



# Chapter 3: Data Pre-Processing

---

## **Based on CSC603.3:**

Students should be able to preprocess the raw data and make it ready for the various data mining tasks

# Topics to be covered

---

- **Data Preprocessing:** Data Cleaning, Data Integration;
- **Data Transformation** : Normalization, Binning, Histogram Analysis and Concept hierarchy generation.
- **Data Reduction:** Attribute subset selection, Histograms, Clustering and Sampling;
- **Data Discretization Concept Description:** Attribute oriented Induction for Data Characterization.

# Why preprocess the data?

- Data in the real world is dirty
  - **incomplete**: lacking *attribute values*, lacking certain *attributes of interest*, or containing only aggregate data
  - **noisy**: containing errors or outliers
  - **inconsistent**: containing discrepancies in codes or names
- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
  - Data warehouse needs consistent integration of quality data
  - Required for both OLAP and Data Mining!

# Data Quality: Why Preprocess the Data?

- Measures for data quality: A multidimensional view
  - Accuracy: correct or wrong, accurate or not
  - Completeness: not recorded, unavailable, ...
  - Consistency: some modified but some not, dangling, ...
  - Timeliness: timely update?
  - Believability: how trustable the data are correct?
  - Interpretability: how easily the data can be understood?

---

Discuss different steps involved in Data  
Preprocessing?

**MAY 2018, 10 MARKS**

# Major Tasks in Data Preprocessing

- **Data cleaning**

- Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies

- **Data integration**

- Integration of multiple databases, data cubes, or files

- **Data transformation and**

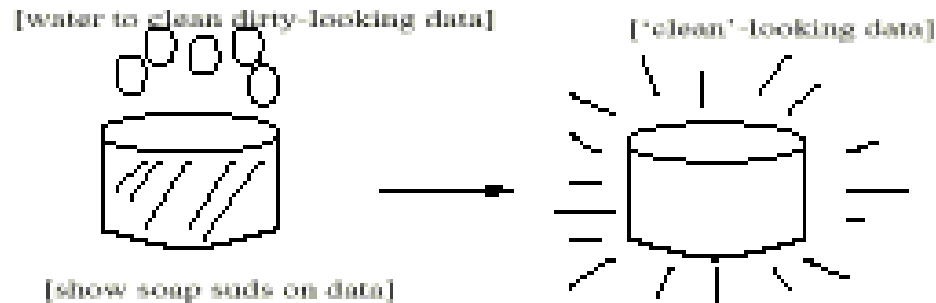
- Smoothing, Aggregation, Generalization, Normalization and attribute construction

- **Data reduction**

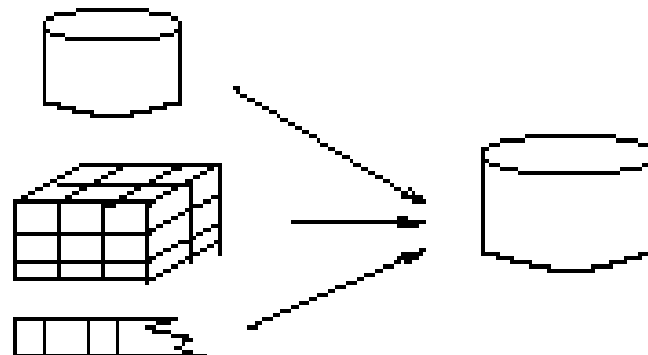
- Data cube aggregation, Attribute Subset selection, Dimensionality reduction, Numerosity reduction, Data discretization & concept hierarchy generation

