DATA MINING
SUPERVISED LEARNING

Regression

Classification

- Decision Trees

Supervised learning

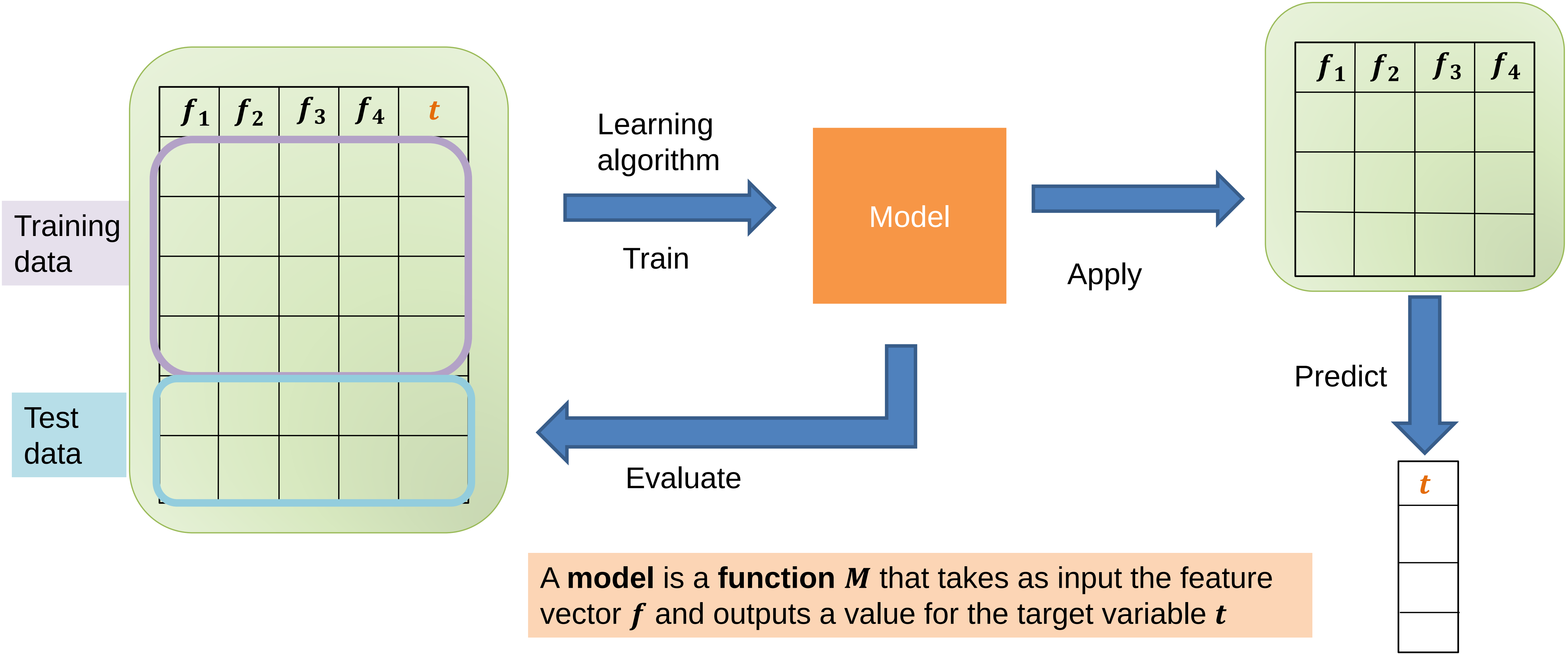
- In supervised learning, except for the feature variables that describe the data, we also have a target variable
- The goal is to learn a function (model) that can estimate/predict the value of the target variable given the features
  - We learn the function using a labeled training set.
- Regression: The target variable (but also the features) is numerical and continuous
  - The price of a stock, the GDP of a country, the grade in a class, the height of a child, the life expectancy etc
- Classification: The target variable is discrete
  - Does a taxpayer cheat or not? Will the stock go up or down? Will the student pass or fail? Is a transaction fraudulent or not? What is the topic of an article?

Applications

- Descriptive modeling: Explanatory tool to understand the data:
  - Regression: How does the change in the value of different factors affect our target variable?
    - What factors contribute to the price of a stock?
    - What factors contribute to the GDP of a country?
  - Classification: Understand what attributes distinguish between objects of different classes
    - Why people cheat on their taxes?
    - What words make an post offensive?

- Predictive modeling: Predict a class of a previously unseen record
  - Regression: What will the life-expectancy of a patient be?
  - Classification: Is this a cheater or not? Will the stock go up or not. Is this an offensive post?

Supervised Learning Overview



A **model** is a **function** $\boldsymbol{M}$ that takes as input the feature vector $\boldsymbol{f}$ and outputs a value for the target variable $\boldsymbol{t}$

LINEAR REGRESSION

Regression

- We assume that we have k feature variables (numeric):
  - Also known as covariates（协变量）, or independent variables（自变量）
- The target variable is also known as dependent variable（因变量）.
- We are given a dataset of the form $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$ where, $\boldsymbol{x}_i$ is a k-dimensional feature vector, and y_i a real value
- We want to learn a function f which given a feature vector $\boldsymbol{x}_i$ predicts a value $y_i' = f(\boldsymbol{x}_i)$ that is as close as possible to the value y_i
- Minimize sum of squares:

$$\sum_i \left(y_i - f(\boldsymbol{x}_i)\right)^2$$

Linear regression

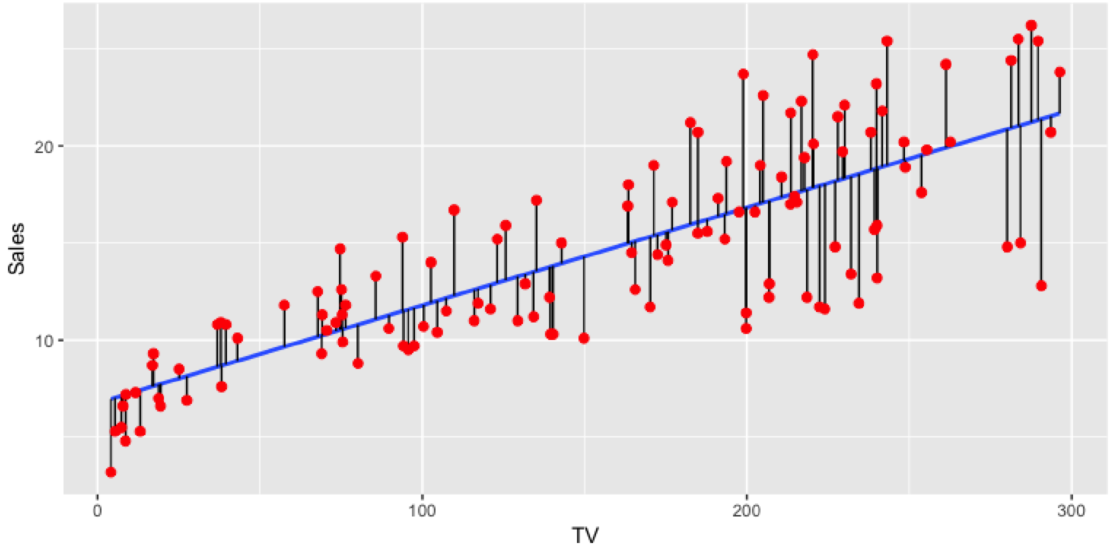
- The simplest form of f is a linear function
- In linear regression the function f is typically of the form:

