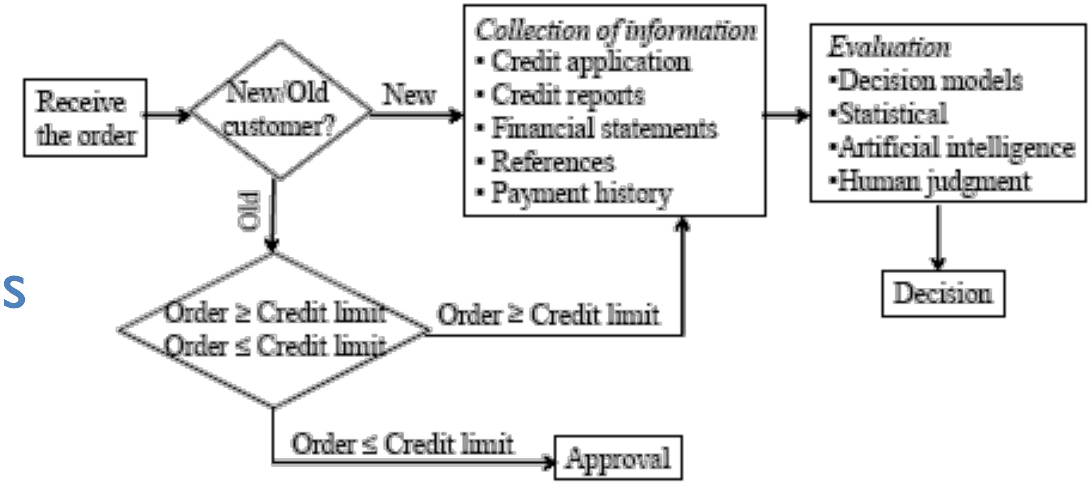
Classification

- What is **classification**?
  - Given is a collection of records (**training set**)
    - Each record consists of a set of attributes, plus a specific **class attribute**
  - Find a model for the class attribute as a **function** of the values of other attributes
  - **Goal:** new records should be assigned to some 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

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification

- How is this useful?
  - Financial field
    - Credit approval process
    - Targeted marketing
    - Classifying credit card transactions as legitimate or fraudulent
  - Categorizing news stories as finance, weather, entertainment, sports, etc.



Classification

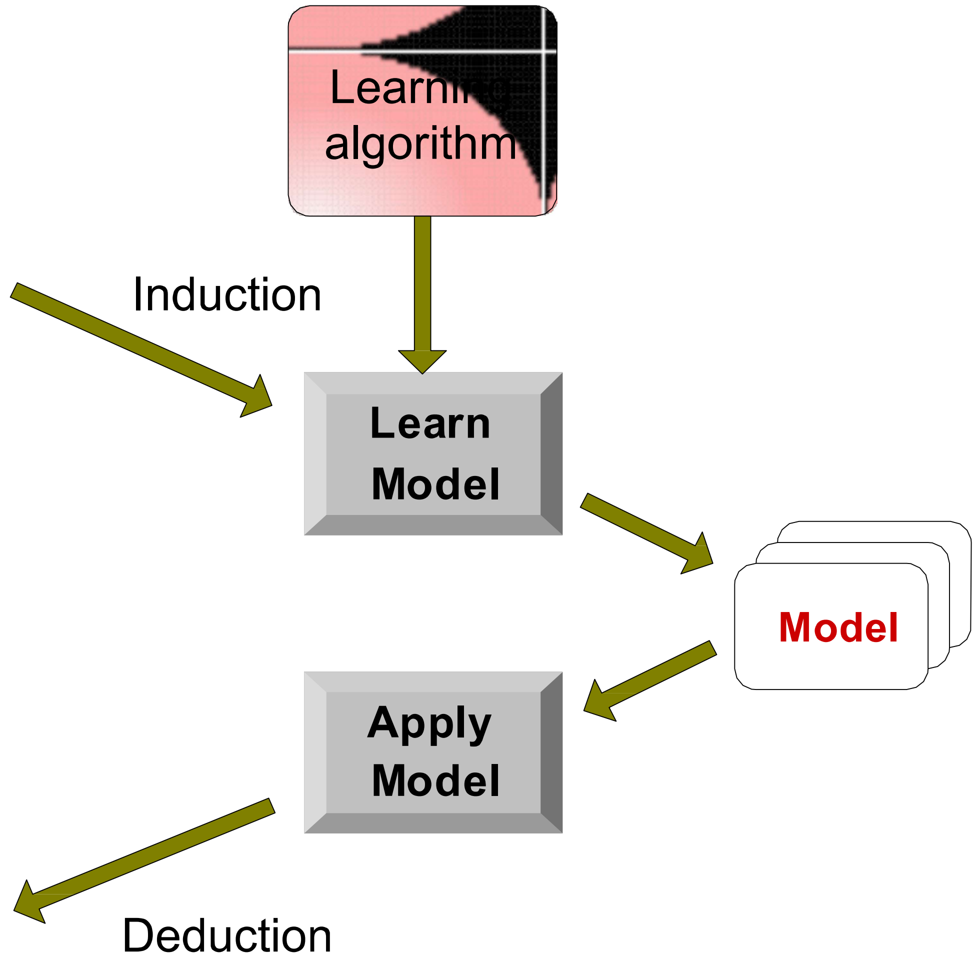
- How does it work?

