# Classical Data Analysis


**Master in Big Data Solutions 2017-2018**

Francisco Gutierres
francisco.gutierres@bts.tech

Sara Hajian
sara.hajian@bts.tech

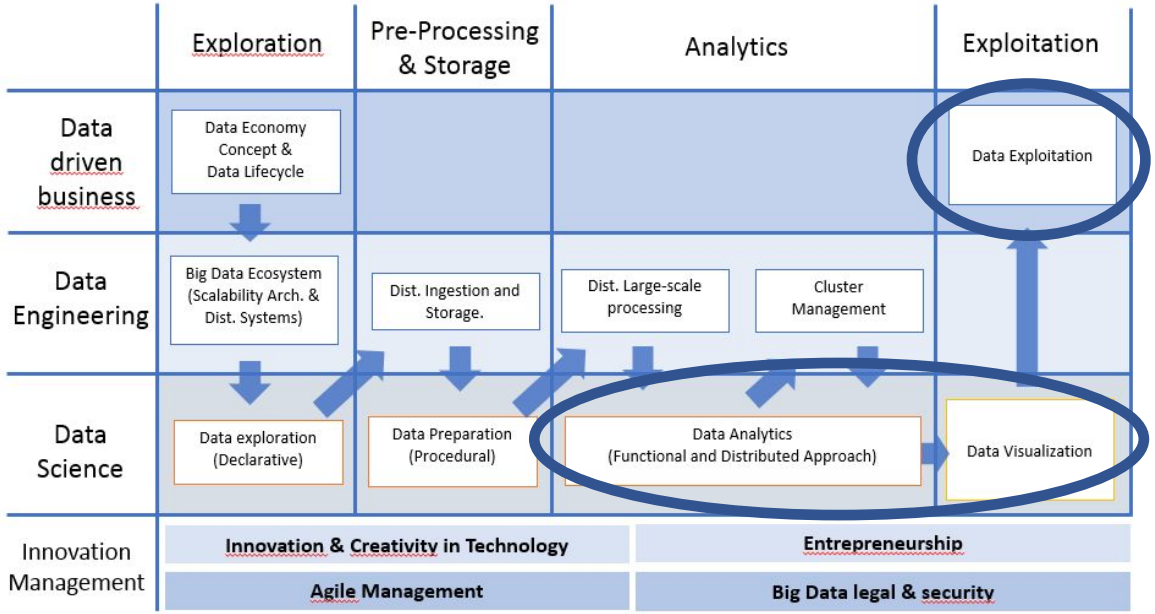
# Session 2.2.2 - Classification Decision Trees

Sara Hajian

# What we will learn

**Session2: Classification**

Decision Trees

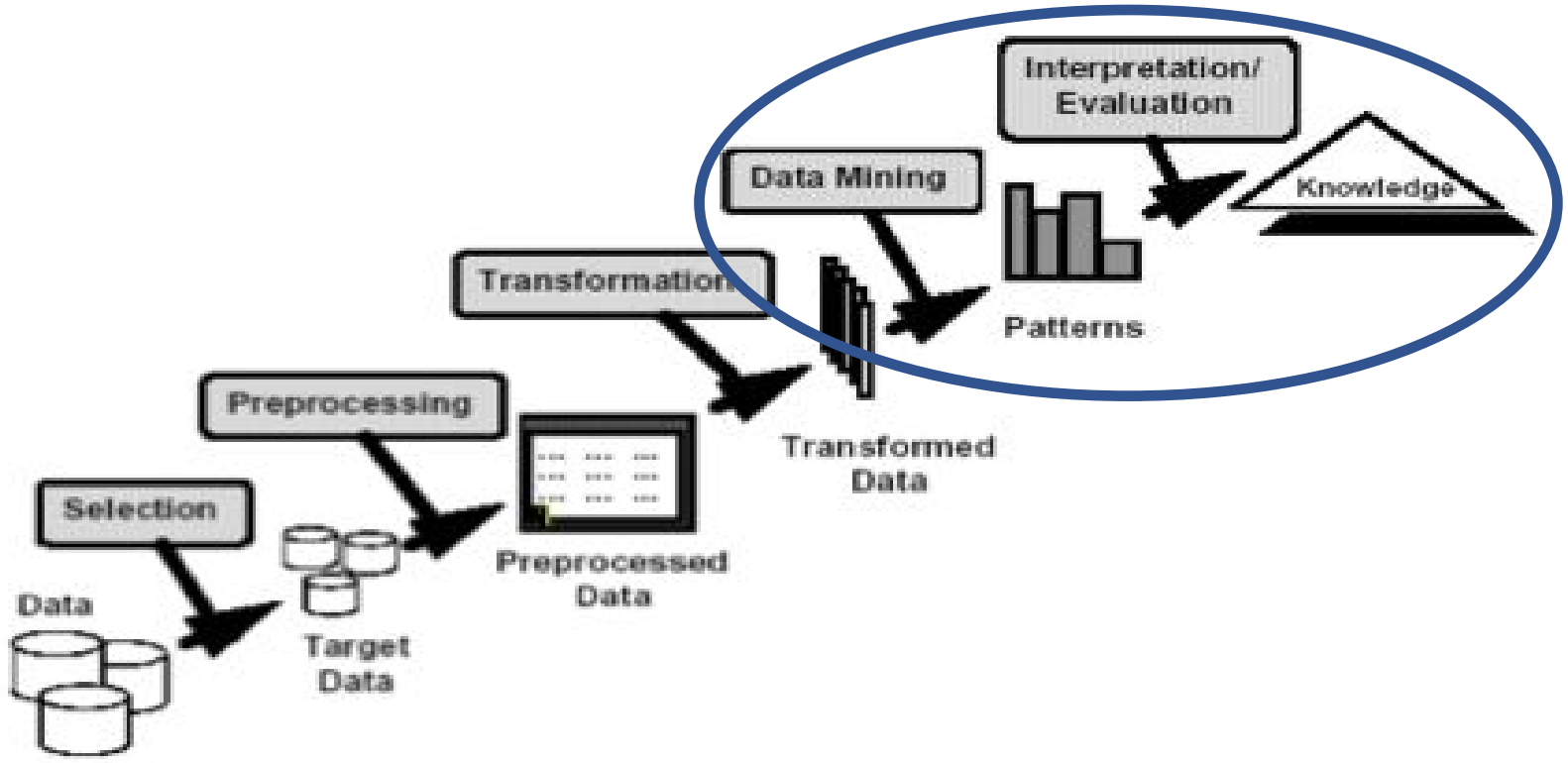



We will learn how to:

- Decision tree induction
- Methods for attribute test conditions
- Measures for selecting the best split
- Algorithm for decision tree induction

# What is data mining

Non-trivial extraction of implicit, previously unknown and potentially useful information from data (i.e., **discovery of meaningful patterns**)




Tan, Pang-Ning. *Introduction to data mining*. Pearson Education India, 2006.

# Data mining algorithms

**Supervised**: Classification and regression

**Unsupervised**: Clustering

# Classical Data analysis: Curriculum

**Part I: Regression**

Linear regression

Logistic regression

**Part II: Classification**

Supervised learning algorithms

Support vector machines (SVM)

Decision trees

Ensemble methods

**Clustering**

# Classification: definition

Given a collection of records (*training set* )

- Each record contains a set of *attributes*, one of the attributes is the *class*.

Find a *model* for class attribute as a function of the values of other attributes.

Goal: previously unseen records should be assigned a class as accurately as possible.

