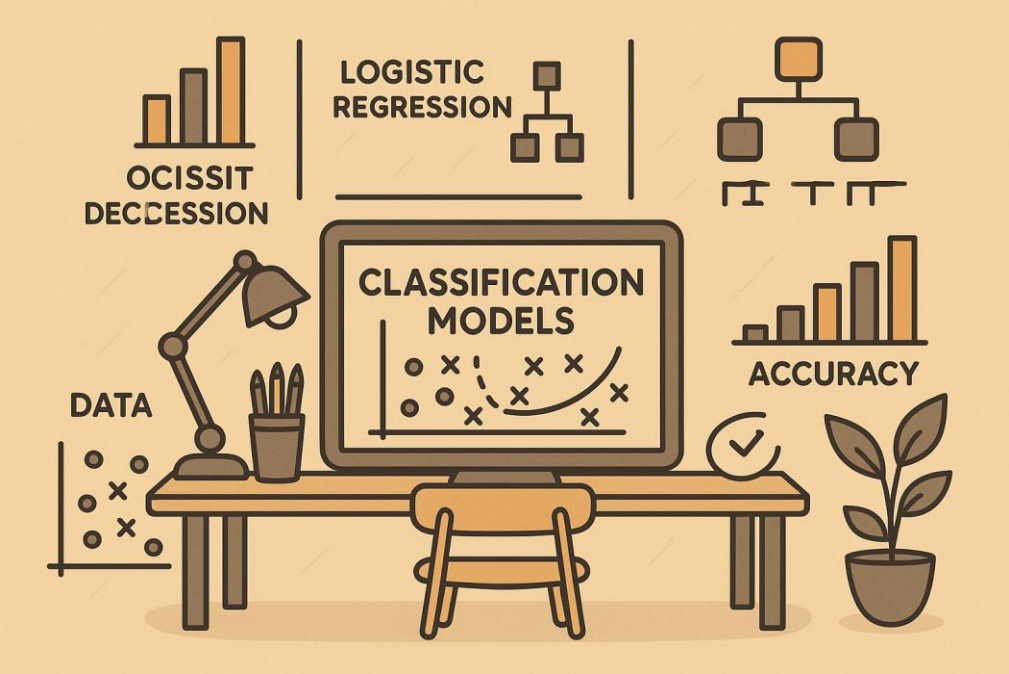
Classification Models Intro

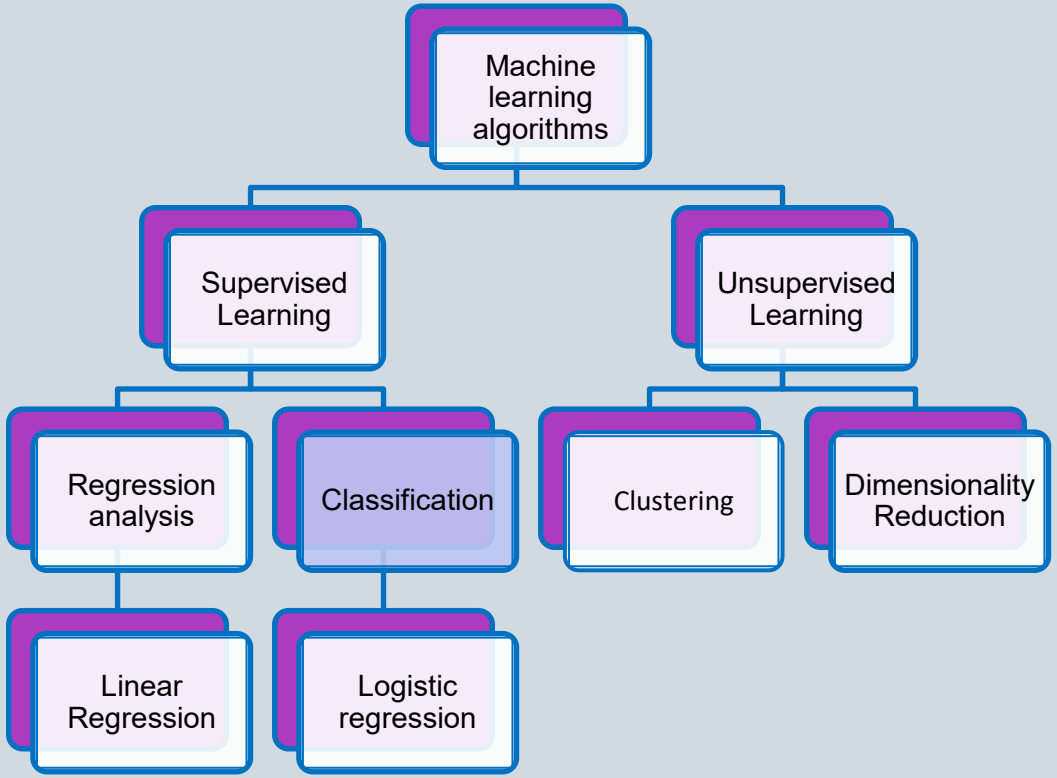


DR. ZVI BEN AMI

Agenda

- **Decision Trees**
  - Decision Tree Induction
  - Decision Tree Deduction
  - Dealing with different attribute types
  - Measures of node impurity
  - Split criteria
  - Stop criteria

Machine Learning Algorithms



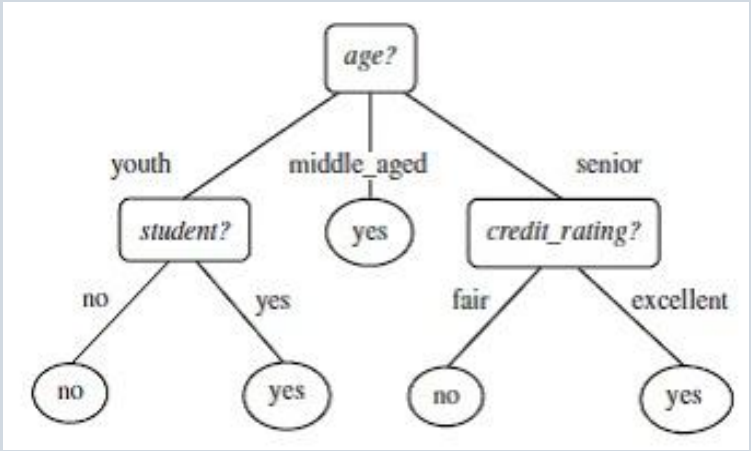
Classification

Given a collection of records, where each record is a set of attributes and a class (category), the objective is to find a model for the class as a function of the values of other attributes.

- The model is inferred from a training set
- Trained model can make accurate predictions for the (previously unseen) test set

Supervised Learning: Uses

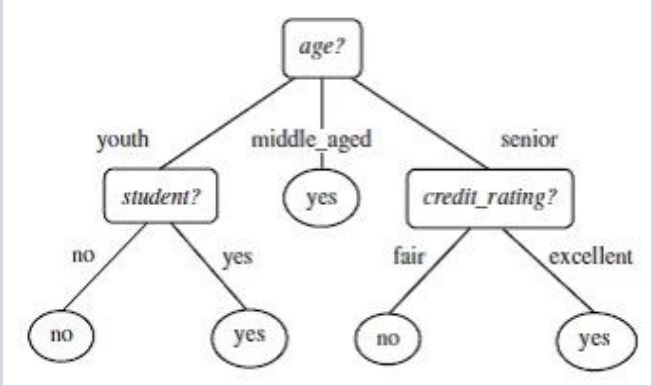
Example: decision trees tools that create rules



- One could fit decision tree to the data.
- How could you use such tree?

Supervised Learning: Uses

Example: decision trees tools that create rules



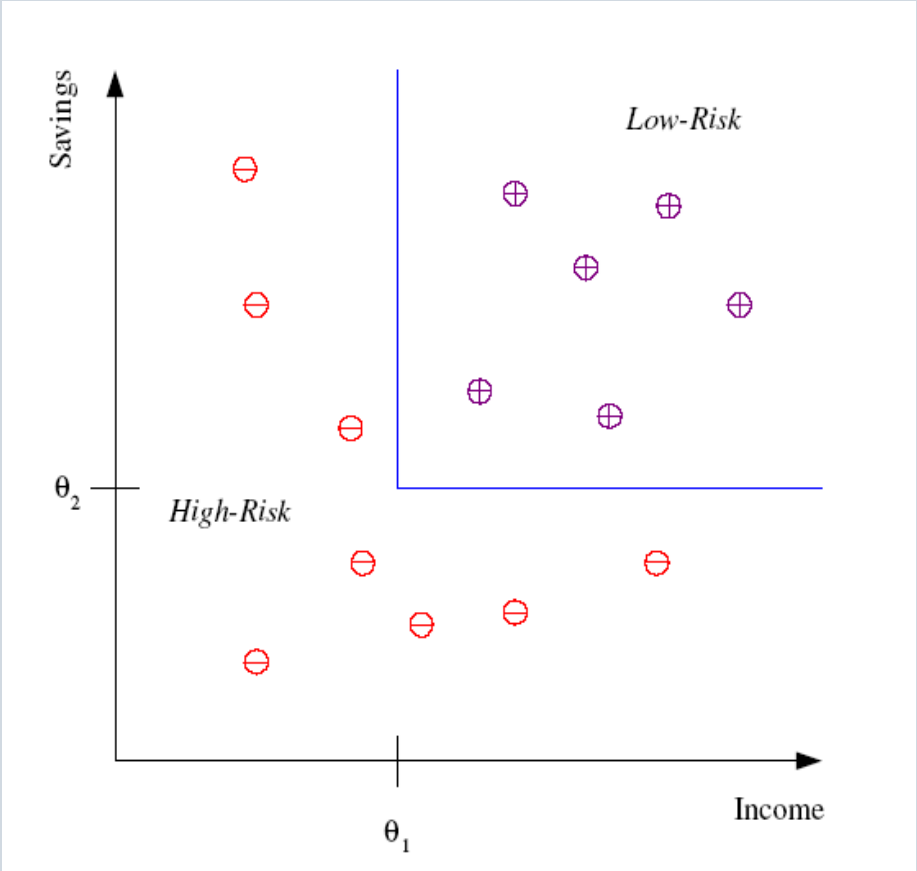
- Prediction of future cases: Use the rule to predict the output for future inputs
- Knowledge extraction: The rule is easy to understand
- Compression: The rule is simpler than the data it explains
- Outlier detection: Exceptions that are not covered by the rule, e.g., fraud

Supervised Learning: Classification

- Example: Credit scoring
- Differentiating between low-risk and high-risk customers from their *income* and *savings*

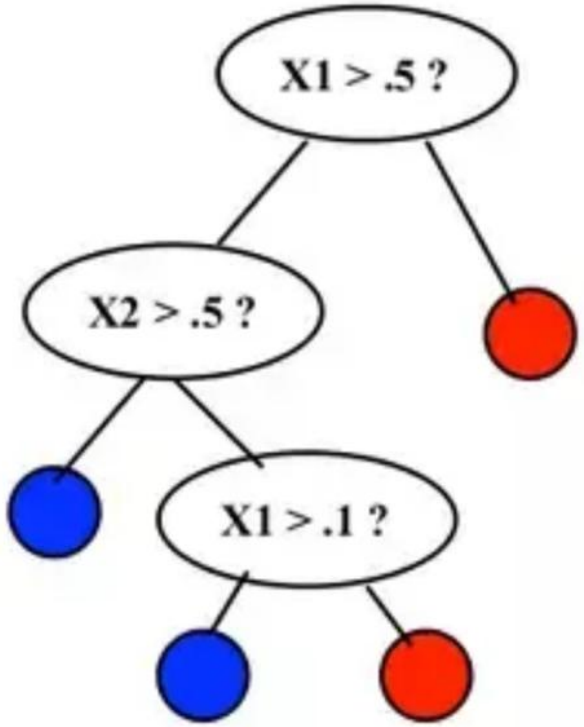
Discriminant: IF *income* > θ_1 AND *savings* > θ_2
THEN low-risk ELSE high-risk