# Tasks of data preprocessing

## Data Cleaning



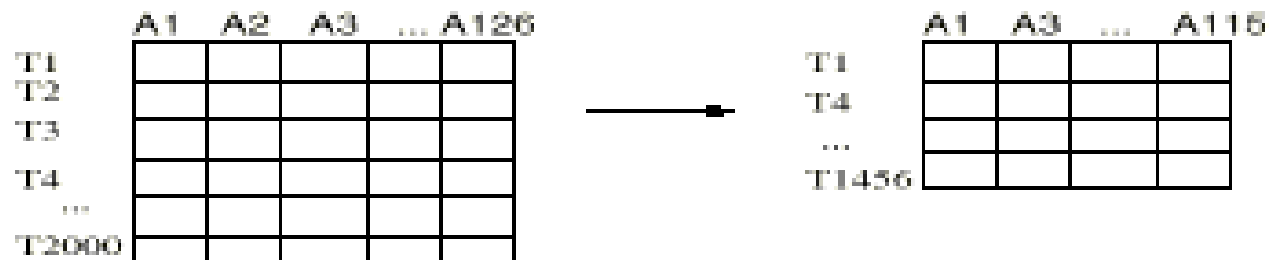
## Data Integration



## Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

## Data Reduction



# Task 1: Data Cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., *Occupation*="" (missing data)
  - noisy: containing noise, errors, or outliers
    - e.g., *Salary*="-10" (an error)
  - inconsistent: containing discrepancies in codes or names, e.g.,
    - *Age*="42", *Birthday*="03/07/2010"
    - Was rating "1, 2, 3", now rating "A, B, C"
    - discrepancy between duplicate records
  - Intentional (e.g., *disguised missing* data)
    - Jan. 1 as everyone's birthday?



# Task 2: Data Integration

- **Data integration:**
  - Combines data from multiple sources into a coherent store
- **Schema integration:** e.g.,  $A.cust-id \equiv B.cust-\#$ 
  - Integrate metadata from different sources
- **Entity identification problem:**
  - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- **Detecting and resolving data value conflicts**
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units

# Task 3: Data Transformation

In data transformation, the data are transformed or consolidated into forms appropriate for mining.

Data transformation can involve the following:

1. **Smoothing**, which works to remove noise from the data. Such techniques include binning, regression, and clustering.
2. **Aggregation**, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.
3. **Generalization of the data**, where low-level or "primitive" (raw) data are replaced by higher-level concepts through the use of concept hierarchies. For example, categorical attributes, like street, can be generalized to higher-level concepts, like city or country. Similarly, values for numerical attributes, like age, may be mapped to higher-level concepts, like youth, middle-aged, and senior.

# Task 3: Data Transformation

4. **Normalization**, where the attribute data are scaled so as to fall within a small specified range, such as 1.0 to 1.0 or 0.0 to 1.0.

There are many methods for data normalization and three of them are :

- **Min-max normalization,**
- **Z-score normalization and**
- **Normalization by decimal scaling.**

5. **Attribute construction (or feature construction)**, where new attributes are constructed and added from the given set of attributes to help the mining process.

# Task 3: Data Transformation- Normalization

**Min-max normalization** performs a linear transformation on the original data.

$\min_A$  and  $\max_A$  are the minimum and maximum values of an attribute,  $A$ .

Min-max normalization maps a value,  $v$ , of  $A$  to  $v'$  in the range  $[\text{new\_min}_A, \text{new\_max}_A]$  by computing

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new\_max}_A - \text{new\_min}_A) + \text{new\_min}_A.$$

Min-max normalization preserves the relationships among the original data values. It will encounter an “out of bounds” error if a future input case for normalization falls outside of the original data range.

**Example:** Suppose that the minimum and maximum values for the attribute *income* are \$12,000 and \$98,000, respectively. By min-max normalization, a value of \$73,600 for *income* is transformed to

$$\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0) + 0 = 0.716.$$

# Task 3: Data Transformation- Normalization

**In z-score normalization** (or zero-mean normalization), the values for an attribute,  $A$ , are normalized based on the mean and standard deviation of  $A$ .

A value,  $v$ , of  $A$  is normalized to  $v'$  by computing 
$$v' = \frac{v - \bar{A}}{\sigma_A}$$

where  $\bar{A}$  and  $\sigma_A$  are the mean and standard deviation, respectively, of attribute  $A$ .

This method of normalization is useful when the actual minimum and maximum of attribute  $A$  are unknown, or when there are outliers that dominate the min-max normalization.

# Task 3: Data Transformation- Normalization

**Normalization by decimal scaling** normalizes by moving the decimal point of values of attribute A. The number of decimal points moved depends on the maximum absolute value of A. A value,  $v$ , of A is normalized to

$v'$  by computing

$$v' = \frac{v}{10^j},$$

where  $j$  is the smallest integer such that  $Max(|v'|) < 1$ .

It is also necessary to save the normalization parameters (such as the mean and standard deviation if using z-score normalization) so that future data can be normalized in a uniform manner.

