IST407/707 Applied Machine Learning

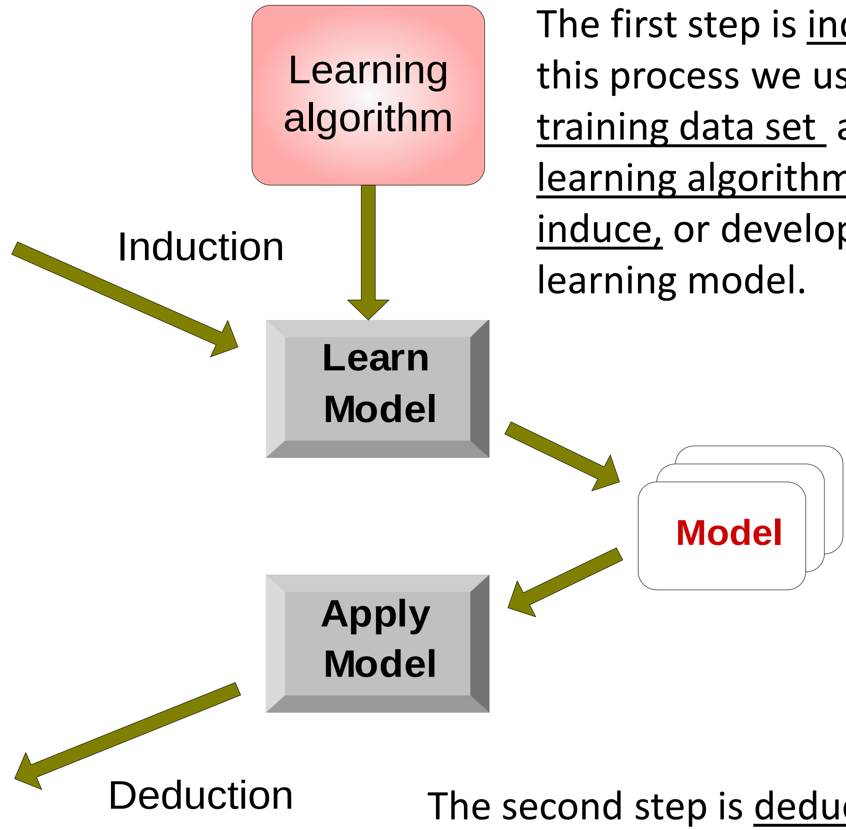
Decision Trees

Syracuse University
School of Information Studies

The Automated Classification Process

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



The first step is induction. In this process we use a training data set and a learning algorithm to induce, or develop a learning model.

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

The second step is deduction, where we apply the learned model to test data, in which the decisions are unknown

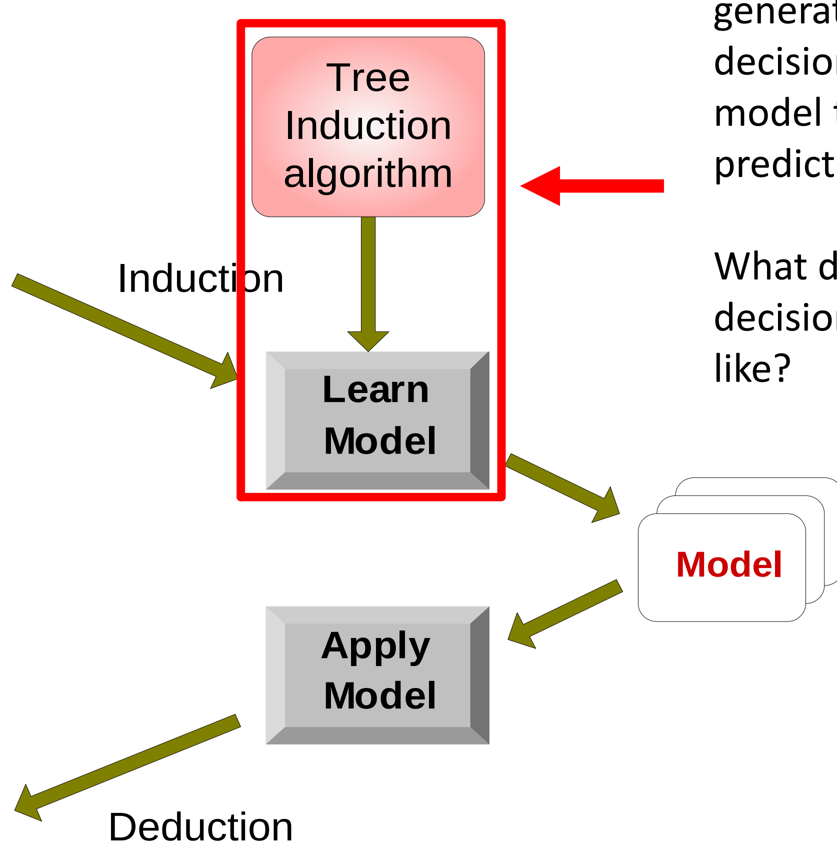
Classification Techniques

- Many classification algorithms have been developed to date.
- This class will introduce the details of several of the most popular algorithms
  - Decision Tree
  - Bayesian method (naïve Bayes)
  - Instance-based learning (k-Nearest Neighbor)
  - Support Vector Machines (SVMs)
- In this week, we illustrate classification tasks using Decision Tree methods

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

A decision tree classification method will generate a decision tree model to make a prediction.

What does a decision tree look like?

An Example of Decision Tree

Problem: to label each person as to whether they will cheat IRS

categorical categorical continuous class

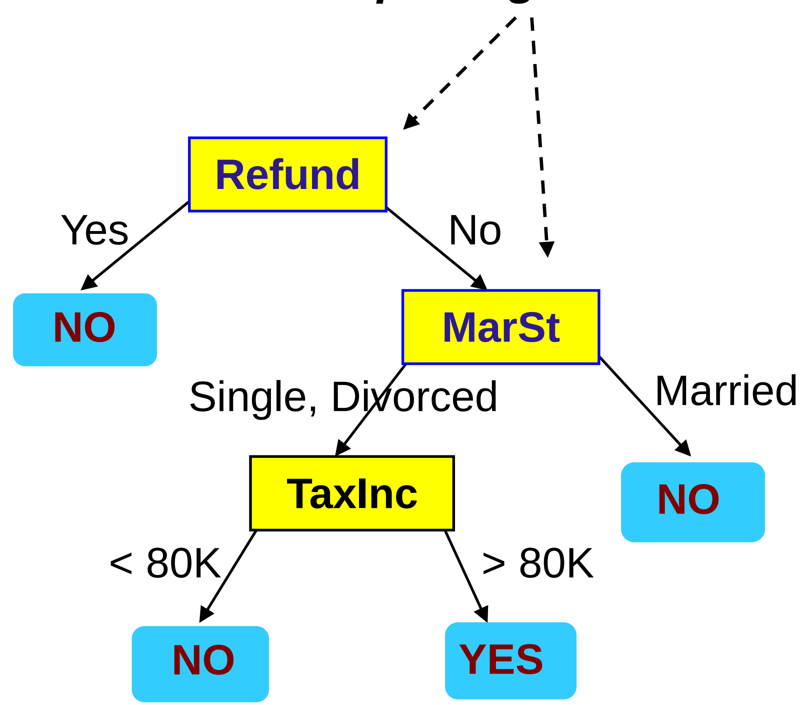
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K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