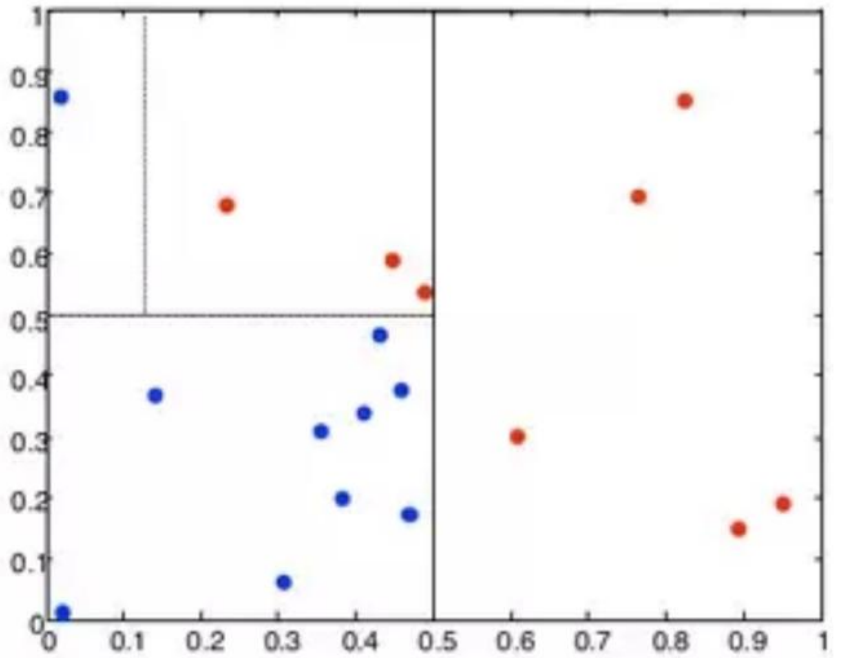
Model



www2.cs.uh.edu/~ceick/ML/ML-Topic1A.ppt

Decision Tree Classifier





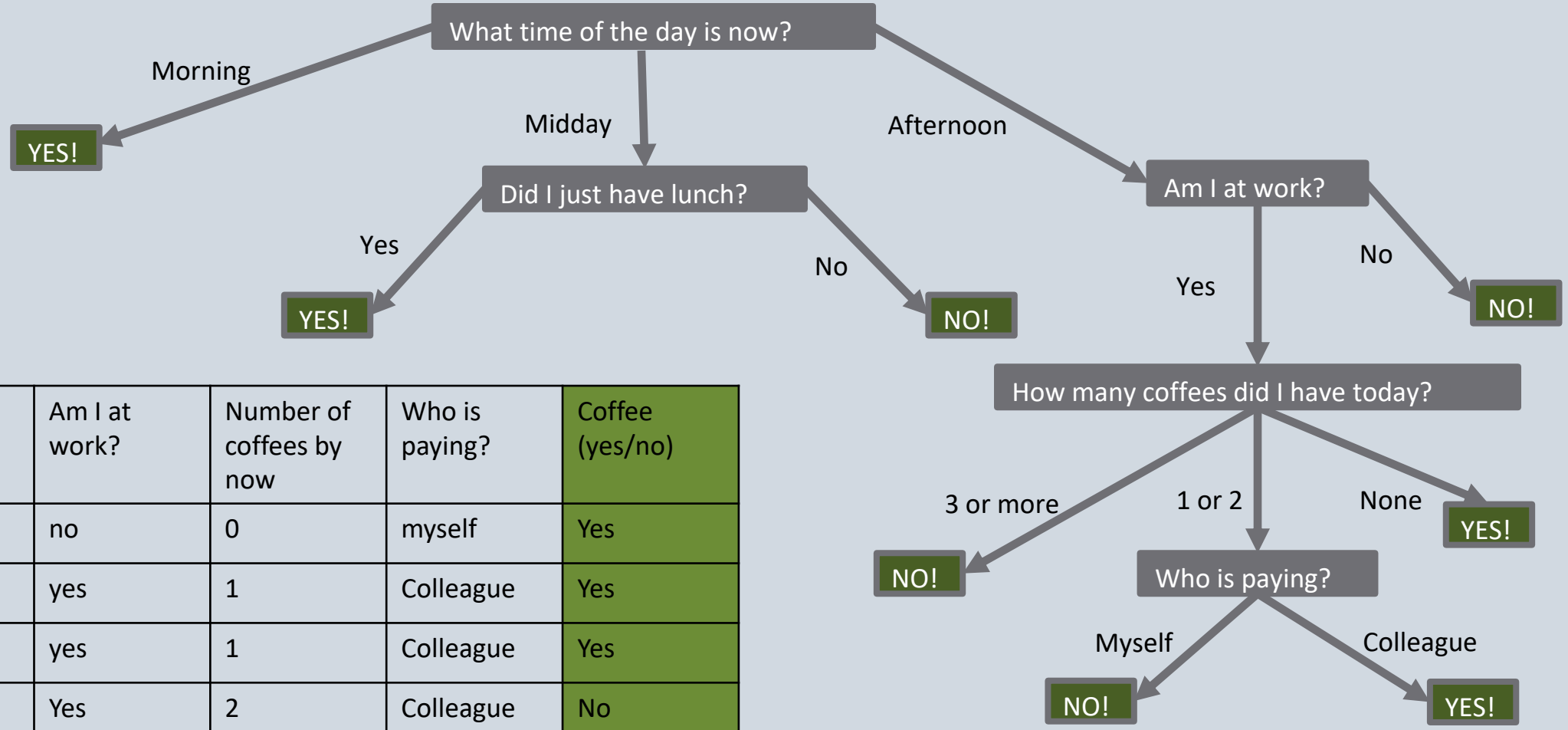
Application of Classification

- Medical diagnosis: From symptoms to illnesses
- Web Advertising: Predict if a user clicks on an ad on the Internet.
- Face recognition: Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Character recognition: Different handwriting styles.
- Speech recognition: Temporal dependency.
  - Use of a dictionary or the syntax of the language.
  - Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech

Classification Problem: Should I Have Coffee?

Time of day	Just had lunch	Am I at work?	Number of coffees by now	Who is paying?	Coffee (yes/no)
Morning	No	No	0	Myself	Yes
Morning	No	Yes	1	Colleague	Yes
Morning	No	Yes	1	Colleague	Yes
Midday	No	Yes	2	Colleague	No
Midday	No	Yes	2	Colleague	No
Midday	Yes	Yes	1	Myself	Yes
Midday	No	No	0	Colleague	No
Afternoon	No	Yes	2	Colleague	Yes

How Do I Decide Whether To Have Coffee?



Time of day	Just had lunch	Am I at work?	Number of coffees by now	Who is paying?	Coffee (yes/no)
morning	No	no	0	myself	Yes
morning	No	yes	1	Colleague	Yes
morning	No	yes	1	Colleague	Yes
Midday	No	Yes	2	Colleague	No
Midday	No	Yes	2	Colleague	No

Decision Tree Classifier – The Concept

- Many problems can be highly non-linear.
- Decision trees offer a way to define predictive model in terms of series of branching (Boolean) tests.

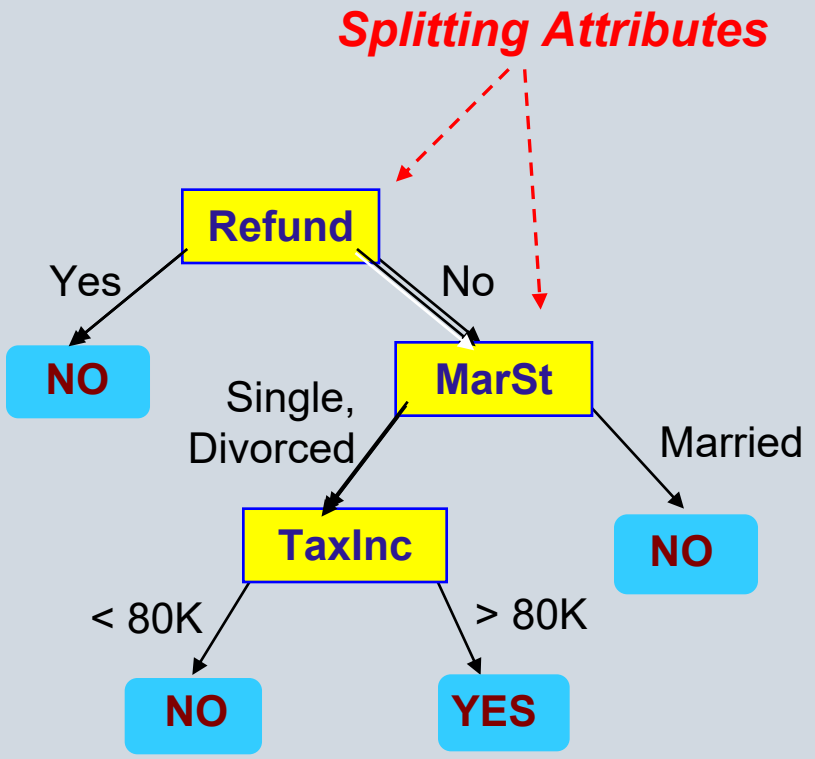
- Boolean trees
  - Can solve highly non-linear models
  - Break complex problems into small tests
- Decision trees are not necessarily Boolean (binary)

Decision Tree Classifier

- We arrive to conclusion by answering a *sequence of questions*
- Decision tree is a set of rules
- Not all attributes have to be used in each path of the decision.
- Does not need to be represented graphically
- In 1970s-80s, decision trees where programmed explicitly – Expert Systems (rule-based)
  - In reality, humans do not formulate algorithms, so it's not easy to define and program them.
  - Nearly impossible to update and maintain.

Fraud Detection: Example of a Decision Tree

	categorical	categorical	continuous	class
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

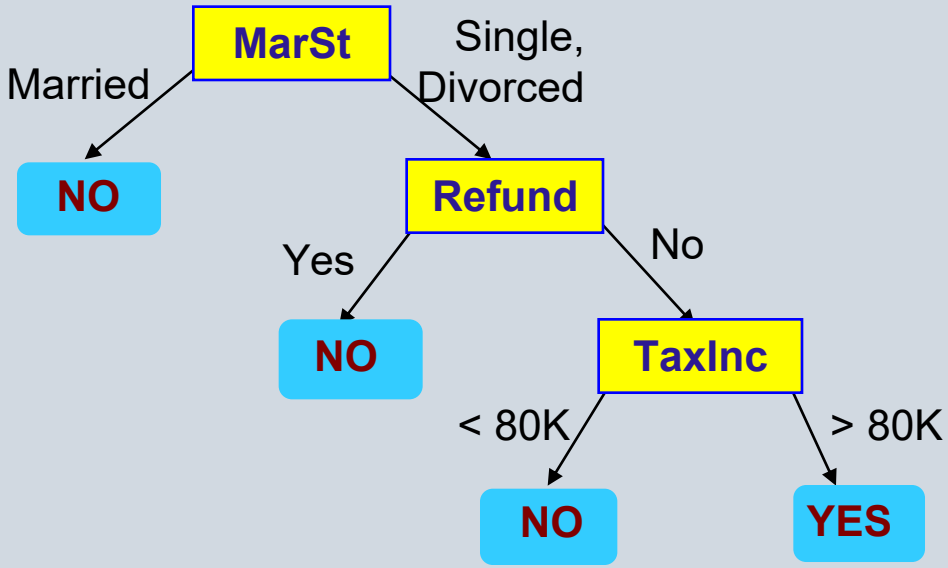


Training Data

Model: Decision Tree

Another Example of a Decision Tree

	categorical	categorical	continuous	class
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



There could be more than one tree that fits the same data!

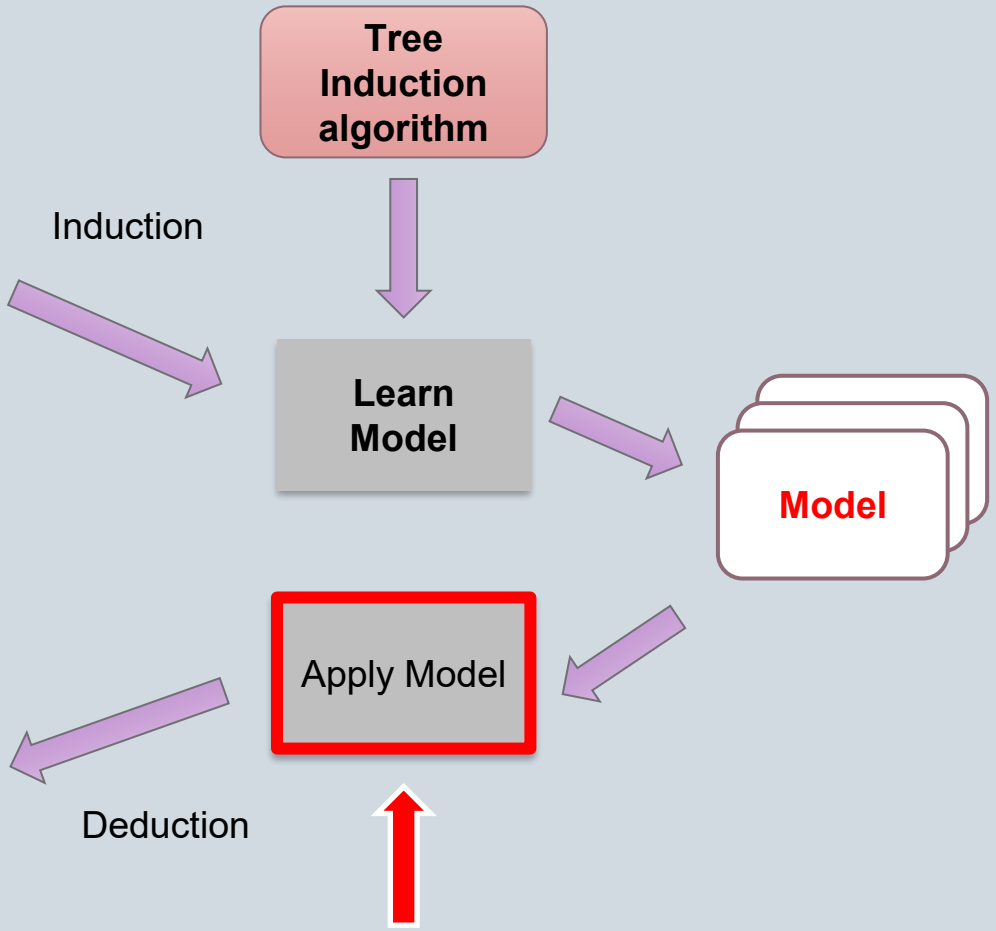
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

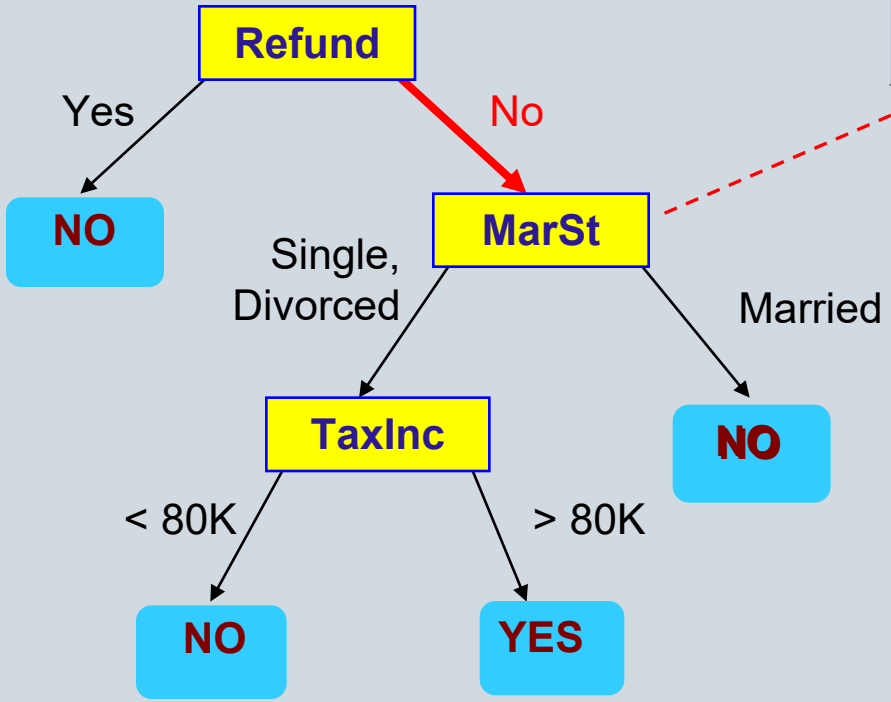
Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