- A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

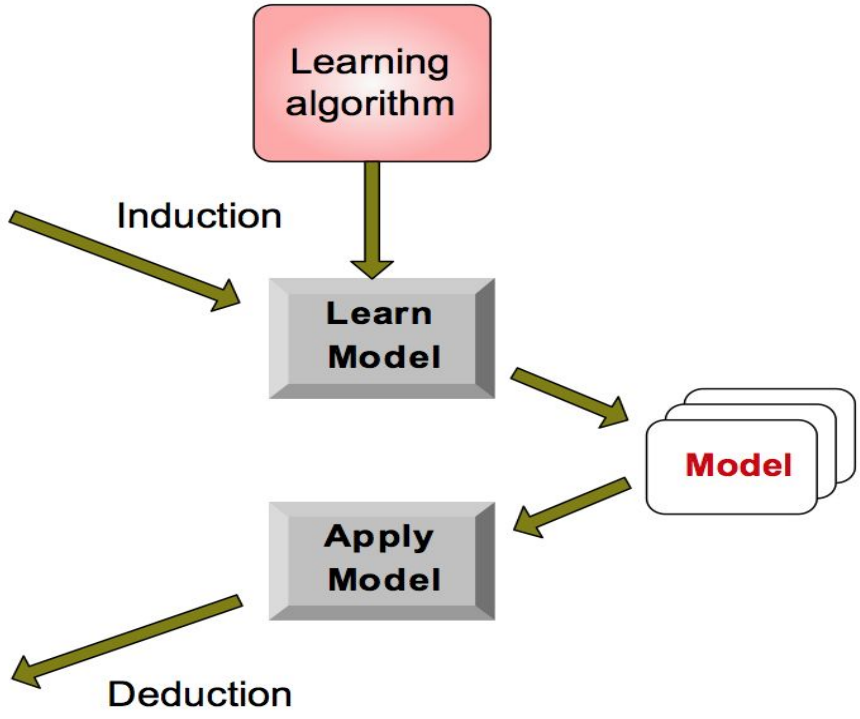
# Classification task

| 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




Tan, Pang-Ning. *Introduction to data mining*. Pearson Education India, 2006.

# Classification: applications

**Health**

- Goal: Predicting tumor cells as benign or malignant

**Fraud Detection**

- Goal: Predict fraudulent cases in credit card transactions.

**News:**

- Goal: Categorizing news stories as finance, weather, entertainment, sports, etc

