



# **DSC478: Programming Machine Learning Applications**

**Roselyne Tchoua**

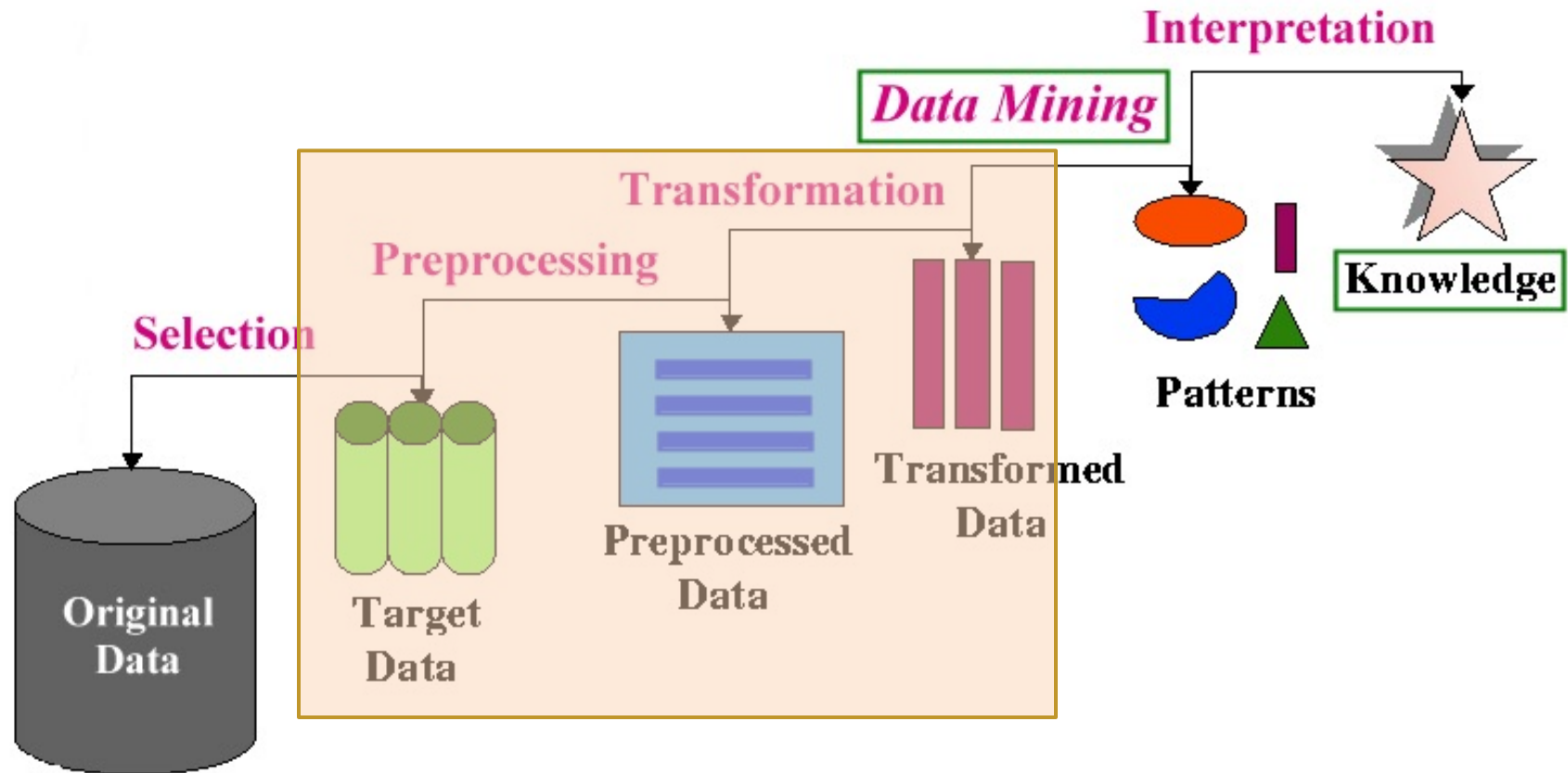
**[rtchoua@depaul.edu](mailto:rtchoua@depaul.edu)**

**School of Computing, CDM, DePaul University**

# Understanding Your Data

- Before doing any machine learning, you should have an idea of what your data looks like
  - What type of data is it?
  - What are the features? What types are they? Do they need to be converted into different types? e.g., categorical to numeric
  - Are some features correlated?
  - Is the data skewed?
  - Are there outliers?
  - If you have class labels, is it **imbalanced**? e.g., a few transactions labeled as fraud in a sea of “normal” transactions
  - Part of visualization is about whether you can easily digest and convey the characteristics of your data