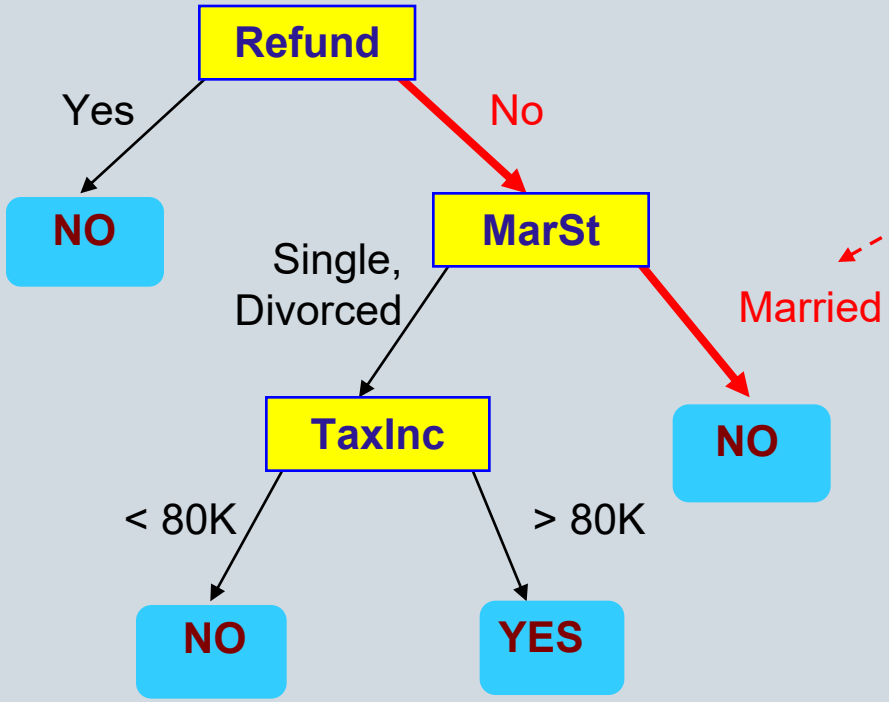
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

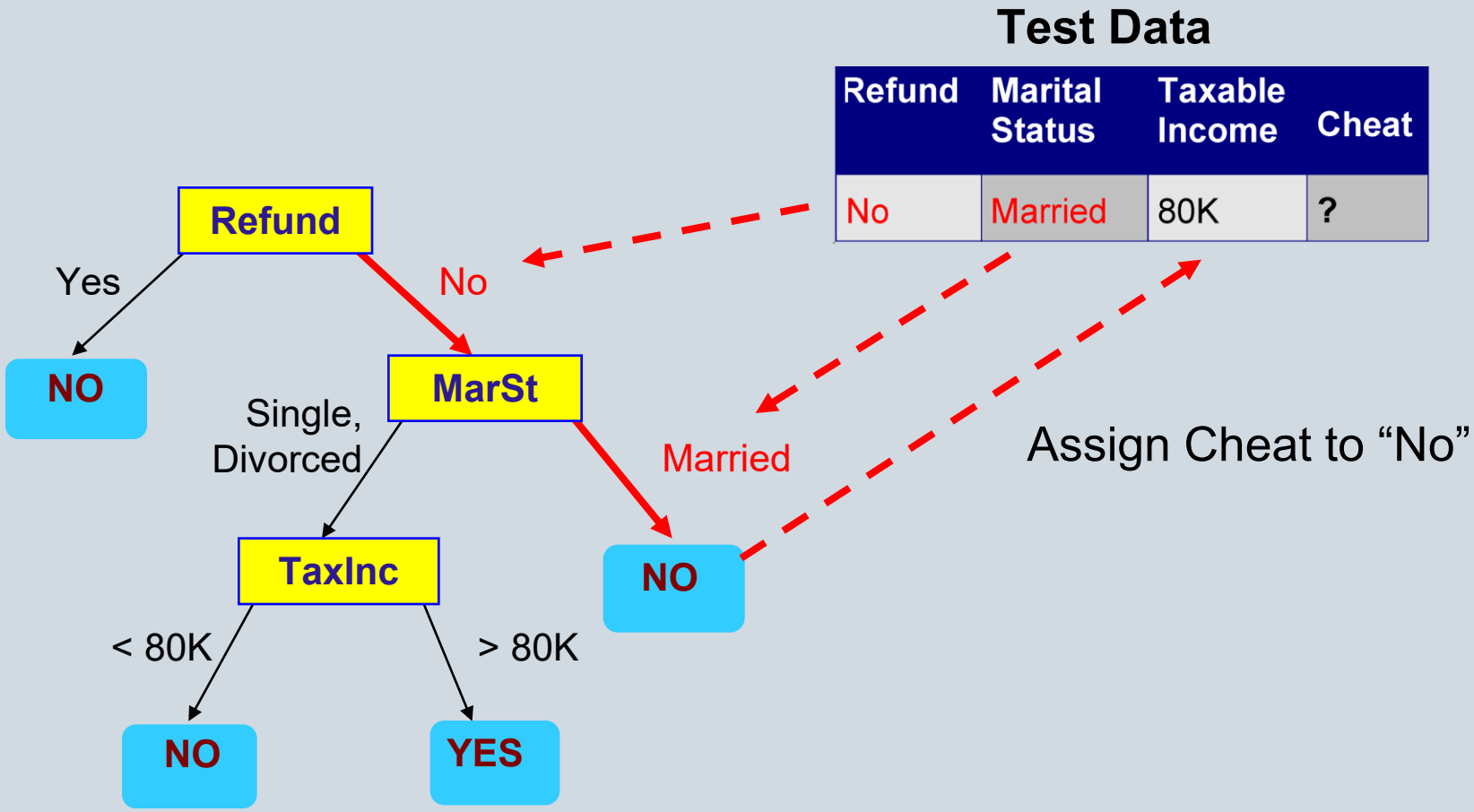
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

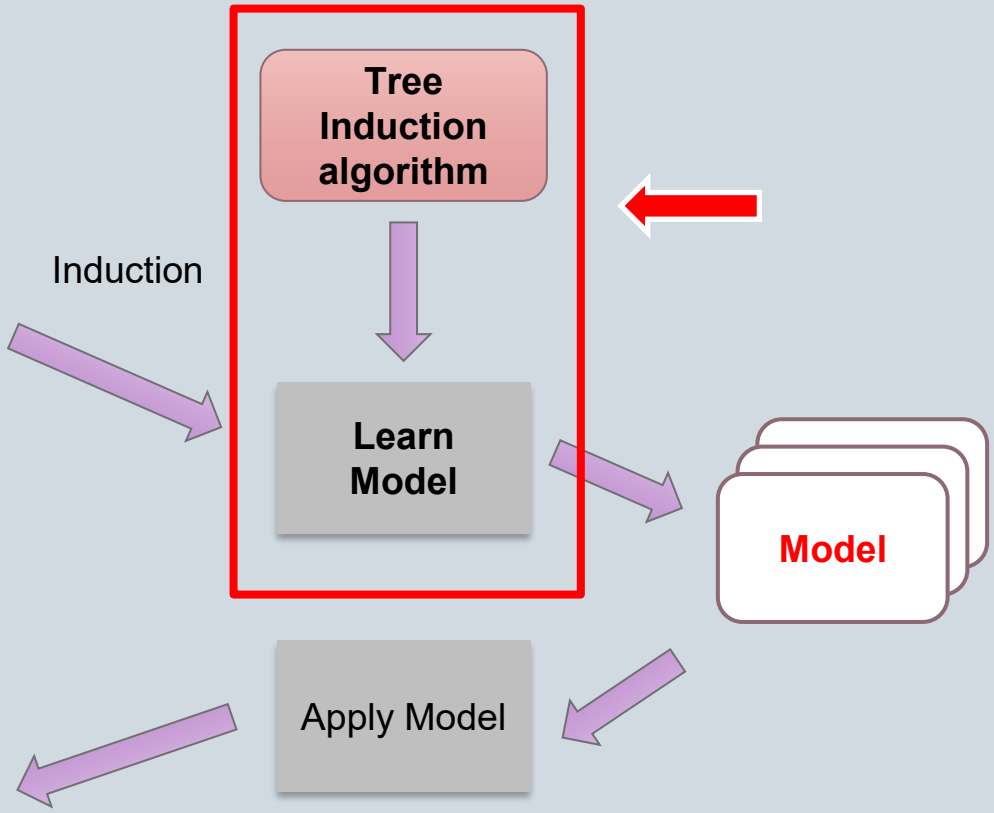
Apply Model to Test Data



Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



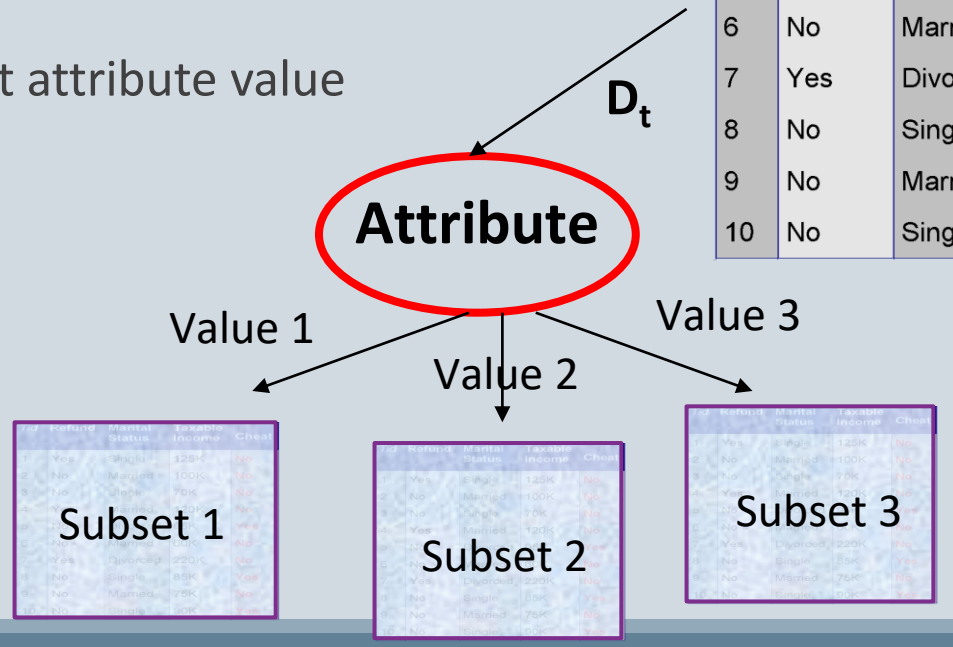
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Decision Tree Induction (Training)

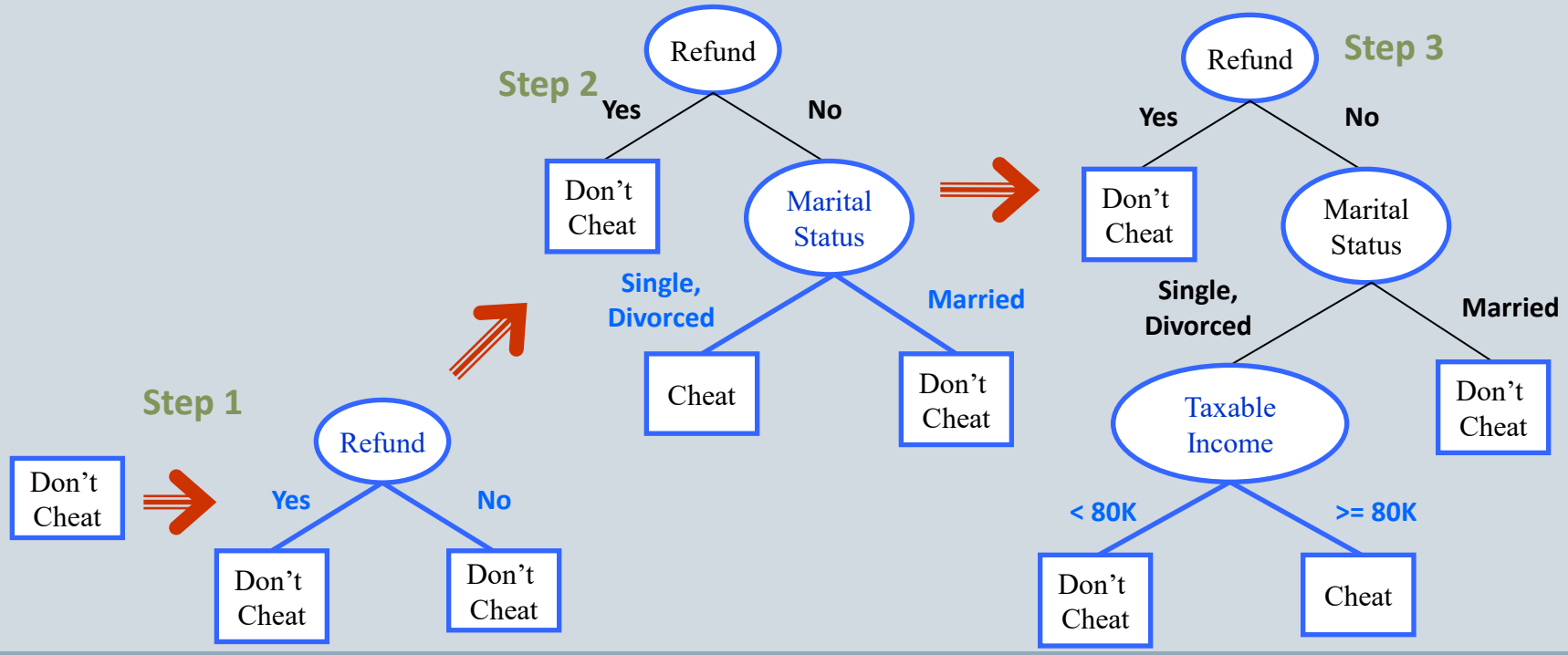
- Choose the *best* attribute(s) to split the remaining observations and make that attribute a decision node
- Repeat this process recursively for each child
- Stop when:
  - All the instances have the same target attribute value
  - There are no more attributes

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



Hunt's Algorithm

- Build a decision tree in a recursive fashion by partitioning the training records into successively purer subsets



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

- The main problem is selection of the next attribute to split the data upon.
- Greedy strategy
  - Split the records based on an attribute test that optimizes certain criterion
  - Optimization is local and does not necessarily lead to a globally optimized tree
- 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
    - avoid overfitting

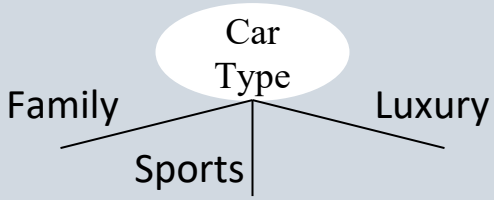
How To Specify Test Condition?

- Depends on attribute types
  - **Nominal** (named, categorical, e.g. marital status)
  - **Ordinal** (contains information about order, e.g. level of satisfaction)
  - **Continuous** (arbitrary numerical value, e.g. taxed income)
- Depends on the number of ways to split
  - 2-way split
  - Multi-way split

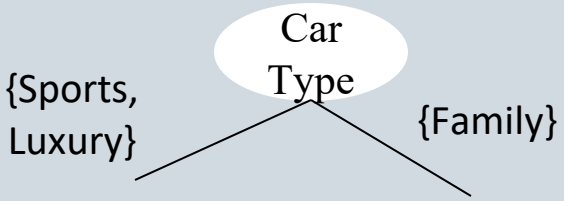
Splitting Based On Nominal Attributes

Nominal
Ordinal
Continuous

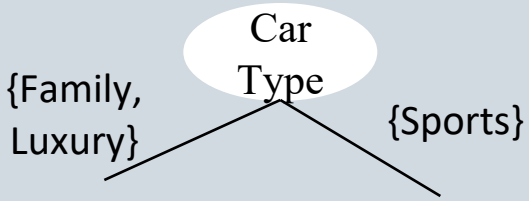
- Multi-way split: Use as many partitions as distinct values.