# Classification: algorithms

**Support vector machines (SVM)**

**Decision trees**

**Ensemble methods**

Rule-based Methods

Neural Networks

Bayesian algorithms such as Naïve Bayes

Instance-based algorithms such kNN

Deep learning

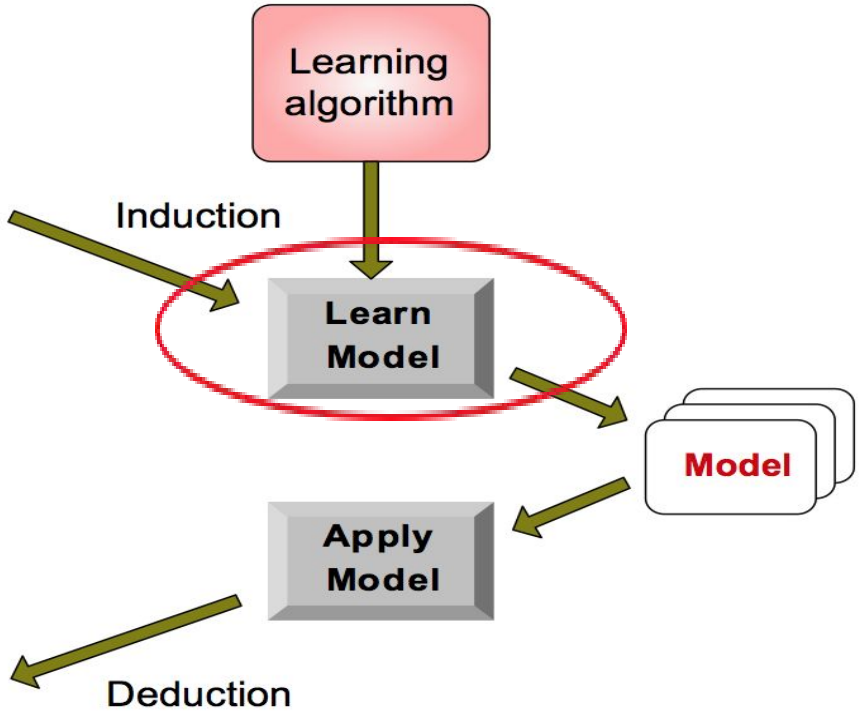
# Decision tree classification task

| 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

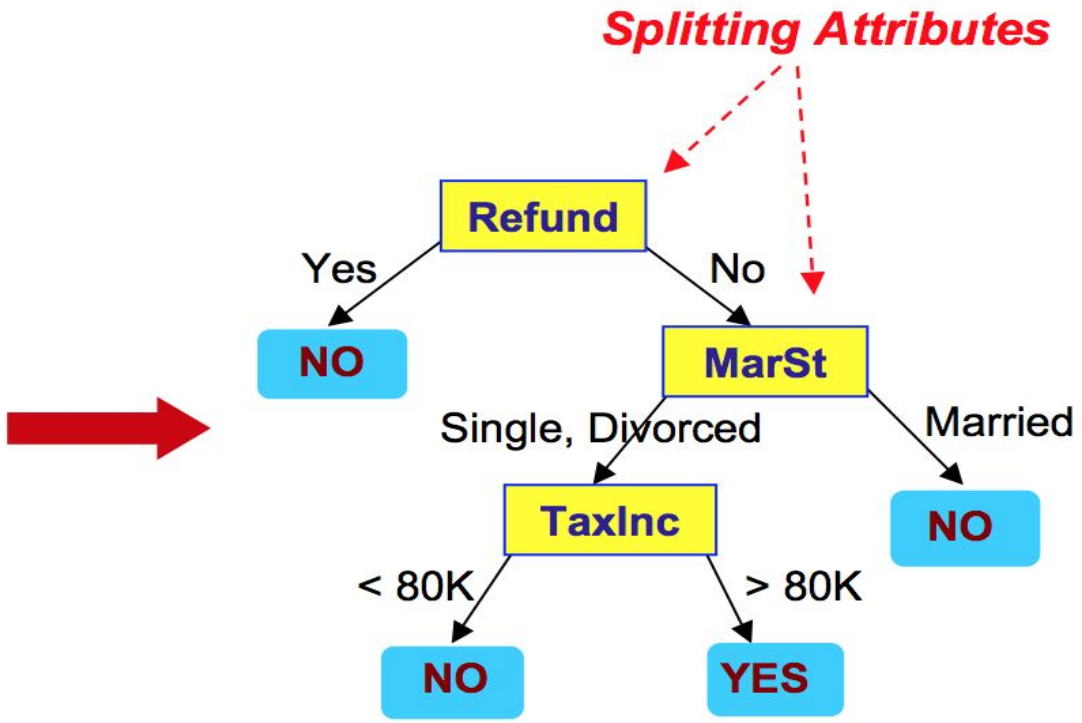


# A Decision Tree: Example

| 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 |

categorical categorical continuous class

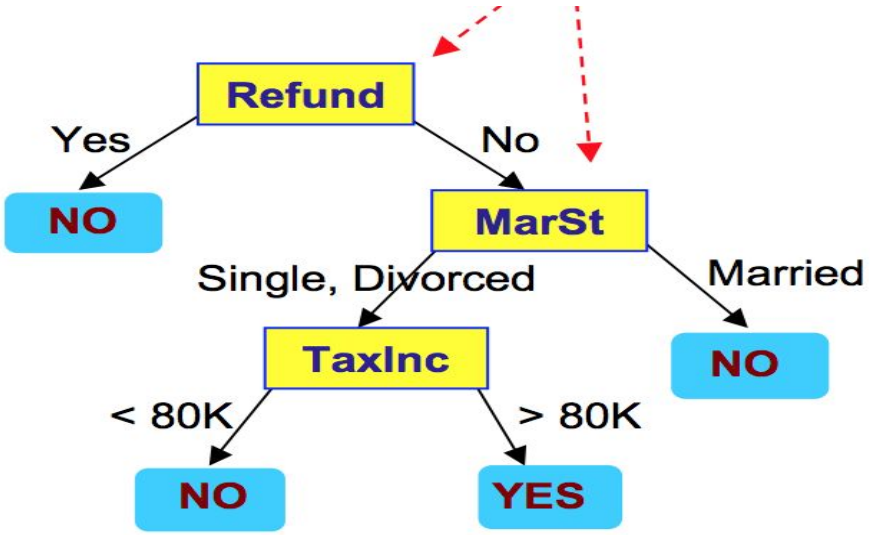
**Training Data**

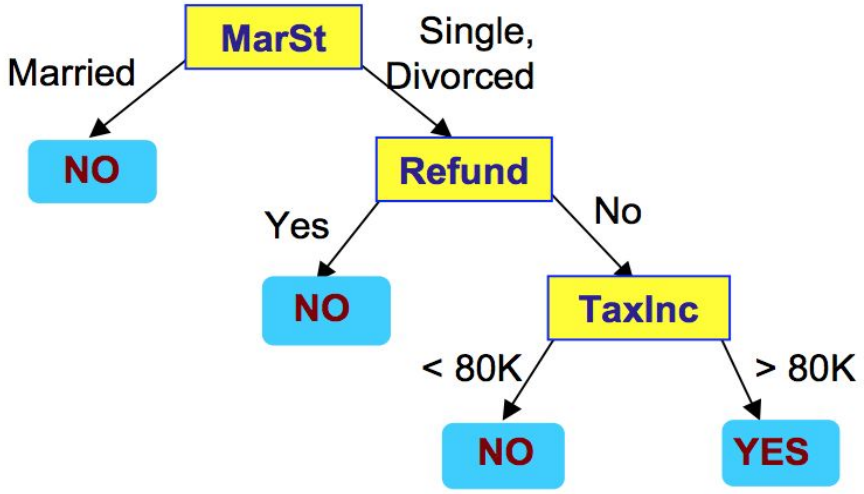



**Model: Decision Tree**

# A Decision Tree: Another example

There might be more that one tree that fits the data!