$$f(\boldsymbol{x_i}) = w_0 + \sum_{j=1}^{k} w_j x_{ij}$$

w_0:截距

x_{ij} : x_i的第j个维度

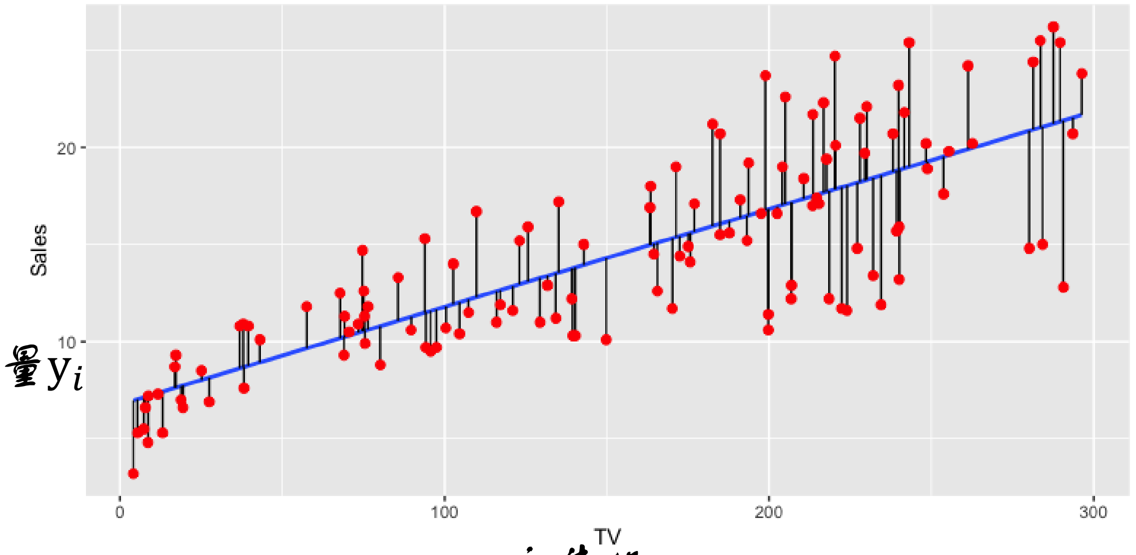
w_j: 第j个维度的系数



One-dimensional linear regression

In the simplest case we have a single variable and the function is of the form:

$$f(x_i) = w_0 + w_1 x_i$$



SSE (loss): $\sum_i \left(y_i - f(\boldsymbol{x}_i)\right)^2$

= $\sum_i (y_i - w_0 - w_1 x_i)^2$

Minimize the loss -> partial derivatives of w_0 and w_1:

∂Loss/∂w_0 = Σ(-2(y_i - ($w_1 x_i$ + w_0))) = 0, $w_0 = \bar{y} - w_1 \bar{x}$

∂Loss/∂w_1 = Σ(-2x_i(y_i - ($w_1 x_i$ + w_0))) = 0,

$$w_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} = r_{xy} \frac{\sigma_y}{\sigma_x}$$

$\bar{x}$: mean value of x_i's

$\bar{y}$: mean value of y_i's

r_{xy}: correlation coefficient between $\boldsymbol{x}, \boldsymbol{y}$

σ_x, σ_y :standard deviation

Multiple linear regression

- In the general case we have k features, and $\boldsymbol{x_i}$,$\boldsymbol{w}$ are vectors.
- We simplify the notation:

$$\boldsymbol{x_i} = (1, x_{i1}, \dots, x_{ik})$$
$$\boldsymbol{w} = (w_0, w_1, \dots, w_k)$$
$$f(\boldsymbol{x_i}, \boldsymbol{w}) = \boldsymbol{x_i}^T \boldsymbol{w}$$

($\boldsymbol{x_i}$和$\boldsymbol{w}$默认列向量)

- Let X be the $n \times (k+1)$ matrix with vectors $\boldsymbol{x_i}$ as rows.
- Let $\boldsymbol{y} = (y_1, \dots, y_n)$; $f(X) = Xw$
- We can write the SSE function as:

$$SSE = \|X\boldsymbol{w} - \boldsymbol{y}\|^2$$

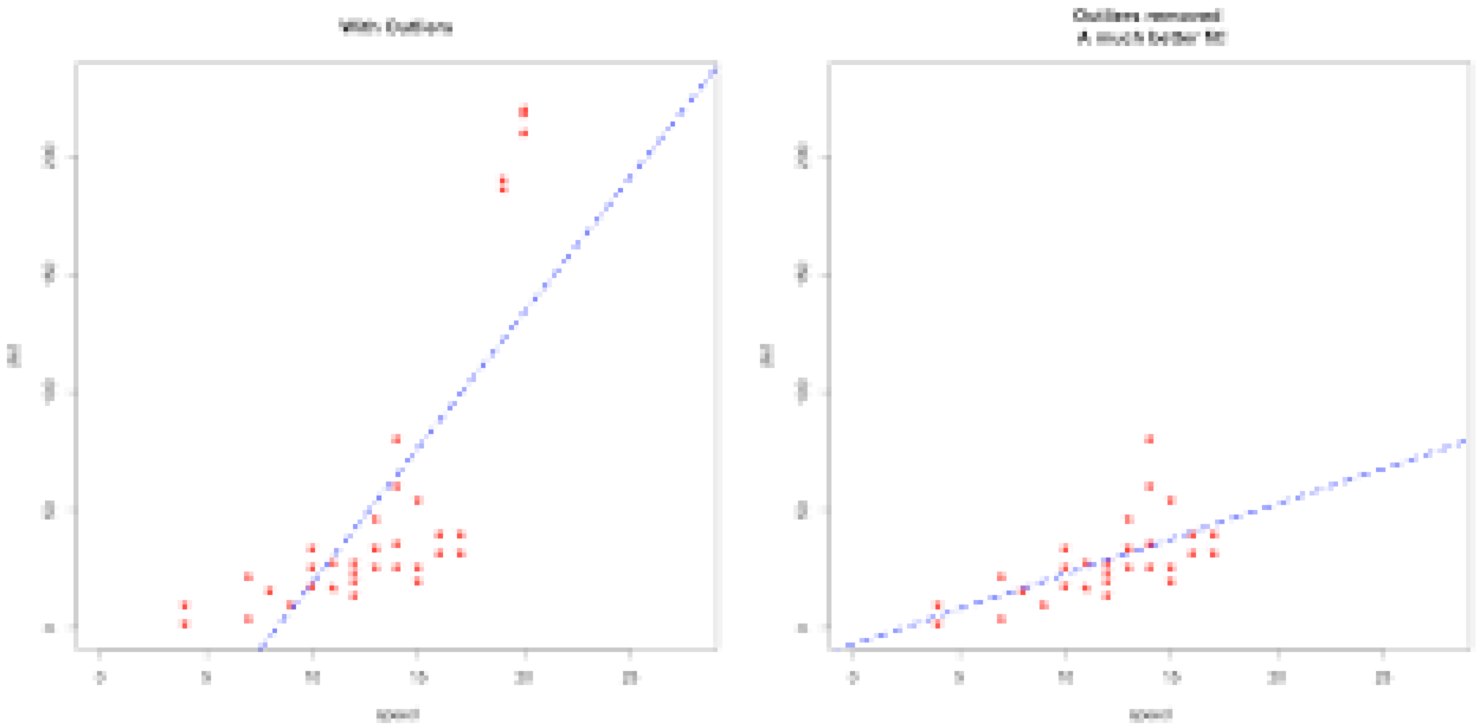
- There is a closed-form solution for $\boldsymbol{w}$:

w的解析解

$$\boldsymbol{w} = (X^T X)^{-1} X^T \boldsymbol{y}$$

- Matrix inversion may be too expensive. Other optimization techniques are often used to find the optimal vector (e.g., Gradient Descent)

Outliers



- Regression is sensitive to outliers:
  - The line will “tilt” to accommodate very extreme values

离群值造成倾斜

- Solution: remove the outliers
  - But make sure that they do not capture useful information

Normalization

- In the regression problem some times our features may have very different scales:
  - For example: predict the GDP of a country using as features the percentage of home owners and the income
  - The weights in this case will not be interpretable
- Solution: Normalize the features by replacing the values with the z-scores
  - Remove the mean and divide by the standard deviation

$$z = \frac{x - \mu}{\sigma}$$

where:

μ is the mean of the population,

σ is the standard deviation of the population.