Training Data

Refund = Yes is pure. No additional split needed

Refund = No is not pure. Additional split needed

Splitting Attributes



Model: Decision Tree

An Example of Decision Tree

Problem: to label each person as to whether they will cheat IRS

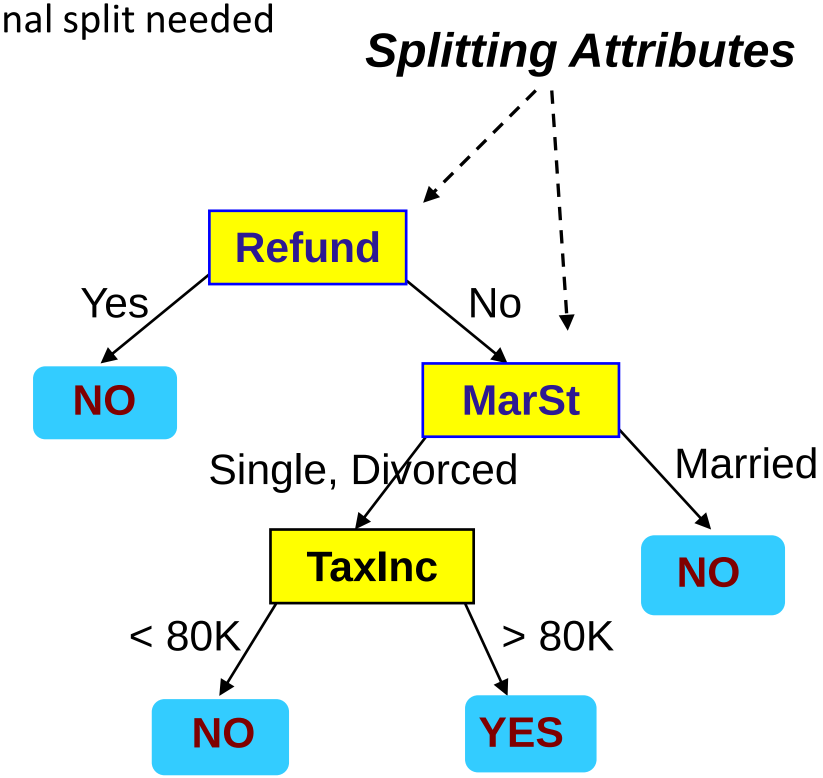
categorical categorical continuous class

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

Refund = Yes is pure. No additional split needed

Refund = No is not pure. Additional split needed



Model: Decision Tree

An Example of Decision Tree

Problem: to label each person as to whether they will cheat IRS

categorical *categorical* *continuous* *class*

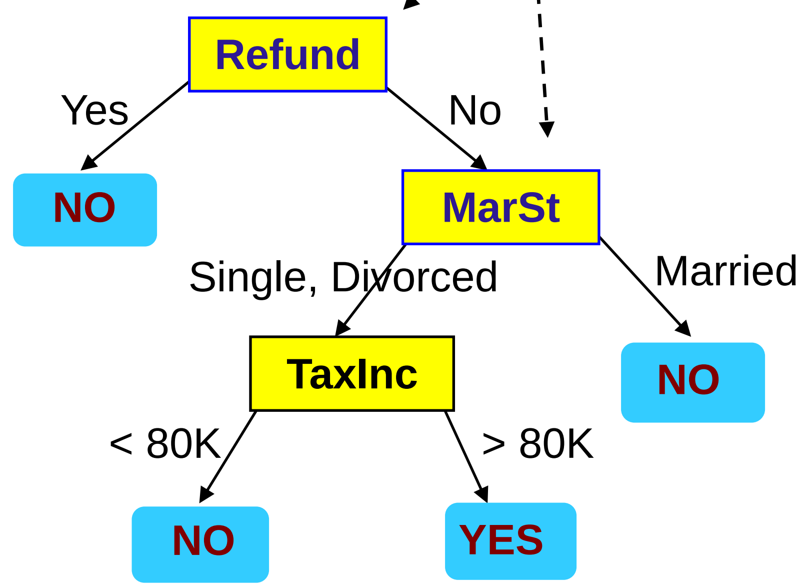
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K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

Training Data

MarSt = Married is pure. No additional split needed

Married = Single / Divorced is not pure. Additional split needed

Splitting Attributes



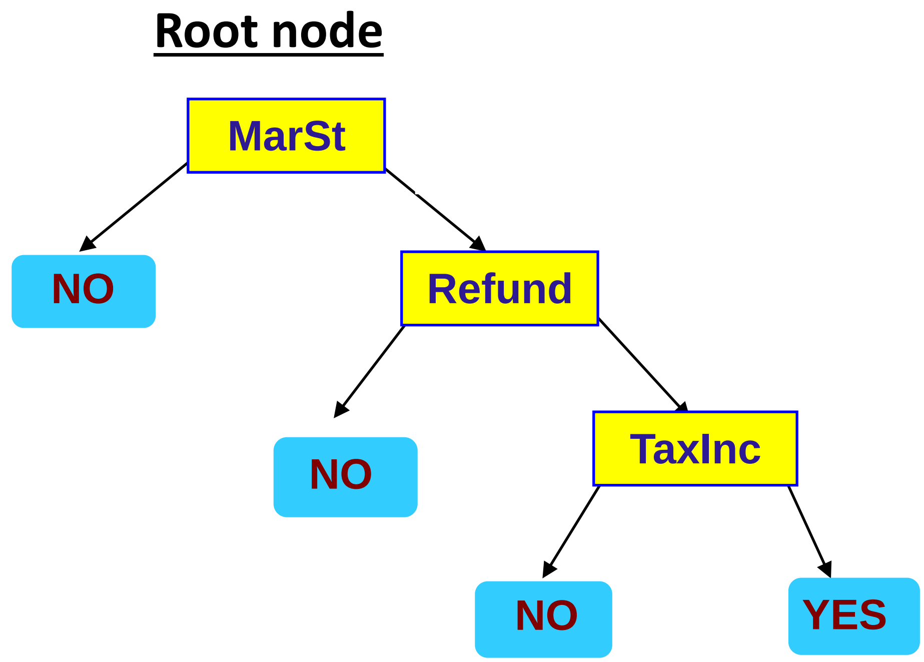
Model: Decision Tree

Syracuse University School of Information Studies

Another Example of 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**

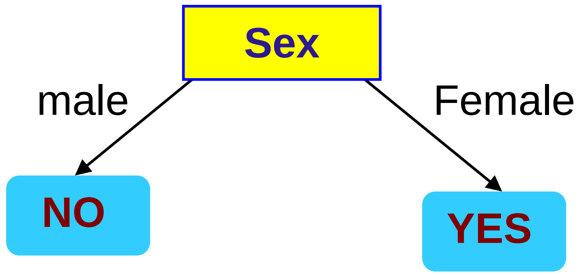


There could be more than one tree that fits the same data!

Which one is the best ?

In Class Exercise 1: Manually build a DT

Task: Open up the Titanic data, observe the patterns, and manually build a decision tree that includes at least two internal nodes. Here is an example of the simplest tree with only one internal node.



Note the goal of this exercise is to check if you understand the concept of a decision tree model. No need to build super-sophisticated trees. Also, don't worry about its actual performance either at this time.

C4.5 ALGORITHM 1: HOW TO SPLIT DATA AT NODE

How to find the best decision tree?

Challenges

- Too many candidate trees
- Manual construction takes too long
- Need some machine intelligence to help

Decision Tree Induction

Many Algorithms:

- Hunt’s Algorithm (one of the earliest)
- CART
- ID3, C4.5
- SLIQ,SPRINT

C4.5 is introduced in this class

Tree Induction

Key questions to build a decision tree model

- Which attribute to pick as internal node?
- How to split the data set at a node?

How to split data at a node?

How many branches?

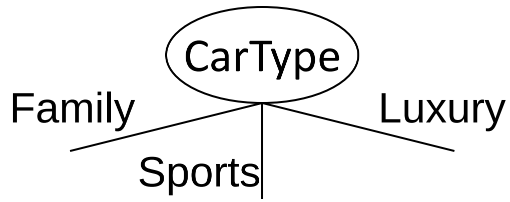
- Splitting can be
  - 2-way split
  - multi-way split

What are the splitting values?

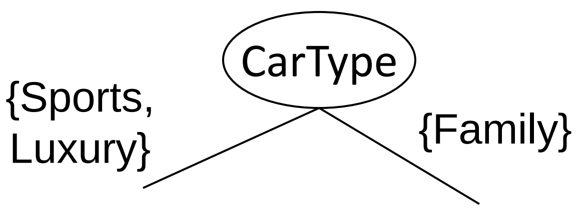
- Splitting conditions depend on attribute type
  - Nominal/categorical
  - Ordinal
  - Continuous

Splitting Based on Categorical Attributes

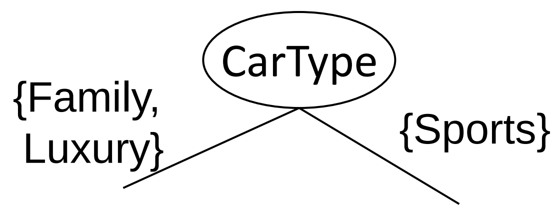
Multi-way split: Use as many partitions as distinct values.