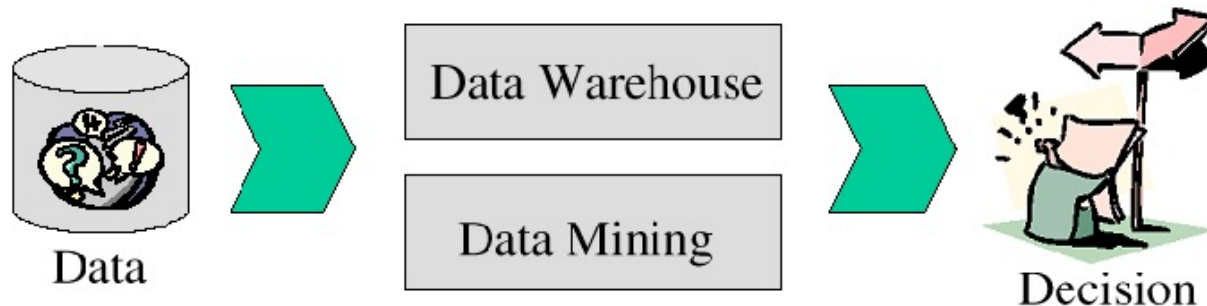
# The Knowledge Discovery Process



*- The KDD Process*

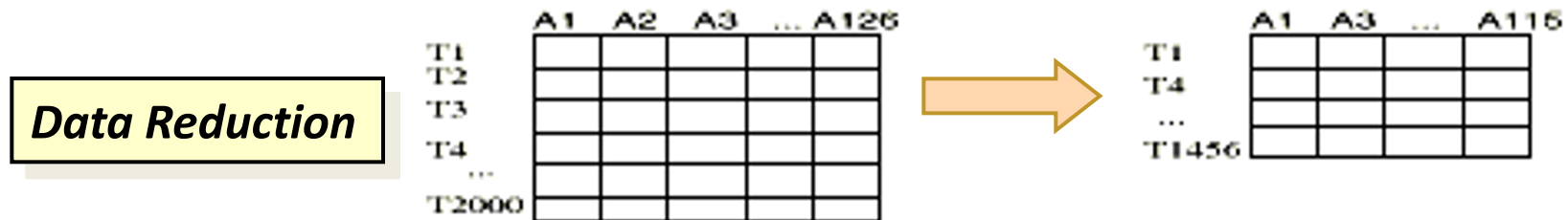
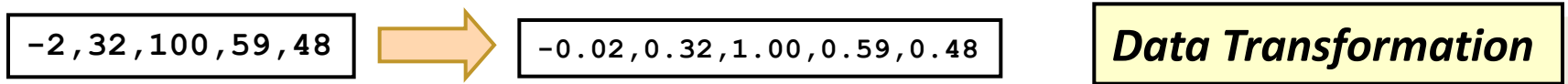
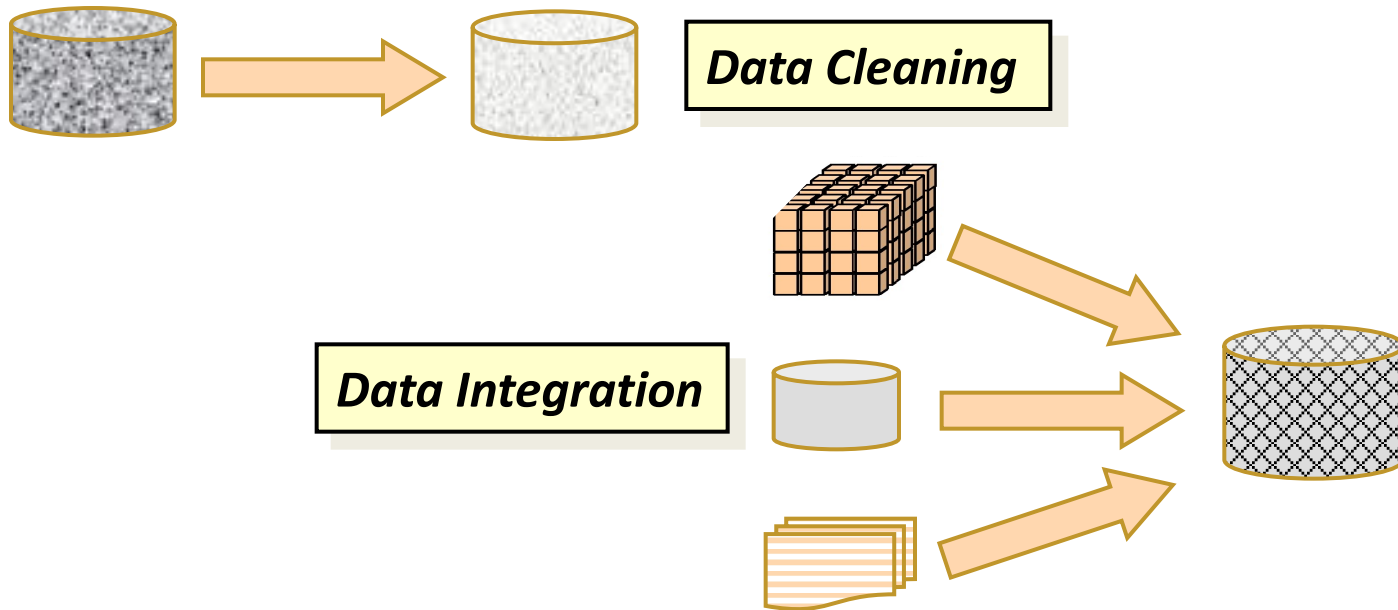
# Data Preprocessing

- Why do we need to prepare the data?
  - In real world applications data can be **inconsistent**, **incomplete** and/or **noisy**
    - Data entry, data transmission, or data collection problems
    - Discrepancy in naming conventions
    - Duplicated records
    - Incomplete or missing data
    - Contradictions in data
- What happens when the data can not be trusted?
  - Can the decision be trusted? Decision making is jeopardized



- Better chance to discover useful knowledge when data is clean

# Data Preprocessing



# Data Cleaning

- Real-world application data can be incomplete, noisy, and inconsistent
  - No recorded values for some attributes
  - Not considered at time of entry
  - Random errors
  - Irrelevant records or fields
- Data cleaning attempts to:
  - Fill in missing values
  - Smooth out noisy data
  - Correct inconsistencies
  - Remove irrelevant data



# Dealing with Missing Values

- Solving the Missing Data Problem
  - **Ignore** the record with missing values; (Can you afford this?)
  - Fill in the missing values manually; (Error-prone, Unsustainable)
  - Use a global constant to fill in missing values (**NULL**, **unknown**, etc.);
  - Use the attribute **value mean** to filling missing values of that attribute;
  - Use the attribute **mean for all samples belonging to the same class** to fill in the missing values;
  - **Infer** the most probable value to fill in the missing value
    - may need to use methods such as Bayesian classification (probabilities) to automatically infer missing attribute values

# Smoothing Noisy Data

The purpose of data smoothing is to eliminate noise and “smooth out” the data fluctuations.

Ex: Original Data for “price” (after sorting): 4, 8, 15, 21, 21, 24, 25, 28, 34

Binning →

**Partition into equi-depth bins**

Bin1: 4, 8, 15  
Bin2: 21, 21, 24  
Bin3: 25, 28, 34

**Each value in a bin is replaced by the mean value of the bin.**

**means**

Bin1: 9, 9, 9  
Bin2: 22, 22, 22  
Bin3: 29, 29, 29

**boundaries**

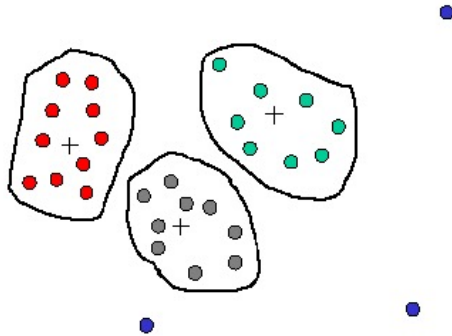
Bin1: 4, 4, 15  
Bin2: 21, 21, 24  
Bin3: 25, 25, 34

**Min and Max values in each bin are identified (boundaries). Each value in a bin is replaced with the closest boundary value.**



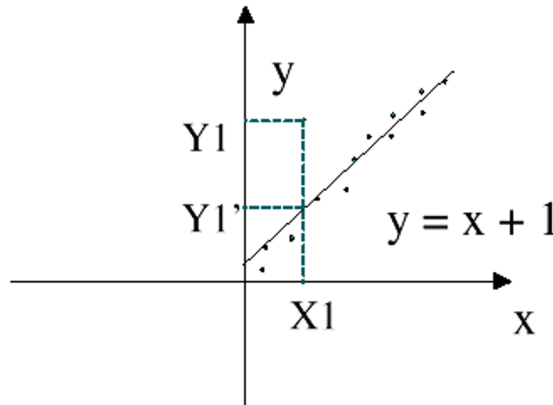
# Smoothing Noisy Data

Clustering



Similar values are organized into groups (clusters). Attributes values may be replaced by “representatives”. Values falling outside of clusters may be considered “outliers” and may be candidates for elimination.