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



Classification

- Example: credit approval
  - Step 1: **learning** (induction)
    - Training data is analyzed by some classification algorithm and the learned model is coded into **classification rules**

Training data

Name		Income	Decision
Jones	Young	Low	Risky
Lee	Young	Low	Risky
Fox	Middle	High	Safe
Field	Middle	Low	Risky
Smith	Senior	Low	Safe
Phips	Senior	Medium	Safe
…	…	…	…

Classification algorithm

Classification rules

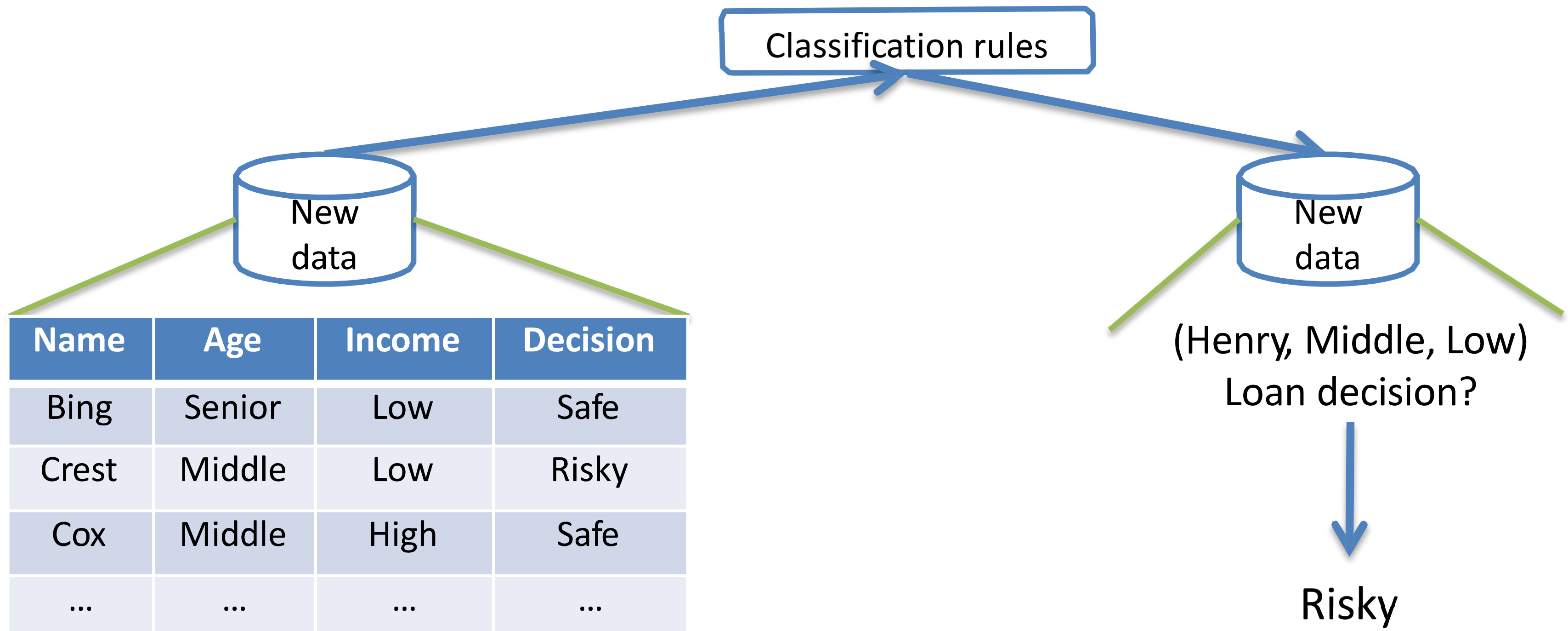
IF age=young THEN decision=risky
IF income=high THEN decision=safe
IF age=middle AND income=low THEN decision=risky
…