Interpretation and significance

- A regression model is useful for making predictions for new data.
- The coefficients for the linear regression model are also useful for understanding the effect of the independent variables to the value of the dependent variable
  - The w_j value is the effect of the increase of x_{ij} by one to the value y_i
- We can also compute the significance of the value of w_j by testing the null hypothesis that $w_j = 0$

Covariate	Least Squares Estimate	Estimated Standard Error	t value	p-value	
(Intercept)	*-589.39*	*167.59*	*-3.51*	*0.001*	**
Age	*1.04*	*0.45*	*2.33*	*0.025*	*
Southern State	*11.29*	*13.24*	*0.85*	*0.399*	
Education	*1.18*	*0.68*	*1.7*	*0.093*	
Expenditures	*0.96*	*0.25*	*3.86*	*0.000*	***
Labor	*0.11*	*0.15*	*0.69*	*0.493*	
Number of Males	*0.30*	*0.22*	*1.36*	*0.181*	
Population	*0.09*	*0.14*	*0.65*	*0.518*	
Unemployment (14-24)	*-0.68*	*0.48*	*-1.4*	*0.165*	
Unemployment (25-39)	*2.15*	*0.95*	*2.26*	*0.030*	*
Wealth	*-0.08*	*0.09*	*-0.91*	*0.367*	

This table is typical of the output of a multiple regression program. The "t-value" is the Wald test statistic for testing $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$. The asterisks denote "degree of significance" with more asterisks being significant at a smaller level. The example raises several important questions. In particular: (1) should we eliminate some variables from this model? (2) should we interpret this relationships as causal? For example, should we conclude that low crime prevention expenditures cause high crime rates? We will address question (1) in the next section. We will not address question (2) until a later Chapter.

Predicting Crime rate

CLASSIFICATION

Classification

- Similar to the regression problem we have features and a target variable that we want to model/predict
- The target variable is now discrete. It is often called the class label
    - In the simplest case, it is a binary variable.
- In classification the features may also be categorical.

Example: Catching tax-evasion

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

Tax-return data for year 2021

A new tax return for 2022
Is this a cheating tax return?

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

An instance of the classification problem: learn a method for discriminating between records of different classes (cheaters vs non-cheaters)

Classification

- **Classification** is the task of ***learning a target function*** **f** that maps attribute set **x** to one of the predefined class labels **y**
- The function may be defined as an algorithm (e.g., if Single and Income < 125K then No)

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**

One of the attributes is the class attribute
In this case: Cheat

Two class labels (or classes): Yes (1), No (0)

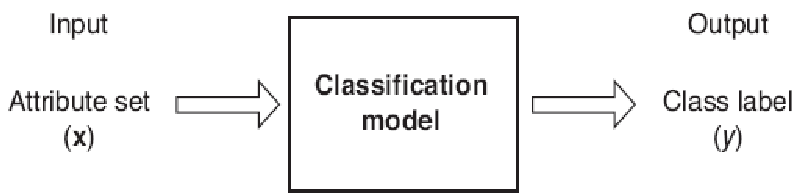


Figure 4.2. Classification as the task of mapping an input attribute set $\mathbf{x}$ into its class label y.

Examples of Classification Tasks

- Categorizing news stories as finance, weather, entertainment, sports
- Identifying spam email, spam web pages
- Predict the direction of stock market
- Classify plant species

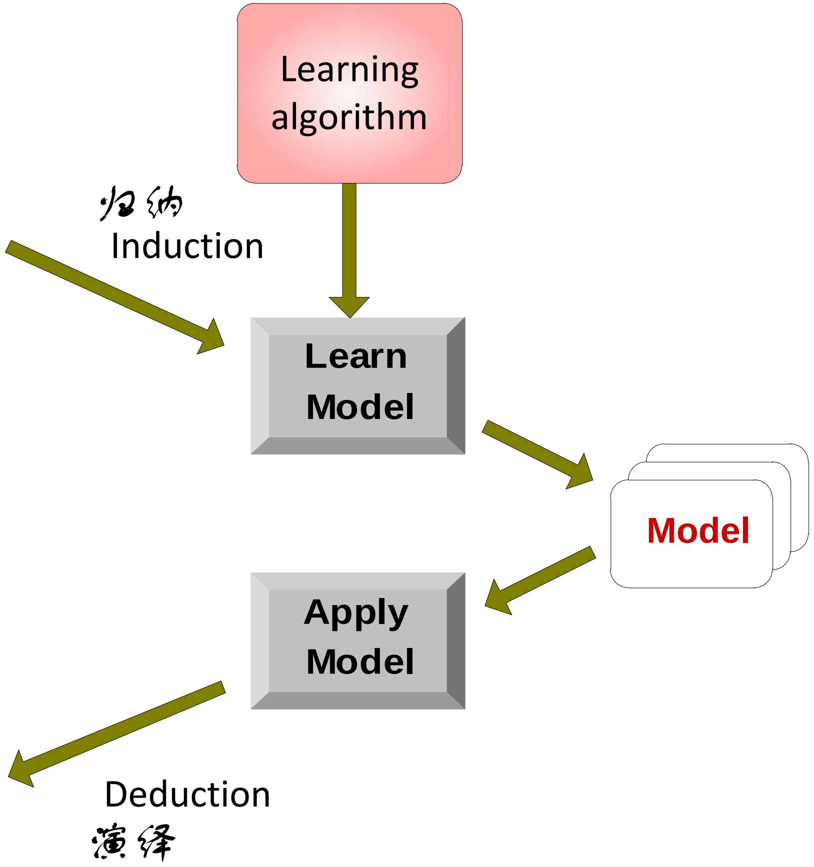
General approach to classification

- Obtain a training set consisting of records with known class labels
- Training set is used to build a classification model
- A labeled test set of previously unseen data records is used to evaluate the quality of the model.
- The classification model is applied to new records with unknown class labels
- Important intermediate step: Decide on what features to use

Illustrating 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	?