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



OR



Splitting Based on Continuous Attributes

Different ways of handling split

- Discretization is used to form an <u>ordinal categorical attribute</u>
    - E.g. age: 1 1 6 7 8 9 9 9 10 10 11 11 12 13 14 15 17 18

- <u>Equal interval</u>: one bin for every six year [0-6][7-12][13-18]
    - 1 1 6 • 7 8 9 9 9 10 10 11 11 12 • 13 14 15 17 18
        - *Problems when data not equally distributed*

- <u>Equal frequency</u>: one bin for every six numbers (could have ties)
    - 1 1 6 7 8 • 9 9 9 10 10 11 11 • 12 13 14 15 17 18

Other custom discretizations are possible, depending on domain knowledge or data distribution

Syracuse University
School of Information Studies

Splitting Based on Continuous Attributes

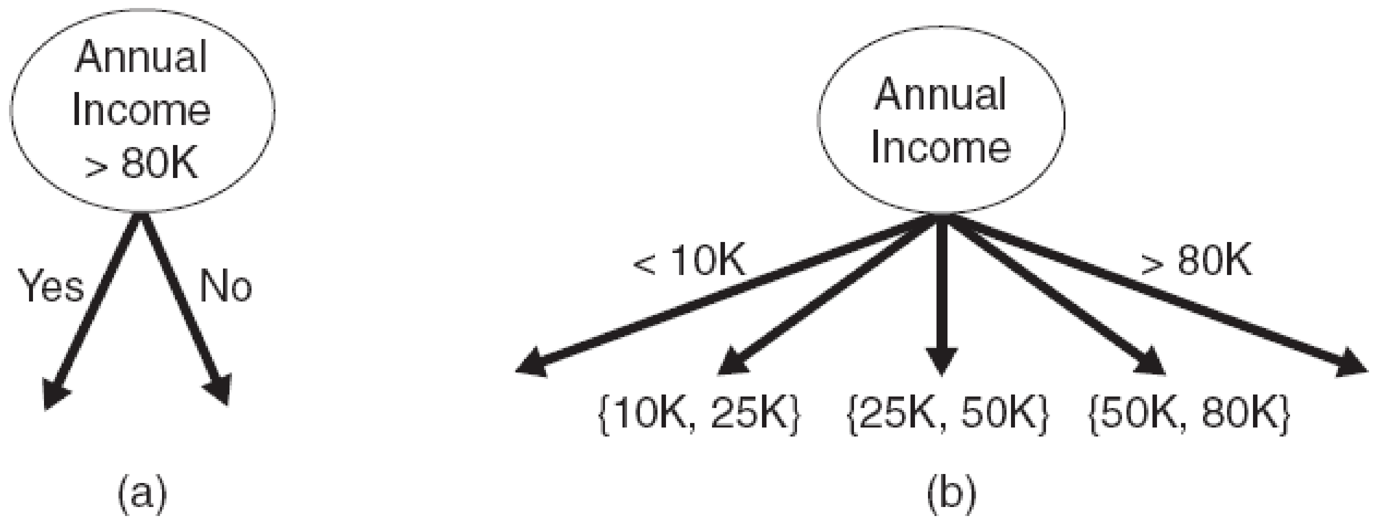


Figure 4.11. Test condition for continuous attributes.

Determine the Best Attribute for Splitting

Information Gain (IG)

- A statistical measure that measures how well a given attribute separates the training examples according to their target classification. (Mitchell, 1990)

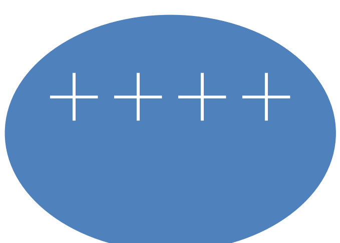
Determine the Best Attribute for Splitting

Entropy

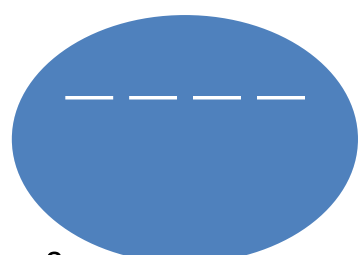
- To measure the impurity of a data set (Noise Level)
- Given a collection S which contains positive (+) and negative (-) examples, p_i is the probability that an example belongs to Class i
- Entropy(S) = $- p_+ \log_2 p_+ - p_- \log_2 p_-$

P+ - Prob pos examples occur
P- - Prob of pos examples occur

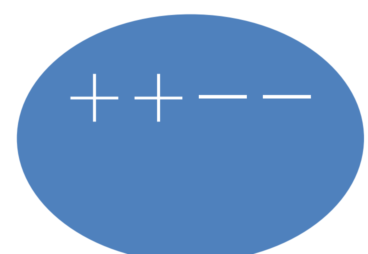
What is the entropy for each of the following collections?



P+ = 1
P- = 0
Ent = 0



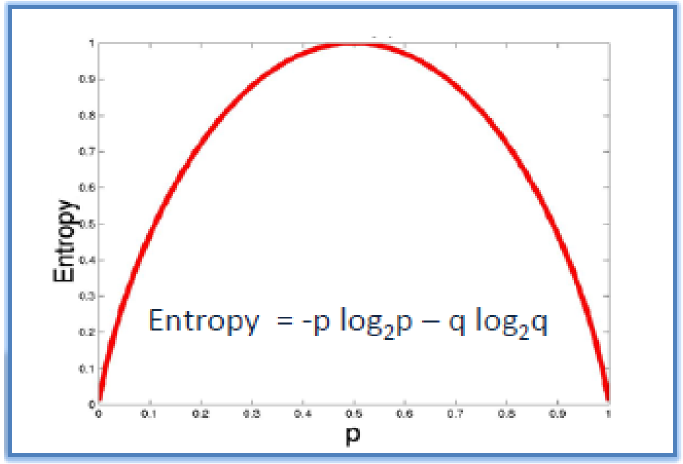
P+ = 0
P- = 1
Ent = 0



P+ = .5
P- = .5
Ent = 1

Determine the Best Attribute for Splitting

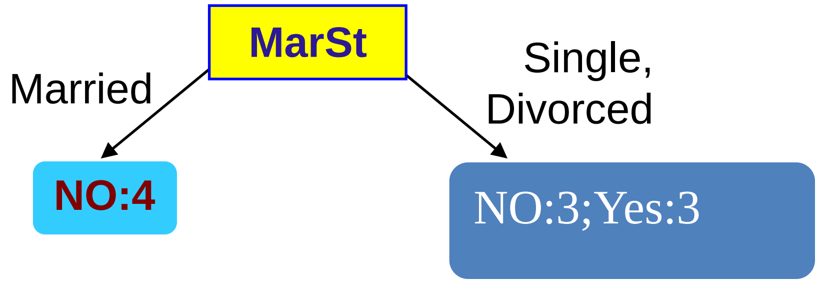
- Entropy
  - A measure that characterizes the impurity of a collection of examples
  - Given a collection S which contains positive (+) and negative (-) examples, pi is the probability that an example belongs to Class i
  - Entropy(S) = - p+log2p+ - p-log2p-
  - A collection of half positive examples and half negative examples
  - Entropy(S) = 1
  - A collection of all positive examples or all negative examples
  - Entropy(S) = 0



Note: entropy is not restricted to [0,1] in cases where |S| > 2

Information Gain: how much improvement toward purity?

	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**



If we choose an attribute to spilt on, how much will it reduce the entropy, or bring the data to more purity level

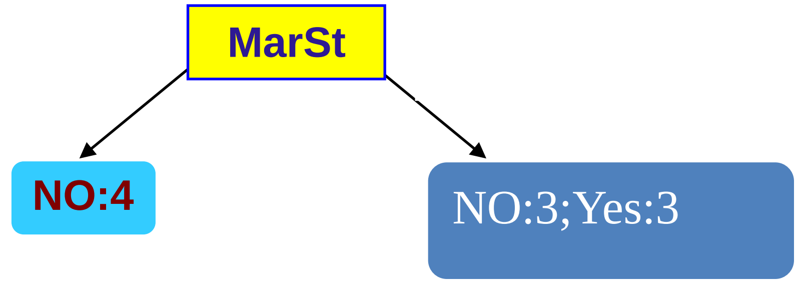
Original data entropy

Weighted sum of the entropy of the subsets that is generated using the split attribute

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \times Entropy(S_v)$$

Information Gain: how much improvement toward purity?