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Classification

– Step 2: **classification** (deduction)

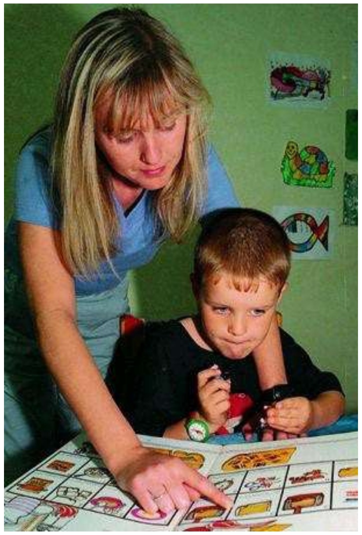
- Test data validates the accuracy of the classification rules
- If the accuracy is considered acceptable, then the rules can be applied to the classification of new records



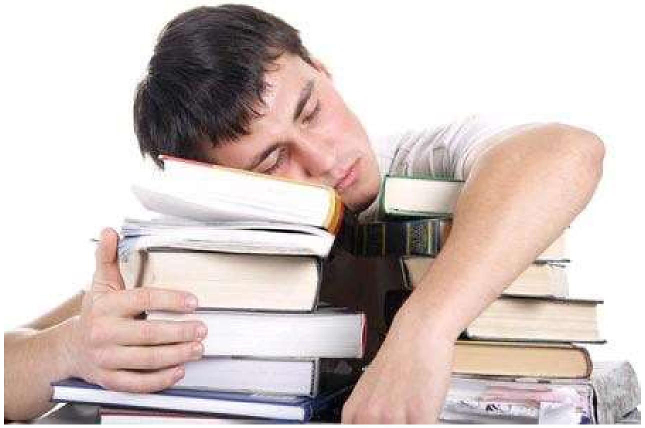
Name	Age	Income	Decision
Bing	Senior	Low	Safe
Crest	Middle	Low	Risky
Cox	Middle	High	Safe
…	…	…	…

Classification

- **Supervised** learning
  - The training data (observations, measurements, etc.) is accompanied by labels indicating the class of the observations
  - New data is classified based on the training set