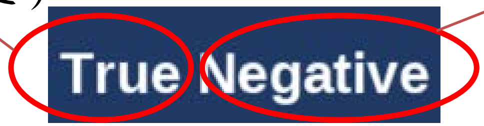
Test Set

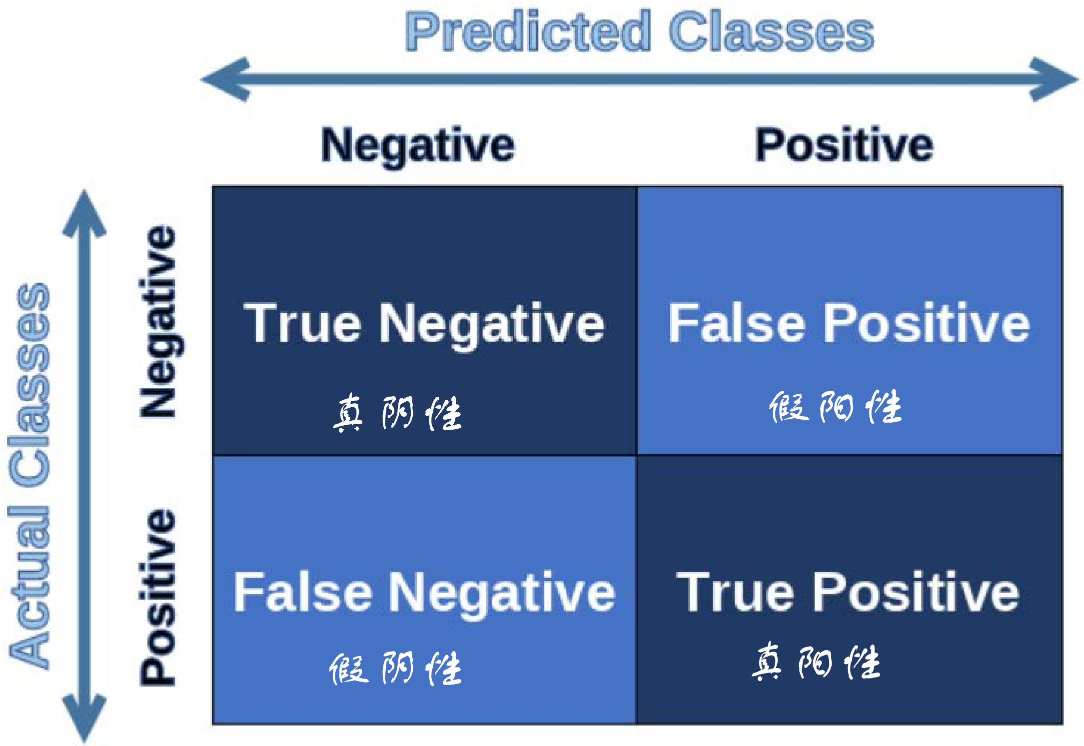
Evaluation of classification models

- **Confusion matrix**

预测值是否正确
（T/F 真/假）

预测值
Negative/Positive 阴/阳





$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

准确率，预测结果正确的比例

Accuracy can be a misleading metric for imbalanced data sets

$$\text{Precision} = \frac{tp}{tp + fp}$$

精确率，预测为阳性的案例里面正确的比例

$$\text{Recall} = \frac{tp}{tp + fn}$$

召回率，所有实际为阳性的案例里面，有被预测正确的比例

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

F measure:
precision和recall的调和平均

Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines
- Logistic Regression

DECISION TREES

Decision Trees

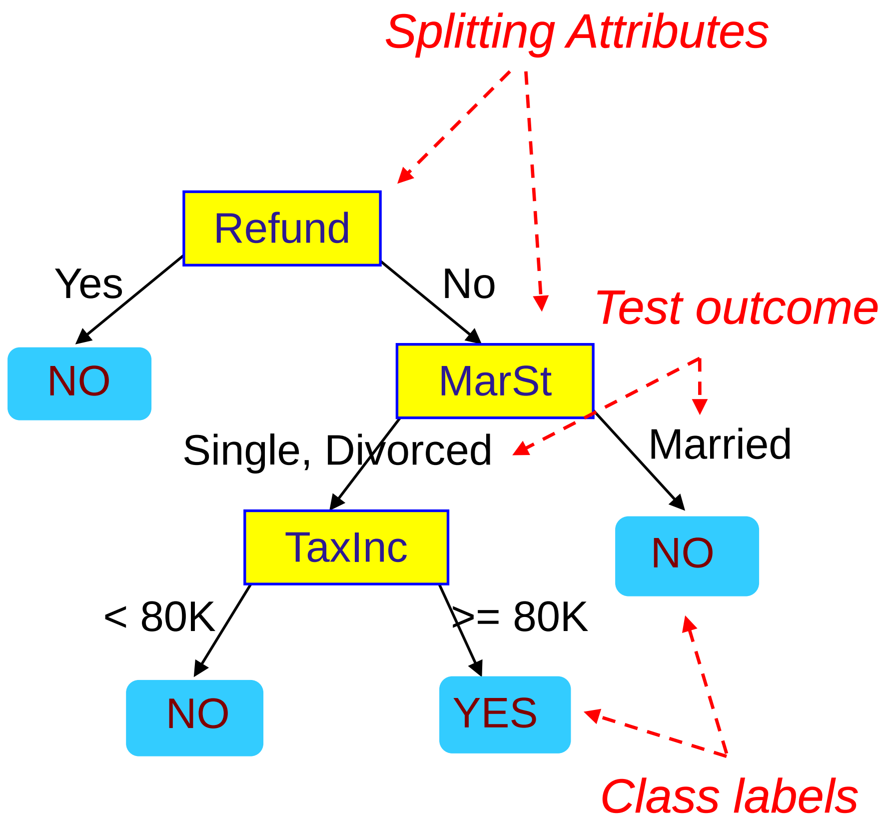
- Decision tree
  - A flow-chart-like tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution

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

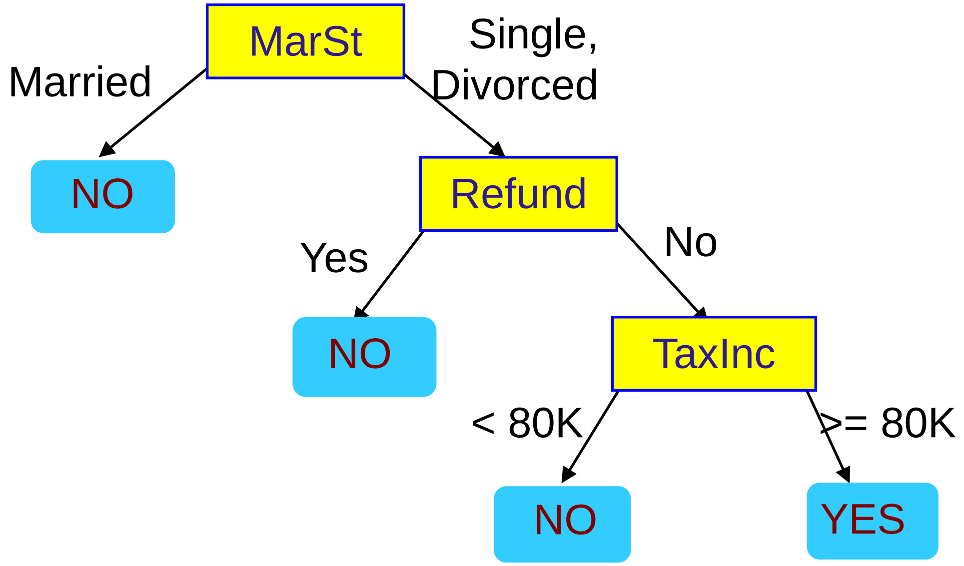


Model: Decision Tree

Another Example of 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

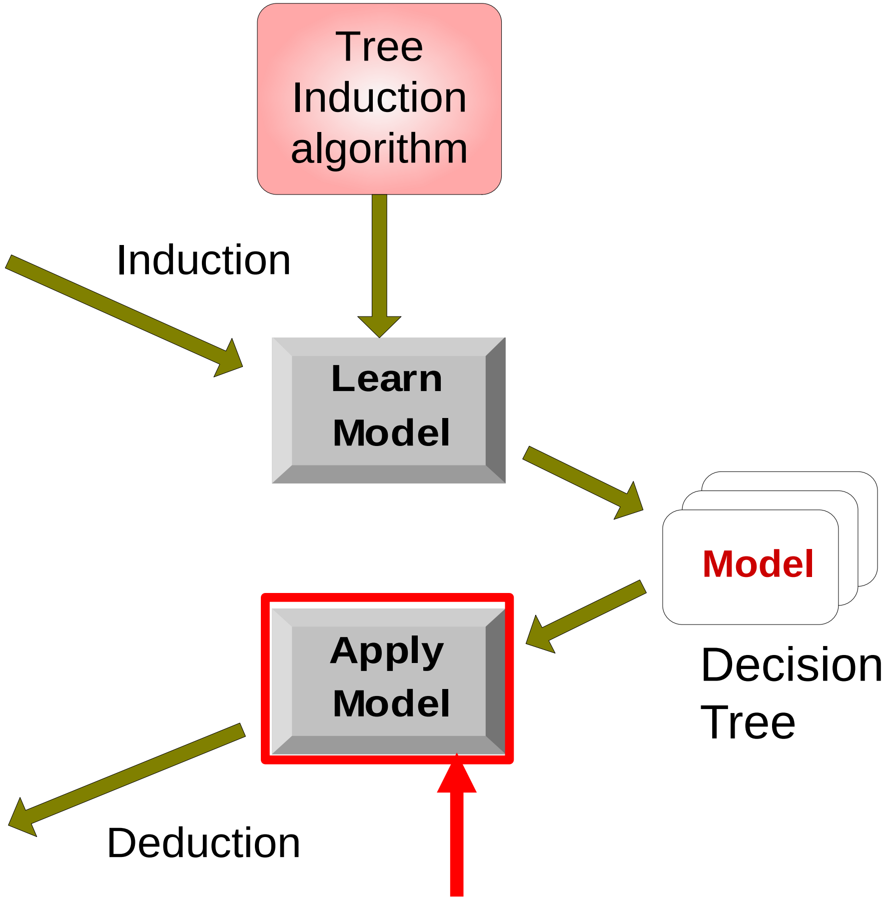


There could be more than one tree that fits the same data!