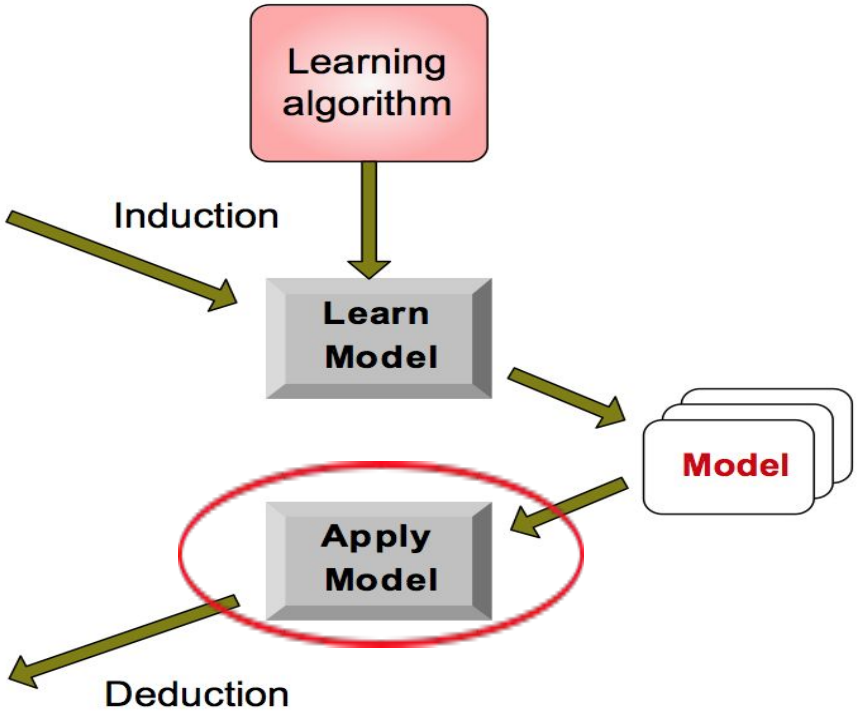
# Decision tree classification task

| 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 |

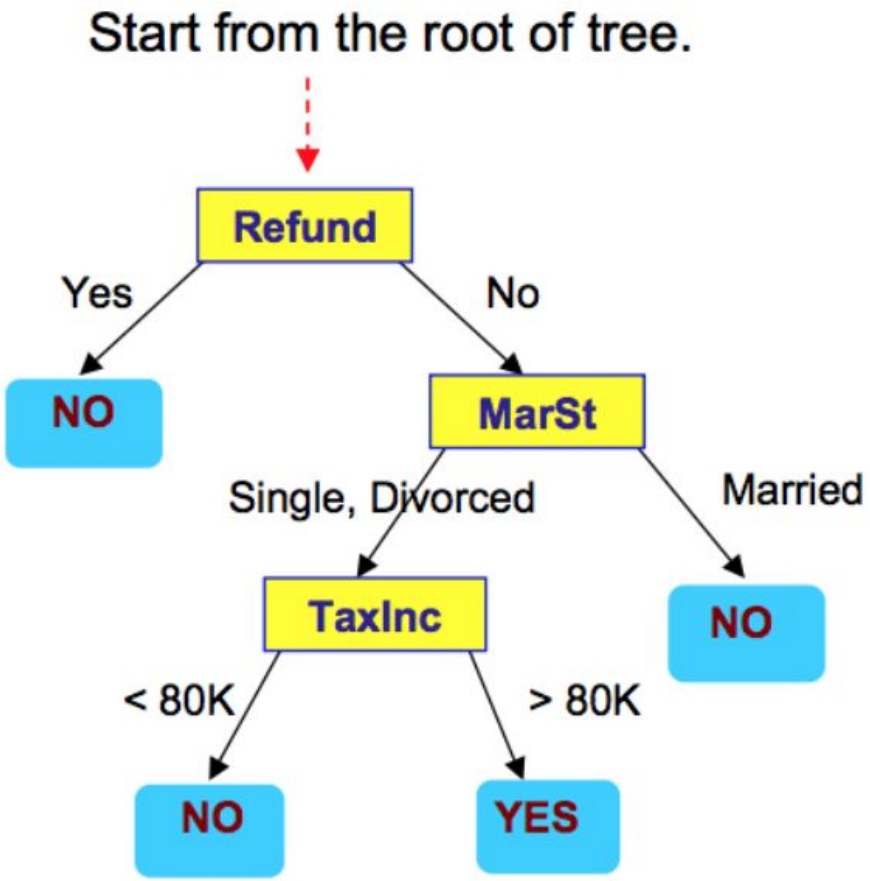
Training Set

| 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



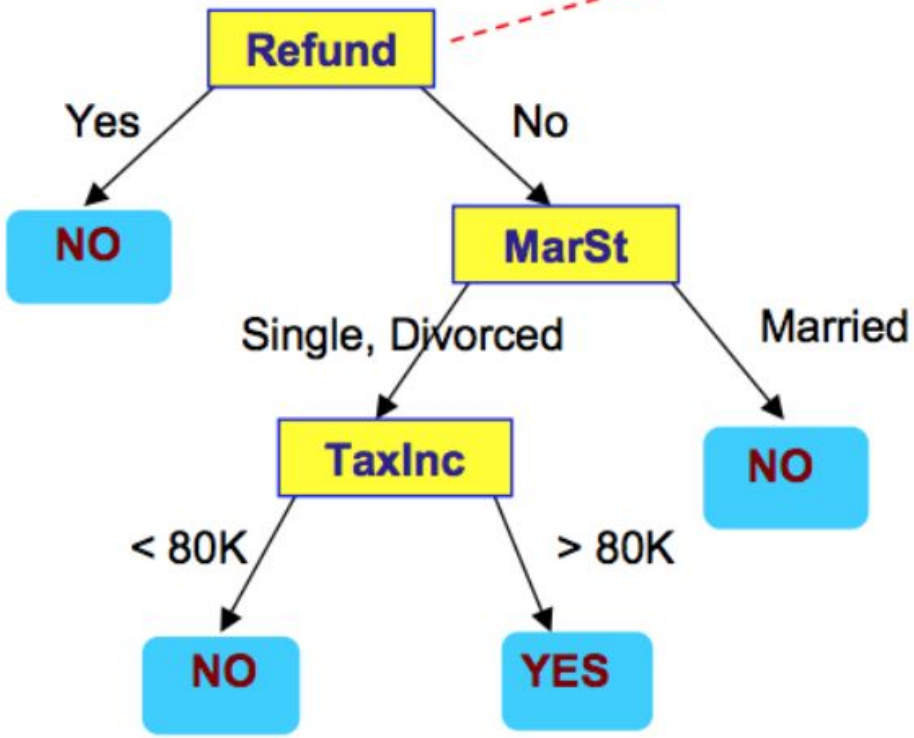

**Test Data**

| Refund | Marital Status | Taxable Income | Cheat |
|---|---|---|---|
| No | Married | 80K | ? |

# Apply Model to Test Data

**Test Data**

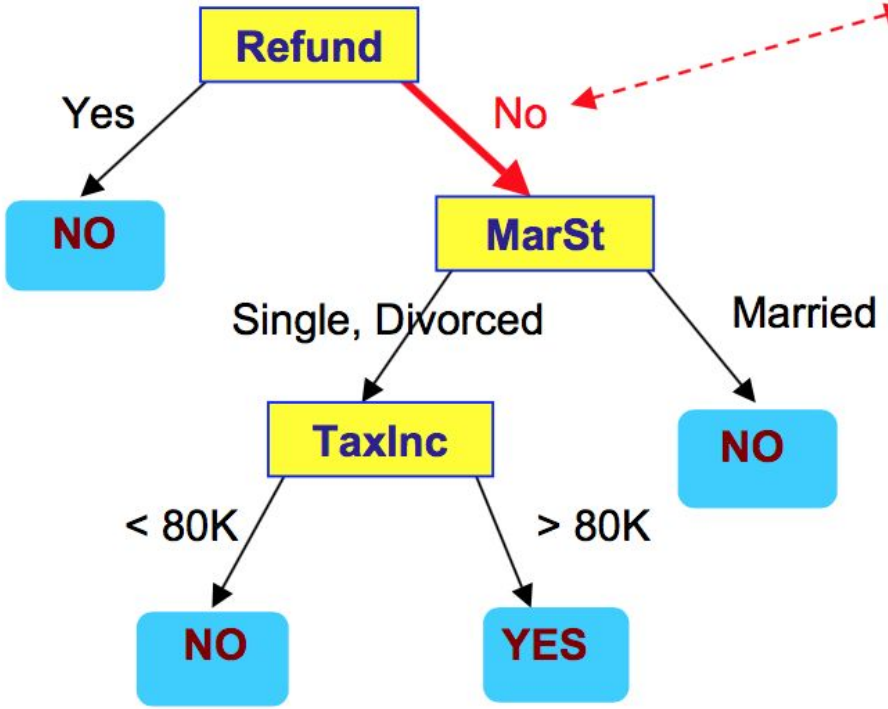
| Refund | Marital Status | Taxable Income | Cheat |
|---|---|---|---|
| No | Married | 80K | ? |



# Apply Model to Test Data

**Test Data**

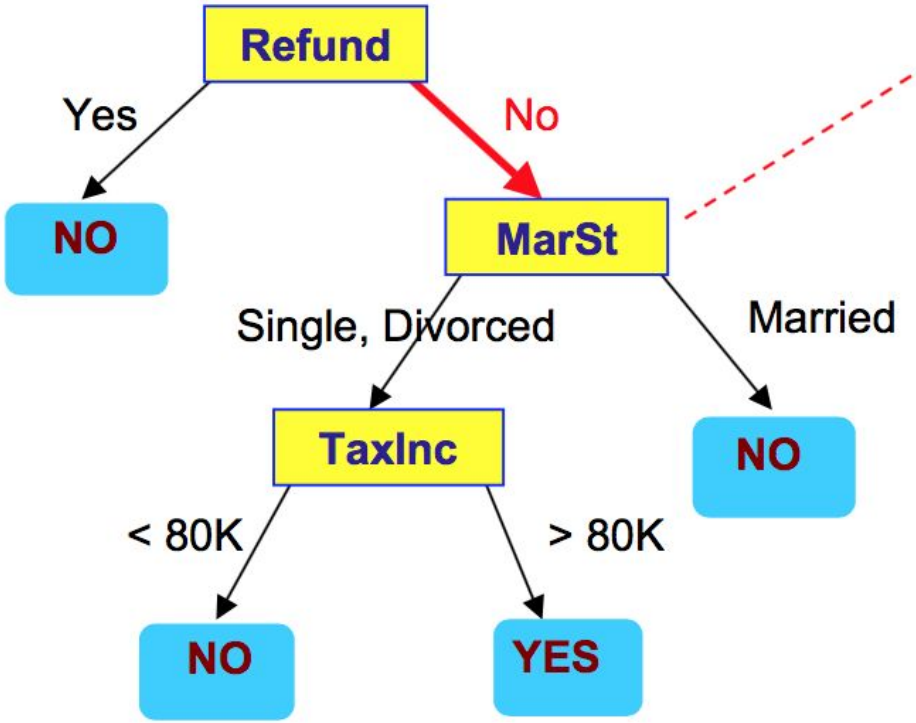
| Refund | Marital Status | Taxable Income | Cheat |
|---|---|---|---|
| No | Married | 80K | ? |