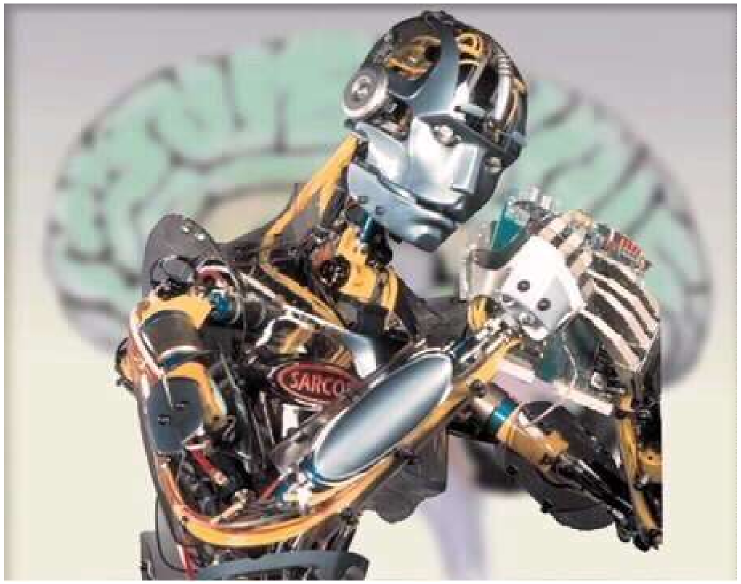
- **Unsupervised** learning (next lecture)
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

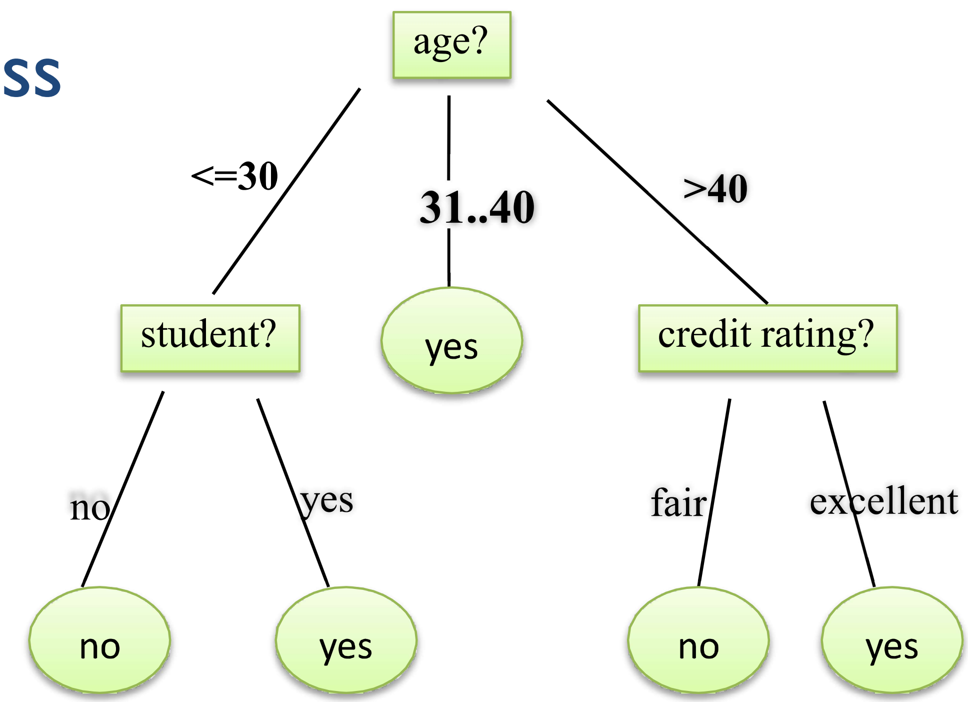
Classification

- Prominent **classification techniques**
  - Decision Tree based Methods
  - Rule-based Methods
  - Naive Bayes and Bayesian Belief Networks
  - Support Vector Machines (SVM)
  - Neural Networks



Decision Tree

- **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
  - E.g., decision making
    - Who buys a computer?