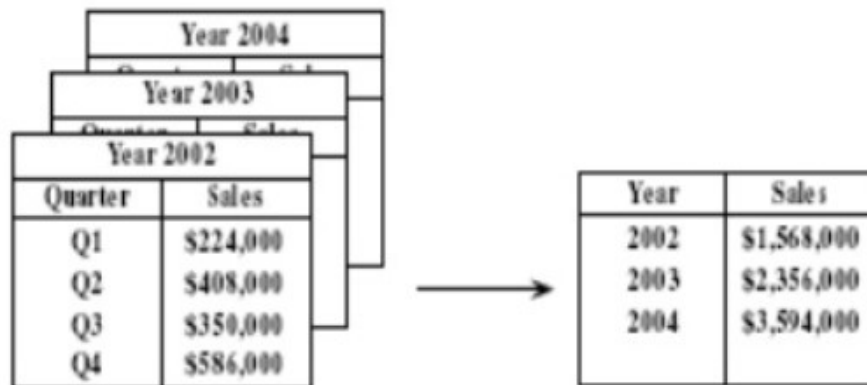
# Task 4: Data Reduction

- Why data reduction?
  - A database/data warehouse may store terabytes of data
  - Complex data analysis/mining may take a very long time to run on the complete data set
- Data reduction
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results
- Strategies for data reduction include the following:
  - Data cube aggregation
  - Attribute subset selection
  - Dimensionality reduction
  - Numerosity reduction
  - Discretization and concept hierarchy generation



# Task 4: Data Reduction- Data cube aggregation

Consider AllElectronics sales per quarter, for the years 2002 to 2004 for analysis. If you are interested in the annual sales (total per year), rather than the total per quarter, the data can be **aggregated** as shown in the below figure.



- Data cubes store multidimensional aggregated information.
- Data cubes are created for varying levels of abstraction.
- Each higher level of abstraction further reduces the resulting data size.
- A cube at the highest level of abstraction is the **apex cuboid**. For the sales data, the apex cuboid would give the total **sales** for all three years, for all item types, and for all branches.

When replying to data mining requests, the **smallest** available cuboids relevant to the given task should be used.



# Task 4: Data Reduction- Attribute subset selection

Attribute subset selection reduces the data set size by removing irrelevant or redundant attributes. Heuristic methods are commonly used for attribute subset selection.

Basic heuristic methods of attribute subset selection include the following techniques:

1. **Stepwise forward selection:**

- The procedure starts with an empty set of attributes.
- At each subsequent iteration or step, the best of the remaining original attributes is added to the set.

2. **Stepwise backward elimination:**

- The procedure starts with the full set of attributes.
- At each step, it removes the worst attribute remaining in the set.

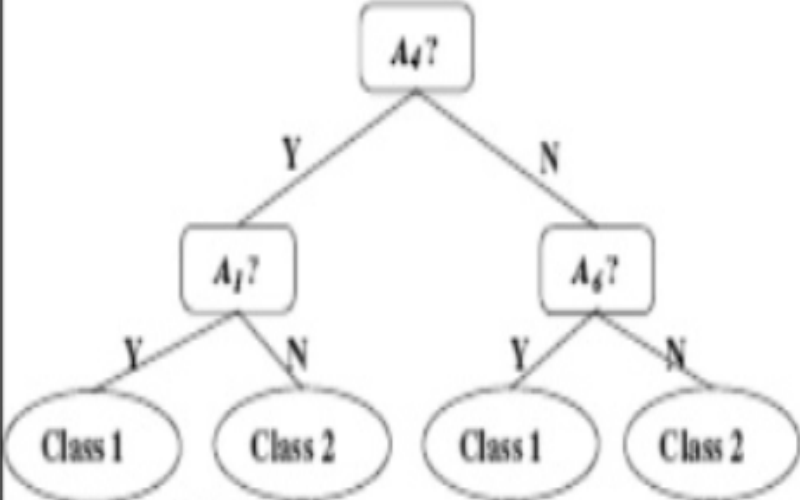
3. **Combination of forward selection and backward elimination:**

- At each step, the procedure selects the best attribute and removes the worst attributes.