# Apply Model to Test Data

**Test Data**

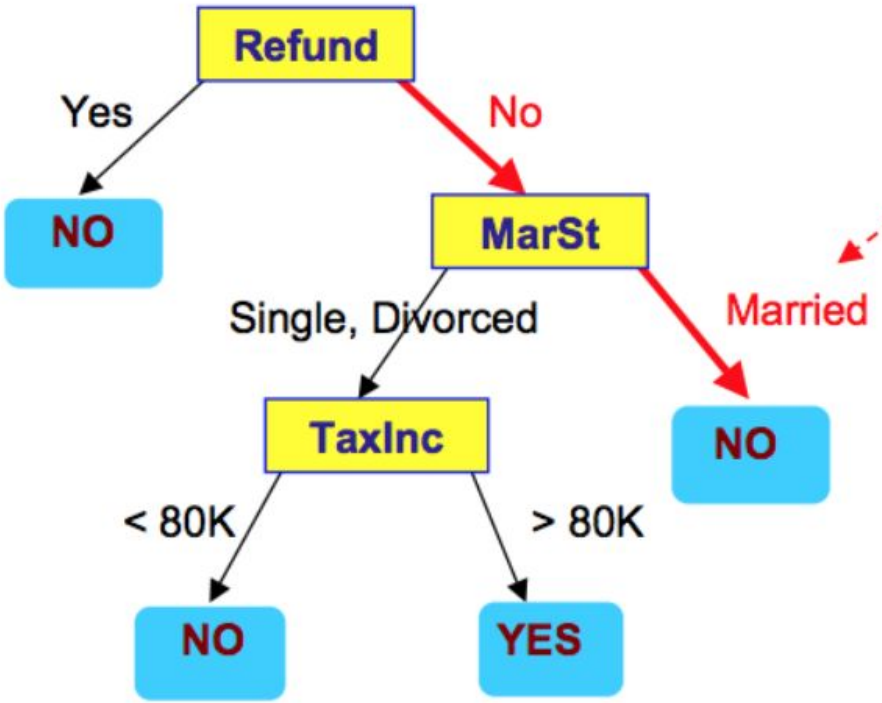
| Refund | Marital Status | Taxable Income | Cheat |
|---|---|---|---|
| No | Married | 80K | ? |



# Apply Model to Test Data

**Test Data**

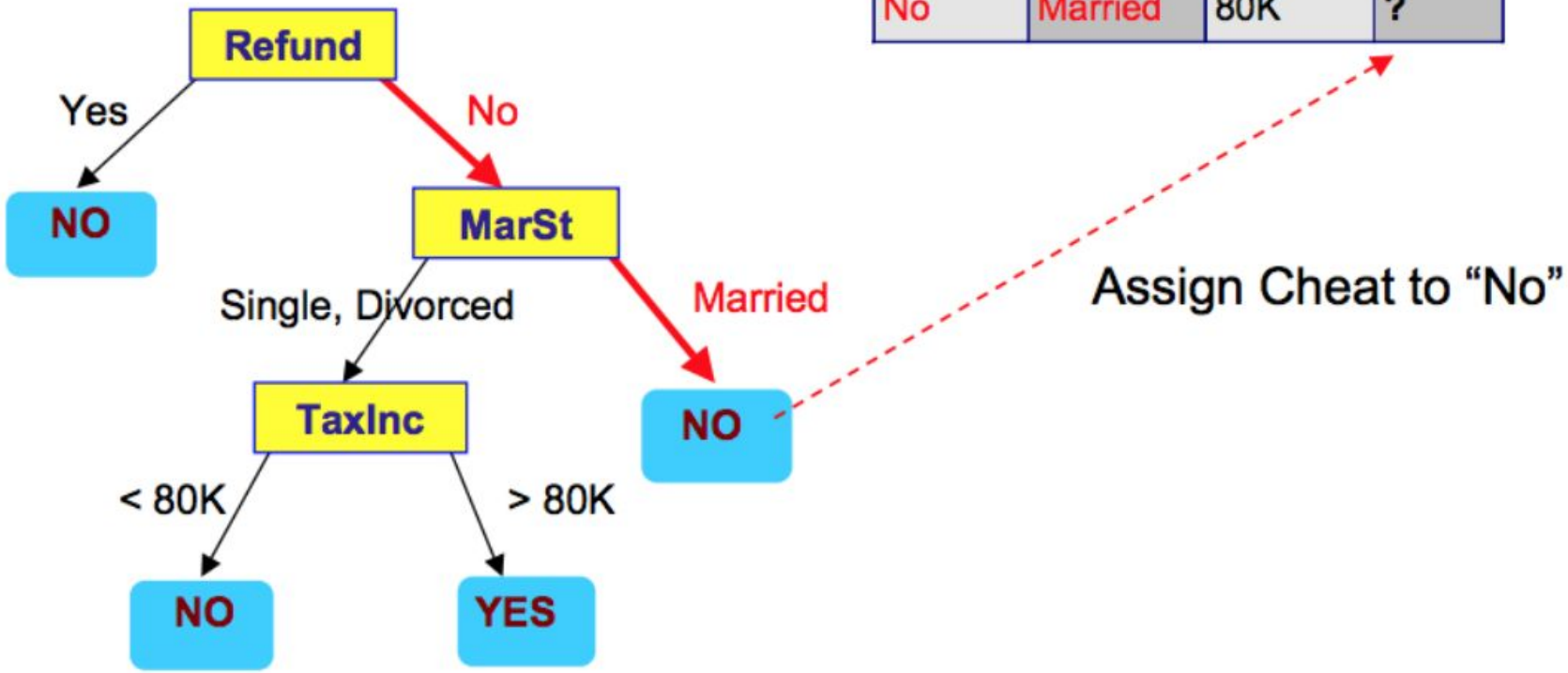
| Refund | Marital Status | Taxable Income | Cheat |
|---|---|---|---|
| No | Married | 80K | ? |



# Apply Model to Test Data

**Test Data**

| Refund | Marital Status | Taxable Income | Cheat |
|---|---|---|---|
| No | Married | 80K | ? |