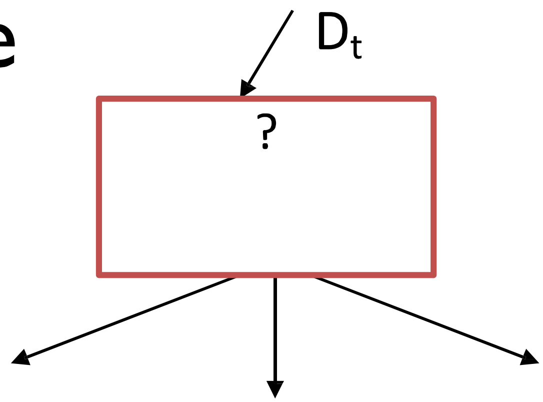
Decision Tree Induction

- Decision tree induction
  - Many Algorithms:
    - Hunt's Algorithm
      - One of the earliest methods
    - ID3 and its successor, C4.5
      - Represents a benchmark to supervised learning algorithms
      - Based on the Hunt algorithm
    - Classification and Regression Trees (CART)
      - Similar to ID3
    - SLIQ, SPRINT

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Decision Tree Induction

- **Hunt's algorithm**, general structure
  - Let D_t be 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 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

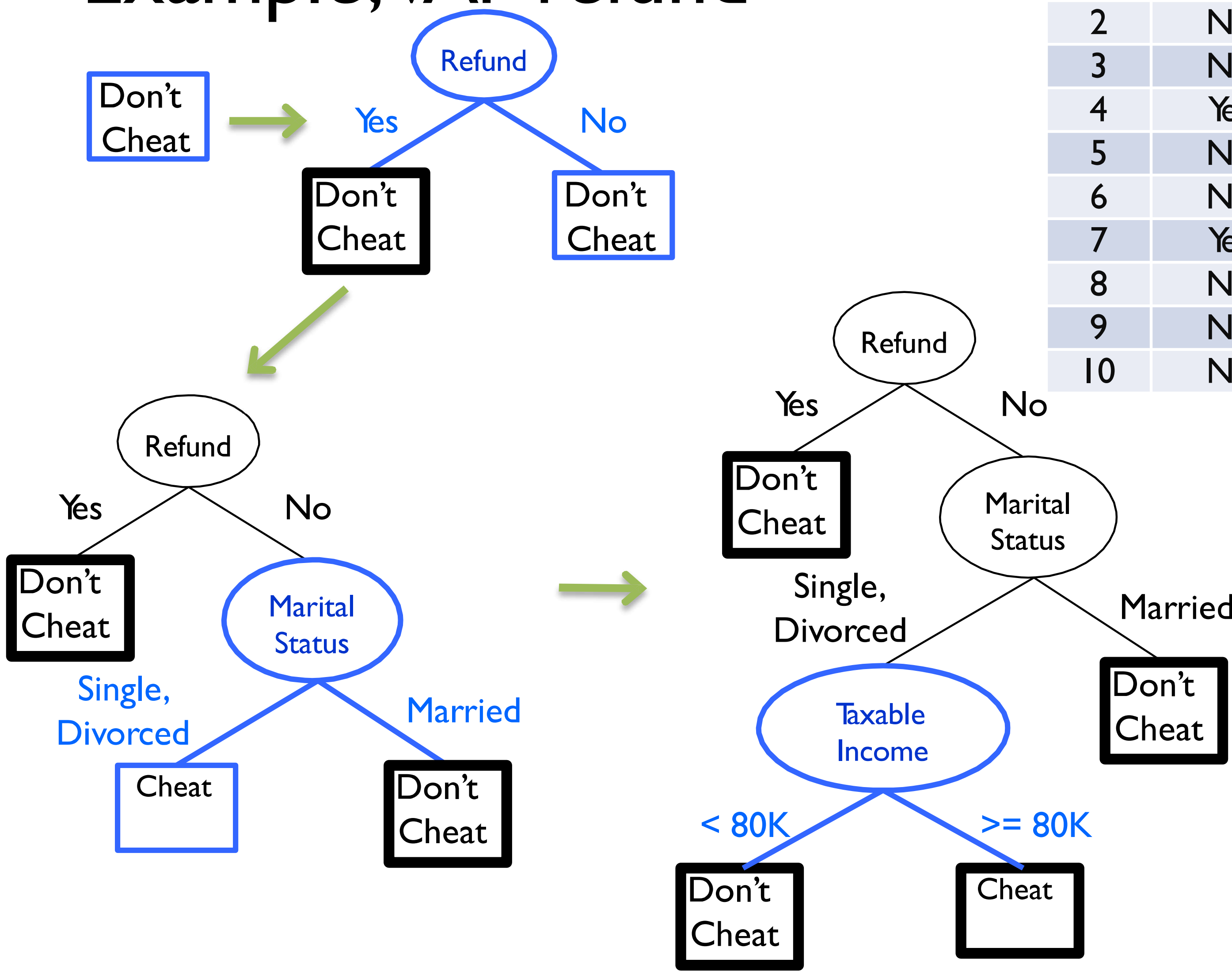


Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Hunts Algorithm

- Example, VAT refund

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



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

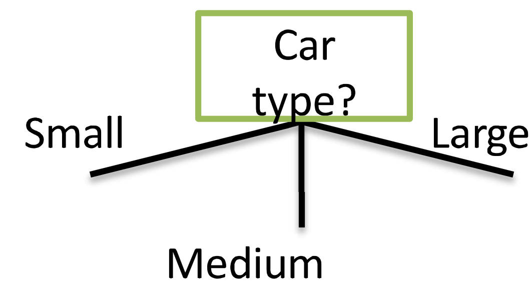
Hunts Algorithm

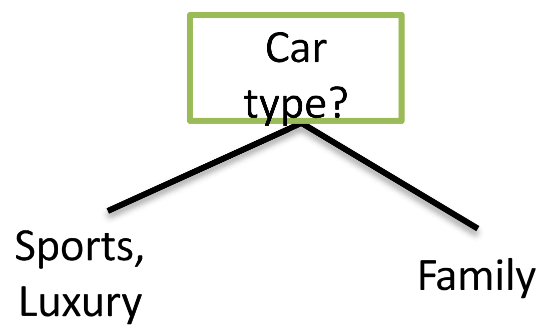
- Greedy strategy
  - **Split** the records based on an **attribute test** that optimizes a certain criterion
