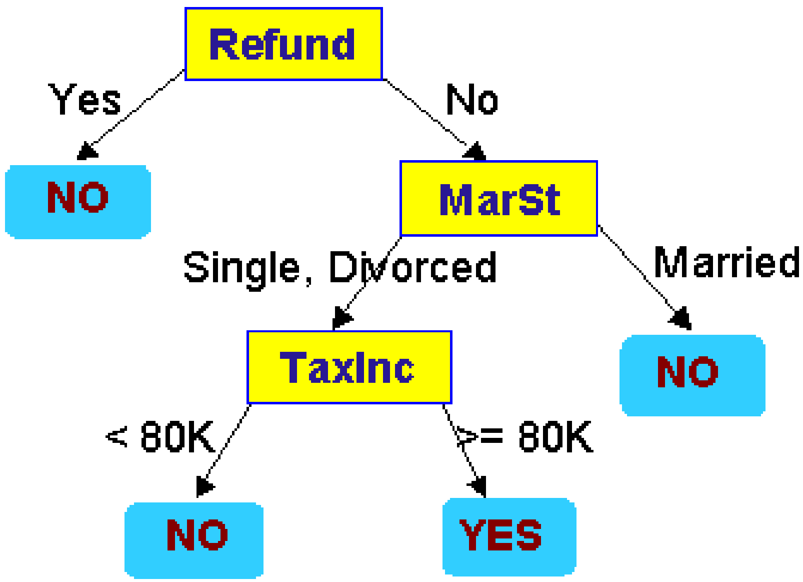
Cheat: Class

Example of a Decision Tree

When we have new data (test data)

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

How can we predict it?



Deduction from the Decision Tree we obtained from training data

Decision Tree Induction

- Large search space
  - Finding the global optimal decision tree is computationally infeasible.
- Efficient feasible algorithm
  - Not optimal, but approximate
  - Greedy strategy
  - Grow the tree by making locally optimal decisions in selecting attributes.