Regression



Fit data to a function. Linear regression finds the best line to fit two variables. Multiple regression can handle multiple variables. The values given by the function are used instead of the original values.

# Smoothing Noisy Data - Example

Want to smooth "Temperature" by bin means with bins of size 3:

1. First sort the values of the attribute (keep track of the ID or key so that the transformed values can be replaced in the original table).
2. Divide the data into bins of size 3 (or less in case of last bin).
3. Convert the values in each bin to the mean value for that bin
4. Put the resulting values into the original table


ID	Outlook	Temperature	Humidity	Windy
1	sunny	85	85	FALSE
2	sunny	80	90	TRUE
3	overcast	83	78	FALSE
4	rain	70	96	FALSE
5	rain	68	80	FALSE
6	rain	65	70	TRUE
7	overcast	58	65	TRUE
8	sunny	72	95	FALSE
9	sunny	69	70	FALSE
10	rain	71	80	FALSE
11	sunny	75	70	TRUE
12	overcast	73	90	TRUE
13	overcast	81	75	FALSE
14	rain	75	80	TRUE



ID	Temperature	
7	58	Bin1
6	65	
5	68	
9	69	Bin2
4	70	
10	71	
8	72	Bin3
12	73	
11	75	
14	75	Bin4
2	80	
13	81	
3	83	Bin5
1	85	

# Smoothing Noisy Data - Example

ID	Temperature	
7	58	Bin1
6	65	
5	68	
9	69	Bin2
4	70	
10	71	
8	72	Bin3
12	73	
11	75	
14	75	Bin4
2	80	
13	81	
3	83	Bin5
1	85	



ID	Temperature	
7	64	Bin1
6	64	
5	64	
9	70	Bin2
4	70	
10	70	
8	73	Bin3
12	73	
11	73	
14	79	Bin4
2	79	
13	79	
3	84	Bin5
1	84	

Value of every record in each bin is changed to the mean value for that bin. If it is necessary to keep the value as an integer, then the mean values are rounded to the nearest integer.

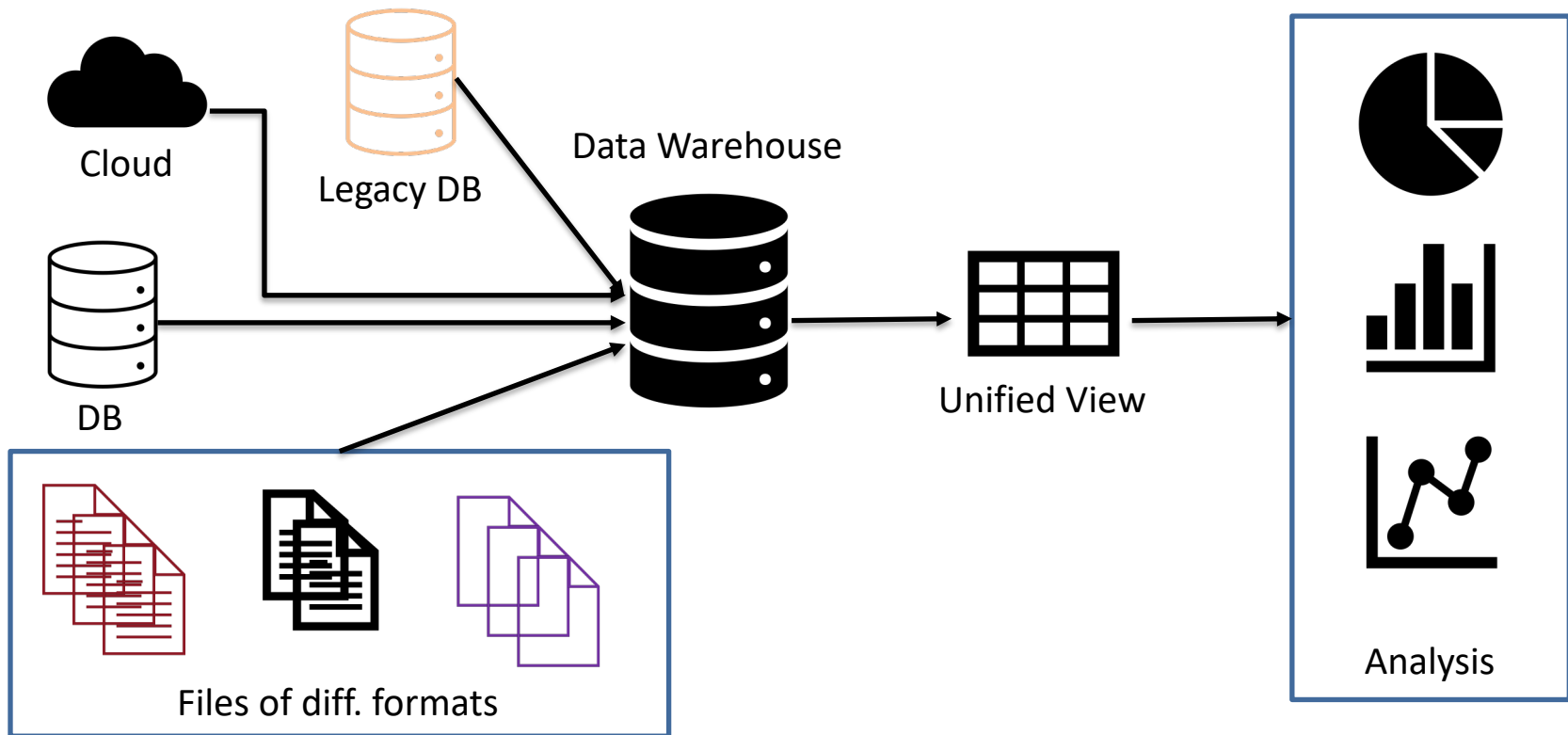
# Smoothing Noisy Data - Example

The final table with the new values for the Temperature attribute.