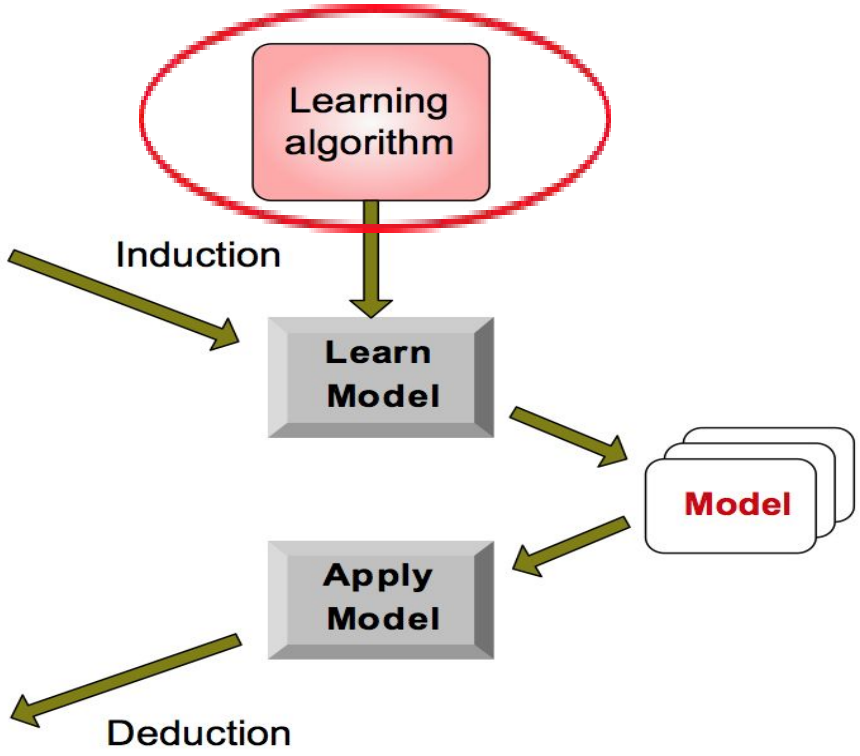
# Decision tree classification task

| 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

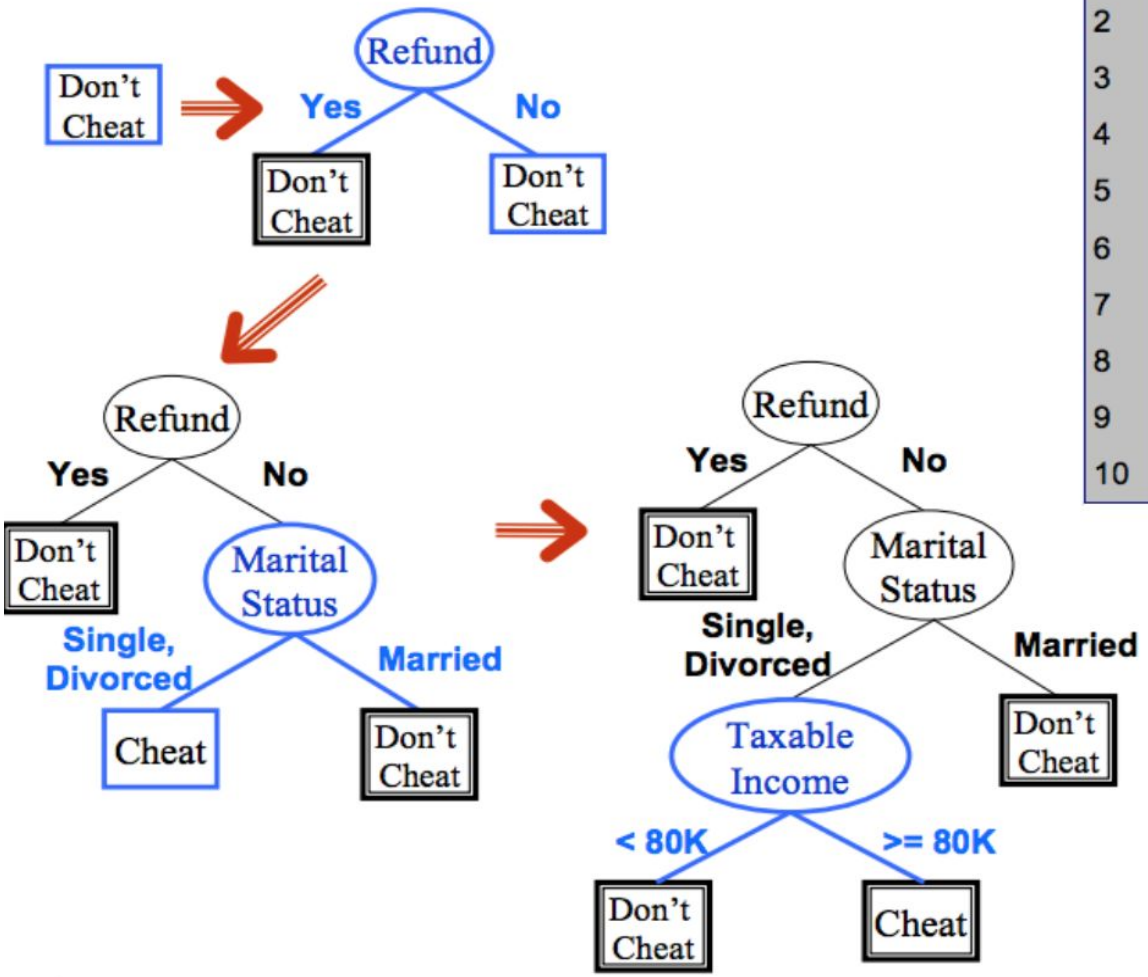


# Decision tree learning algorithms

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

# 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 |



# Decision 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

# How to specify test condition?

Depends on **attribute types**

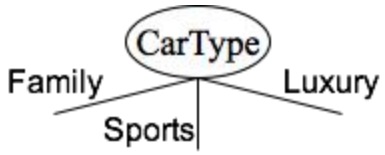
- Nominal
- Ordinal
- Continuous

Depends on **number of ways to split**

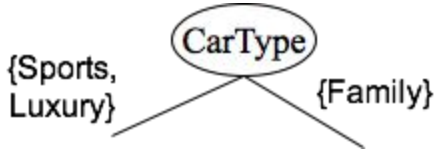
- 2-way split
- Multi-way split

# Splitting Based on Nominal Attributes

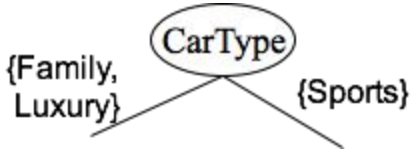
Multi-way split: Use as many partitions as distinct values.




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

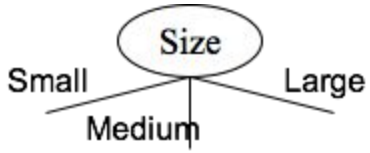



OR



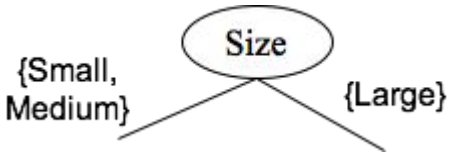
# Splitting Based on Ordinal Attributes

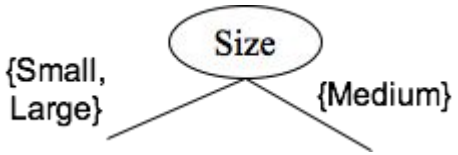
Multi-way split: Use as many partitions as distinct values.

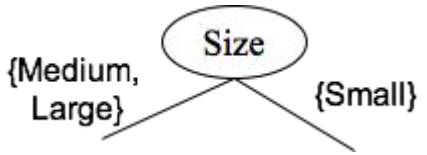



Binary split: Divides values into two subsets.

Need to find optimal partitioning.



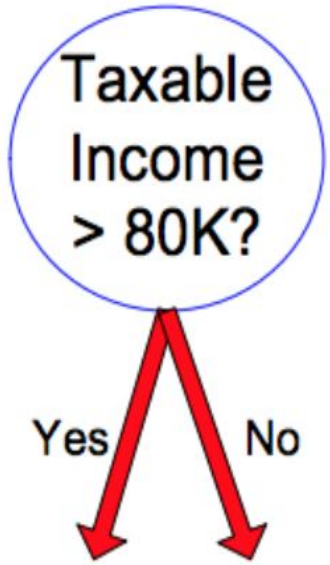



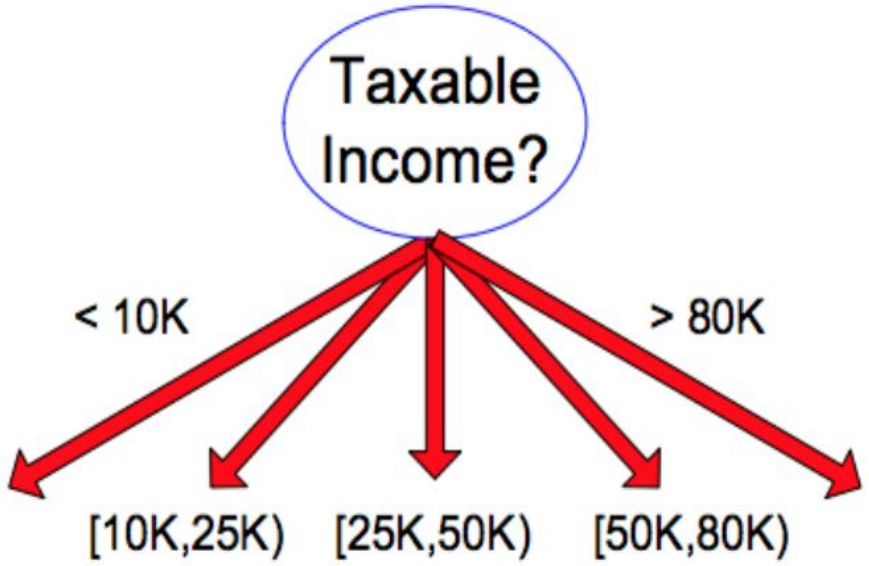
# Splitting Based on Continuous Attributes

- 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




(i) Binary split




(ii) Multi-way split

# Decision tree induction

**Greedy strategy**

Split the records based on an attribute test that optimizes certain criterion.