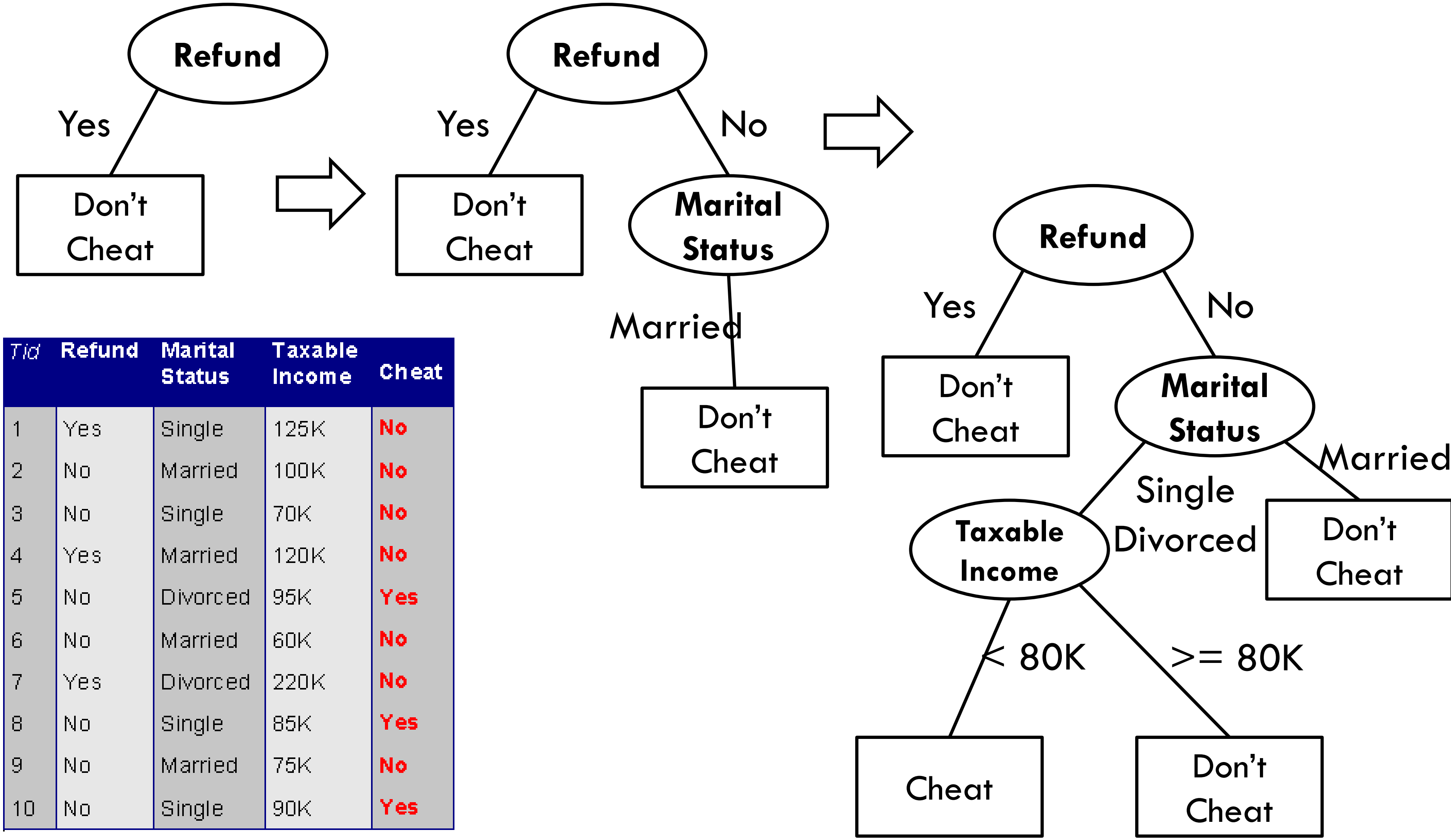
Decision Tree Induction

- Decision Tree Algorithms:
  - Hunt’s algorithm (one of the earliest)
  - CART
  - ID3, C4.5
  - SLIQ, SPRINT

Hunt's algorithm

- Most decision tree induction algorithms are based on Hunt's algorithm
- Let $\boldsymbol{D_t}$ be the set of training data and $\boldsymbol{y}$ be class labels, $\boldsymbol{y} = \{y_1, y_2, \dots, y_c\}$
  - If $\boldsymbol{D_t}$ contains data that belong to y_k, its decision tree consists of leaf node labeled as y_k
  - If $\boldsymbol{D_t}$ is an empty set, the decision tree is a leaf node of default class
  - If $\boldsymbol{D_t}$ contains data that belong to more than one classes, perform "attribute test" to split the data into smaller and more homogenous subsets

Example of Hunt's algorithm



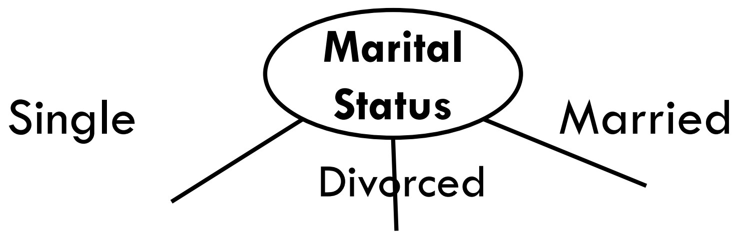
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Tree Induction

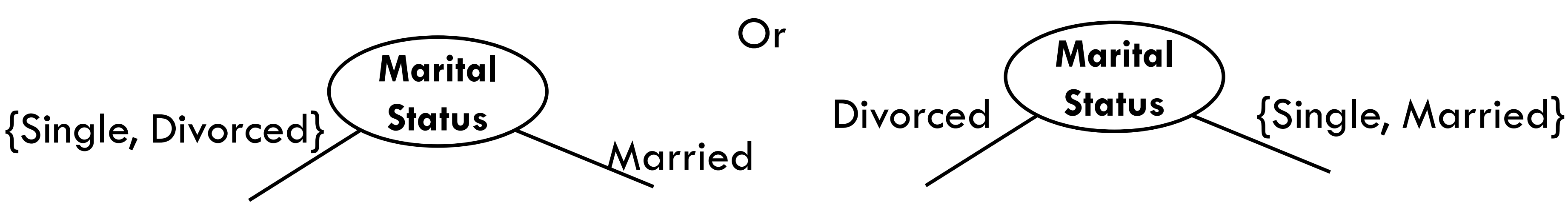
- Greedy strategy
  - Split the data based on the attribute test that locally optimizes certain criterion
- Determine how to split the data
  - How to specify the attribute test condition?
  - How to determine the best split?
    - 2-way split vs Multi-way split
- Determine when to stop splitting