4. **Decision tree induction:**

It constructs a flow-chart-like structure where  
Each internal (nonleaf) node denotes a test on an attribute,  
Each branch corresponds to an outcome of the test,  
Each external (leaf) node denotes a class prediction.  
The set of attributes appearing in the tree form the reduced subset of attributes.

# Task 4: Data Reduction- Attribute subset selection

Forward selection	Backward elimination	Decision tree induction
<p>Initial attribute set:  <math>\{A_1, A_2, A_3, A_4, A_5, A_6\}</math></p> <p>Initial reduced set:  <math>\{\}</math>  <math>\Rightarrow \{A_1\}</math>  <math>\Rightarrow \{A_1, A_4\}</math>  <math>\Rightarrow</math> Reduced attribute set:  <math>\{A_1, A_4, A_6\}</math></p>	<p>Initial attribute set:  <math>\{A_1, A_2, A_3, A_4, A_5, A_6\}</math>  <math>\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}</math>  <math>\Rightarrow \{A_1, A_4, A_5, A_6\}</math>  <math>\Rightarrow</math> Reduced attribute set:  <math>\{A_1, A_4, A_6\}</math></p>	<p>Initial attribute set:  <math>\{A_1, A_2, A_3, A_4, A_5, A_6\}</math></p>  <pre> graph TD     A4["A4?"] -- Y --&gt; A1["A1?"]     A4 -- N --&gt; A6["A6?"]     A1 -- Y --&gt; C1_1("Class 1")     A1 -- N --&gt; C2_1("Class 2")     A6 -- Y --&gt; C1_2("Class 1")     A6 -- N --&gt; C2_2("Class 2")     </pre> <p><math>\Rightarrow</math> Reduced attribute set:  <math>\{A_1, A_4, A_6\}</math></p>

# Task 4: Data Reduction- Dimensionality Reduction

---

Data encoding or transformations are applied for data reduction and compression.  
Data reduction is

- **Lossless data reduction:** If the original data can be reconstructed from the compressed data without any loss of information.
- **Lossy data reduction:** If we can reconstruct only an approximation of the original data.

There are two popular and effective methods of lossy reduction:

- Wavelet transforms and
- Principal components analysis.

# Task 4: Data Reduction- Numerosity Reduction

Numerosity reduction reduces the data volume by choosing 'smaller' forms of data representation.

These techniques can be

- Parametric

In parametric methods, a model is used to estimate the data, so that only the data parameters need be stored, instead of the actual data.