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



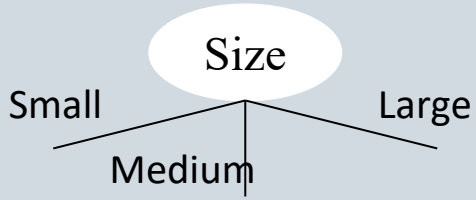
OR



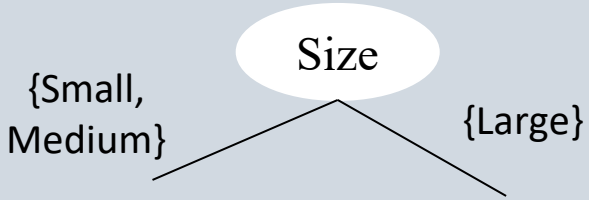
Splitting Based On Ordinal Attributes

Nominal
Ordinal
Continuous

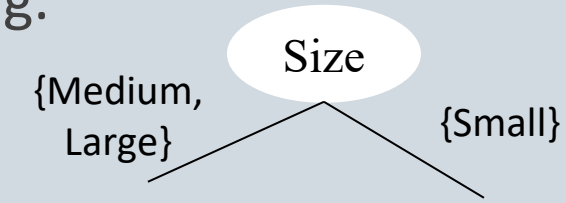
- Multi-way split: Use as many partitions as distinct values.



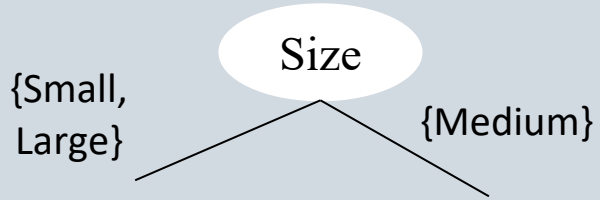
- Binary split: Divides values into two subsets.
  Need to find optimal partitioning.



OR



- What about this split?

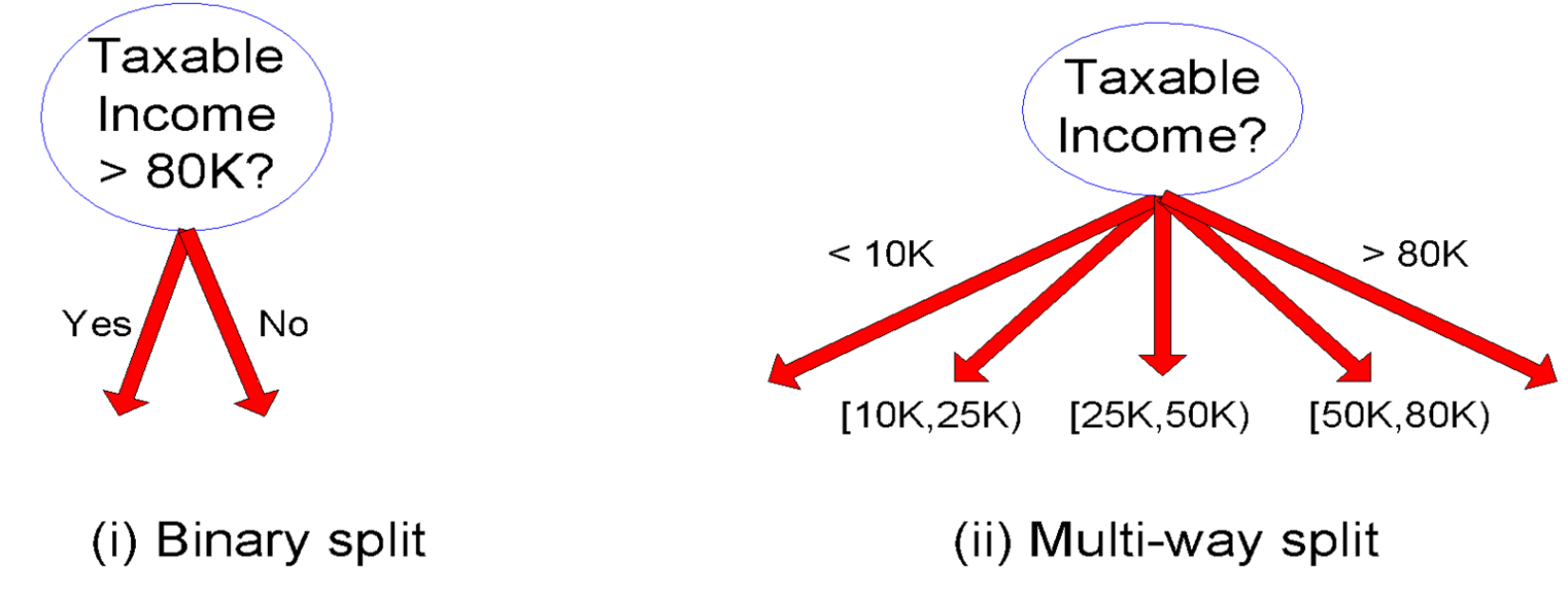


Splitting Based On Continuous Attributes

Nominal
Ordinal
Continuous

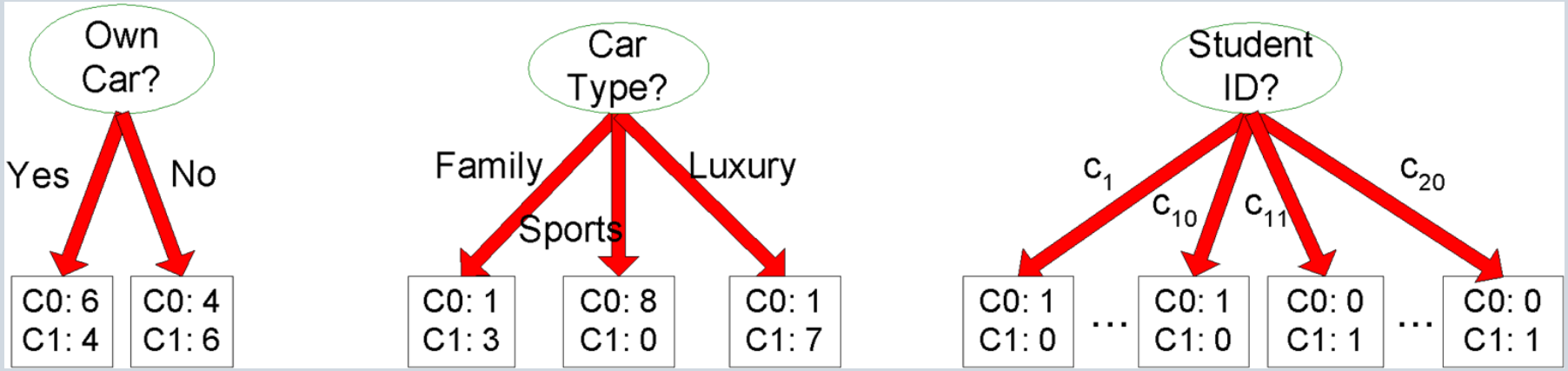
- Different ways of handling
  - Discretization to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
  - Binary Decision: $(A < v)$ or $(A \geq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

Splitting Based On Continuous Attributes



How To Determine The Best Split

Before Splitting: 10 records of class 0,
10 records of class 1



Which test condition is the best?

How To Determine The Best Split

- Main questions:
  - How can we judge whether a variable contains important information about the target variable?
  - How can we (automatically) obtain a selection of the more informative variables with respect to predicting the value of the target variable?
  - Even better, can we obtain the ranking of the variables?
- Greedy approach:
  - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity:

C0: 5
C1: 5

Non-homogeneous,

High degree of impurity

C0: 9
C1: 1

Homogeneous,

Low degree of impurity

Measures of Node Impurity

- Entropy

$$\boldsymbol{Entropy(t) = -\sum p(x)\log[p(x)]}$$

Gini Index

$$\boldsymbol{Gini(t) = 1 - \sum [p(j|t)]^2}$$

- Misclassification Error:

$$\boldsymbol{Error(t) = 1 - \max P(i|t)}$$

Measures of Node Impurity: Entropy

Target variable has two (or more) categories: 1,2 (,…m)

Probability P1 for category 1

Probability P2 for category 2

….

Entropy:

$$S = -p_1 log(p_1) - p_2 log(p_2) - \ \ldots - p_m log(p_m)$$

$$\boldsymbol{Entropy(t) = -\sum p(x) \log[p(x)]}$$

Maximum (log n_c) when records are equally distributed among all classes, implying least information

Minimum (0, 0) when all records belong to one class, implying most information

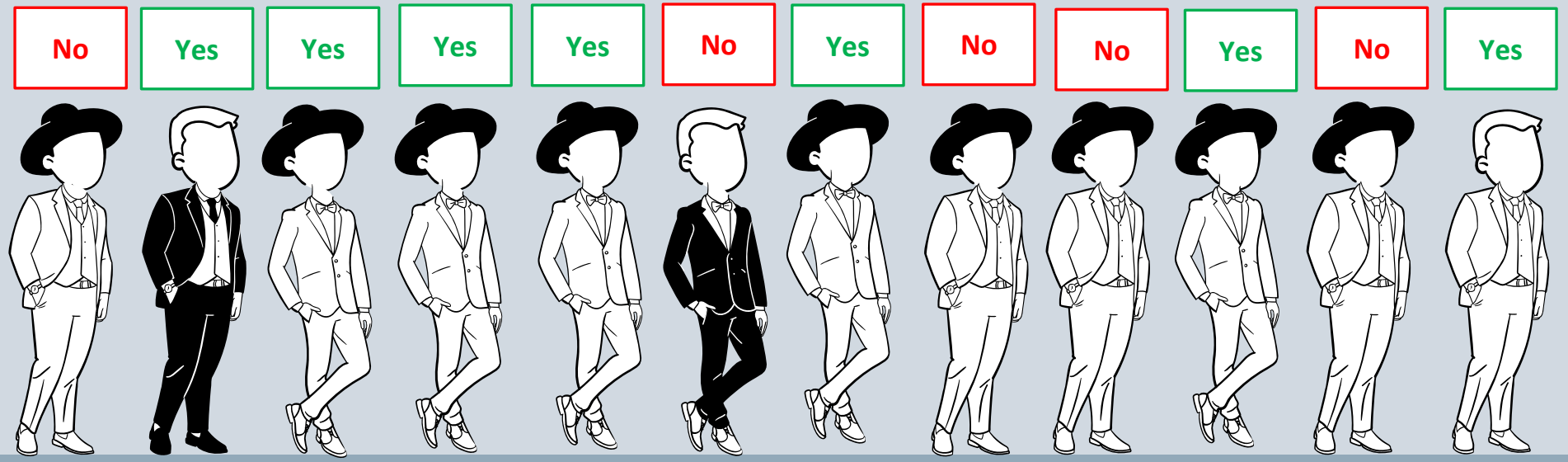
Measures of Node Impurity: Dataset

Attributes:

- Hat: Yes, No
- Wear: tie, bow-tie
- Clothing color: Black ,White

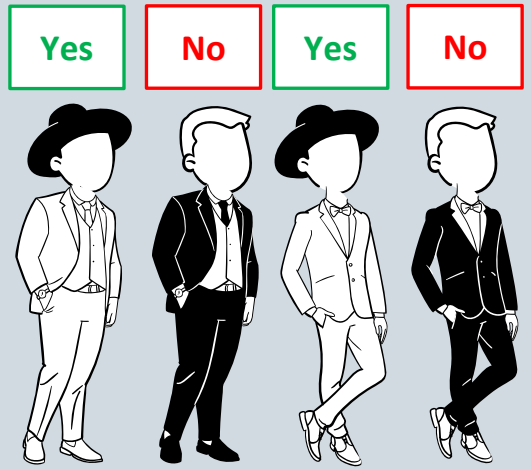
Target variable:

- Will buy the product? : Yes, No

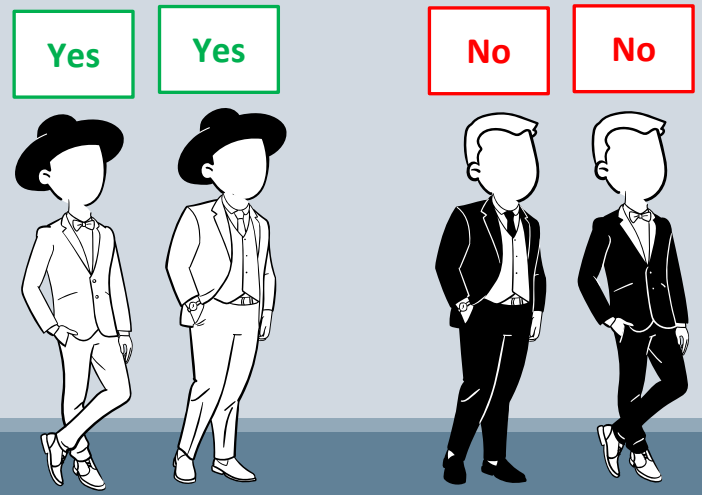


Splitting the Data

- Example, if this was our entire data:



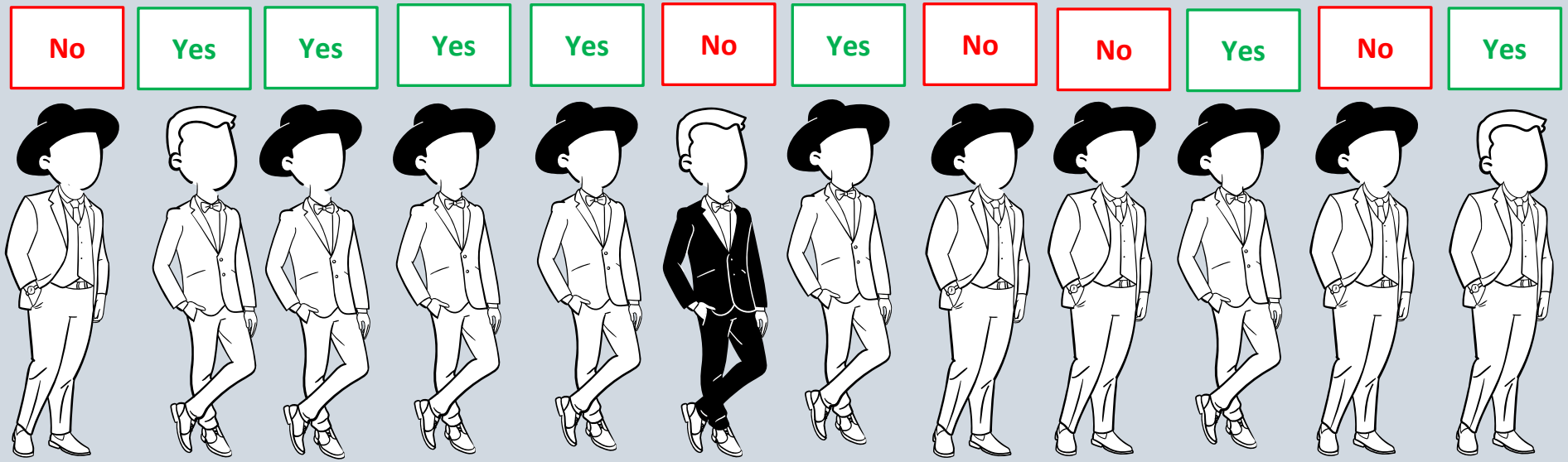
- Then, we can obtain two pure groups by splitting according to whether the person ware a hat or not



Challenges

Attribute rarely split a group perfectly

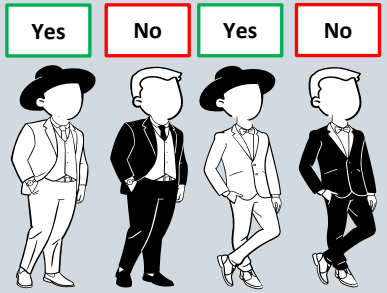
Even if one subgroup happens to be pure, the other may not



Is a very small, pure group, a good thing?