ID	Outlook	Temperature	Humidity	Windy
1	sunny	84	85	FALSE
2	sunny	79	90	TRUE
3	overcast	84	78	FALSE
4	rain	70	96	FALSE
5	rain	64	80	FALSE
6	rain	64	70	TRUE
7	overcast	64	65	TRUE
8	sunny	73	95	FALSE
9	sunny	70	70	FALSE
10	rain	70	80	FALSE
11	sunny	73	70	TRUE
12	overcast	73	90	TRUE
13	overcast	79	75	FALSE
14	rain	79	80	TRUE

# Data Integration

Ideal case → Access to data warehouse, in which data integration has already **combined data from multiple sources into a coherent store**



# Data Integration

- Reality and Research:
  - Data Lakes: A data lake is a centralized repository that allows you to store all your structured and unstructured data at any scale. You can store your data as-is, without having to first structure the data, and run different types of analytics.
    - Low-cost storage and maintenance
    - Scalable
    - Store structured and non-structured data
    - Usually have some organization (metadata), some data integration tools
  - Data swamps: no organization, no system, no curation, no or broken metadata/context data
    - Here, integration is difficult and requires advanced solutions.
    - Meta-data is often necessary for successful data integration

# Data Integration

- Data analysis may require a combination of data from multiple sources into a coherent data store
- Challenges in Data Integration:
  - Schema integration: CID = C\_number = Cust-id = cust#
  - Semantic heterogeneity (diagnosis, medical condition in diff. ontologies)
  - Data value conflicts (different representations or scales, etc.)
  - Synchronization (especially important in Web usage mining)
  - Redundant attributes (redundant if it can be derived from other attributes) -- may be able to identify redundancies via correlation analysis:

$$\Pr(A,B) / (\Pr(A).\Pr(B))$$

= 1: independent,

> 1: positive correlation,

< 1: negative correlation.

# Data Transformation: Normalization

- Min-max normalization: linear transformation from  $v$  to  $v'$ 
  - $v' = [(v - \text{min}) / (\text{max} - \text{min})] \times (\text{newmax} - \text{newmin}) + \text{newmin}$
  - Note that if the new range is  $[0..1]$ , then this simplifies to  $v' = [(v - \text{min}) / (\text{max} - \text{min})]$
  - Ex: transform \$30000 between  $[10000..45000]$  into  $[0..1] \Rightarrow [(30000 - 10000) / 35000] = 0.514$
- z-score normalization: normalization of  $v$  into  $v'$  based on attribute value mean and standard deviation
  - $v' = (v - \text{Mean}) / \text{StandardDeviation}$
- Normalization by decimal scaling
  - moves the decimal point of  $v$  by  $j$  positions such that  $j$  is the minimum number of positions moved so that absolute maximum value falls in  $[0..1]$ .
  - $v' = v / 10^j$
  - Ex: if  $v$  in  $[-56 .. 9976]$  and  $j=4 \Rightarrow v'$  in  $[-0.0056 .. 0.9976]$



# Normalization: Example

- z-score normalization:  $v' = (v - \text{Mean}) / \text{Stdev}$
- Example: normalizing the “Humidity” attribute:

Humidity
85
90
78
96
80
70
65
95
70
80
70
90
75
80



Mean = 80.3  
Stdev = 9.84



Humidity
0.48
0.99
-0.23
1.60
-0.03
-1.05
-1.55
1.49
-1.05
-0.03
-1.05
0.99
-0.54
-0.03

# Normalization: Example II

- Min-Max normalization on an employee database
  - max distance for salary: 100000-19000 = 81000
  - max distance for age: 52-27 = 25
  - New min for age and salary = 0; new max for age and salary = 1

$$x'_i = \frac{x_i - \min x_i}{\max x_i - \min x_i} (\text{new max} - \text{new min}) + \text{new min}$$

ID	Gender	Age	Salary
1	F	27	19,000
2	M	51	64,000
3	M	52	100,000
4	F	33	55,000
5	M	45	45,000

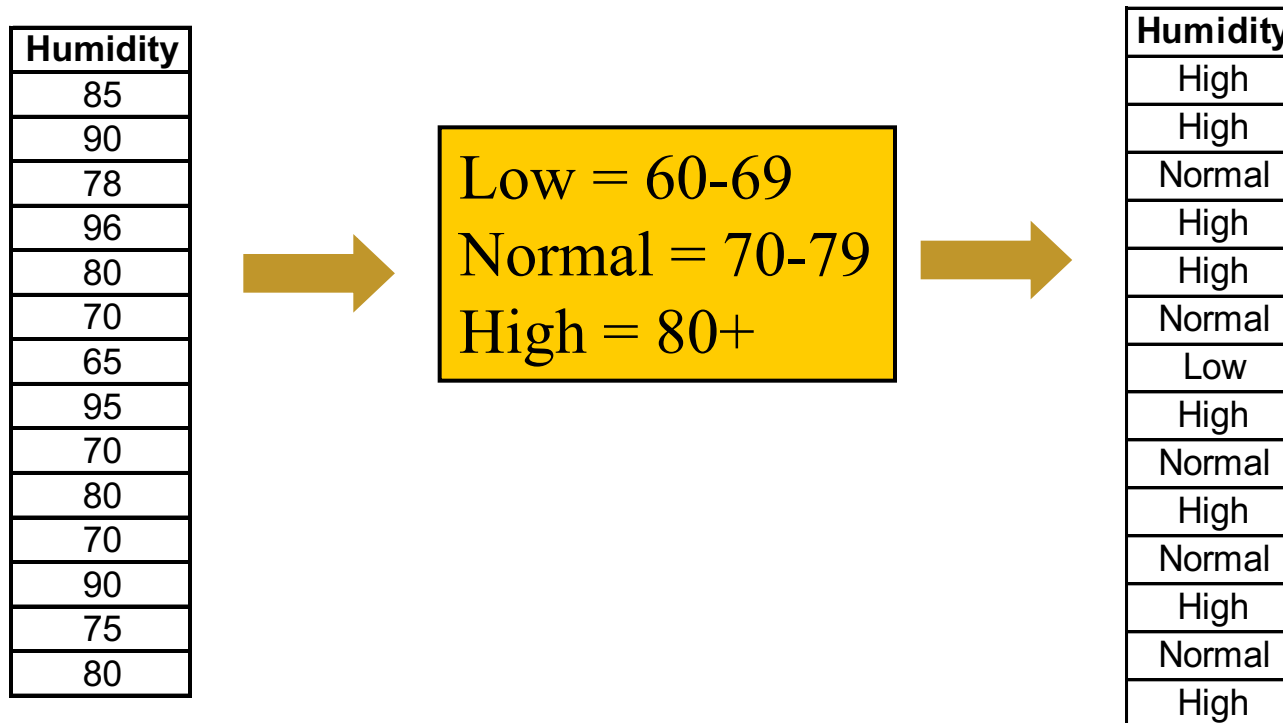
ID	Gender	Age	Salary
1	1	0.00	0.00
2	0	0.96	0.56
3	0	1.00	1.00
4	1	0.24	0.44
5	0	0.72	0.32