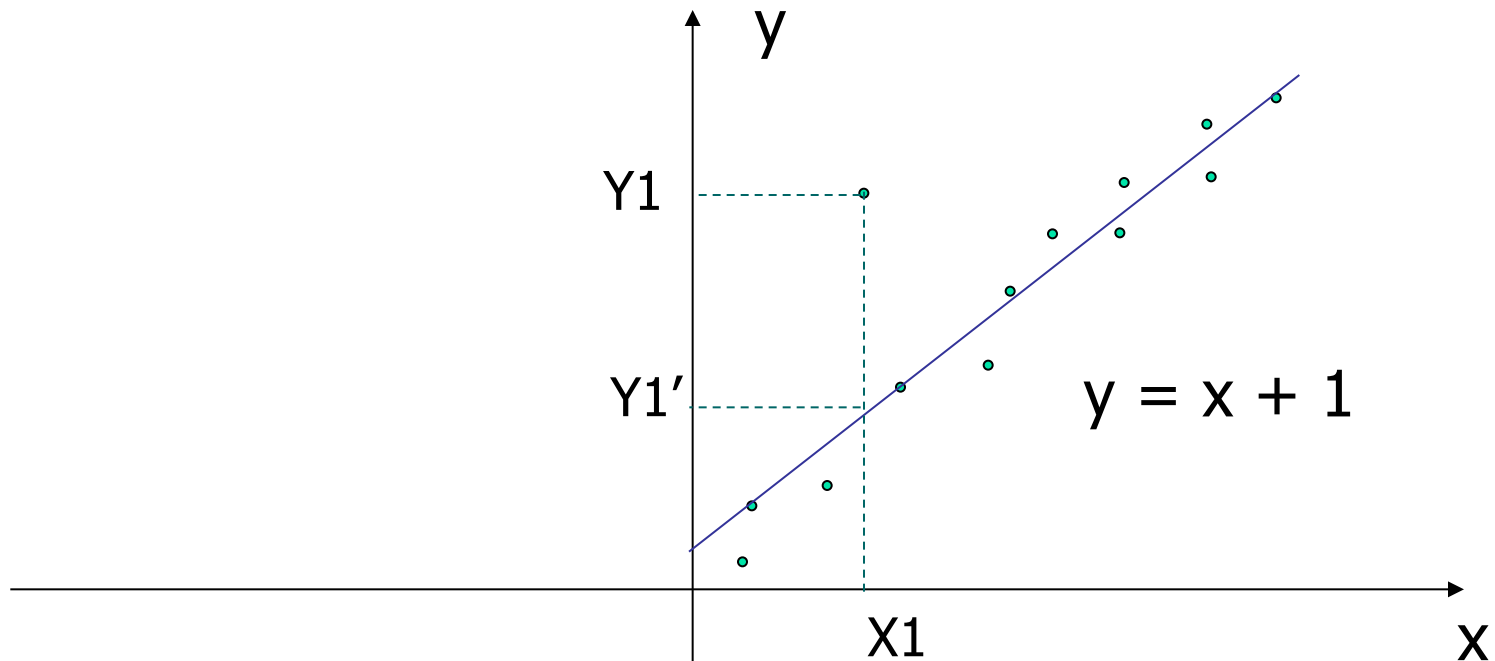
*Ex: Regression and Log-linear models*

- Non-parametric

Nonparametric methods for storing reduced representations of the data include *histograms, clustering, and sampling*.

# Task 4: Data Reduction- Numerosity

## Reduction: Regression



- Linear regression (best line to fit two variables)
- Multiple linear regression (more than two variables, fit to a multidimensional surface)

# Task 4: Data Reduction- Numerosity

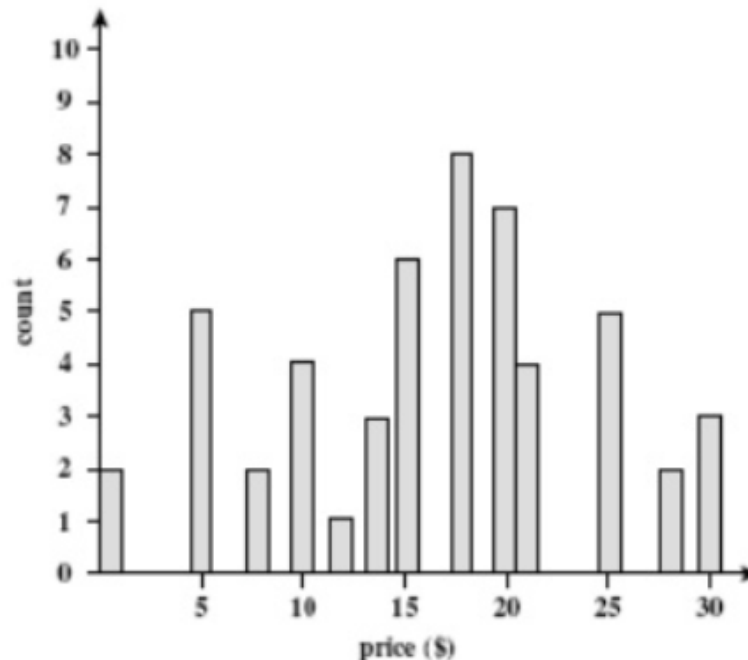
## Reduction: Histogram

### Histograms

A histogram for an attribute,  $A$ , partitions the data distribution of  $A$  into disjoint subsets, or buckets.

**Example:** The following data are a list of prices of commonly sold items at AllElectronics.

The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.