	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**



Entropy(S)=-0.7*log2(0.7)-0.3*log2(0.3)=0.88

Entropy(MarSt = No)=0
Entropy(MarSt = Yes)=1

IG=0.88-(0.4*0 + 0.6*1) = 0.28

Repeat this calculation to find the attribute that provides the highest IG

Exercise: calculate Info Gain

Let's start with "age", , see if the entropy gets smaller after using age to split the data.

Step 1: calculate the entropy of the entire training data set S, which contains <u>9 positive examples and 5 negative examples.</u>

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

Exercise: calculate Info Gain

Let's start with "age", , see if the entropy gets smaller after using age to split the data.

Step 1: calculate the entropy of the entire training data set S, which contains <u>9 positive examples and 5 negative examples.</u>

Entropy(S) = I(9,5)

$$=-\frac{9}{14}\log_2\frac{9}{14}-\frac{5}{14}\log_2\frac{5}{14}$$

=.940

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

Exercise: calculate Info Gain

Step 2: count the numbers of positive examples (column p_i) and negative examples (column n_i) in each subset, and then calculate the entropy for each subset, $I(p_i, n_i)$.

For example, for the "<=30" subset S_1,

$$Entropy(S_1) = I(2,3)$$
$$= -\frac{2}{5}\log_2(\frac{2}{5}) - \frac{3}{5}\log_2(\frac{3}{5})$$
$$=0.971$$

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

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

Similarly,
Entropy(S_2) =0;
Entropy(S_3) = Entropy(S_1)=0.971

Exercise: calculate Info Gain

Step 3: calculate the weighted average entropy after using age to split the data into three subsets "<=30", "31..40", and ">40".

$$Entropy(age, S) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2)$$

$$= \frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971$$

$$= 0.694$$

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

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

Exercise: calculate Info Gain

Step 4: calculate the information gain of using age to split the data into three subsets "<=30", "31..40", and ">40".

$$Gain(age) = Entropy(S) - Entropy\ (age, S)$$

$$= 0.940 - 0.694 = 0.246$$

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

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

Which attribute should be the first node?

Step 5: repeat the process for each attribute, and then pick the attribute with highest IG as the first node.

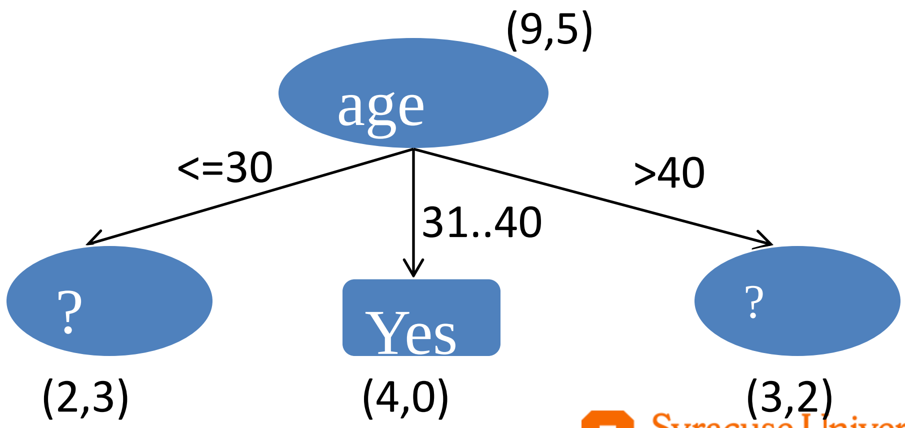
$$Gain(age) = 0.246$$

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

The DT now has one leaf node
And two subsets that need
to be further split.

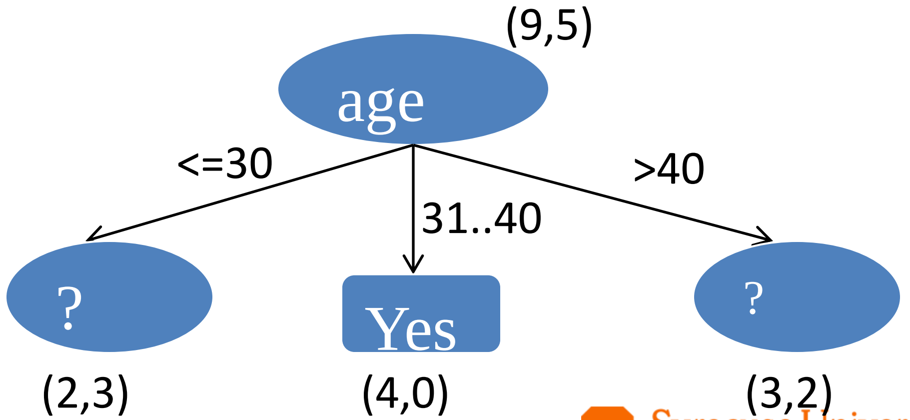


What's the next step?

Repeat the prior steps for the subsets (2,3) and (3,2).

- For subset (2,3), calculate IG for each attribute, pick the attribute with highest IG to replace the question mark.
- Do the same thing to the subset (3,2)

Until all nodes are "pure" with all positive examples, or all negative examples.

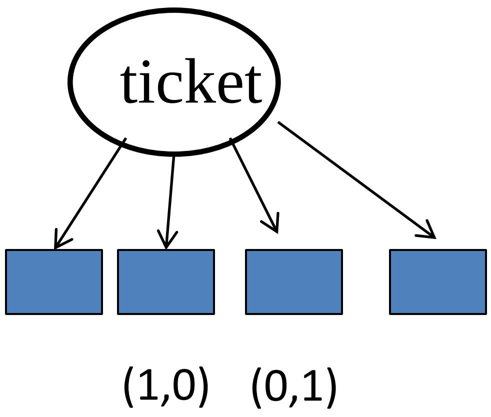


Syracuse University
School of Information Studies

Gain Ratio

Impurity measures tend to favor attributes that have a large number of distinct values

- E.g. the “ticket” attribute in the Titanic data set means the ticket number. Assuming every passenger has unique ticket number, the ticket attribute has many distinct values, and impurity measures like IG favor such attributes.



Gain Ratio

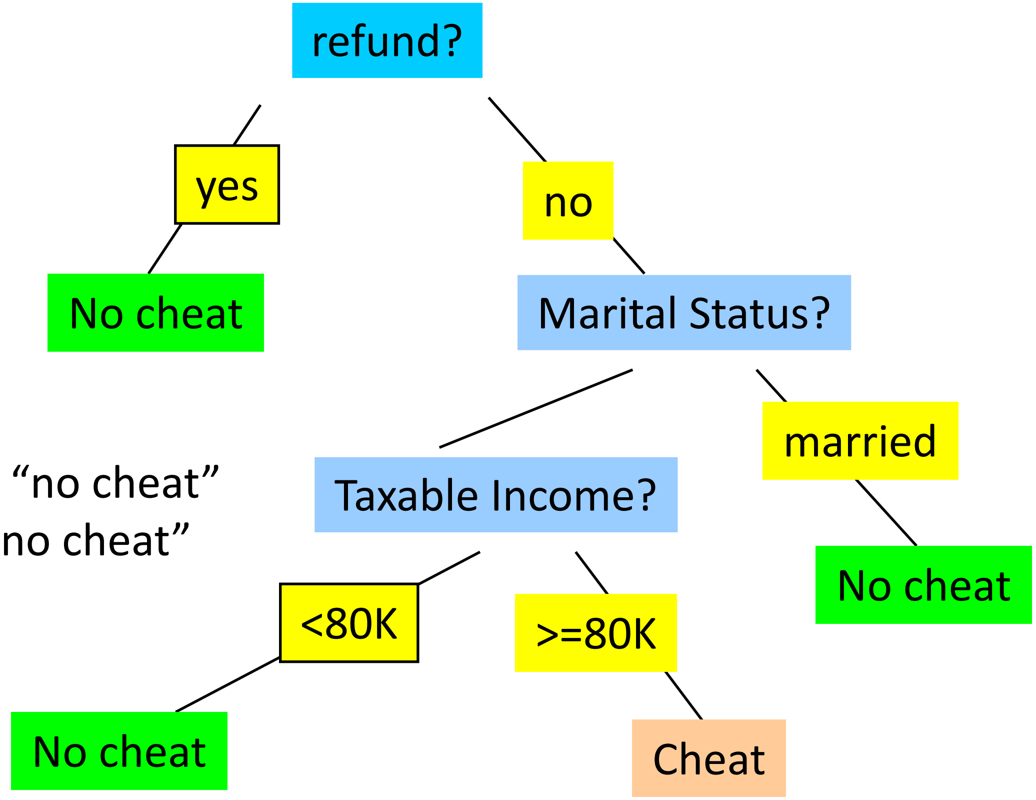
What to do?