Splitting on Nominal Attributes

- Multi-way split
  - Use a many partitions as distinct values

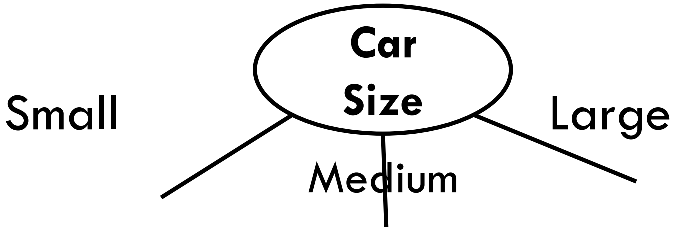


- Binary split
  - Divides values into two subsets. Need to optimize



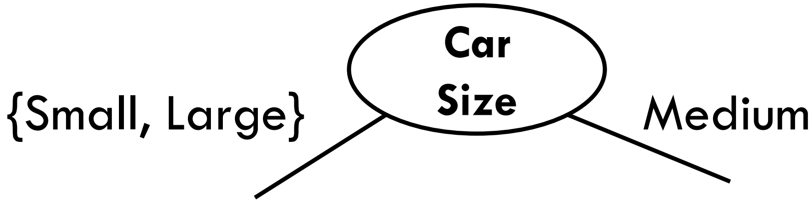
Splitting on Ordinal Attributes

- Multi-way split
  - Use a many partitions as distinct values



- Binary split
  - Divides values into two subsets

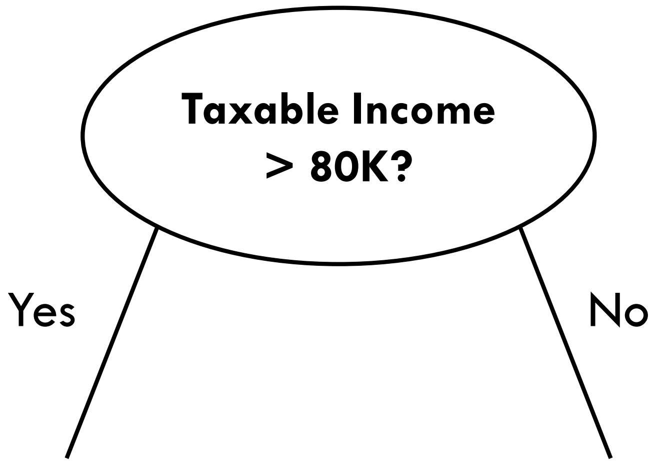
However, how about this partition?



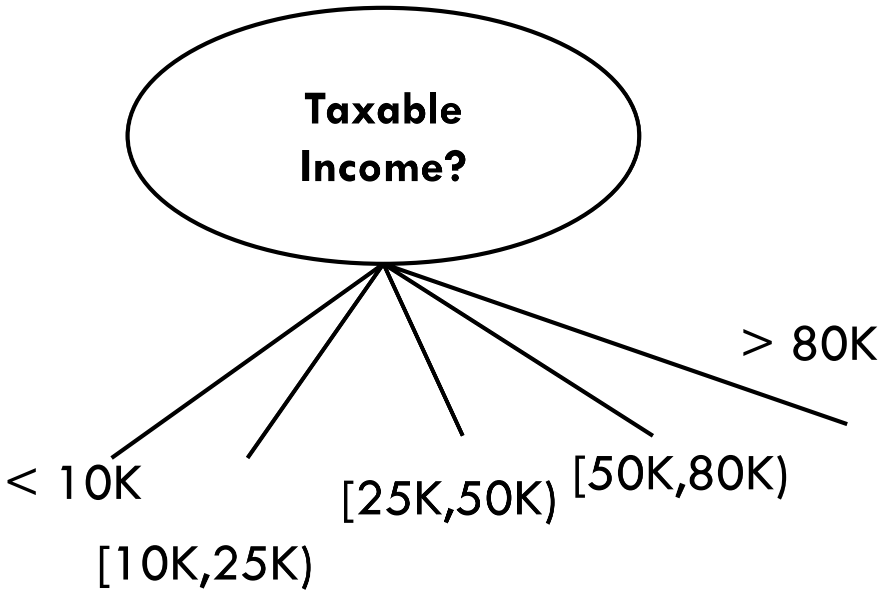
Splitting on Continuous Attributes

- Discretization
  - Convert continuous to an ordinal categorical attribute
- Binary Decision
  - Split into two, $(A < v)$ or $(A >= v)$
  - Consider all possible splits and finds the best cut

Splitting on Continuous Attributes



Binary split



Multi-way split

How to determine the best split?

- Find nodes with homogeneous class distribution
- Measure of node impurity

C0: 5 C1: 5	C0: 9 C1: 1
Non-homogeneous, High degree of impurity	Homogeneous, Low degree of impurity

How to measure of node impurity?