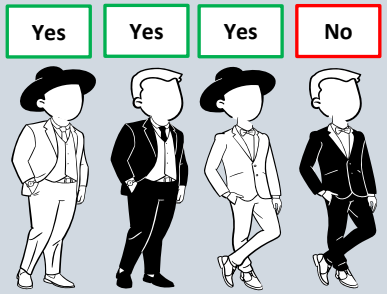
Handling continuous and categorical attributes

Lower Entropy =>“Purer” Group

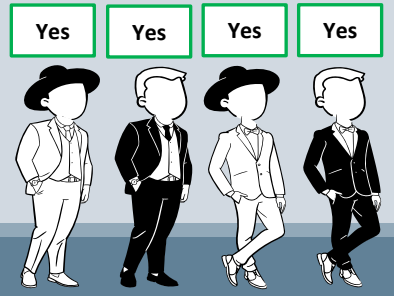


$$H(X) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \cdots - p_m \log_2 p_m$$

$$H(X) = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$



$$H(X) = -0.75 \log_2 0.75 - 0.25 \log_2 0.25 = 0.81$$



$$H(X) = -1 \log_2 1 = 0$$

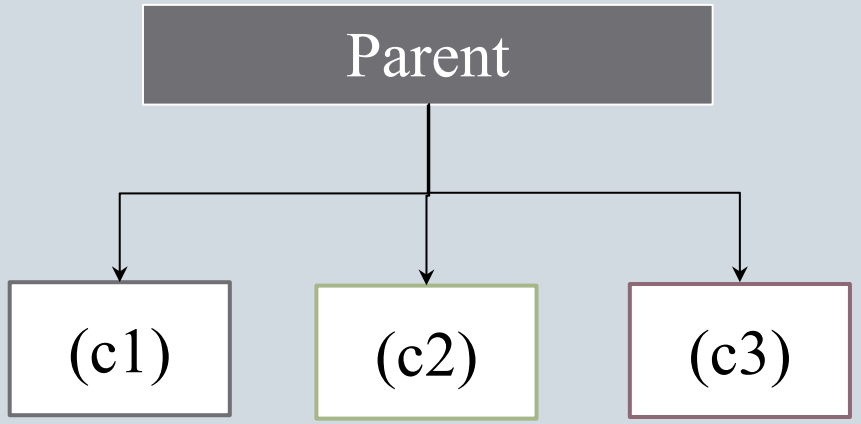
Information Gain

- Specifically, the definition of information gain (IG) is:
- IG (parent, children) =

S(parent)−[p(c1)×entropy(c1)+p(c2)×entropy(c2) + · ·]

$$\boldsymbol{Information\ Gain(split) = entropy(p) - \sum_{i=1}^{k} \frac{n_i}{N} entropy(i)}$$

Parent Node, p is split into k partitions;
n_i is number of records in partition



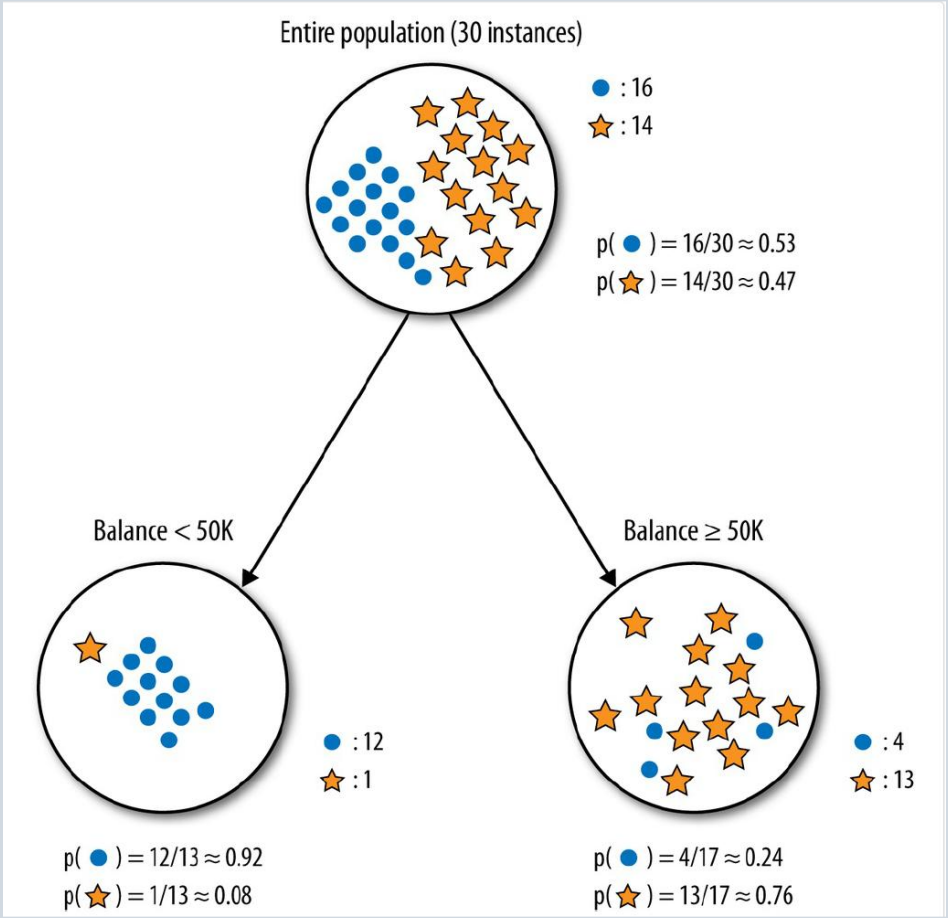
Note: Higher IG indicates a more informative split by the variable

- Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN).
- Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

Information Gain

person id	age>50	gender	residence	Balance>=50,000	mortgage payment delay
123213	N	F	own	N	delayed
17824	Y	M	own	Y	OK
232897	N	F	rent	N	delayed
288822	Y	M	other	N	delayed
….	….	….	….	….	….

Information Gain

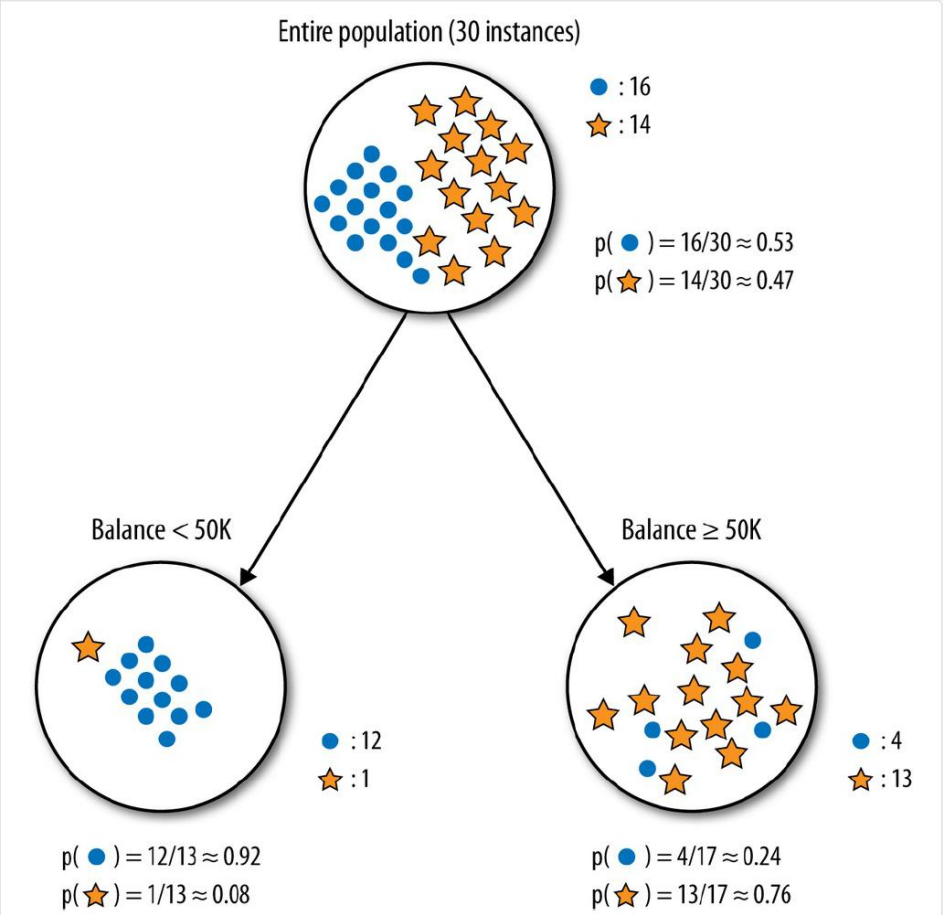


● - delay
★ - OK

Source: “Data Science for Business”, Provost, Fawcett, 2013

Information Gain

- delay
- OK



Entropy(parent)

$= -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)]$

$= -[0.53 \times (-0.9) + 0.47 \times (-1.1)]$

$= 0.99$ (very impure!)

Left child: entropy(Balance < 50K)

$= -[p(\bullet)\ \log_2 p(\bullet) + p(\star)\ \log_2 p(\star)]$

$= -[0.92 \times (-0.12) + 0.08 \times (-3.7)]$

$= 0.39$

Right child: entropy(Balance ≥50K)

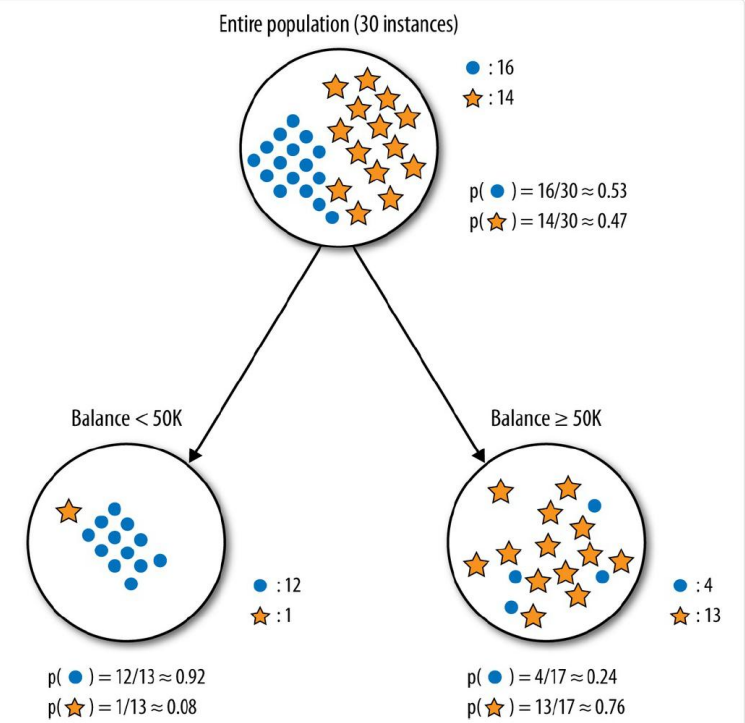
$= -[p(\bullet)\ \log_2 p(\bullet) + p(\star)\ \log_2 p(\star)]$

$= -[0.24 \times (-2.1) + 0.76 \times (-0.39)]$

$= 0.79$

Source: “Data Science for Business”, Provost, Fawcett, 2013

Information Gain



Entropy(parent)

= 0.99

Left child: entropy(Balance < 50K)

= 0.39

Right child: entropy(Balance ≥50K)

= 0.79

IG for the split based on "balance" variable :

IG = entropy(parent) − [p(Balance < 50K) × entropy(Balance < 50K)

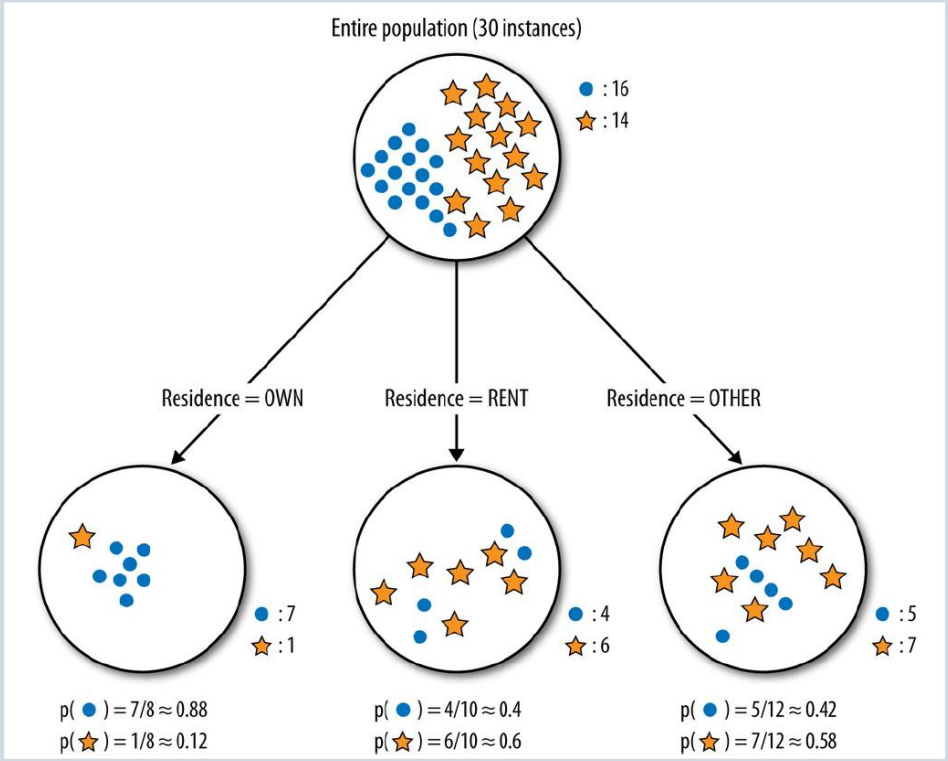
+p(Balance ≥ 50K) × entropy(Balance ≥ 50K)]