- Issues
  - Determine **how to split** the records
    - How to specify the **attribute test condition?**
    - How to **determine the best split?**
  - Determine **when to stop** splitting

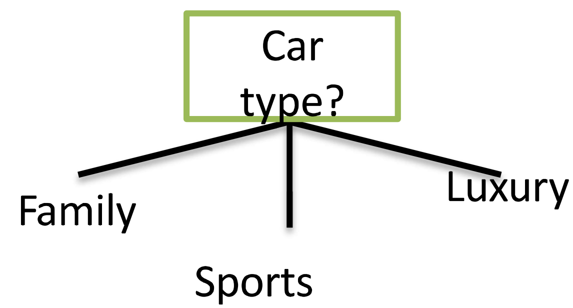
Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Attribute test condition

- **Splitting:** how to specify the attribute test condition?
  - Depends on **attribute types**
    - Nominal e.g., car: sports, luxury, family
    - Ordinal e.g., small, medium large
    - Continuous e.g, age
  - Depends on **number of ways** to split
    - Binary split
    - Multi-way split







Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

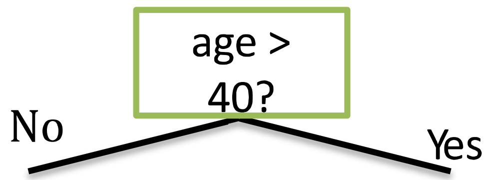
Attribute test condition

- What about splitting **continuous attributes**?
  - **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), clustering, or supervised clustering
  - Binary decision
    - Consider all possible splits and finds the best cut
    - Can be quite computationally expensive