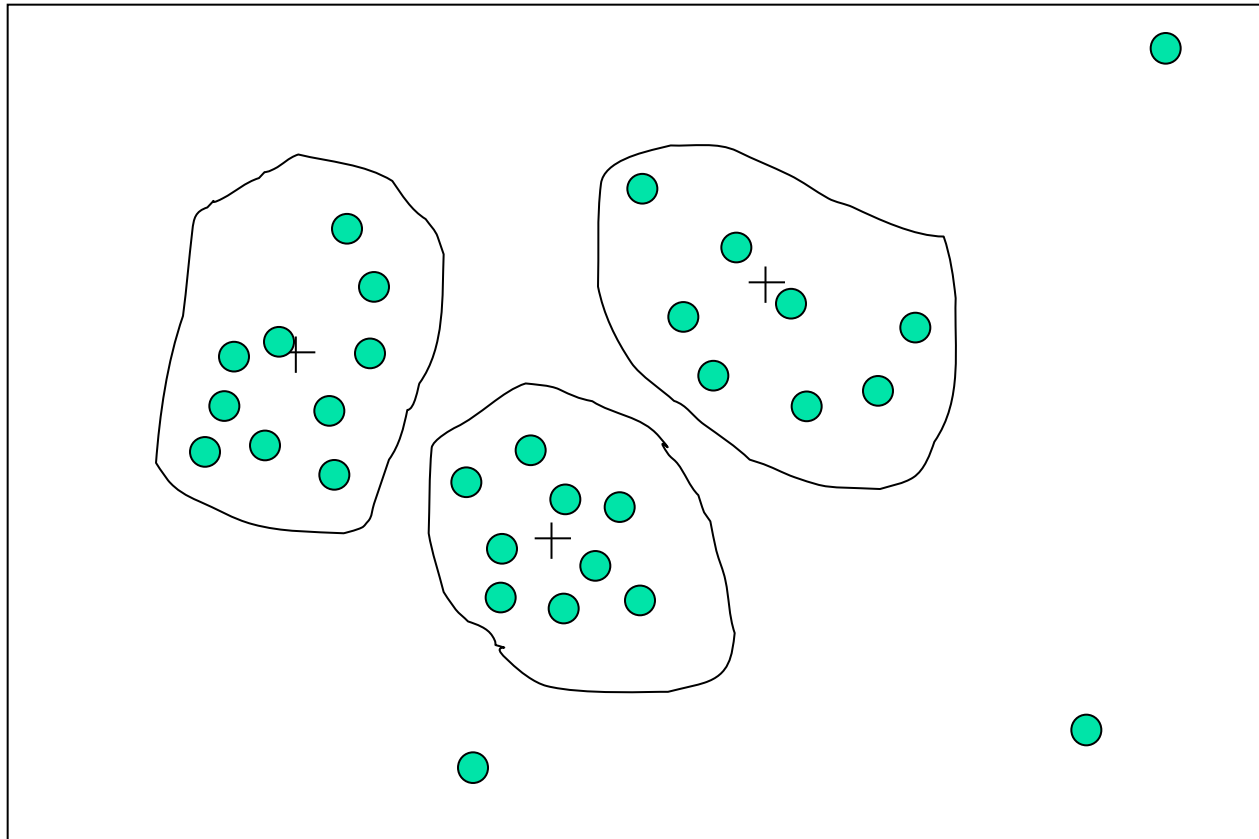


A histogram for *price* using singleton buckets—each bucket represents one price-value/frequency pair.

# Task 4: Data Reduction- Numerosity

## Reduction: Cluster Analysis

Partition data set into clusters based on similarity and store cluster representation only



# Task 4: Data Reduction- Discretization & Concept hierarchy generation

## Data discretization techniques

- Divide the range of the attribute into intervals.
- Interval labels can then be used to replace actual data values.

Based on how the discretization is performed data discretization techniques are divided into

- Supervised discretization - uses class information
- Unsupervised discretization – based on which direction it proceeds
  - ▮ **Top-down or Splitting** - Splits entire attribute range by one or a few points.
  - ▮ **Bottom-up or Merging** - Merges neighborhood values to form intervals.
- Concept hierarchies can be used to reduce the data by collecting and replacing low-level concepts (such as numerical values for the attribute **age**) by higher-level concepts (such as youth, middle-aged, or senior)..
- Mining on a reduced data set requires fewer input/output operations and is more efficient than mining on a larger, ungeneralized data set.



# Discretization & Concept hierarchy generation for Numerical data

- Typical methods: All the methods can be applied recursively
  - Binning
    - Top-down split, unsupervised
  - Histogram analysis
    - Top-down split, unsupervised
  - Clustering analysis (unsupervised, top-down split or bottom-up merge)
  - Decision-tree analysis (supervised, top-down split)
  - Correlation (e.g.,  $\chi^2$ ) analysis (unsupervised, bottom-up merge)

# Simple Discretization Methods: Binning

---

- **Equal-width (distance) partitioning:**

- It divides the range into  $N$  intervals of equal size: uniform grid
- if  $A$  and  $B$  are the lowest and highest values of the attribute, the width of intervals will be:  $W = (B - A) / N$ .
- The most straightforward
- But outliers may dominate presentation
- Skewed data is not handled well.