# Data Transformation: Discretization

- 3 Types of attributes
  - **nominal** - values from an unordered set (also “categorical” attributes)
  - **ordinal** - values from an ordered set
  - **numeric/continuous** - real numbers (but sometimes also integer values)
- Discretization is used to **reduce the number of values for a given continuous attribute**
  - usually done by dividing the range of the attribute into intervals
  - interval labels are then used to replace actual data values
- Some data mining algorithms only accept categorical attributes and cannot handle a range of continuous attribute value
- Discretization can also be used to generate **concept hierarchies**
  - reduce the data by collecting and replacing low level concepts (e.g., numeric values for “age”) by higher level concepts (e.g., “young”, “middle aged”, “old”)

# Discretization - Example

Example: discretizing the “Humidity” attribute using 3 bins.



# Data Discretization Methods

- 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: Binning

- Equal-width (distance) partitioning
  - 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
  - Divides the range into N intervals, each containing approximately same number of samples
  - Good data scaling

# Discretization by Classification & Correlation Analysis

- Classification (e.g., decision tree analysis)
  - Supervised: Given class labels, e.g., cancerous vs. benign
  - Using entropy to determine split point (discretization point)
  - Top-down, recursive split
- Correlation analysis (e.g., Chi-merge:  $\chi^2$ -based discretization)
  - Supervised: use class information
  - Bottom-up merge: merge the best neighboring intervals (those with similar distributions of classes, i.e., low  $\chi^2$  values)
  - Merge performed recursively, until a predefined stopping condition

# Converting Categorical Attributes to Numerical Attributes – Dummy variables

ID	Outlook	Temperature	Humidity	Windy
1	sunny	85	85	FALSE
2	sunny	80	90	TRUE
3	overcast	83	78	FALSE
4	rain	70	96	FALSE
5	rain	68	80	FALSE
6	rain	65	70	TRUE
7	overcast	58	65	TRUE
8	sunny	72	95	FALSE
9	sunny	69	70	FALSE
10	rain	71	80	FALSE
11	sunny	75	70	TRUE
12	overcast	73	90	TRUE
13	overcast	81	75	FALSE
14	rain	75	80	TRUE

## Attributes:

**Outlook (overcast, rain, sunny)**

**Temperature real**

**Humidity real**

**Windy (true, false)**

## Standard Spreadsheet Format

Create separate columns for each value of a categorical attribute (e.g., 3 values for the Outlook attribute and two values of the Windy attribute). There is no change to the numerical attributes.

OutLook	OutLook	OutLook	Temp	Humidity	Windy	Windy
overcast	rain	sunny			TRUE	FALSE
0	0	1	85	85	0	1
0	0	1	80	90	1	0
1	0	0	83	78	0	1
0	1	0	70	96	0	1
0	1	0	68	80	0	1
0	1	0	65	70	1	0
1	0	0	64	65	1	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.





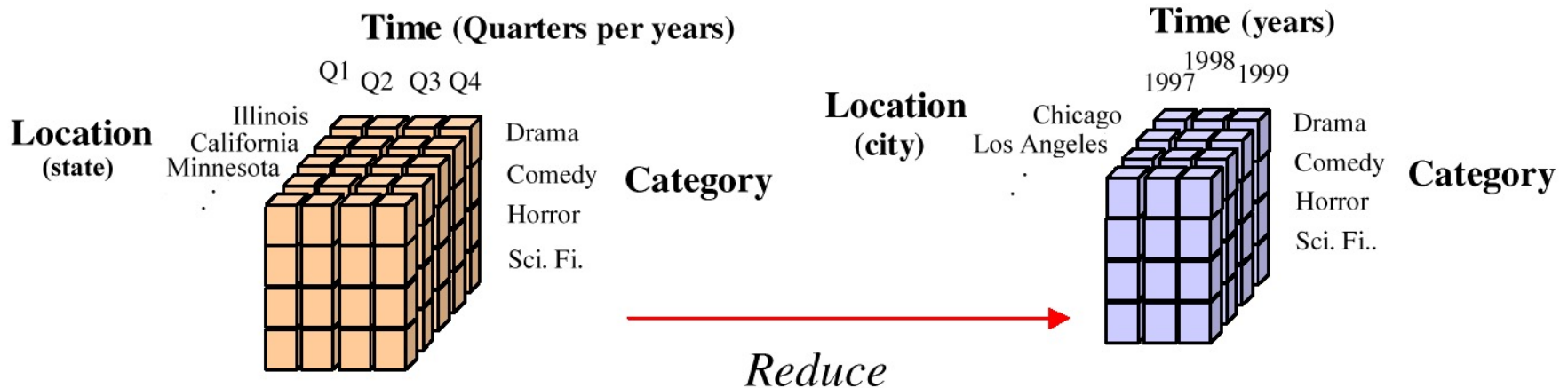
# Data Reduction

- Data is often too large; reducing data can improve performance
- Data reduction consists of reducing the representation of the data set while producing the same (or almost the same) results
- Data reduction includes:
  - Data (cube) aggregation
  - Dimensionality reduction
  - Discretization
  - Numerosity reduction
    - Regression
    - Histograms
    - Clustering
    - Sampling



# Data Cube Aggregation

- Reduce the data to the concept level needed in the analysis
  - Use the smallest (most detailed) level necessary to solve the problem

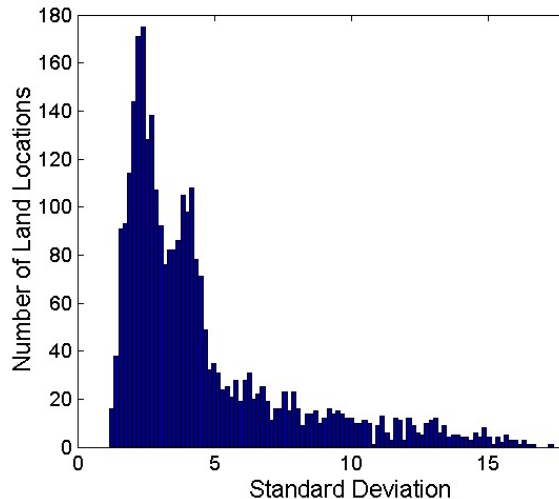


- Queries regarding aggregated information should be answered using data cube when possible