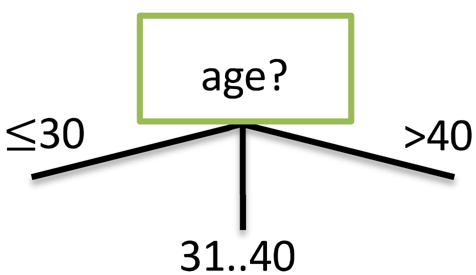
Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Attribute test condition

- Splitting **continuous attributes** (e.g., age)
  - Binary split



  - Multi-way split

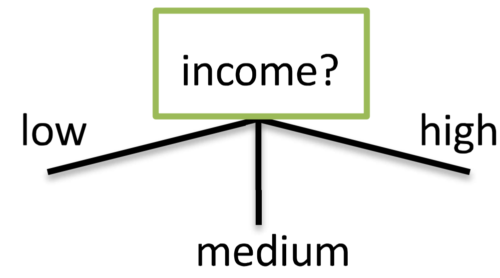


age	income	student	Credit rating	Buys computer
27	high	no	fair	no
28	high	no	excellent	no
31	high	no	fair	yes
45	medium	no	excellent	yes
43	low	yes	excellent	yes
56	low	yes	fair	no
37	low	yes	excellent	yes
20	medium	no	fair	no
20	low	yes	fair	yes
60	medium	yes	excellent	yes
24	medium	yes	excellent	yes
36	medium	no	excellent	yes
31	high	yes	fair	yes
41	medium	no	fair	no

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Determine the best split

- How do we determine the best split?
  - We can split on **any** of the 4 attributes!
  - E.g., income
    - Low – yes:3, no:1
    - Medium – yes:4, no:2
    - High – yes:2, no:2
  - E.g., student
    - No – yes:3, no:4
    - Yes – yes:6, no:1
  - Which split is **better?**



age	income	student	Credit rating	Buys computer
27	high	no	fair	no
28	high	no	excellent	no
31	high	no	fair	yes
45	medium	no	excellent	yes
43	low	yes	excellent	yes
56	low	yes	fair	no
37	low	yes	excellent	yes
20	medium	no	fair	no
20	low	yes	fair	yes
60	medium	yes	excellent	yes
24	medium	yes	excellent	yes
36	medium	no	excellent	yes
31	high	yes	fair	yes
41	medium	no	fair	no

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Determine the best split

- What does better mean?
  - Nodes with **homogeneous** class distribution (pure nodes) are preferred
    - E.g., homogeneous nodes
      - Student attribute, Yes – yes:6, no:1
    - E.g., heterogeneous nodes
      - Income attribute, High – yes:2, no:2
- How do we measure node **impurity**?

Determine the best split

- Methods to measure impurity
  - **Information gain** (used in C4.5)
    - All attributes are assumed to be categorical
    - Can be modified for continuous-valued attributes
    - Also Kullback–Leibler divergence
  - Gini index
    - All attributes are assumed continuous-valued
    - Assume there exist several possible split values for each attribute
    - May need other tools, such as clustering, to get the possible split values

Decision Tree Induction

- **Information gain**
  - Method
    - Assume there are two classes, P and N
      - Let the set of examples S contain p elements of class P and n elements of class N
      - The amount of information, needed to decide whether an arbitrary example in S belongs to P or N is defined as

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

    - Select the attribute with the **highest** information gain

estimates the probability that label is p

estimates the probability that label is n

Information Gain

- Information gain in decision tree induction
  - Assume that using attribute A a set S will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$
  - If S_i contains p_i examples of P and n_i examples of N, the **entropy**, or the expected information needed to classify objects in all subtrees S_i is

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

  - The encoding information that would be gained by branching on A is

$$Gain(A) = I(p, n) - E(A)$$

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Information Gain

- Attribute selection by gain computation, example:
  - Class P: buys_computer = "yes"
  - Class N: buys_computer = "no"

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

  - $I(p, n) = I(9, 5) = 0.94$

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	excellent	yes
>40	low	yes	excellent	yes
>40	low	yes	fair	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	excellent	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	fair	no