- **Equal-depth (frequency) partitioning:**

- It divides the range into  $N$  intervals, each containing approximately same number of samples
- Good data scaling
- Managing categorical attributes can be tricky.

# Binning Methods for Data Smoothing

---

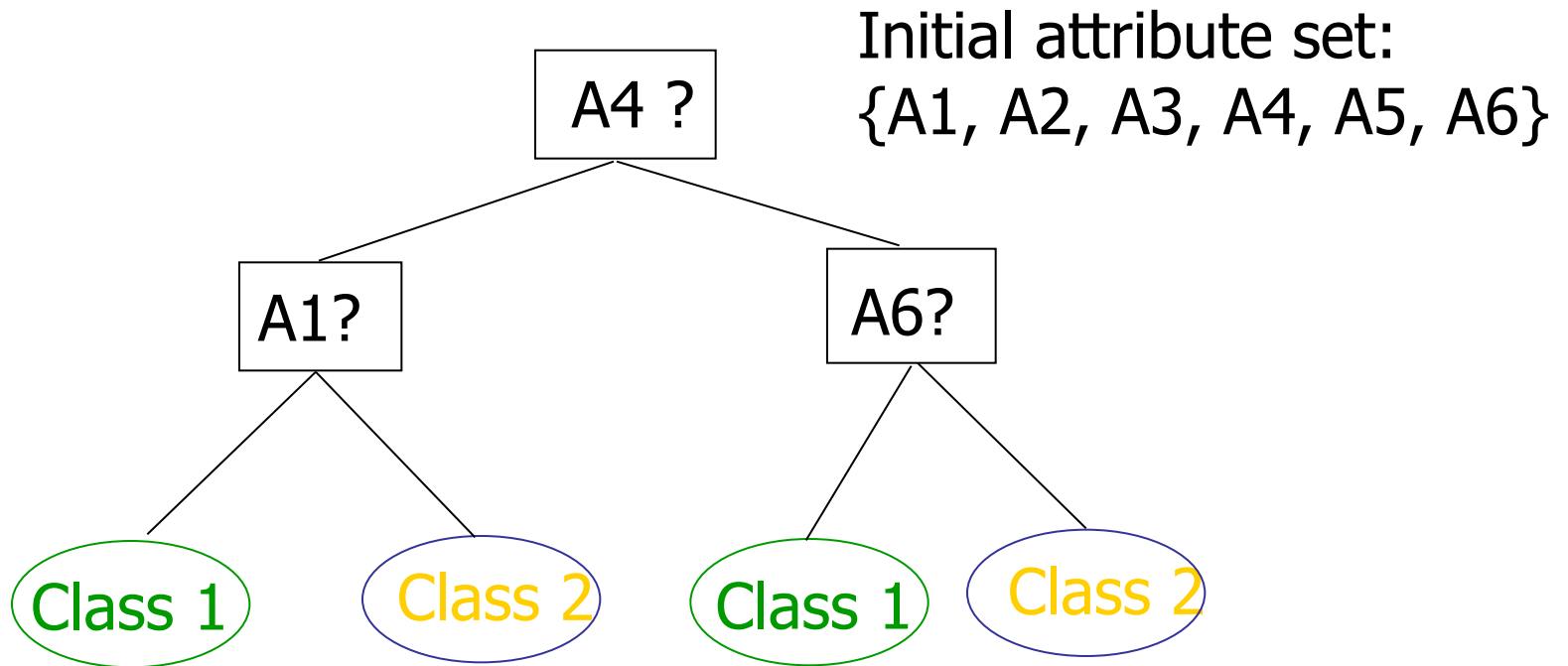
- \* Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
- \* Partition into (equi-depth) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
- \* Smoothing by bin means:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
- \* Smoothing by bin boundaries:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

# Example of Decision Tree Induction

Non-leaf nodes: tests

branches: outcomes of tests

leaf nodes: class prediction



-----> Reduced attribute set:  $\{A1, A4, A6\}$

---

**THANK YOU**