= 0.99 − [0.43 × 0.39 + 0.57 × 0.79]

= 0.37

Source: "Data Science for Business", Provost, Fawcett, 2013

Information Gain



- delay
- OK

entropy(parent) =0.99

entropy(Residence=OWN) =0.54

entropy(Residence=RENT) =0.97

entropy(Residence=OTHER) =0.98

IG = 0.13

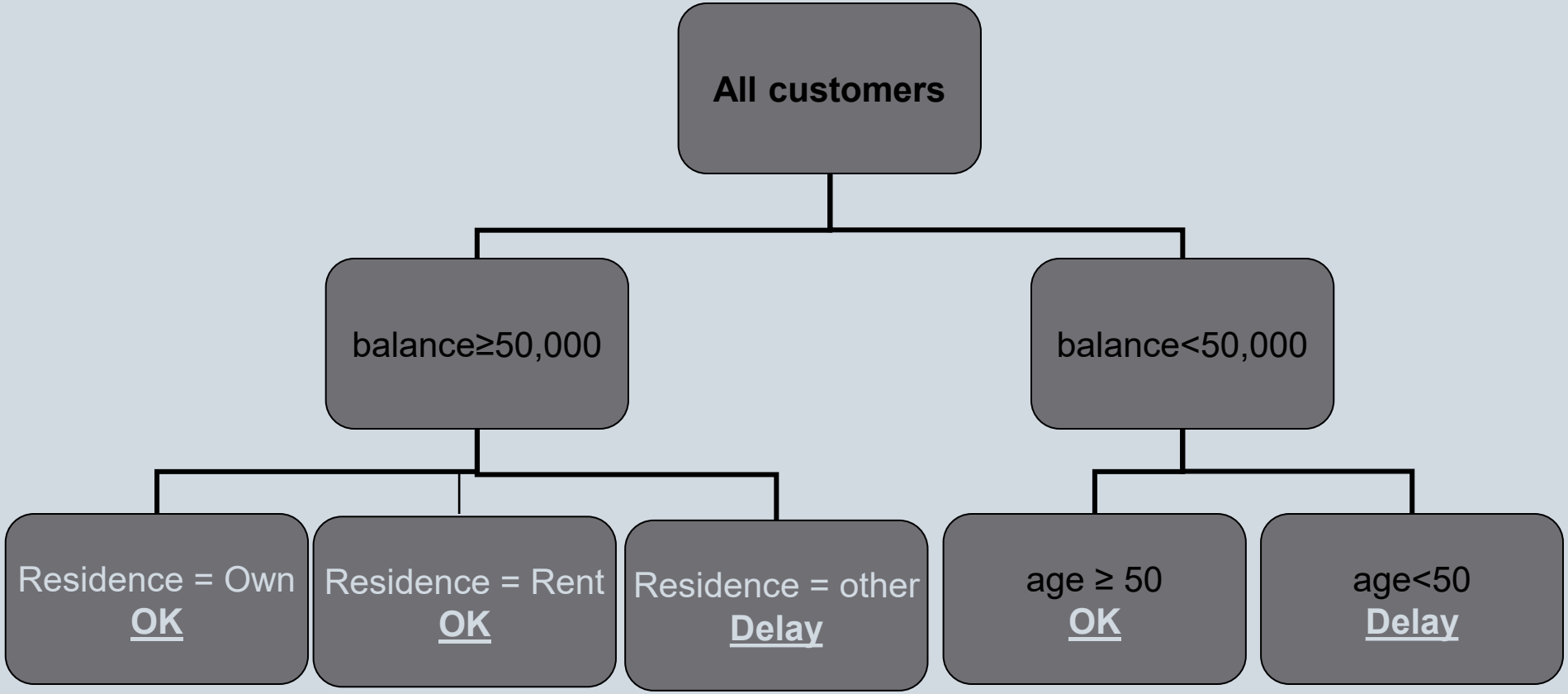
Source: “Data Science for Business”, Provost, Fawcett, 2013

41

Sample Tree Fit

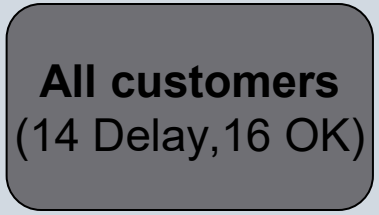
person id	age>50	gender	residence	Balance>= 50,000	mortgage payment delay
123213	N	F	own	N	delayed
17824	Y	M	own	Y	OK
232897	N	F	rent	N	delayed
288822	Y	M	other	N	delayed
….	….	….	….	….	….

Sample Tree Fit



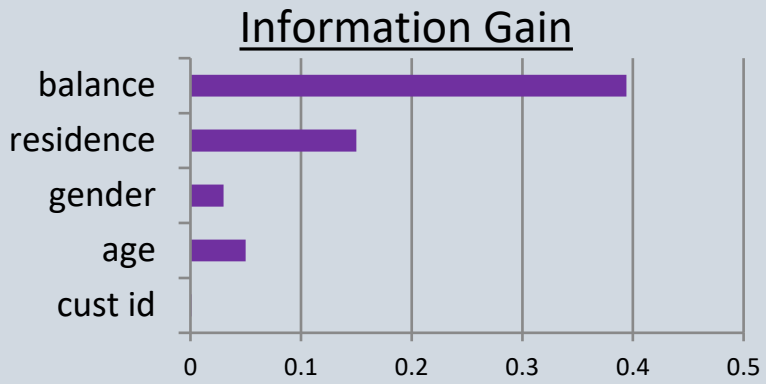
43

Sample Tree Fit



44

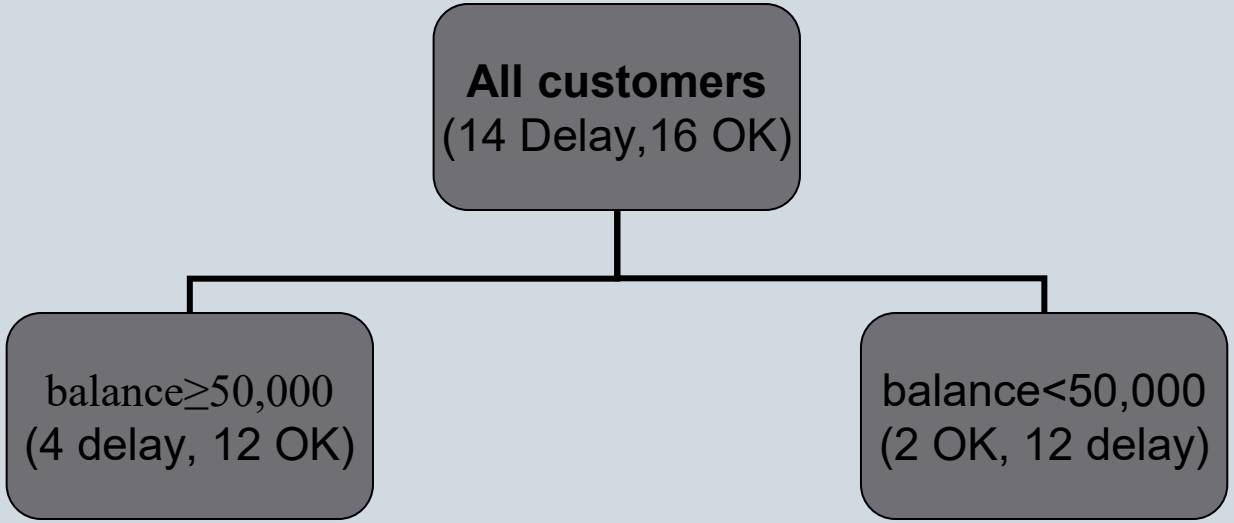
Sample Tree Fit



All customers
(14 Delay,16 OK)