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Information Gain

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

– Compute the entropy for the following 3 age partitions age <= 30, 30 < age <= 40 and 40 < age

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

- Gain(age) = I(p, n) – E(age) $\Longrightarrow$ 0.94 – 0.694 = 0.246

– The same we can calculate also the gains for

- Gain(income) = 0.029
- Gain(student) = 0.151
- Gain(credit_rating) = 0.048

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Information Gain

– Since age brings the highest information gain, it becomes the splitting node

– Continue recursively to grow the tree until **stop conditions** are met

age?

youth

income	student	credit	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

senior

income	student	credit	class
medium	no	excellent	yes
low	yes	excellent	yes
low	yes	fair	no
medium	yes	excellent	yes
medium	no	fair	no

middle_aged

income	student	credit	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

Decision Tree Induction

- Stop conditions
  - All the records **belong to the same class**
    - E.g.,

income	credit	class
high	fair	no
high	excellent	no
medium	fair	no

    - In this case a leaf node is created with the corresponding class label (here“no”)

Decision Tree Induction

- Stop conditions
  - All the records **have similar attribute values**
    - E.g., perform split by student but all records are students

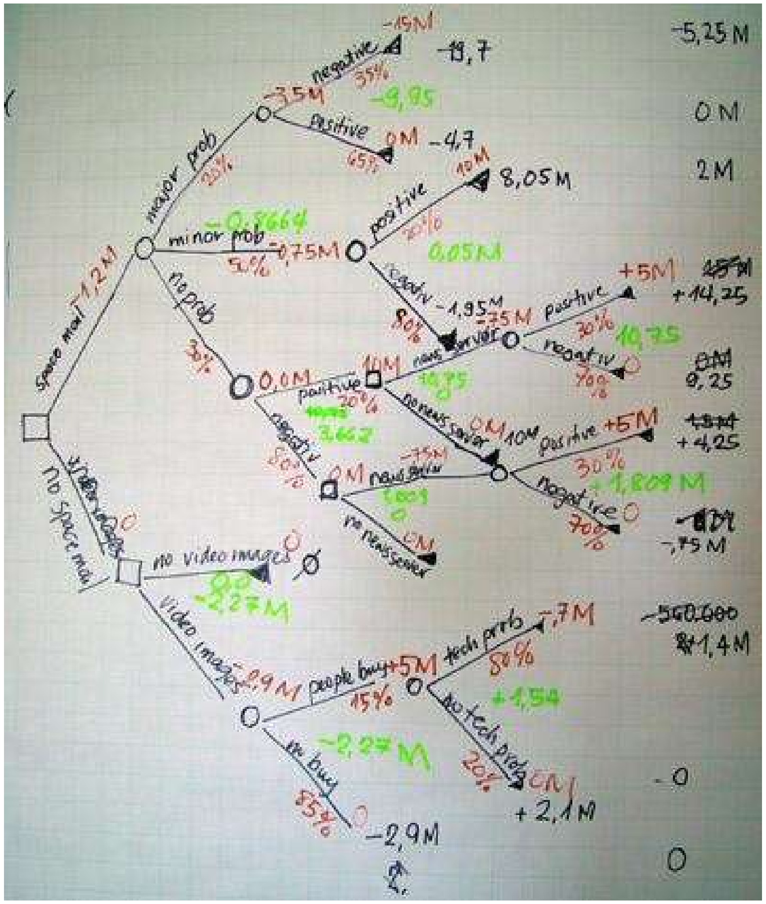
student	credit	class
yes	fair	no
yes	fair	no
yes	fair	no
yes	fair	yes
yes	excellent	yes

    - In this case instead of performing the split, a leaf node is created with the **majority** class as label (here "no")

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Decision Trees

- Decision tree **deduction**
  - Use the decision tree rules to classify new data
  - Exemplified together with induction in the **detour section**



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Summary

- Classification
  - Decision Trees based Classification