- Use domain knowledge: ticket number has nothing to do with survival chance?
- Use Gain Ratio, which is IG divided by "split info"
  - "Split info" is a penalty to a large number of splits
- Some algorithms use gain ratio or other means to avoid this problem (e.g., allows one to specify min number of leaves)

CONVERTING DECISION TREES TO DECISION RULES

Converting Decision Tree to Decision Rules

Tree can be displayed as a set of rules:



If refund = yes then then “no cheat”
else if martital_status = “married” then “no cheat”
    else if taxable_income < 80K then “no cheat”
    else “cheat”

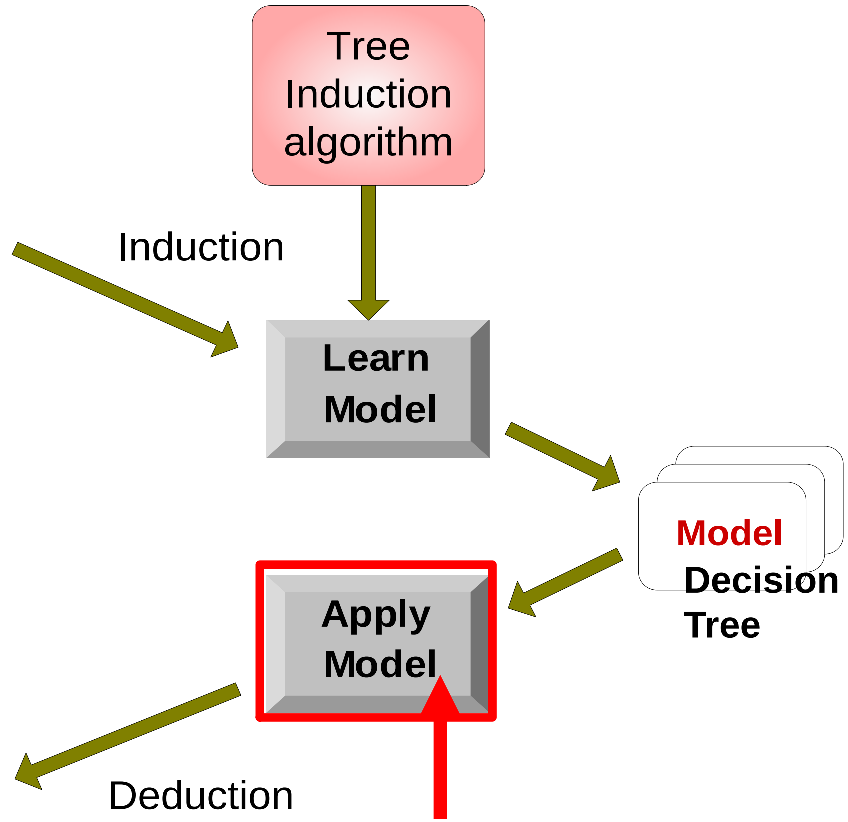
Decision Tree Classification (Prediction)Task

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

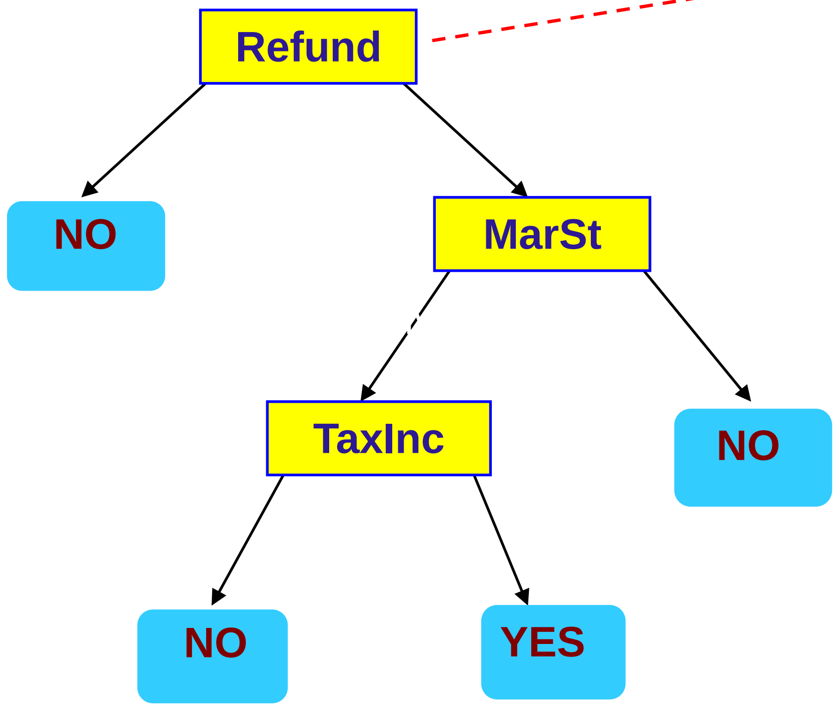
Training Set

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



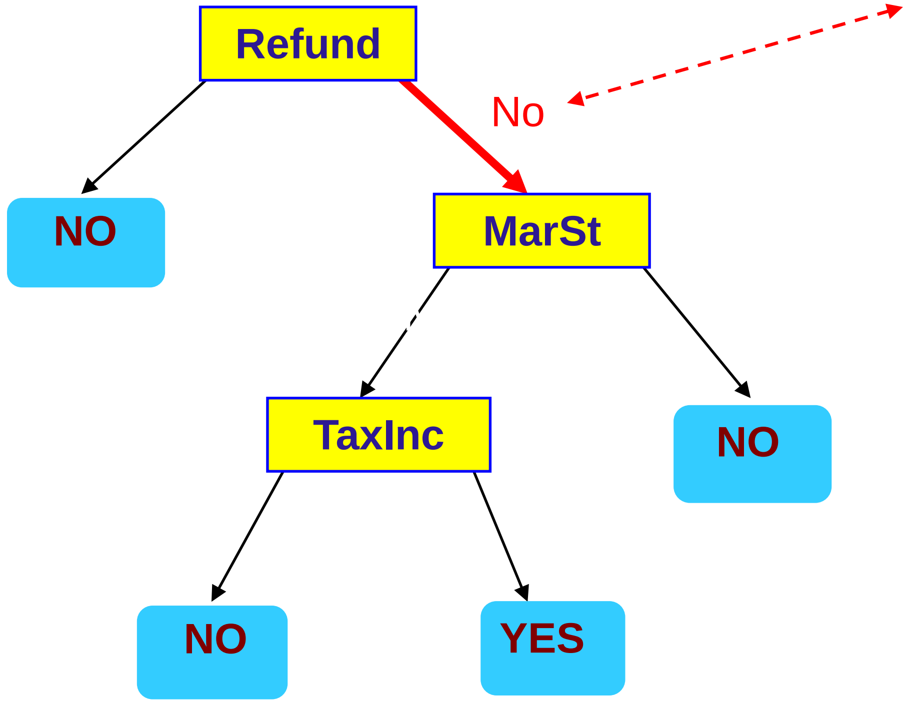
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