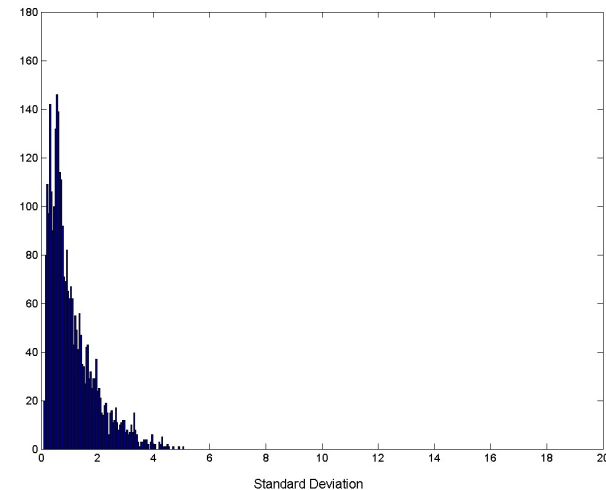
# Data Aggregation

- Change of scale: Cities aggregated into regions, states, countries, etc.
- More “stable” data: Aggregated data tends to have less variability
- Data reduction: Reduce the number of objects

**Variation of Precipitation in Australia**



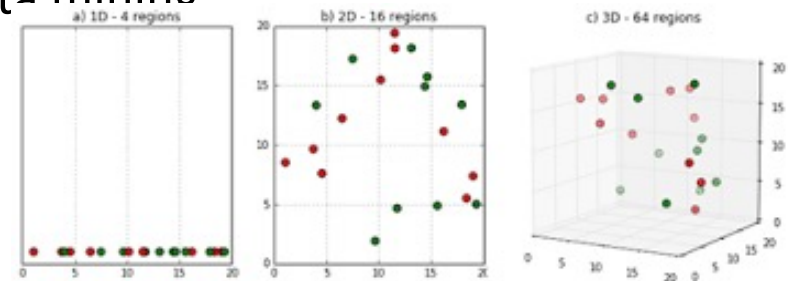
**Standard Deviation of Average  
Monthly Precipitation**



**Standard Deviation of Average  
Yearly Precipitation**

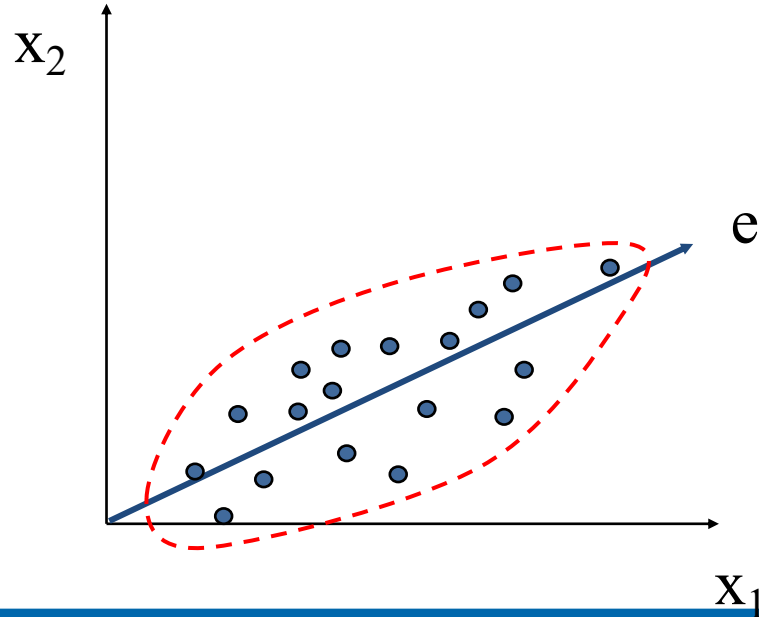
# Dimensionality Reduction

- **Curse of dimensionality**
  - When dimensionality increases, data becomes increasingly sparse
  - **Density** and **distance** between points, which is critical to clustering, outlier analysis, becomes less meaningful
  - The possible combinations of subspaces will grow exponentially
- Dimensionality reduction
  - Improve **interpretability**
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization
- Dimensionality reduction techniques
  - Principal Component Analysis
  - Attribute subset selection
  - Attribute or feature generation



# Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction
  - Done by finding the eigenvectors of the covariance matrix, and these eigenvectors define the new space