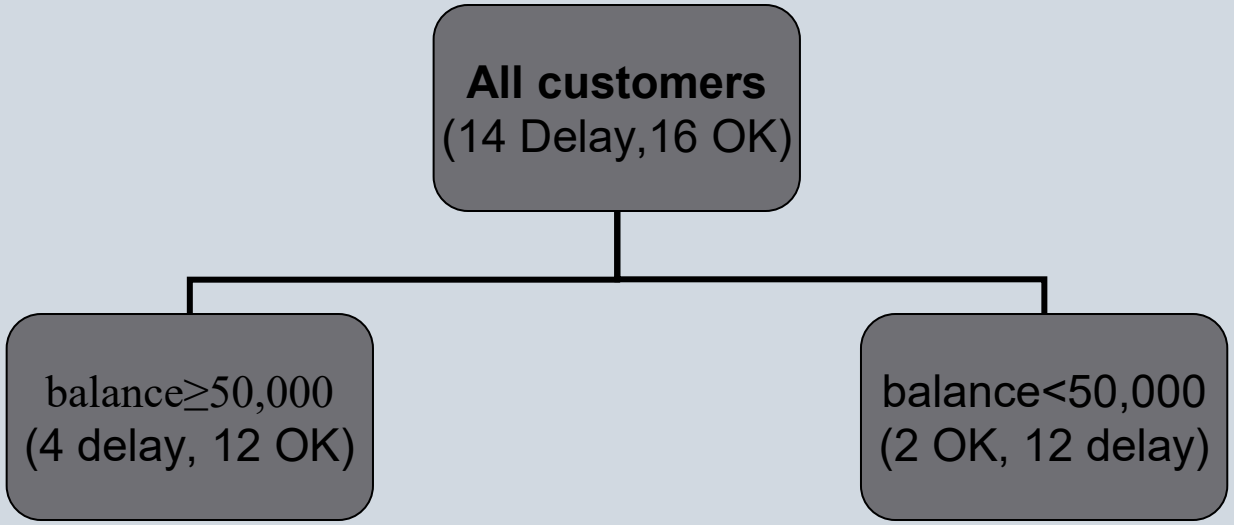
45

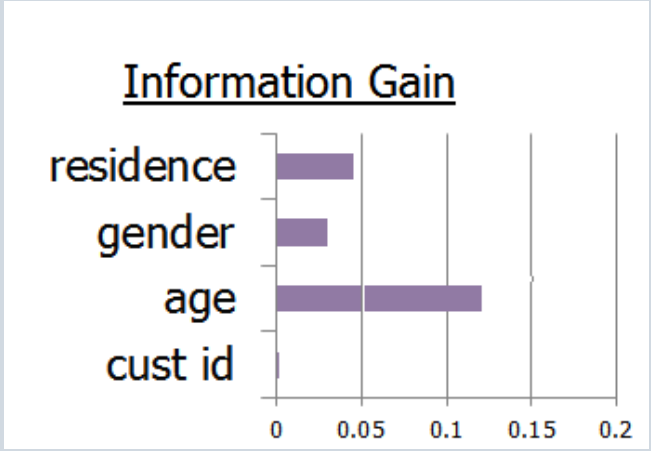
Sample Tree Fit



46

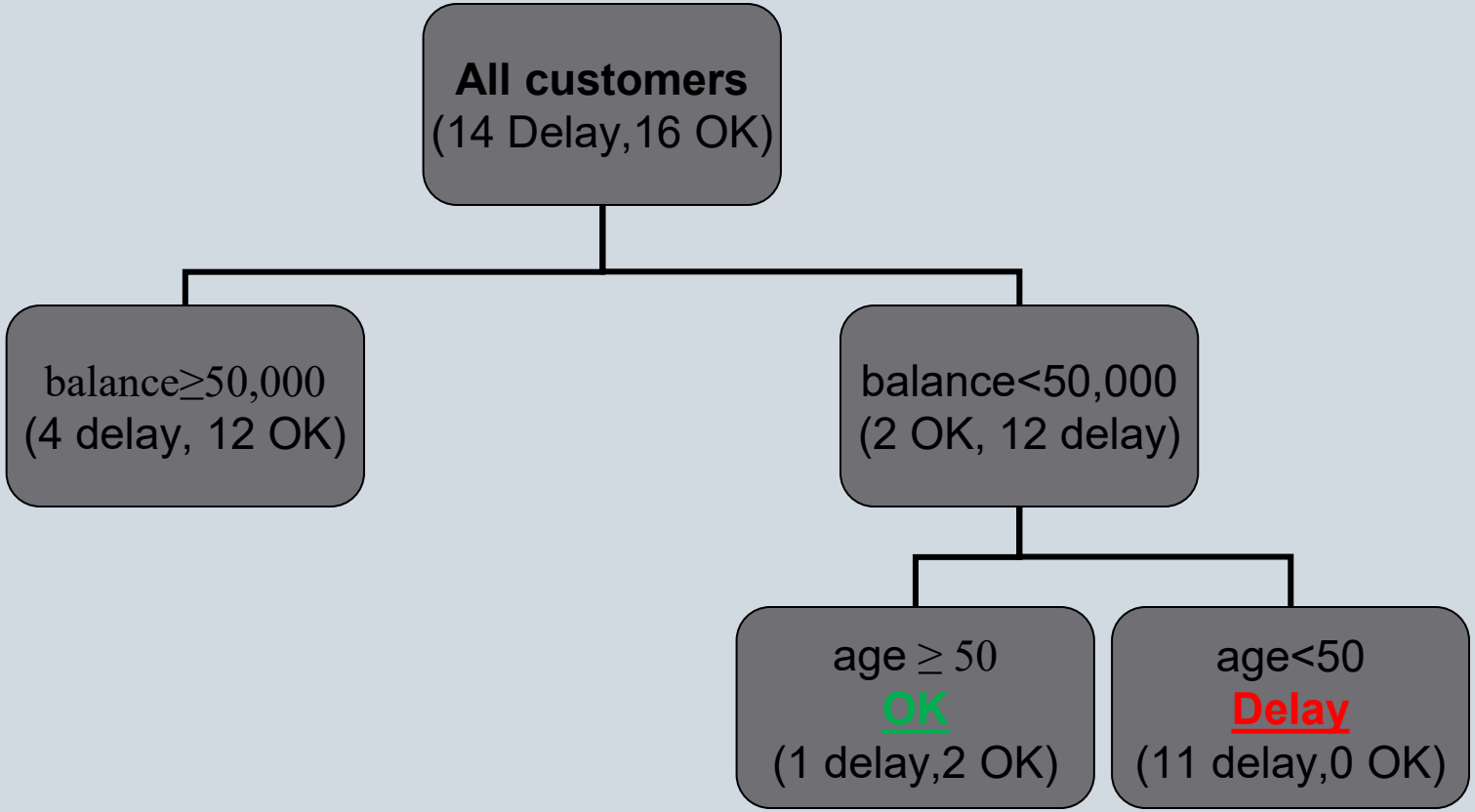
Sample Tree Fit



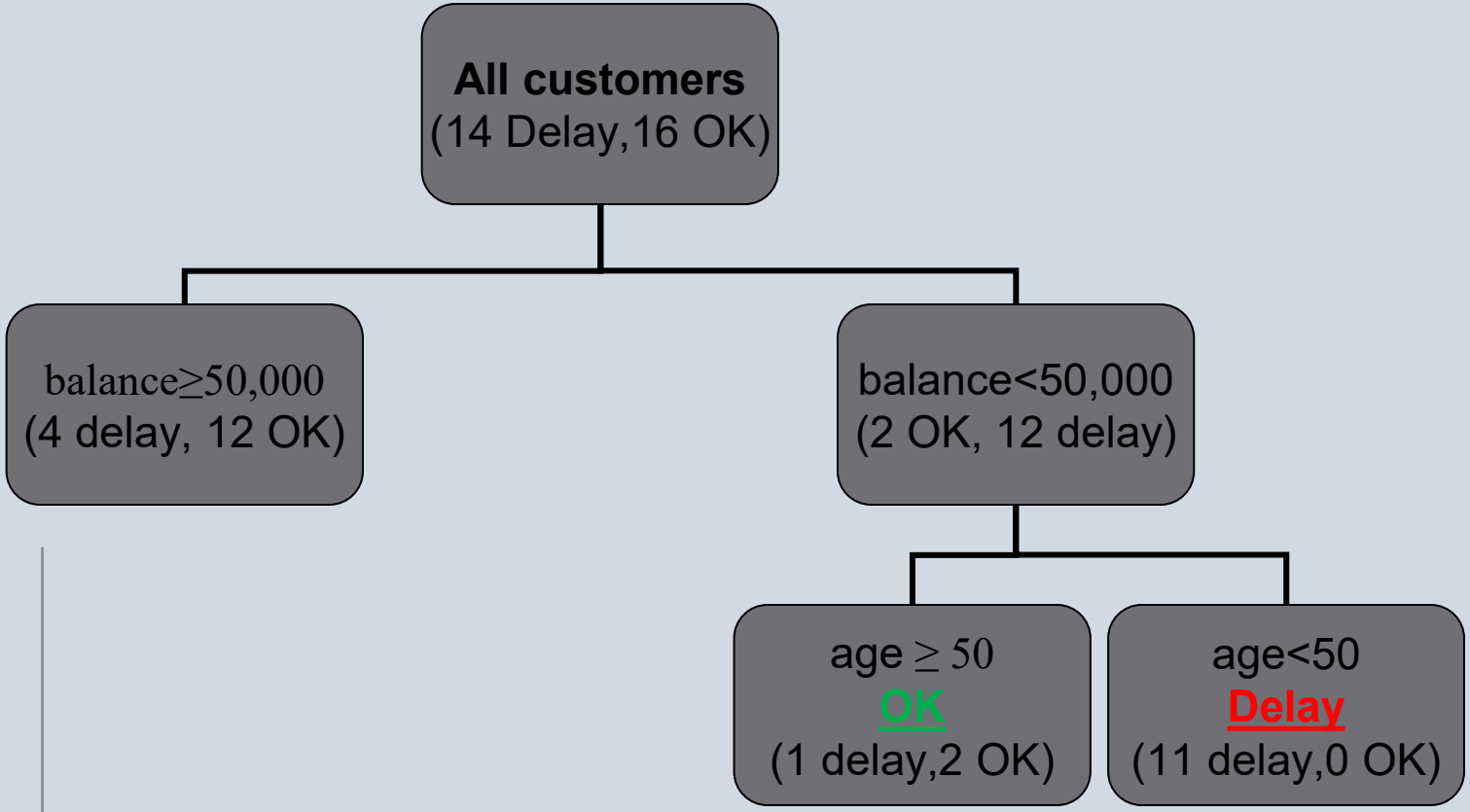


47

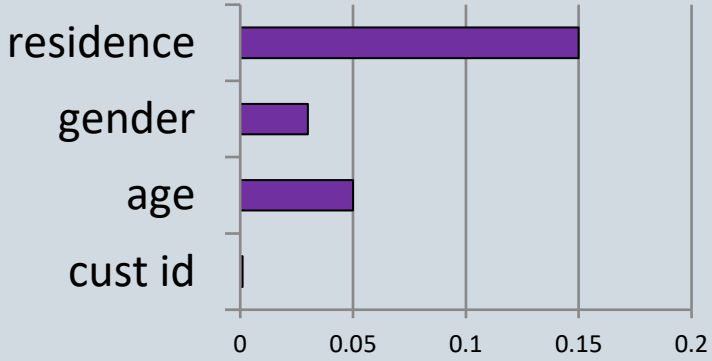
Sample Tree Fit



Sample Tree Fit

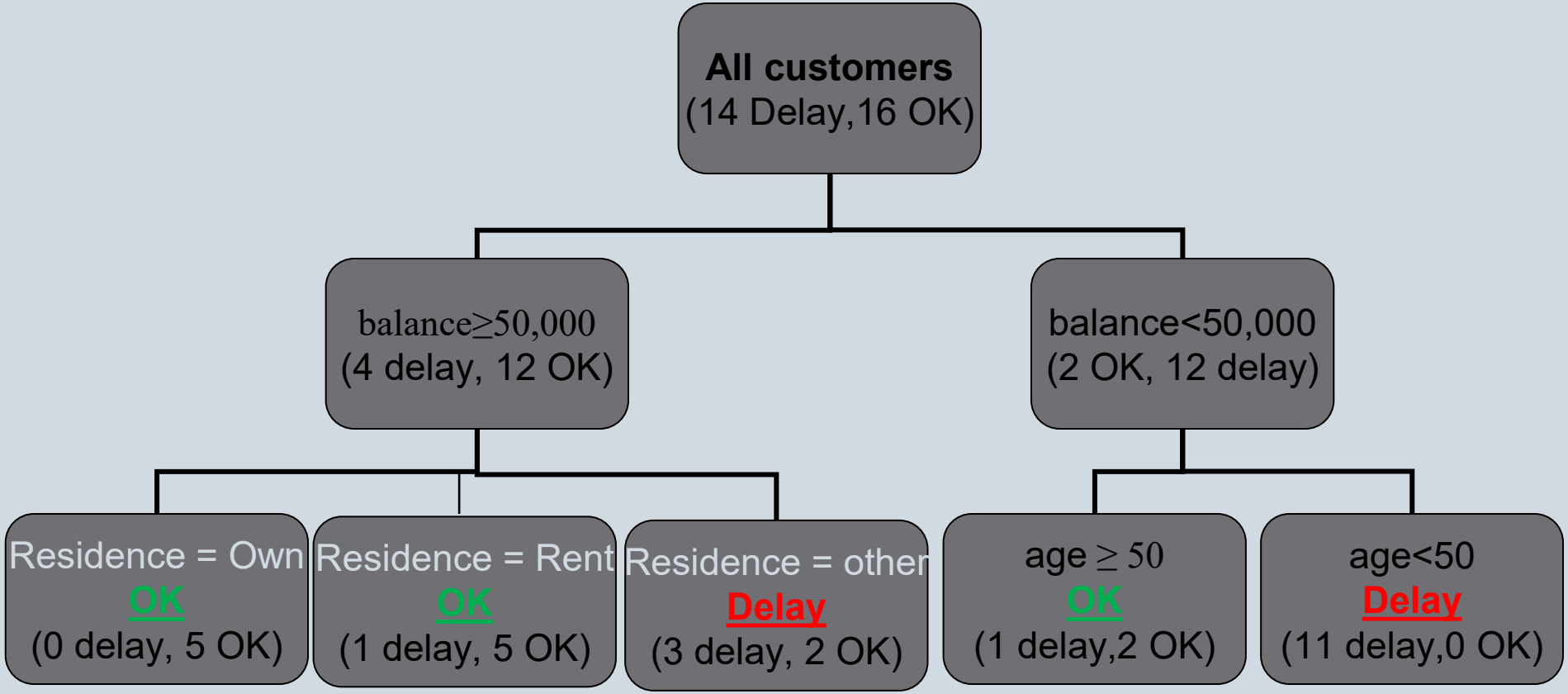


Information Gain

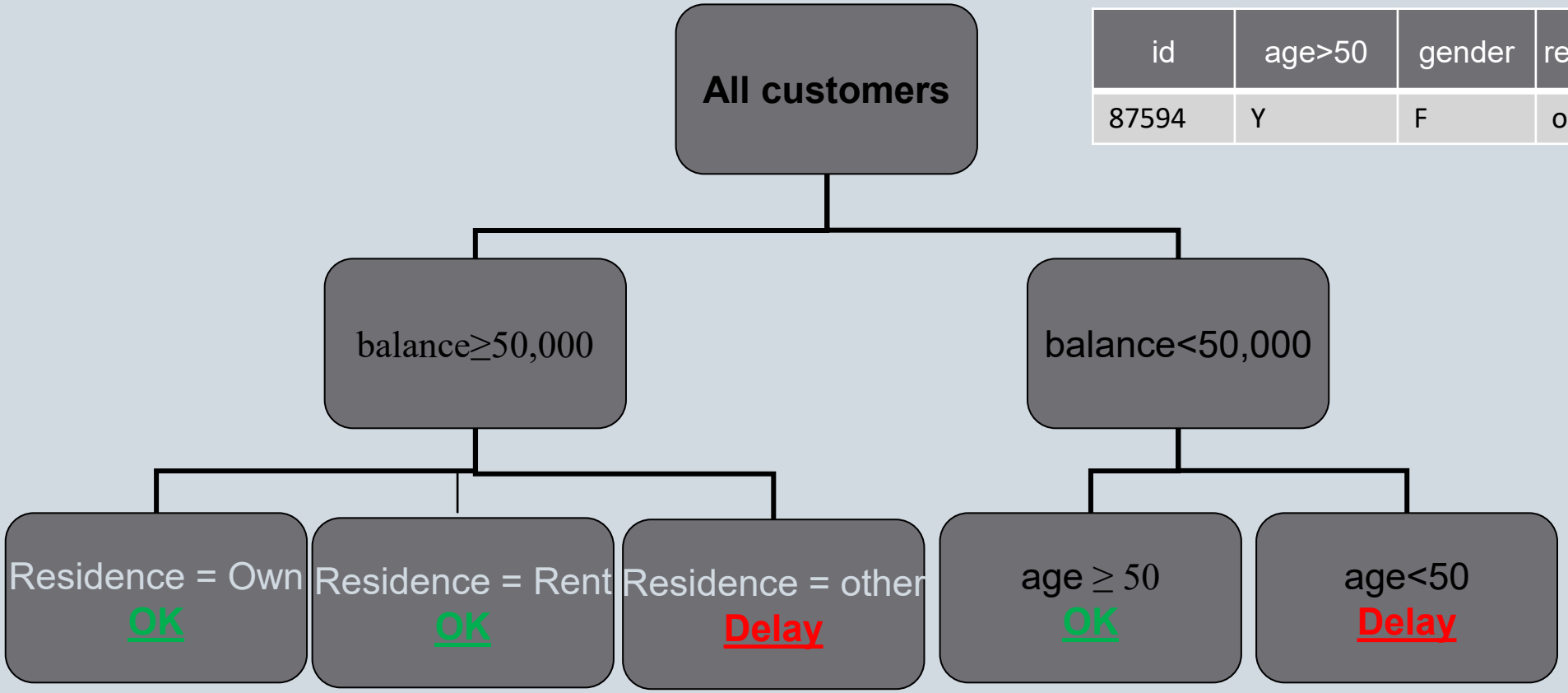


49

Sample Tree Fit



Sample Tree Fit



id	age>50	gender	residence	Balance>=50K	delay
87594	Y	F	own	<50K	???

Splitting Based On Info

- Gain Ratio

$$GainRATIO_{split} = \frac{Gain_{split}}{SplitINFO} \qquad SplitINFO = \sum_{i=1}^{k} \frac{n_i}{N} log \frac{n_1}{N}$$

  - Present Node, p is split into k partitions
  - n_i is the number of records in partition I

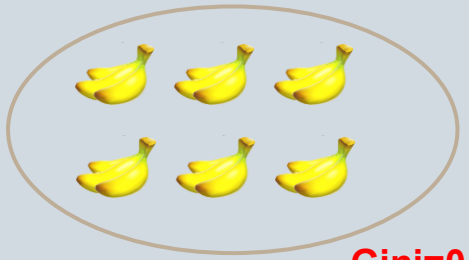
- Adjust Information Gain by the entropy of the partitioning (SplitINFO)
  - Higher entropy partitioning (large number of small partitions) is penalized!
- Designed to overcome the disadvantage of Information Gain

Measures of Node Impurity: Gini

- Gini Impurity Index for a given node t: $$Gini(t) = 1 - \sum [p(j|t)]^2$$
  - P(J|t) is the relative frequency of class j at node t

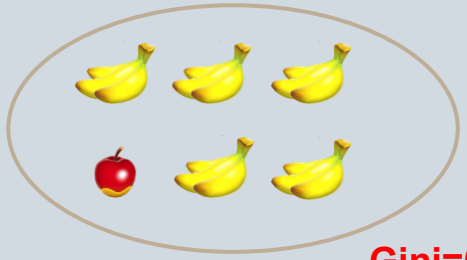
When building a tree, pick the split that minimizes Gini of the produced separation

- Maximum $(1 - \frac{1}{n_c})$ when records are equally distributed among all classes, implying less interesting information.
- Minimum (0.0) when all records belong to one class, implying most interesting information.