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

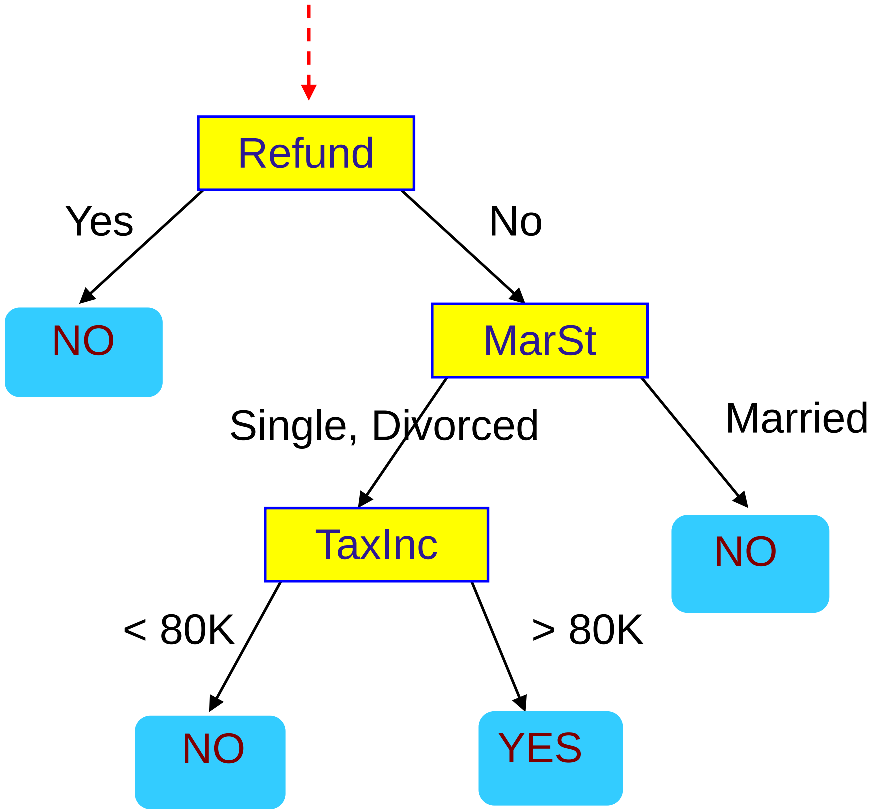


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Apply Model to Test Data

Start from the root of tree.



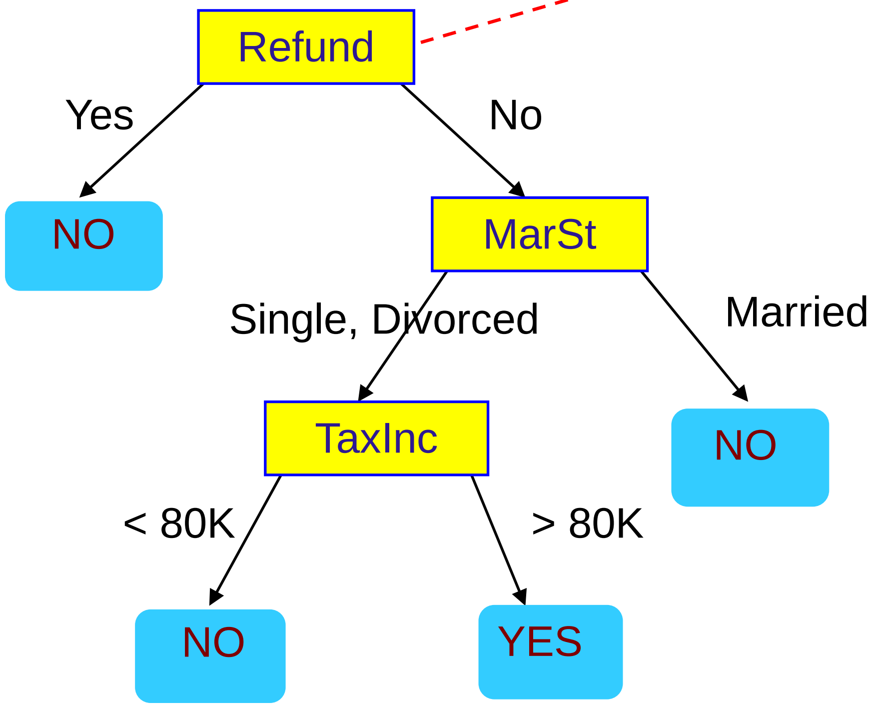
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	**?**

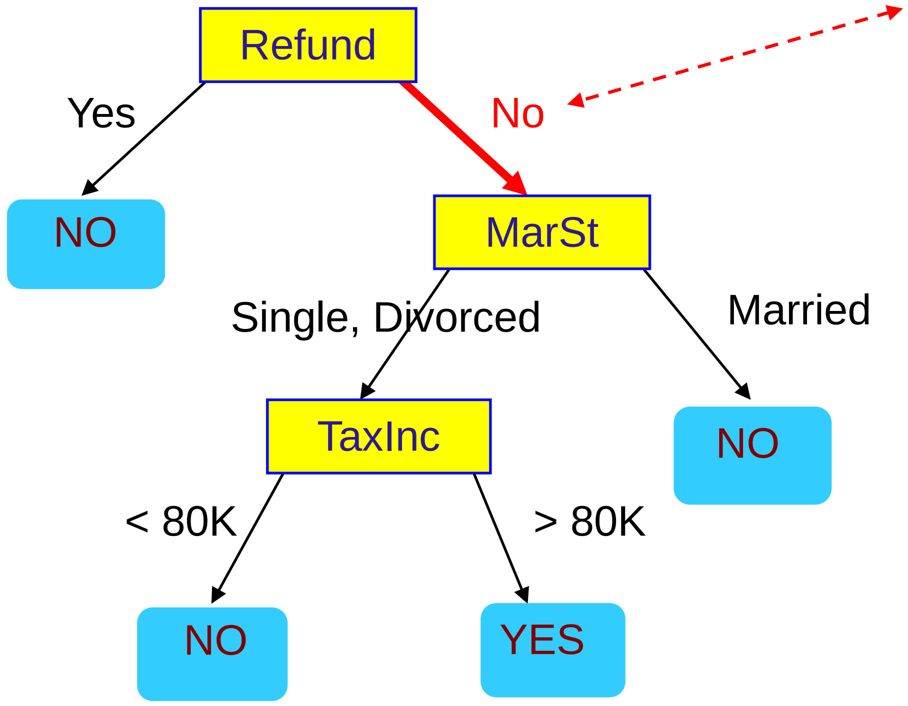
Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