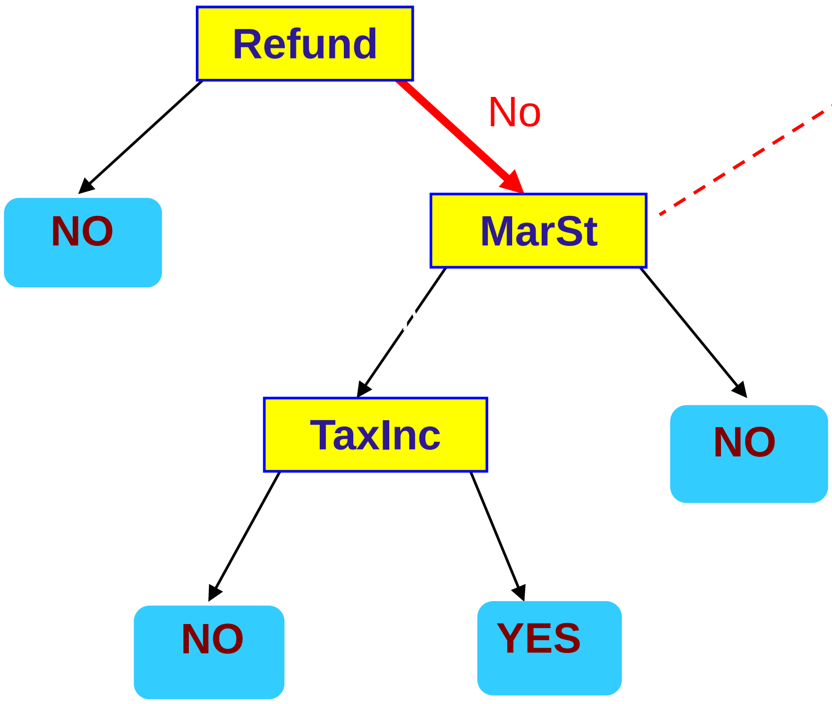
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

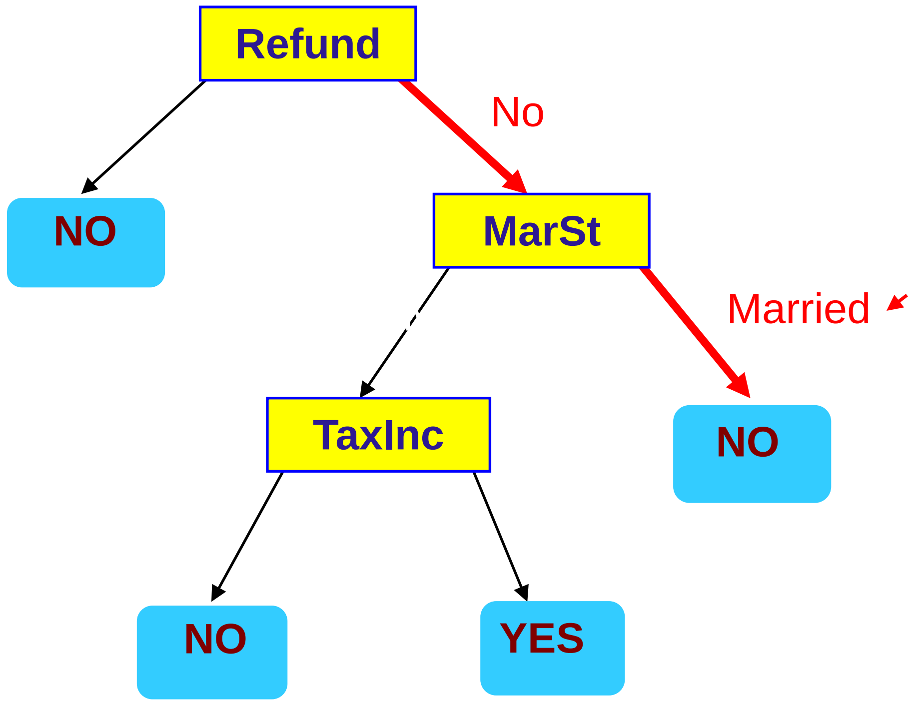
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

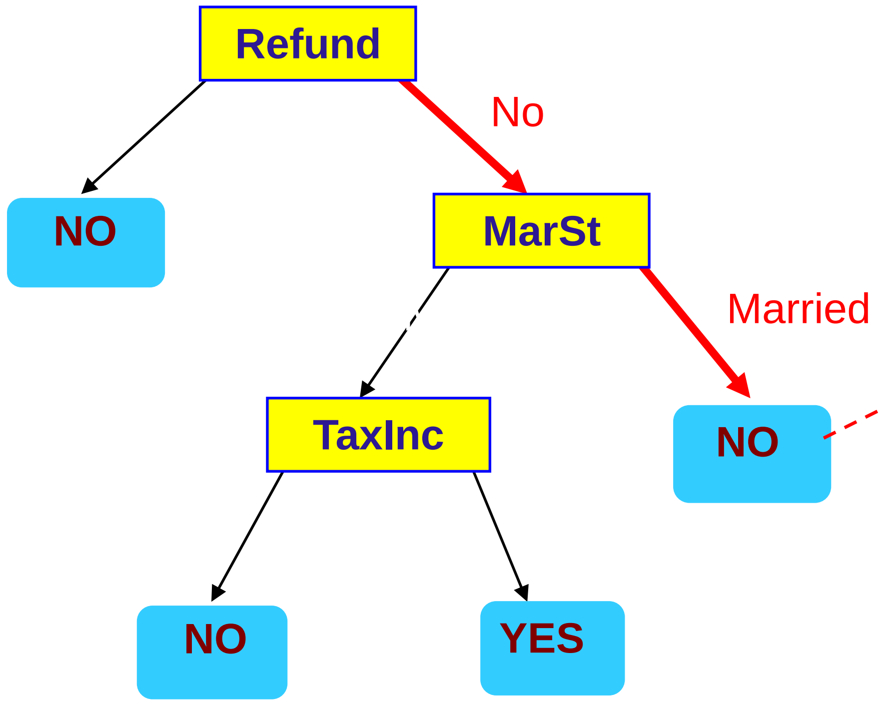
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



Test Data

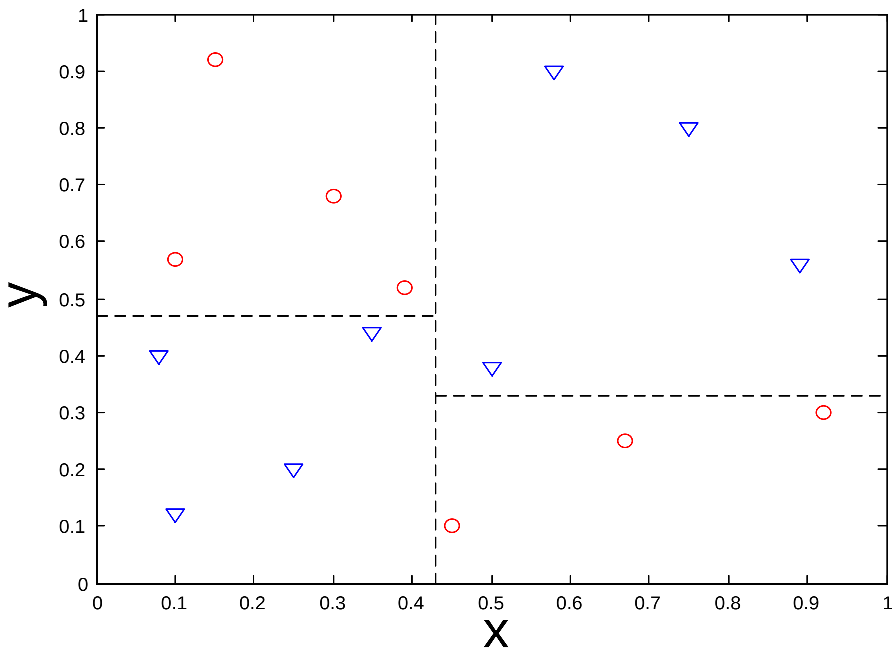
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

OVERFITTING AND PRUNING

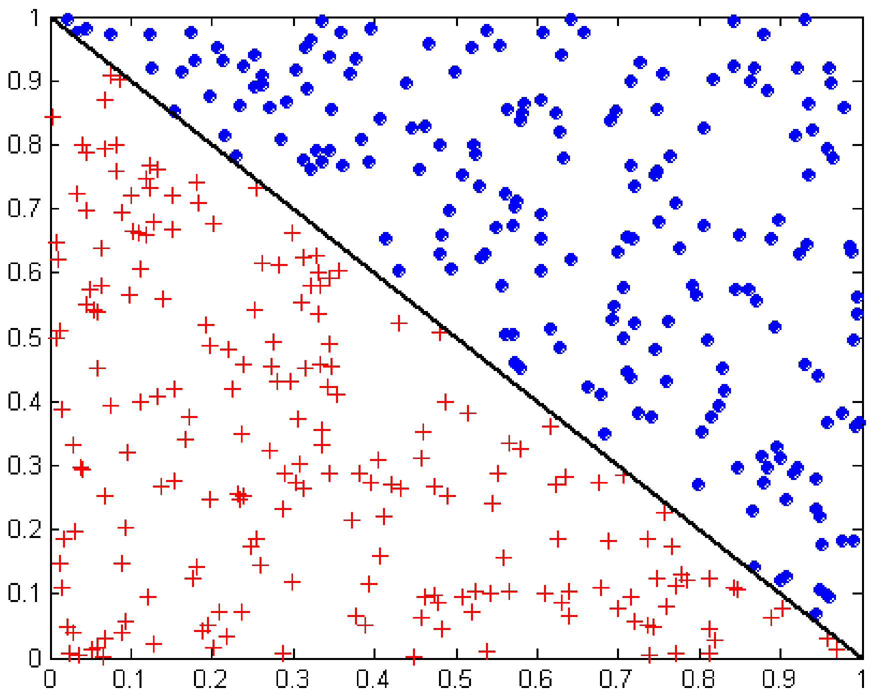
Characteristics of decision tree induction