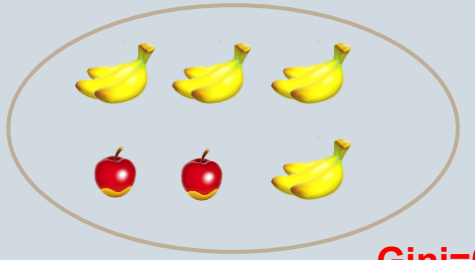
Gini=0

$$1 - \left(\frac{0}{36} + \frac{36}{36}\right) = 0$$



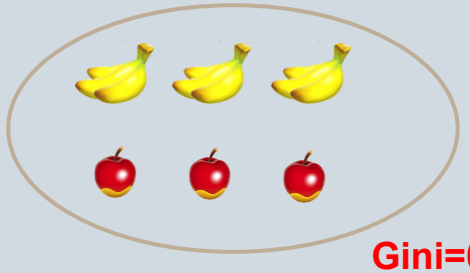
Gini=0.278

$$1 - \left(\frac{1}{36} + \frac{25}{36}\right) = 0.278$$



Gini=0444

$$1 - \left(\frac{4}{36} + \frac{16}{36}\right) = 0.444$$



Gini=0.5

$$1 - \left(\frac{9}{36} + \frac{9}{36}\right) = 0.5$$

DO NOT CONFUSE WITH GINI COEFFICIENT

Splitting Based on Gini

- Used in decision trees classifiers such as CART, SLIQ, SPRINT.
- When a node p is split into k partitions (children), the quality of the split is computed as:

$$Gini_{split} = \sum_{i=1}^{k} \frac{n_i}{N} Gini(i)$$

Where, n_i = number of records in child i,

n= number of records at node p.

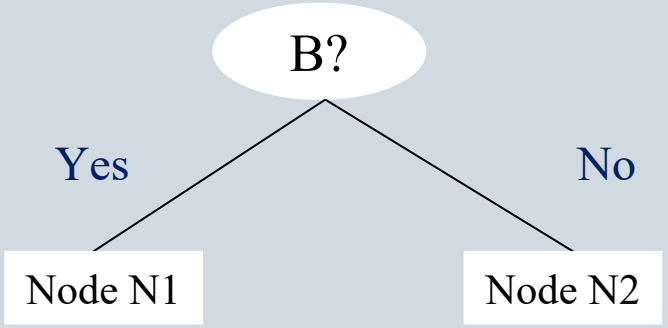
Binary Attributes: Computing Gini Impurity Index

- Splits into two partitions
- Effect of Weighing partitions:

  Larger and Purer Partitions are sought for.

	Parent
C1	6
C2	6
Gini=0.5	

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



$$Gini_{split} = \sum_{i=1}^{k} \frac{n_i}{N} Gini(i)$$

$$Gini(n_1) = 1 - \left(\left[\frac{5}{7}\right]^2 + \left[\frac{2}{7}\right]^2\right) = 0.408$$

$$Gini(n_2) = 1 - \left(\left[\frac{1}{7}\right]^2 + \left[\frac{4}{7}\right]^2\right) = 0.320$$

	N1	N2
C1	5	1
C2	2	4
Gini=0.371		

Gini(Children)
= 7/12 * 0.408 +
5/12 * 0.320
= 0.371

Categorical Attributes: Computing Gini Index

- For each distinct value, gather counts for each class in the dataset
- Use the count matrix to make decisions

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

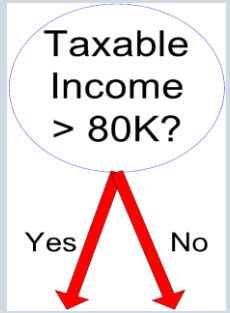
	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Continuous Attributes: Computing Gini Index

- Approaches to split:
  - Use Binary Decisions based on one value
  - Several Choices for the splitting value
    - Number of possible splitting values = Number of distinct values
- Each splitting value has a count matrix associated with it
  - Class counts in each of the partitions, A < v and A ≥ v
- Simple method to choose best v
  - For each v, scan the database to gather count matrix and compute its Gini index
  - Computationally Inefficient! Repetition of work.

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



Continuous Attributes: Computing Gini Index

- For efficient computation: for each attribute,
  - Sort the attribute on values
  - Linearly scan these values, each time updating the count matrix and computing gini index
  - Choose the split position 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	

Splitting Criteria Based On Classification Error

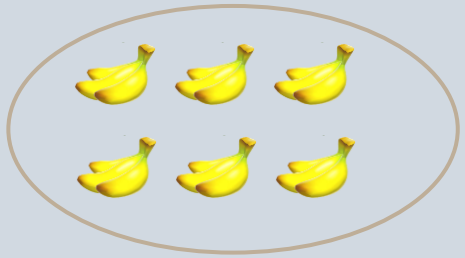
- Classification error at a node t:

$$\boldsymbol{Error(t) = 1 - \max P(i|t)}$$

- Measures misclassification error made by a node.
  - Maximum (1 - 1/n_c) when records are equally distributed among all classes, implying least interesting information
  - Minimum (0.0) when all records belong to one class, implying most interesting information

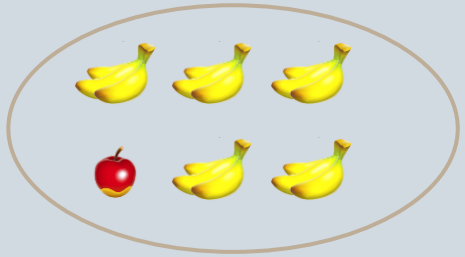
Examples for Computing Error

$$Error(t) = 1 - \max P(i|t)$$



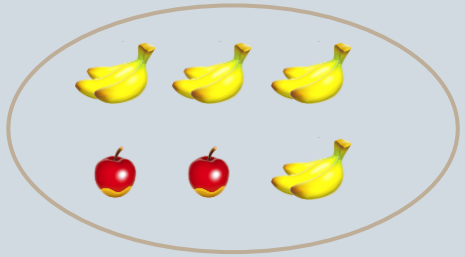
P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0



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

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

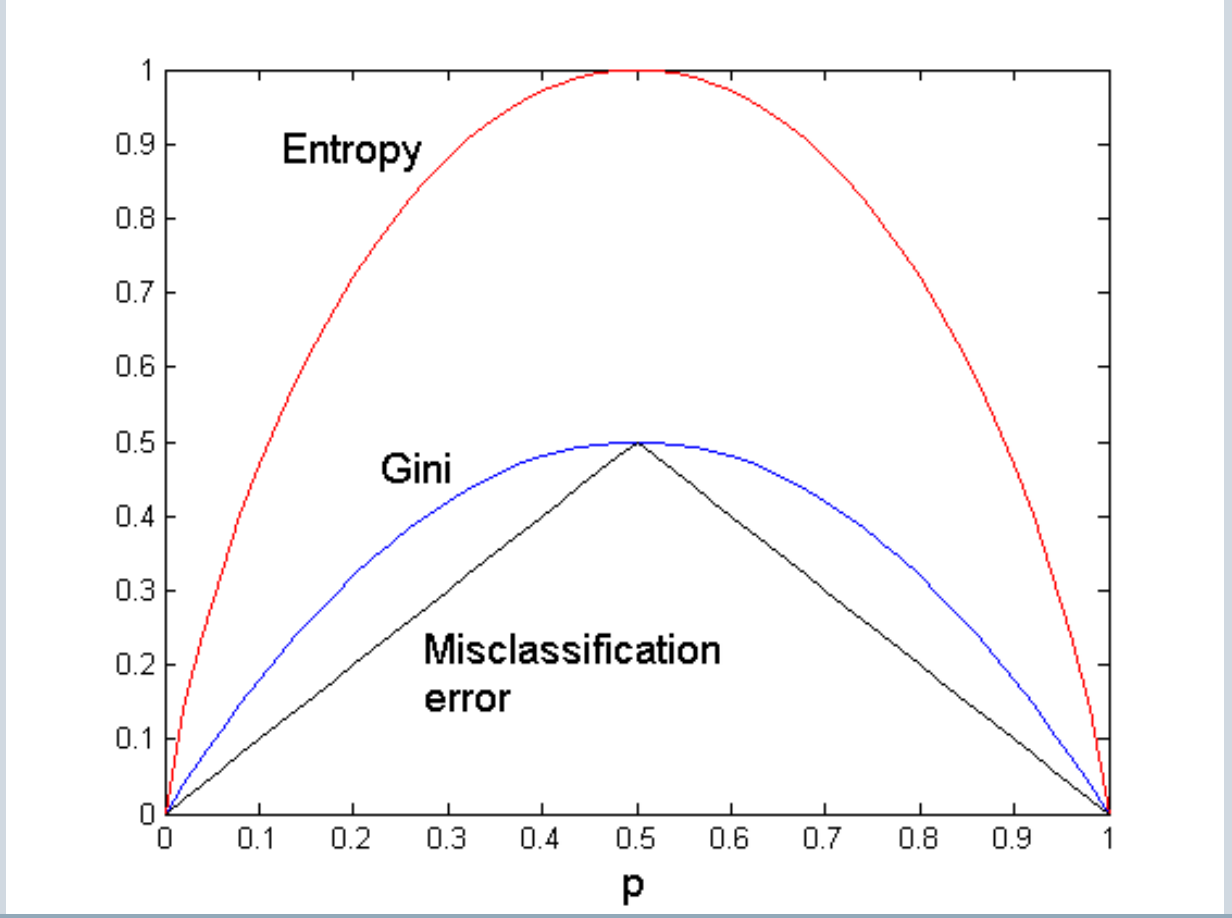


P(C1) = 2/6 P(C2) = 4/6

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

Comparison Among Splitting Criteria

For a 2-class problem:



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

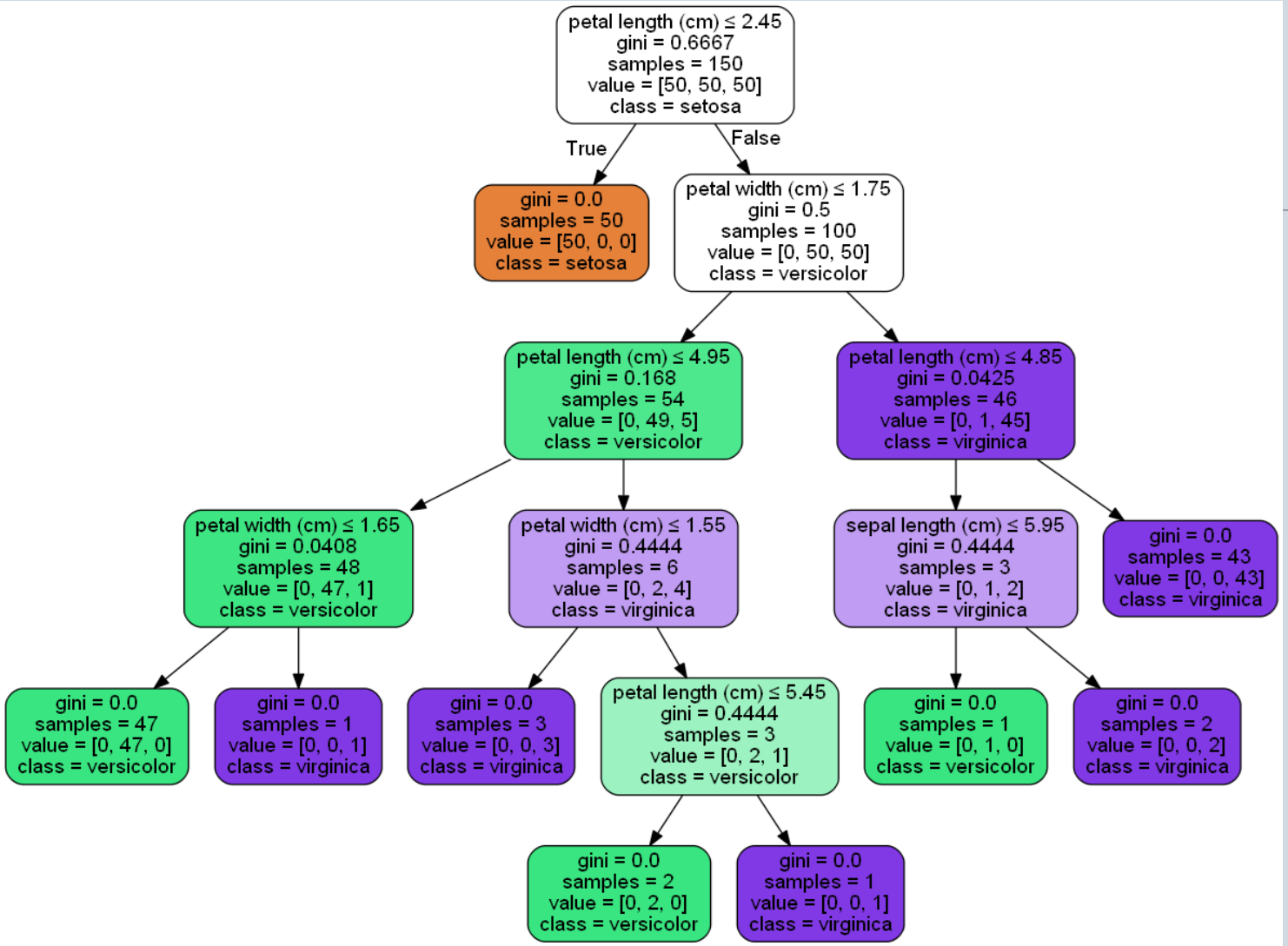
Stopping Criteria For Tree Induction

- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- Early termination:
  - Maximal depth reached **(max_depth)**
  - Fragmentation **(min_samples_split – don't split nodes with fewer samples, min_samples_leaf – leafs can't be too small)**
  - Sufficient impurity (**min_impurity_decrease – don't split unless gain more than this**)

Decision Tree Based Classification

- Advantages:
  - Inexpensive to construct
  - Extremely fast at classifying unknown records
  - Can handle a mixture of nominal(categorical), ordinal and continues data
  - Easy to interpret for small-sized trees
  - Accuracy is comparable to other classification techniques for many simple data sets
- Disadvantages:
  - Not stable – variations in data or algorithm parameters can produce entirely different tree
  - Complex trees are difficult to interpret
  - Tends to overfit (avoid by limiting depth or set minimal leaf size)
- Notes:
  - Works best if the boundaries between classes are sharp (and not smooth)

Classification_Models_Part_A.ipynb



Decision Tree vs Logit Regression

- If the relationship between dependent & independent variable is well approximated by a linear model, linear regression will outperform tree-based model.

- If there is a **high non-linearity** & complex relationship between dependent & independent variables, a tree model will outperform a classical regression method.

Regression Trees

- Classification trees are used when dependent variable is categorical.
- Regression trees are used when dependent variable is continuous.
- Leaves of regression trees contain mean value of all observations falling into the corresponding leaf.
- Recursive binary splitting of the data (contraction of the tree) continues until
  - the leaves are sufficiently homogenous or
  - their population is small enough (based on user-defined criteria).
- Splitting:
  - One common splitting criterion – reduction of variance: $\boldsymbol{Variance} = \frac{\sum(X-\bar{X})^2}{n}$
  - At each step t, the regression tree tries splitting data across all possible features f and computes

    $$\boldsymbol{V_{left}(t,f) + V_{right}(t,f)}$$
- Selects the split that minimizes the variance.
- The algorithm is greedy – maximizes immediate payoff.

THANK YOU FOR LISTENING

ZVI.BENAMI@MAIL.HUJI.AC.IL