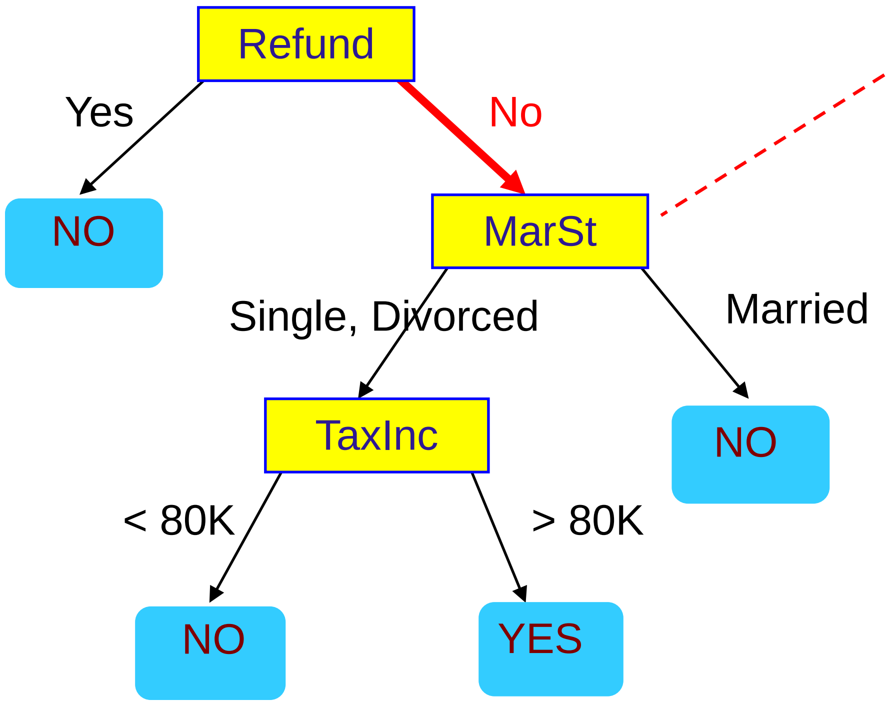
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

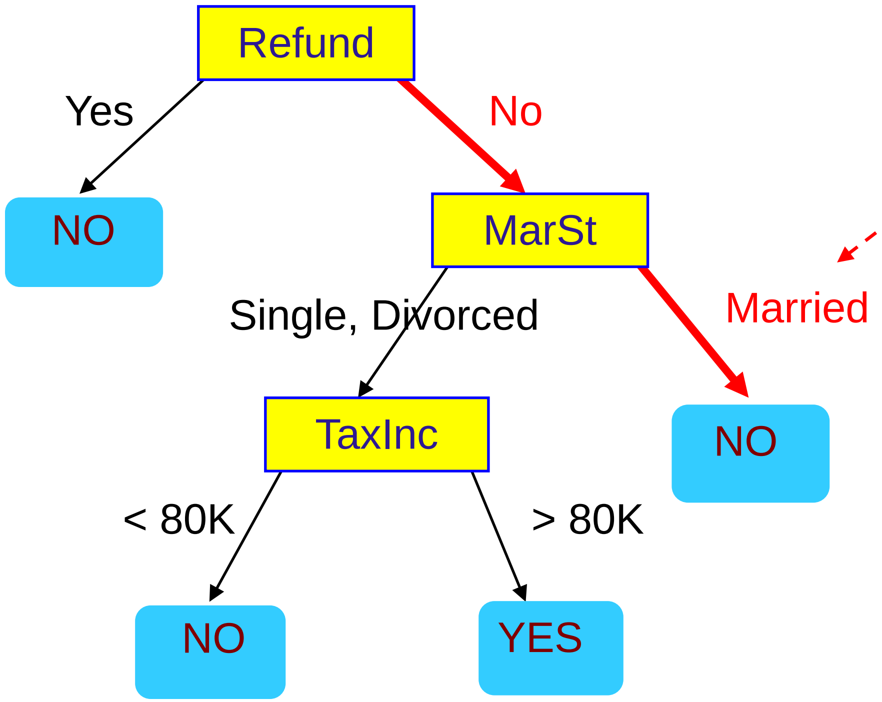
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



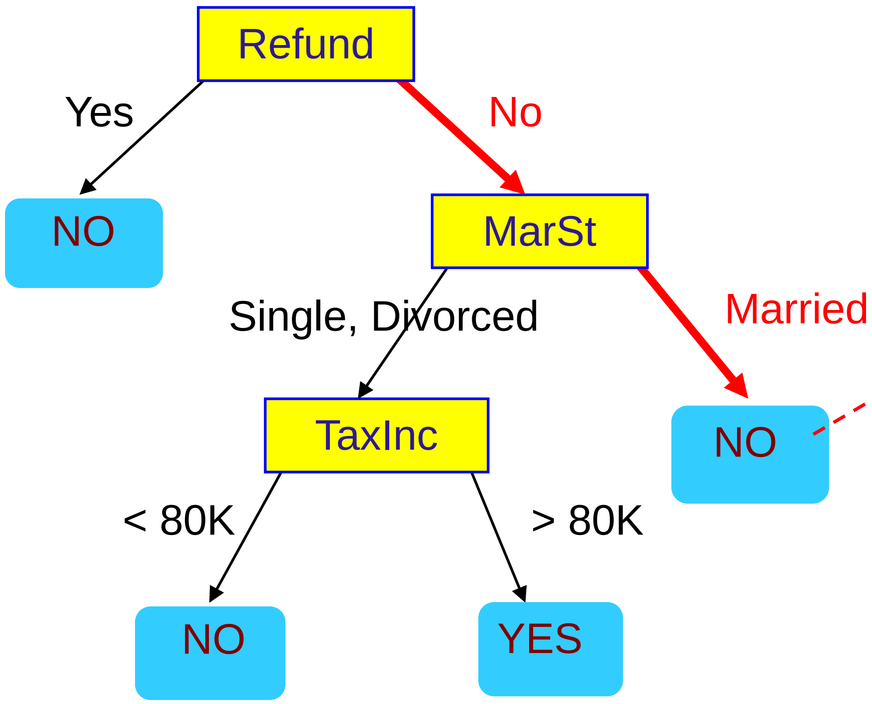
Test Data

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	?



Assign Cheat to “No”

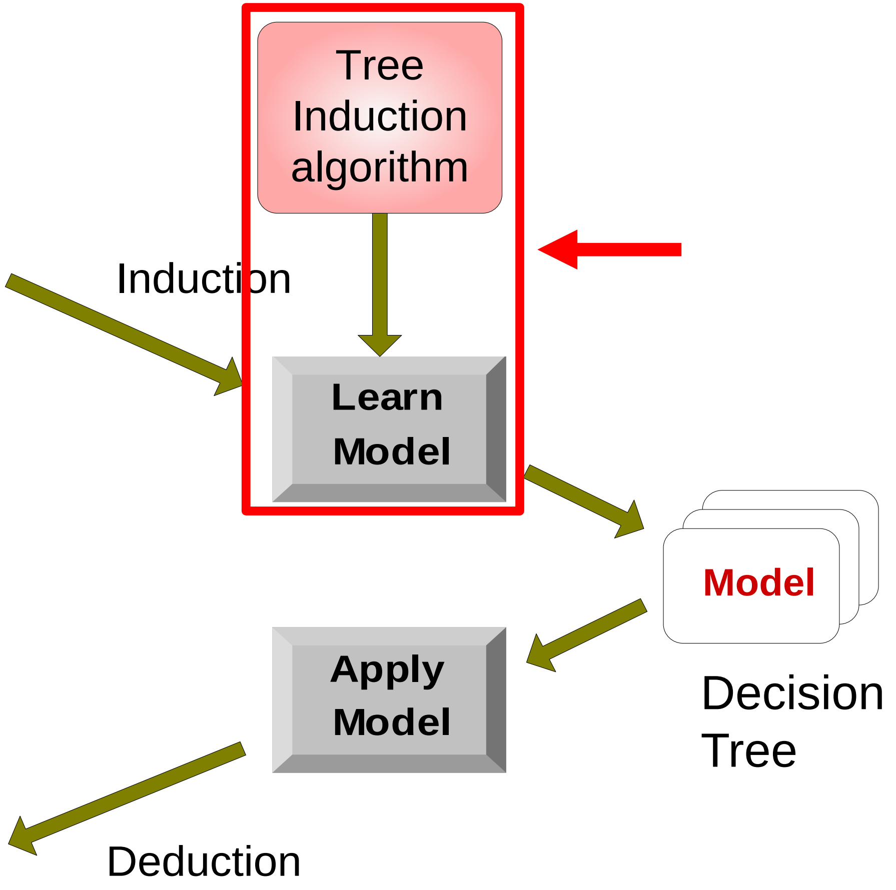
Decision Tree Classification Task

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



Tree Induction

- Goal: Find the tree that has low classification error in the training data (training error)