- DT is a **nonparametric algorithm**, meaning it does not require any prior assumptions regarding the type of probability distributions satisfied by the class and other attributes
- Linear classification algorithms are parametric algorithms because they assume the decision boundary is linear, such as a line in 2-dimensional space
- "decision boundary" means the border between two neighboring regions of different classes.

There is no silver bullet...



nonlinear



linear

Model Overfitting

Decision trees have the particular problem of overfitting

- There may not be enough examples to fully represent all possible cases that may arise in the future
- If decision tree is fully developed, it may be too detailed a fit to the training data and lead to more errors on the test data
- E.g. assume we are looking for patterns of buyers for a certain product. In the training data set, no women purchased a product, the DT algorithm may learn a pattern that "if women, no purchase".
- However, in real life, there were women who bought this product. In such case, the DT model overfits the training data and lost precision in future prediction.
- **Occam's razor (preference of small trees)**

Model Overfitting

Generally speaking, complex models are more likely to overfit than simple models

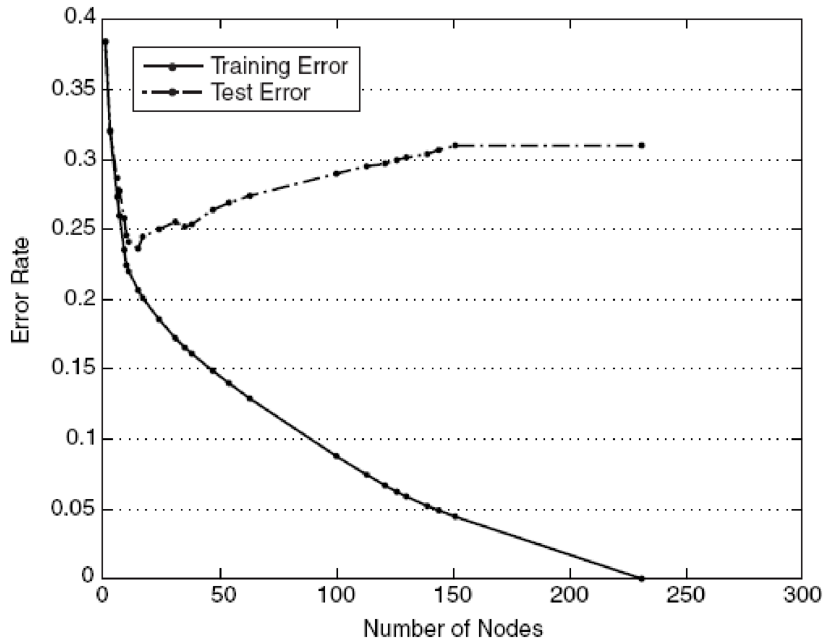


Figure 4.23. Training and test error rates.

For decision tree, #nodes indicates model complexity.

In this figure, the higher the #nodes, the lower the training error, and the higher the test error, meaning the increasingly complex models are increasingly overfitting.

Overfitting and Tree Pruning

Two approaches to avoid overfitting

- Prepruning: Halt tree construction early—do not split a node if information gain falls below a threshold
  - Difficult to choose an appropriate threshold
- Postpruning: Remove branches from a "fully grown" tree—get a sequence of progressively pruned trees
  - Use a set of data different from the training data to decide which is the "best pruned tree"

Summary of Decision Trees

Strengths of decision trees are that they:

- Fast in prediction
- Interpretable patterns
- Robust to noise

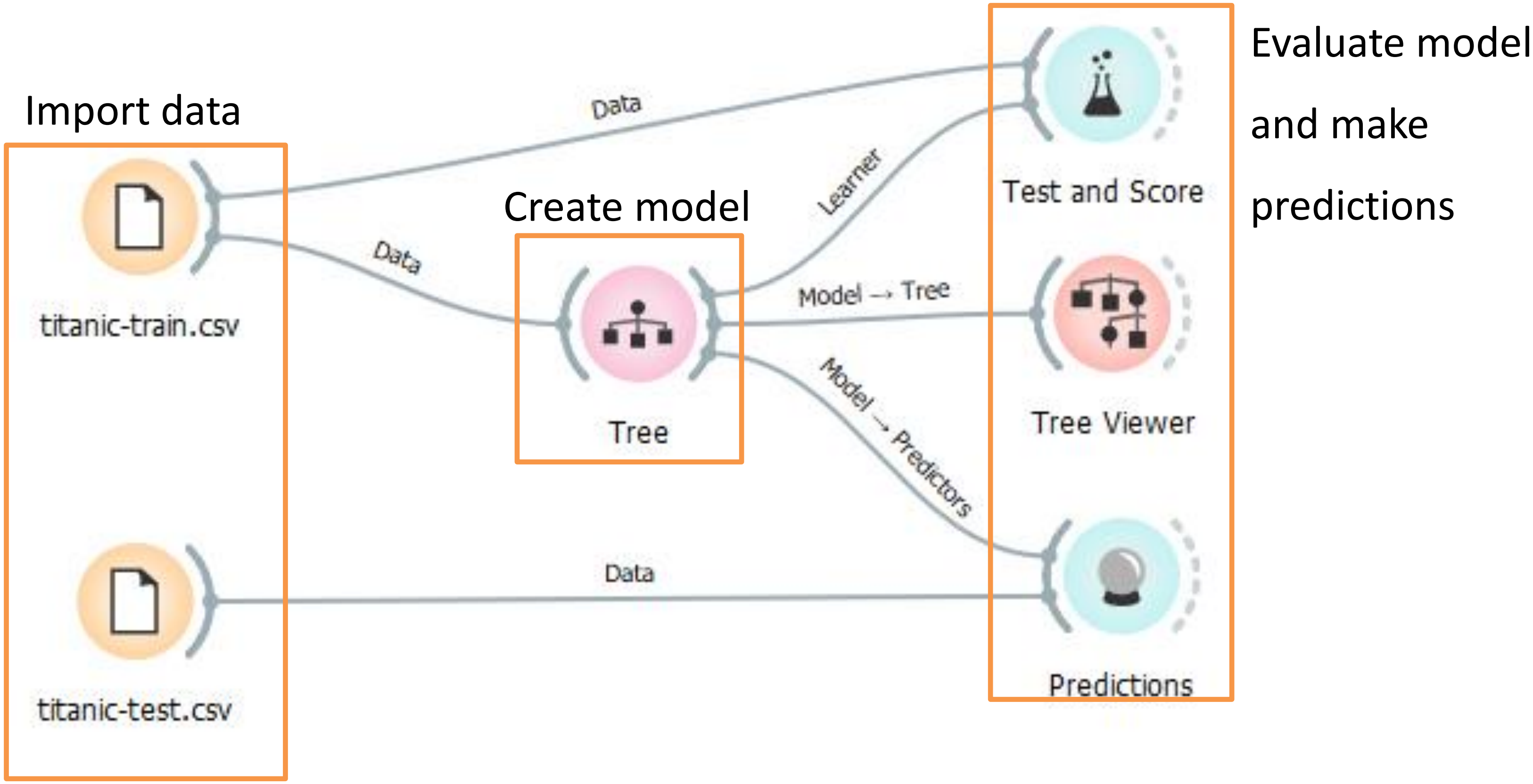
Weaknesses of decision trees are that they:

- Tend to overfit (pruning helps)
- Are error prone with too many classes
- Are computationally expensive in training (compared to the low cost in prediction)