- Gini Index
  - Most commonly used to measure inequality of income or wealth.
  - A value between zero and one, where one expresses maximal inequality
- Entropy
  - Information Theory
- Misclassification error

GINI Index

- Gini index for a given node t:

$$GINI(t) = 1 - \sum_j [p(j|t)]^2$$

$p(j|t)$ is a conditional probability, but can be measured by the relative frequency of class j at node t)

- Minimum (0) when all data belong to one class only
  - Imply most interesting information
- Maximum (1) when all data are equally distributed among all classes

C1	0
C2	6
Gini=0.000	

C1	1
C2	5
Gini=0.278	

C1	2
C2	4
Gini=0.444	

C1	3
C2	3
Gini=0.500	

Splitting based on GINI

- Used in CART, SLIQ, SPRINT
- When a node *p* is split into *k* partitions, the quality of split is computed as,

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

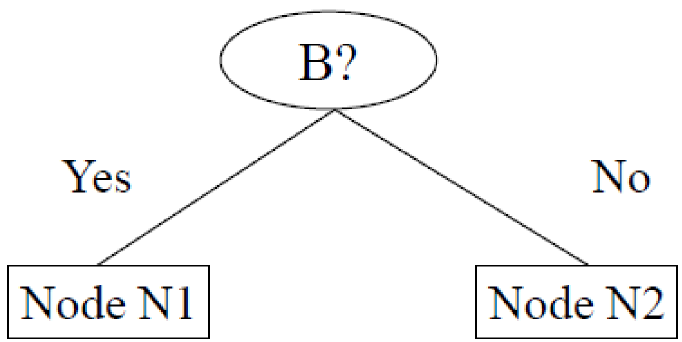
n_i = number of data at child *i*,

n= number of data at node p.

GINI on Binary attributes

- Splits into two partitions
- Effect of Weighting partitions:
  - Larger and Purer partitions are sought for

	Parent
C1	6
C2	6
Gini = 0.500	



Gini(N1)
$= 1 – (5/7)^2 – (2/7)^2$
$= 0.408$

Gini(N2)
$= 1 – (1/5)^2 – (4/5)^2$
$= 0.32$

	N1	N2
C1	5	1
C2	2	4
Gini=0.371		

Gini(Children)
$= 7/12 * 0.408 +$
$5/12 * 0.32$
$= 0.371$

GINI on Categorical Attributes

- For each distinct value, count for each class
- Use the count matrix to make decision

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

GINI on Continuous Attributes

- Use binary decisions based on one value
- Several Choices for the splitting value
- A count matrix for each splitting value
  - Counts in each of the partitions, A<v and A >= v
- Compute its Gini index each

GINI on Continuous Attributes

- Sort the attribute
- Linearly scan these values and compute gini index
- Choose the split cut that has the least gini index

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
	Taxable Income									
Sorted Values	60	70	75	85	90	95	100	120	125	220

Split Positions	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		*0.300*		0.343		0.375		0.400		0.420	

Entropy

- Adapted from a thermodynamic system
- Measure of <u>molecular disorder</u> within a macroscopic system
- Entropy is zero when a outcome is certain.

$$Entropy(t) = -\sum_j p(j|t) \log p(j|t)$$

- Computations are similar to the GINI index

Entropy

$$Entropy(t) = -\sum_j p(j\,|\,t)\log_2 p(j\,|\,t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (5/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Splitting based on Information Gain

- Information Gain:
  - Expected reduction in entropy caused by partitioning

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

$Entropy(p)$: entropy at parent node *p*

- First term: entropy of the original collection

- Second term: expected value of the entropy after S is partitioned using attribute *i*

Splitting based on Information Gain

- Choose the split so that maximize gain
- Used in ID3
- Drawback: tends to splits that result in large number of partitions, each being small but pure.

Splitting based on information theory

- Gain Ratio
- $GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$
- $SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$
- Adjusts information gain by the entropy of the partitioning (SplitINFO). Higher entropy partitioning (large number of small partitions) is penalized
- Used in C4.5

Splitting based on Classification Error

- Classification error at a node t:

- $Error(t) = 1 - max_i P(i|t)$
- Measures misclassification error at a node
  - Maximum when records are equally distributed among all classes
  - Minimum when all data belong to one class.

Splitting based on Classification Error

$$Error(t) = 1 - \max_i P(i \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Stopping criteria for tree induction

- Stop expanding a node when all data belong to the same class
- Stop expanding a node when all data have same (or similar) attribute values

Decision Tree

- Advantage:
  - Inexpensive to construct
  - Extremely fast at classifying new data
  - Easy to interpret the decision process
- Issues in Decision Tree
  - Overfitting problem
  - Training data with missing values

C4.5

- Simple depth-first construction
- Information Gain for splitting criteria
- Sorts continuous attributes at each node

- J. Ross Quinlan, C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning), 1st Edition, 1992, ISBN: 1558602380