- Finding the best decision tree (lowest training error) is NP-hard

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

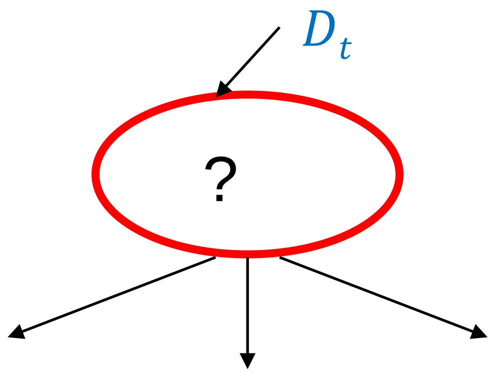
- Many Algorithms:
  - Hunt's Algorithm (one of the earliest)
  - CART
  - ID3, C4.5

《导论》P70

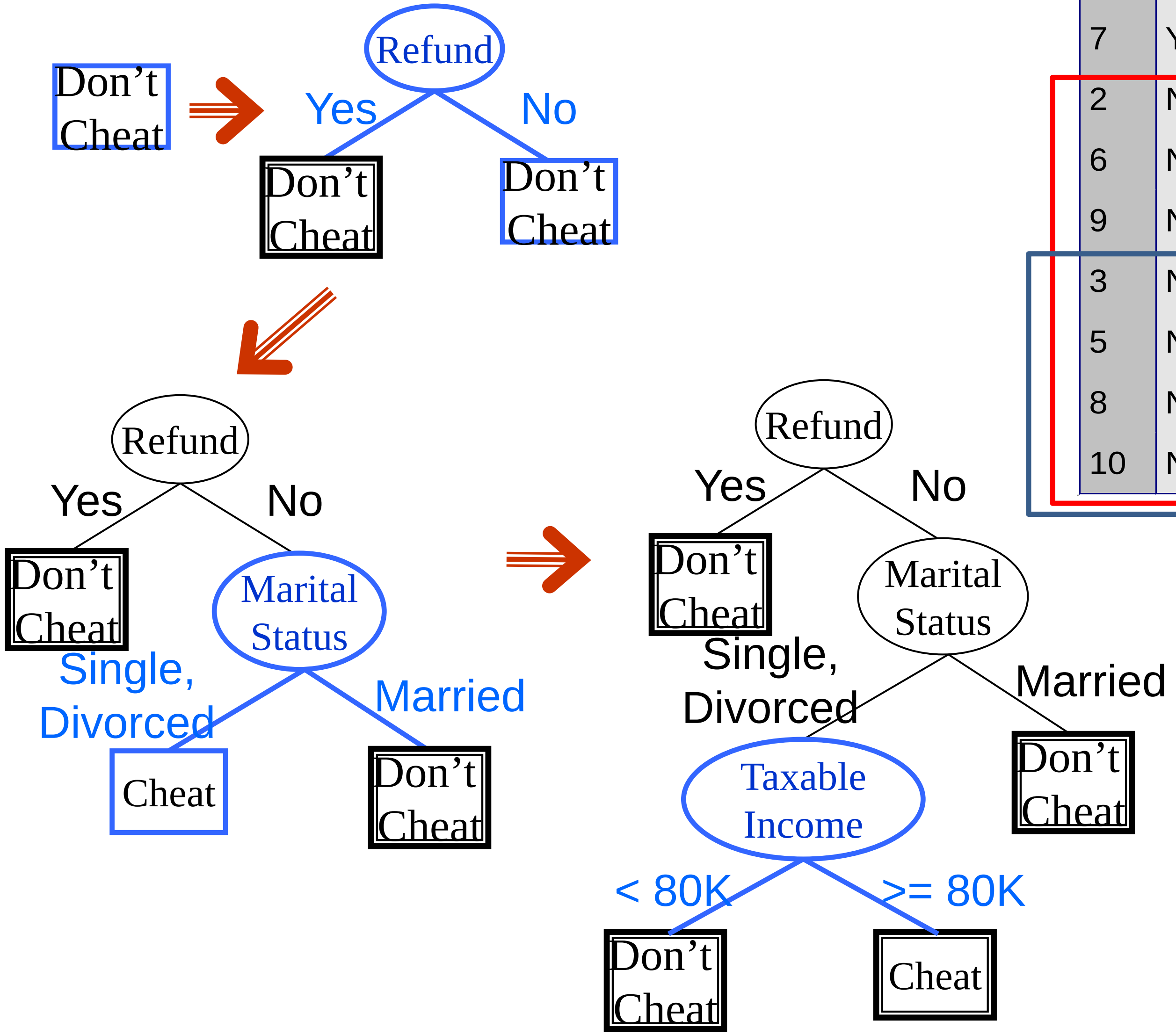
General Structure of Hunt's Algorithm

- D_t : the set of training records that reach a node t
- General Procedure:
  - If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
  - If D_t contains records with the same attribute values, then t is a leaf node labeled with the majority class y_t
  - If D_t is an empty set, then t is a leaf node labeled by the default class, y_d
  - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets.
- Recursively apply the procedure to each subset.

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



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

Constructing decision-trees (pseudocode)

GenDecTree(Sample **S**, Features **F**)

1. **If stopping_condition(S,F)** = true **then**
   a. **leaf = createNode()**
   b. **leaf.label= Classify(S)**
   c. **return leaf**

满足终止条件，创建并返回叶节点

2. **root = createNode()**
3. **root.test_condition = findBestSplit(S,F)**

创建根节点，找到最佳拆分属性和测试条件

4. **V = {v| v** a possible outcome of **root.test_condition}**
5. **for *each*** value **vϵV**:
   a. **S_v: = {s | root.test_condition(s)** = **v** and **s ϵ S}**;
   b. **child = GenDecTree(S_v ,F)** ;
   c. Add **child** as a descent of **root** and label the edge **(root→child) as v**

根据测试条件进行拆分，对于每个子数据集，递归调用**GenDecTree**，生成一个节点，并作为根节点的子节点

6. **return root**

返回根节点

Tree Induction

- Issues
  - How to Classify a leaf node
    - Assign the majority class
    - If leaf is empty, assign the default class – the class that has the highest popularity (overall or in the parent node).
  - Determine how to split the records
    - How to specify the attribute test condition? 属性测试条件是什么？
    - How to determine the best split? 选哪个属性进行分割
  - Determine when to stop splitting
    - Same labels or same attribute values
    - Early stop to avoid overfitting 《导论》P85

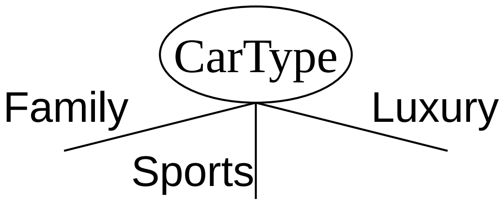
How to Specify Test Condition?