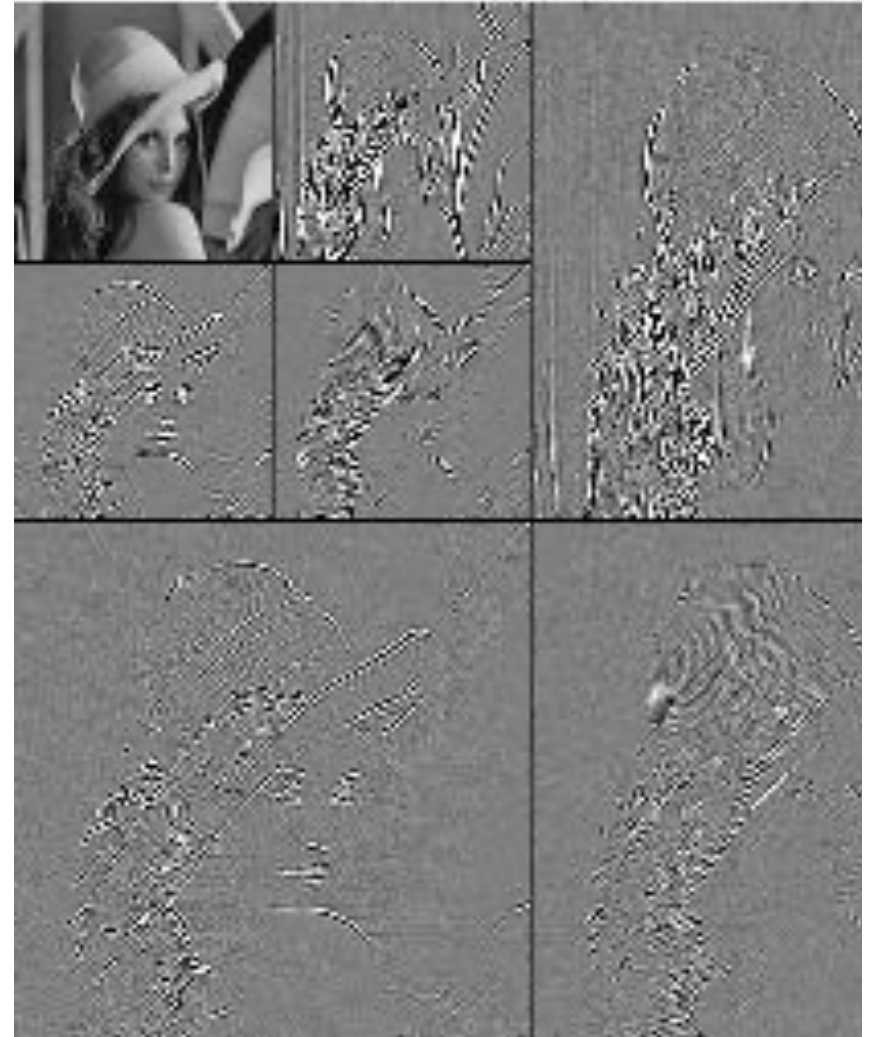


# Principal Component Analysis (Steps)

- Given  $N$  data vectors (rows in a table) from  $n$  dimensions (attributes), find  $k \leq n$  orthogonal vectors (*principal components*) that can be best used to represent data
  - Normalize input data: Each attribute falls within the same range
  - Compute  $k$  orthonormal (unit) vectors, i.e., *principal components*
  - Each input data (vector) is a linear combination of the  $k$  principal component vectors
  - The principal components are sorted in order of decreasing “significance” or strength
  - The size of the data can be reduced by eliminating the *weak components*, i.e., those with low variance
    - Using the strongest principal components, it is possible to reconstruct a good approximation of the original data
- Works for numeric data only

# Other Feature Reduction Methods

- Discrete wavelet transform (DWT): linear signal processing, multiresolutional analysis
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
- Idea: summaries an image (average and difference)
- Method:
  - Length,  $L$ , must be an integer power of 2 (padding with 0s, when necessary)
  - Each transform has 2 functions: smoothing, difference
  - Applies to pairs of data, resulting in two set of data of length  $L/2$
  - Applies two functions recursively, until reaches the desired length
- Non-linear methods: neural-network based (e.g., text data)



# Attribute Subset Selection

- Start here. Remove:
  - Attribute for which values change for every object
  - Attributes for which the values are the same for every object
    - Or almost the same (spread on box plot maybe an indication)
  - Redundant attributes
    - Duplicate much or all of the information contained in one or more other attributes
    - e.g., age and date of birth
  - Irrelevant attributes
    - Contain no information that is useful for the data mining task at hand
    - e.g., student's area code to predict GPA



# Attribute Relevance Analysis

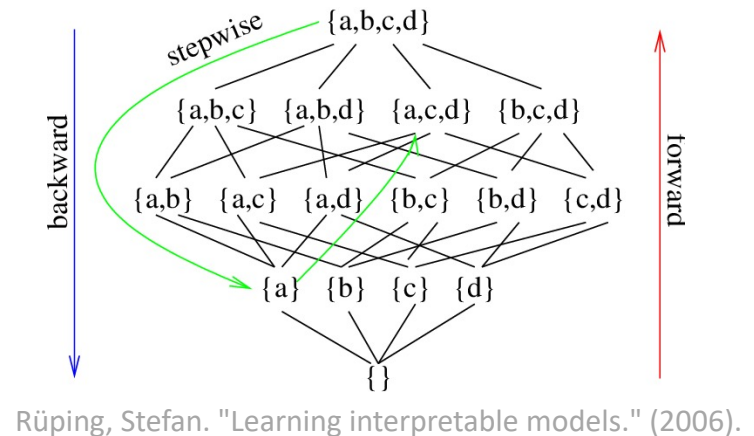
- Idea: compute quantified measure for the attribute given *class* (e.g., info gain, Gini index, correlation coef.)
- Rank attributes from most to least discriminating
- Set an arbitrary threshold for selection
- Problem?

# Feature Subset Selection

- Techniques:
  - Brute-force approach:
    - Try all possible feature subsets as input to data mining algorithm
  - Embedded approaches:
    - Feature selection occurs naturally as part of the data mining algorithm
  - Filter approaches:
    - Features are selected before data mining algorithm is run
  - Wrapper approaches:
    - Use the data mining algorithm as a **black box** to find best subset of attributes
  - We revisit this topic later in the course

# Heuristic Search in Attribute Selection

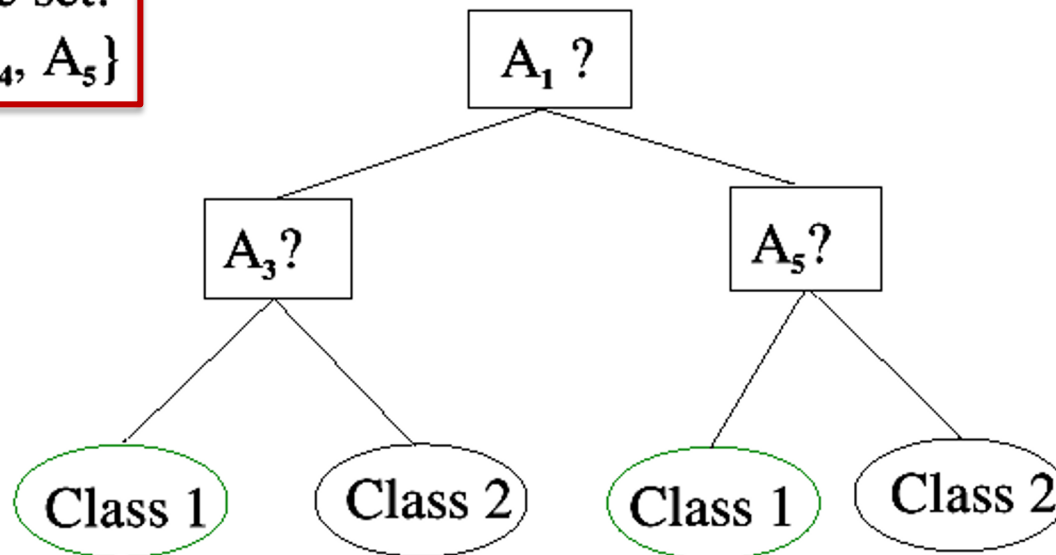
- There are  $2^d$  possible attribute combinations of  $d$  attributes
- Typical heuristic attribute selection methods:
  - “Best” single attribute under the attribute independence assumption: choose by significance tests
  - Best step-wise feature selection:
    - The best single-attribute is picked first. Then next best attribute condition to the first, ...
    - $\{\{A1\}\{A1, A3\}\{A1, A3, A5\}$
  - Step-wise attribute elimination:
    - Repeatedly eliminate the worst attribute:  $\{A1, A2, A3, A4, A5\}\{A1, A3, A4, A5\} \{A1, A3, A5\}, \dots$
  - Combined attribute selection and elimination
  - The stopping criteria for the methods may vary
  - Decision Tree Induction