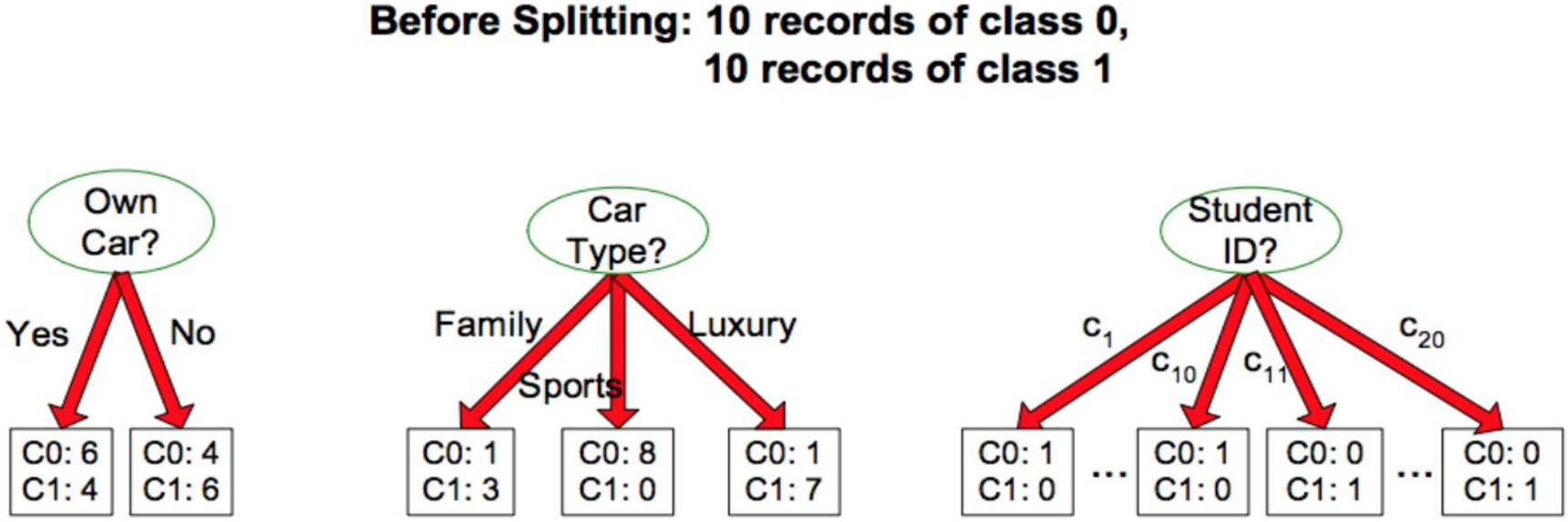
**Issues**

Determine how to split the records

- How to specify the attribute test condition?
- How to determine the best split?

Determine when to stop splitting

# How to determine the best split?



# How to determine the best split?

Greedy approach:

- Nodes with homogeneous class distribution are preferred

Need a measure of node impurity:

| C0: 5<br>C1: 5 | C0: 9<br>C1: 1 |
|---|---|
| **Non-homogeneous,**<br>**High degree of impurity** | **Homogeneous,**<br>**Low degree of impurity** |

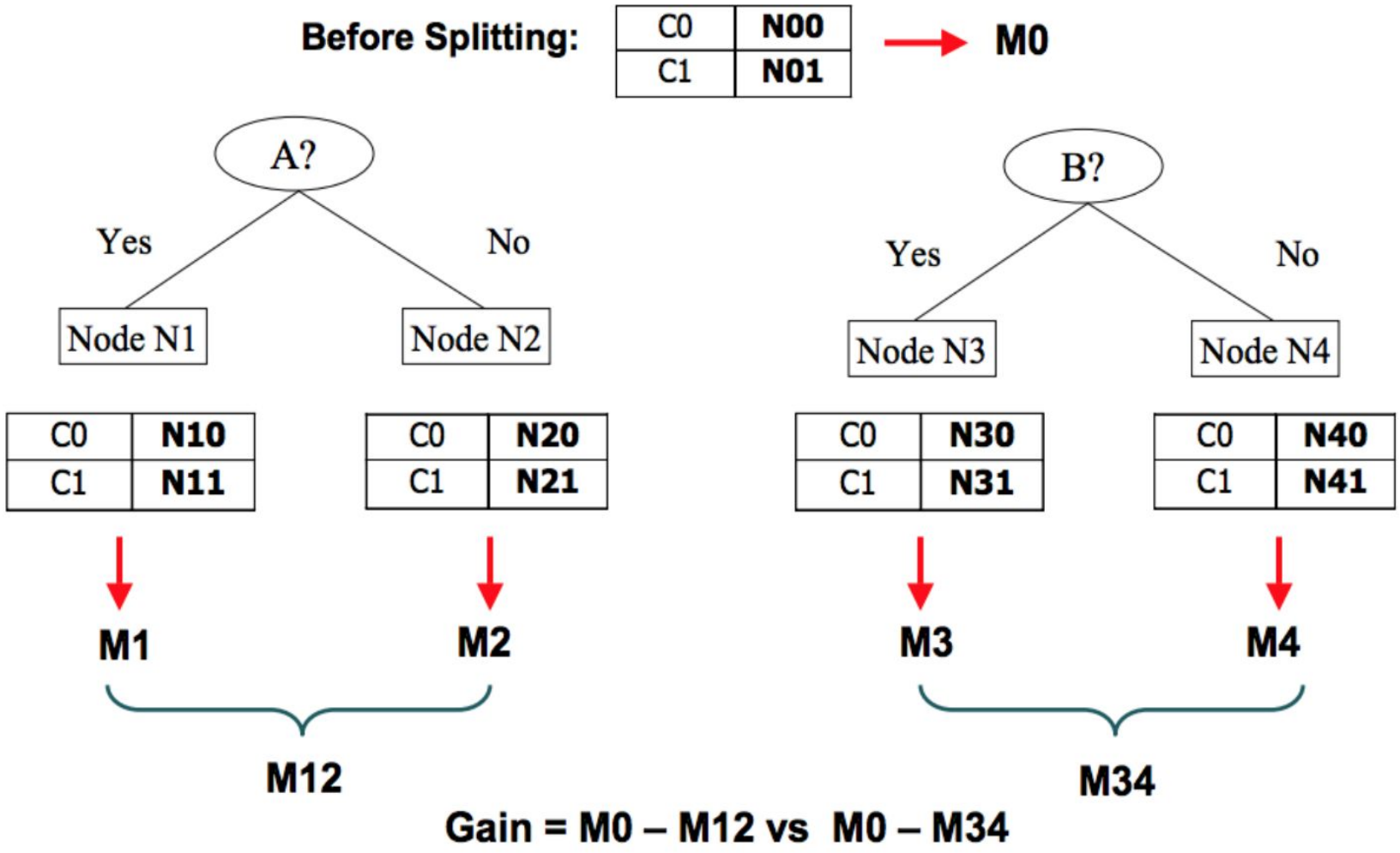
# Measures of Node Impurity

Gini Index

Entropy

Misclassification error

# How to Find the Best Split



# Measure of impurity: GINI

Gini Index for a given node t :