- Depends on attribute types
  - Nominal 标称属性，不带顺序
  - Ordinal 序数属性，带顺序
  - Continuous 连续属性，数值型

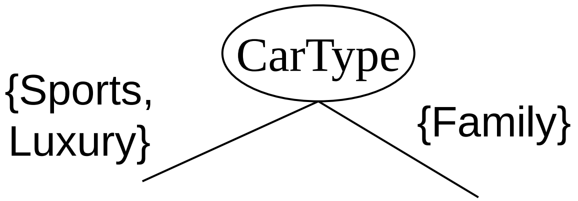
- Depends on number of ways to split
  - 2-way split 二元划分
  - Multi-way split 多路划分

Splitting Based on Nominal Attributes

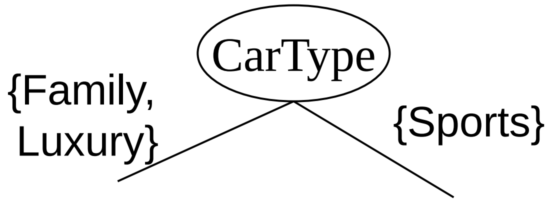
- Multi-way split: Use as many partitions as distinct values.



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

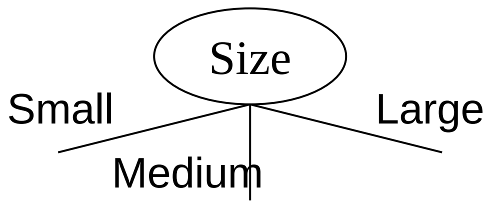


OR

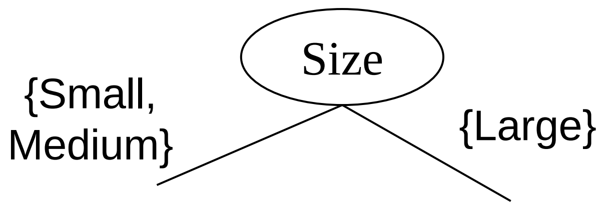


Splitting Based on Ordinal Attributes

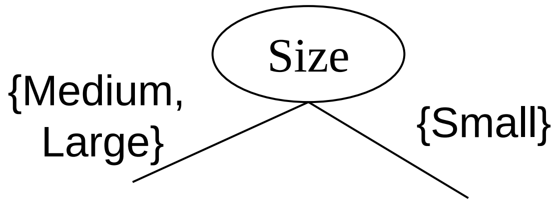
- Multi-way split: Use as many partitions as distinct values.



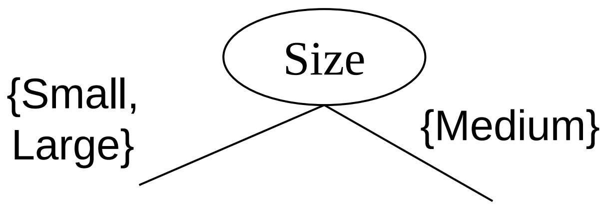
- Binary split: Divides values into two subsets – respects the order. Need to find optimal partitioning.



OR



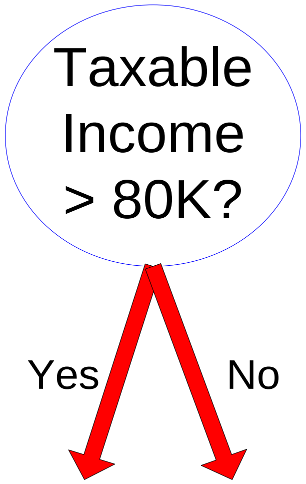
- What about this split?



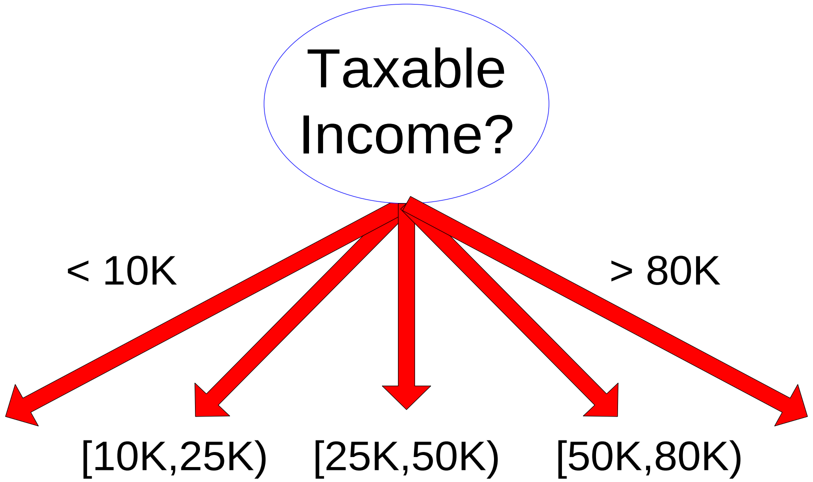
Splitting Based on Continuous Attributes

- 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 computationally intensive

Splitting Based on Continuous Attributes



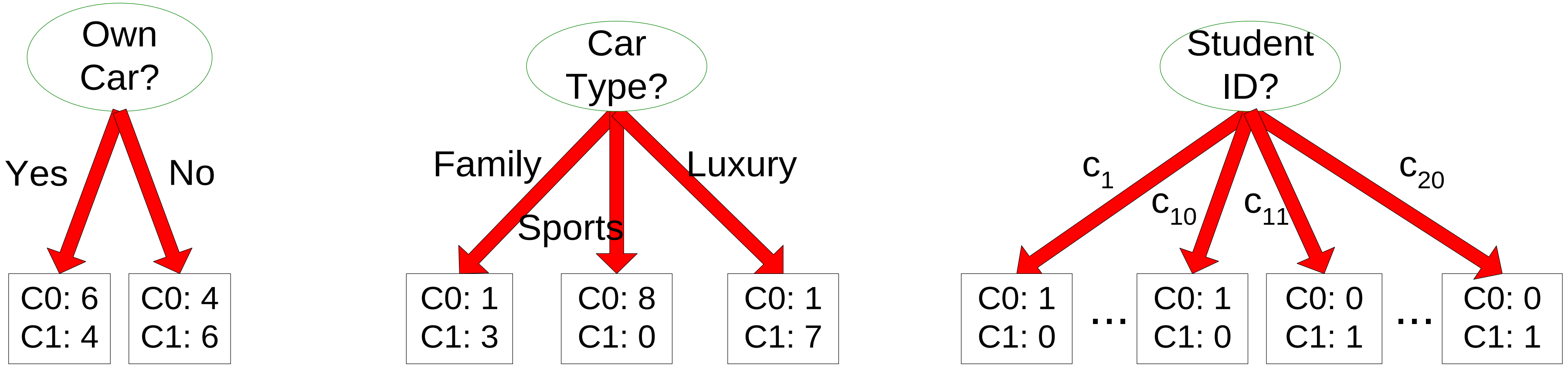
(i) Binary split



(ii) Multi-way split

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

- Greedy approach:
  - Creation of nodes with homogeneous class distribution is preferred
- Need a measure of node impurity: 不纯性

C0: 5 C1: 5

Non-homogeneous,

High degree of impurity

C0: 9 C1: 1

Homogeneous,

Low degree of impurity

Smaller tree, less likely to be overfitted, less training and testing time

Measuring Node Impurity

- We are at a node D_t and the samples belong to classes $\{1, \dots, c\}$
  - $p(i|t)$: fraction of records associated with node D_t belonging to class $\boldsymbol{i}$
- Impurity measures:

$$Entropy(D_t) = -\sum_{i=1}^{c} p(i|t) \log p(i|t)$$

  - Used in ID3 and C4.5

$$Gini(D_t) = 1 - \sum_{i=1}^{c} p(i|t)^2$$

$$Classification\ Error(D_t) = 1 - \max p(i|t)$$

  - Used in CART.

For more, see 《导论》P73