# Decision Tree Induction

Use information theory techniques to find the most “informative” attributes

Initial attribute set:  
 $\{A_1, A_2, A_3, A_4, A_5\}$



-----> Reduced attribute set:  $\{A_1, A_3, A_5\}$

# Attribute/Feature Creation/Generation

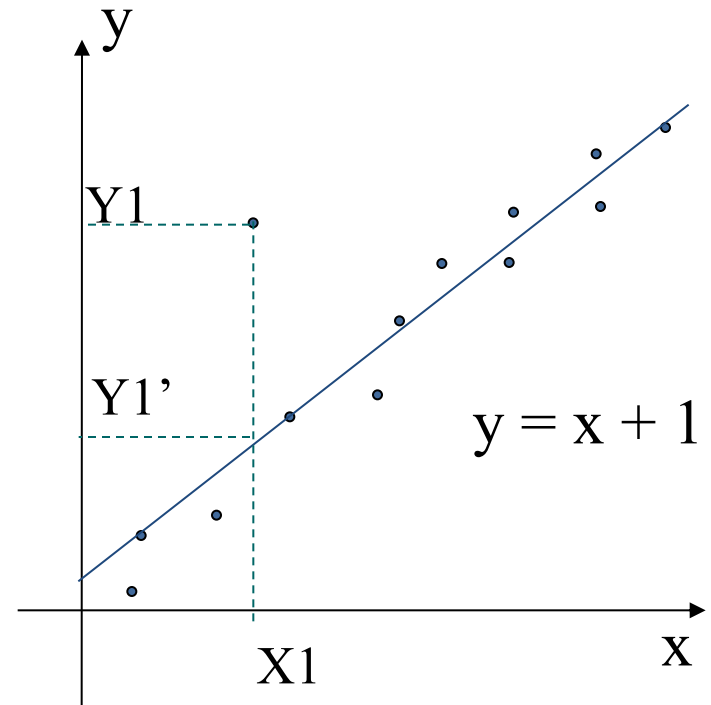
- Create new attributes (features) that can capture the important information in a data set more effectively than the original ones
- Three general methodologies
  - Attribute extraction
    - Domain-specific
  - Mapping data to new space (see data reduction)
    - E.g., Fourier transformation, wavelet transformation, etc.
  - Attribute construction
    - Combining features
    - Data discretization

# Data Reduction: Numerosity Reduction

- Reduce data volume by choosing alternative, *smaller forms* of data representation
- **Parametric methods** (e.g., regression)
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
  - Ex.: Linear models — keep equation, discard points
- **Non-parametric methods**
  - Do not assume models
  - Major families: histograms, clustering, sampling, ...

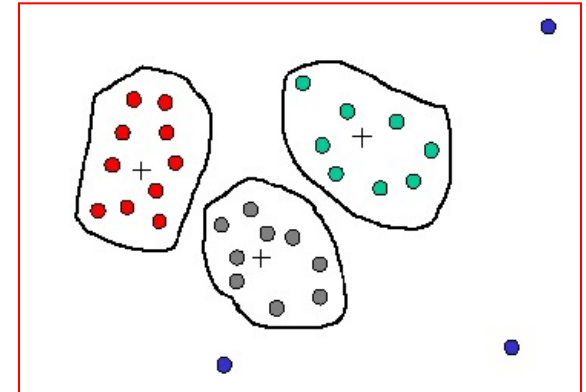
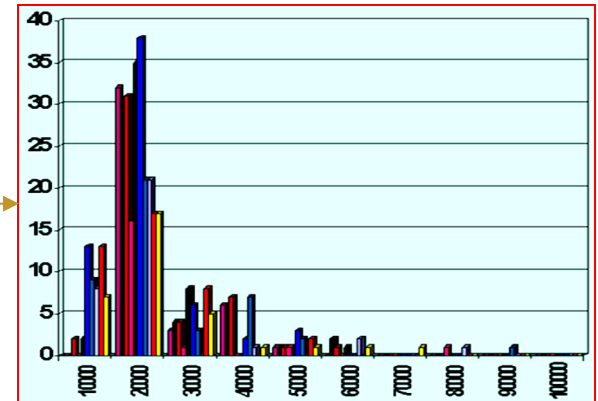
# Regression Analysis

- Collection of techniques for the modeling and analysis of numerical data consisting of values of a *dependent variable* (also *response variable* or *measurement*) and of one or more *independent variables* (aka. *explanatory variables* or *predictors*)
- The parameters are estimated to obtain a "best fit" of the data
- Typically, the best fit is evaluated by using the least squares method, but other criteria have also been used
- Used for prediction (including forecasting of time-series data), inference, hypothesis testing, and modeling of causal relationships



# Numerocity Reduction

- Reduction via histograms:
  - Divide data into buckets and store representation of buckets (sum, count, etc.)
- Reduction via clustering
  - Partition data into clusters based on “closeness” in space
  - Retain representatives of clusters (centroids) and outliers
- Reduction via sampling
  - Will the patterns in the sample represent the patterns in the data?
  - Random sampling can produce poor results
  - Stratified sample (stratum = group based on attribute value)





# Sampling

- The key principle for effective sampling is the following:
  - using a sample will work almost as well as using the entire data sets, if the sample is representative
  - A sample is representative if it has approximately the same property (of interest) as the original set of data
- Potential problems:
  - Imbalanced classes
  - Not enough data

# Types of Sampling

- Simple Random Sampling
  - There is an equal probability of selecting any particular item
- Sampling without replacement
  - As each item is selected, it is removed from the population
- Sampling with replacement
  - Objects are not removed from the population as they are selected for the sample.
    - In sampling with replacement, the same object can be picked up more than once
- Stratified sampling
  - Split the data into several partitions; then draw random samples from each partition

# Sampling Techniques