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

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- 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

| 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 | |

# Example of GINI measure

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

| | |
|---|---|
| C1 | **0** |
| C2 | **6** |

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

**Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0**

| | |
|---|---|
| C1 | **1** |
| C2 | **5** |

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

**Gini = 1 – (1/6)² – (5/6)² = 0.278**

| | |
|---|---|
| C1 | **2** |
| C2 | **4** |

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

**Gini = 1 – (2/6)² – (4/6)² = 0.444**

# Splitting based on GINI

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

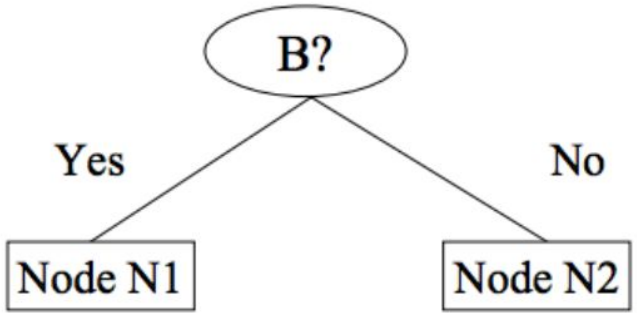
$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where, $n_i$ = number of records at child i,

n = number of records at node p.

# Computing GINI index for Binary Attributes

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.




| | Parent |
|---|---|
| C1 | 6 |
| C2 | 6 |
| Gini = 0.500 | |

**Gini(N1)**
**= 1 – (5/6)² – (2/6)²**
**= 0.194**

**Gini(N2)**
**= 1 – (1/6)² – (4/6)²**
**= 0.528**

| | N1 | N2 |
|---|---|---|
| C1 | 5 | 1 |
| C2 | 2 | 4 |
| Gini=0.333 | | |

**Gini(Children)**
**= 7/12 * 0.194 +**
**5/12 * 0.528**
**= 0.333**

# Computing GINI index for Categorical Attributes

- 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 | | |

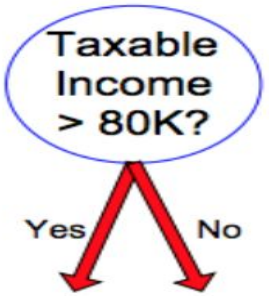
Two-way split
(find best partition of values)

| | 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 | |

# Computing GINI index for Continuous 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 |



# Computing GINI index for Continuous Attributes

- 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 | | 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 | |

# Measures of Node Impurity

Gini Index

**Entropy**

Misclassification error

# Measure of impurity: Entropy

Entropy at a given node t:

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

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Measures homogeneity of a node.
  - 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
- Entropy based computations are similar to the GINI index computations

# Example of Entropy measure

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

| | |
|---|---|
| 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) $\log_2$ (1/6) = 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 GINI

- Information Gain:

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

Parent Node, p is split into k partitions;

$n_i$ is number of records in partition i

  - Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
  - Used in ID3 and C4.5
  - Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

# Splitting based on GINI

- Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions

$n_i$ is the number of records in partition i

- Adjusts Information Gain by the entropy of the partitioning (SplitINFO). Higher entropy partitioning (large number of small partitions) is penalized!
- Used in C4.5
- Designed to overcome the disadvantage of Information Gain

# Measures of Node Impurity

Gini Index

Entropy

**Misclassification error**

# Measure of impurity: Misclassification error

- Classification error at a node t :

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

- 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

# Decision 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

# Tree induction: stopping criteria

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 (avoid overfitting)

# Classification: algorithms

**Decision Trees**

**Advantages:**

- Inexpensive to construct
- Extremely fast at classifying unknown records
- Easy to interpret for small-sized trees
- Accuracy is comparable to other classification techniques for many simple data sets

# Individual assignment

Describe the similarities and differences between logistic regression, SVM and decision trees from different aspects

Given a small dummy data sample. Compute different measures of best split such as Gini index, entropy and information gain.

# Next class

**Decision trees:**

- Model overfitting
- estimation of generalization errors
- handling overfitting in decision tree induction
- evaluating the performance of a classifier
- methods for comparing classifiers

**Decision trees with Python:**

- get the data
- split the data into a training and a test set
- train a single decision tree
- evaluate the decision trees and the prediction results
- decision tree visualization

# Resources


- Tan, Pang-Ning. *Introduction to data mining*. Pearson Education India, 2006.

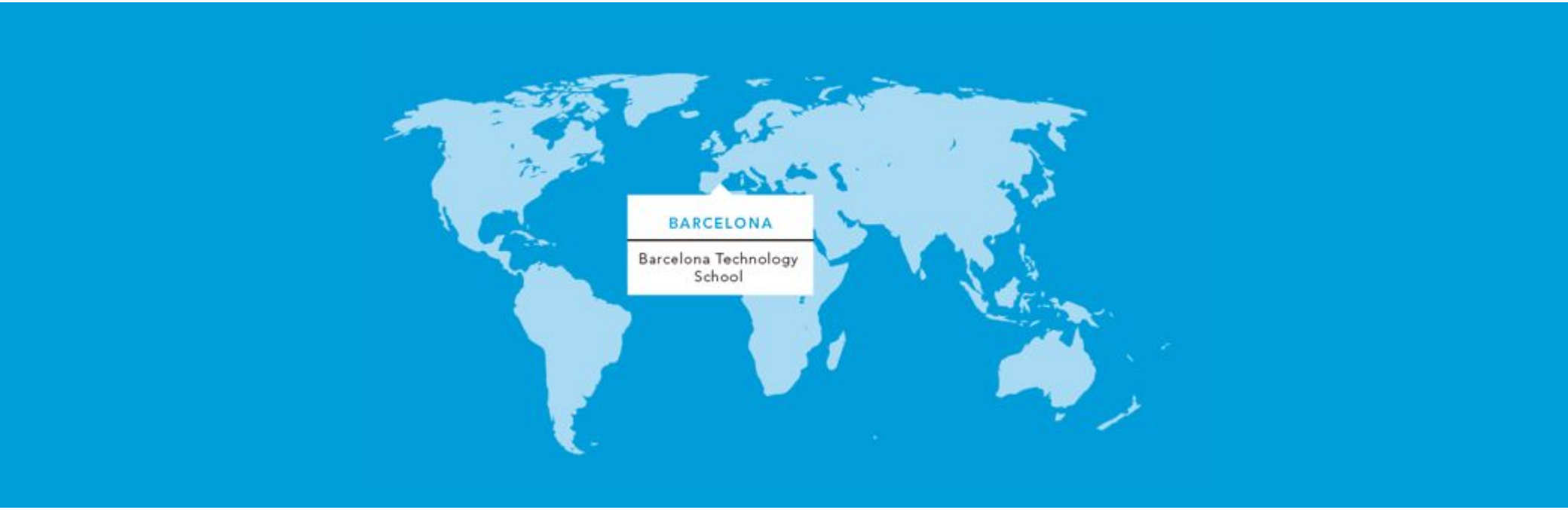
- Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. New York: Springer series in statistics, 2001.

- Examples of SVM with different learning parameters

https://chrisalbon.com/machine-learning/svc_parameters_using_rbf_kernel.html

BTS | Barcelona Technology School

eurecat Centre Tecnològic de Catalunya




# Thank you

Barcelona, 2017