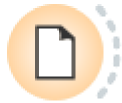
USING ORANGE FOR DECISION TREES

Syracuse University
School of Information Studies

Workflow — Decision Trees



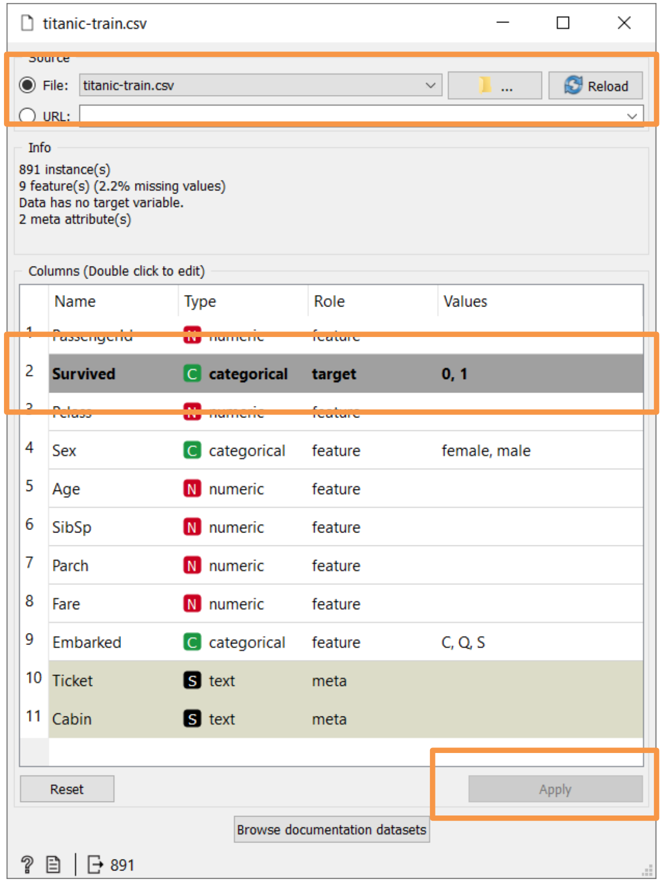
Importing

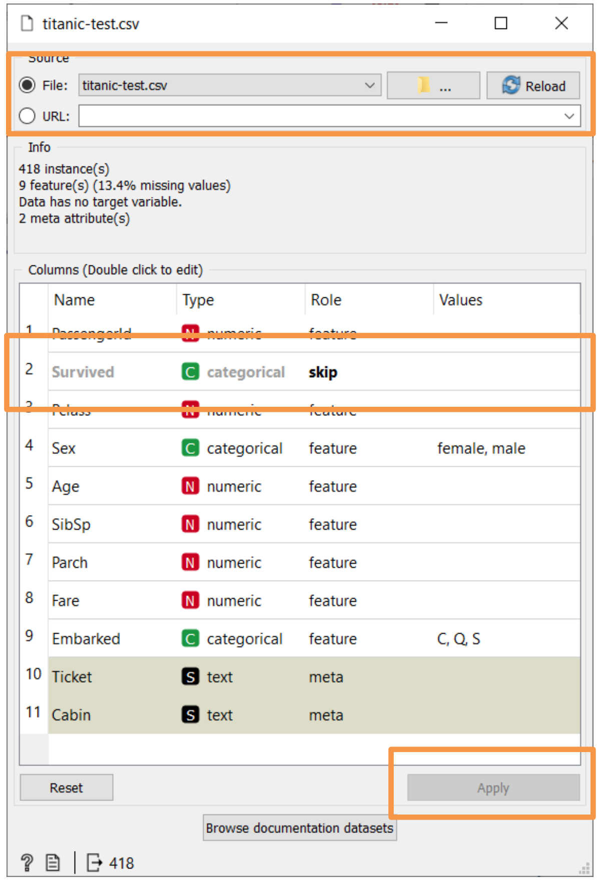


Select file

Set target variable

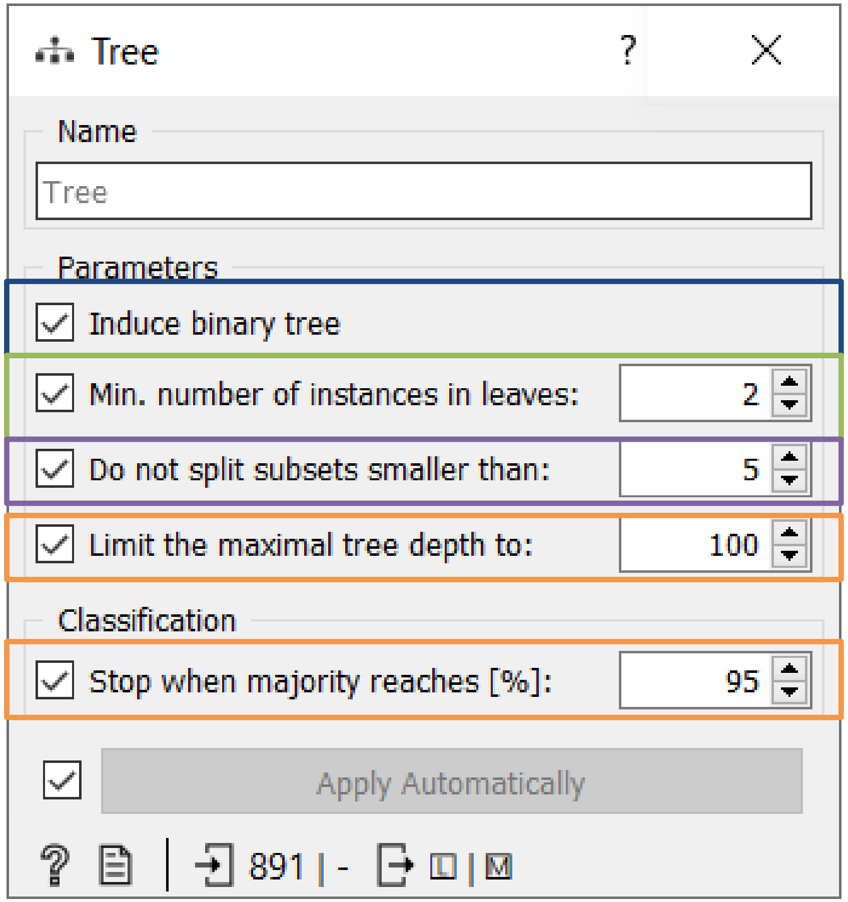
Apply





For our purposes, Orange does not expect the test data to have a target variable, so be sure to set that variable to "skip" instead!

Modeling



Limits each node to *at most* two children

Minimum number of data points in each leaf

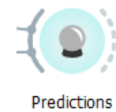
Each node split will have at least *x* data points

The tree will have no more than *x* levels

Stop splitting the nodes when *x*%

majority threshold is reached

Evaluating and making predictions

- Connect test data and model to 

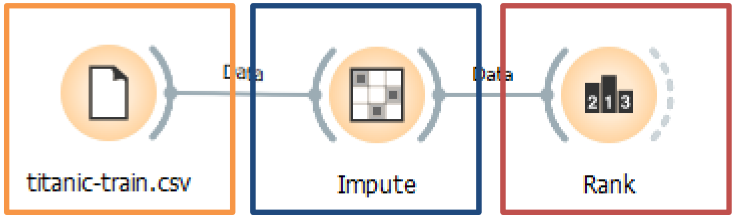
 click to see the predictions
- Connect the model to 

 view the tree
    - You can even use it to adjust your model and see those effects in real time!
- Connect training data and the model to to eva

e the model using cross-validation, random sampling, etc.

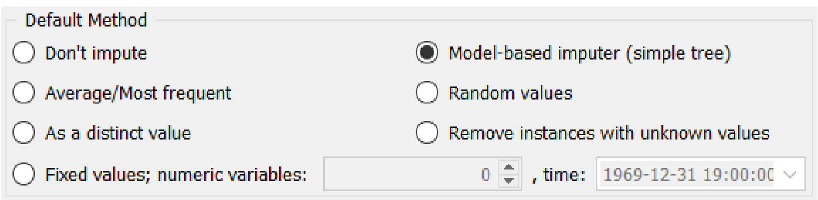
Workflow — Feature Selection

Import data



Rank calculation

Calculate



Available impute methods

Rank scoring