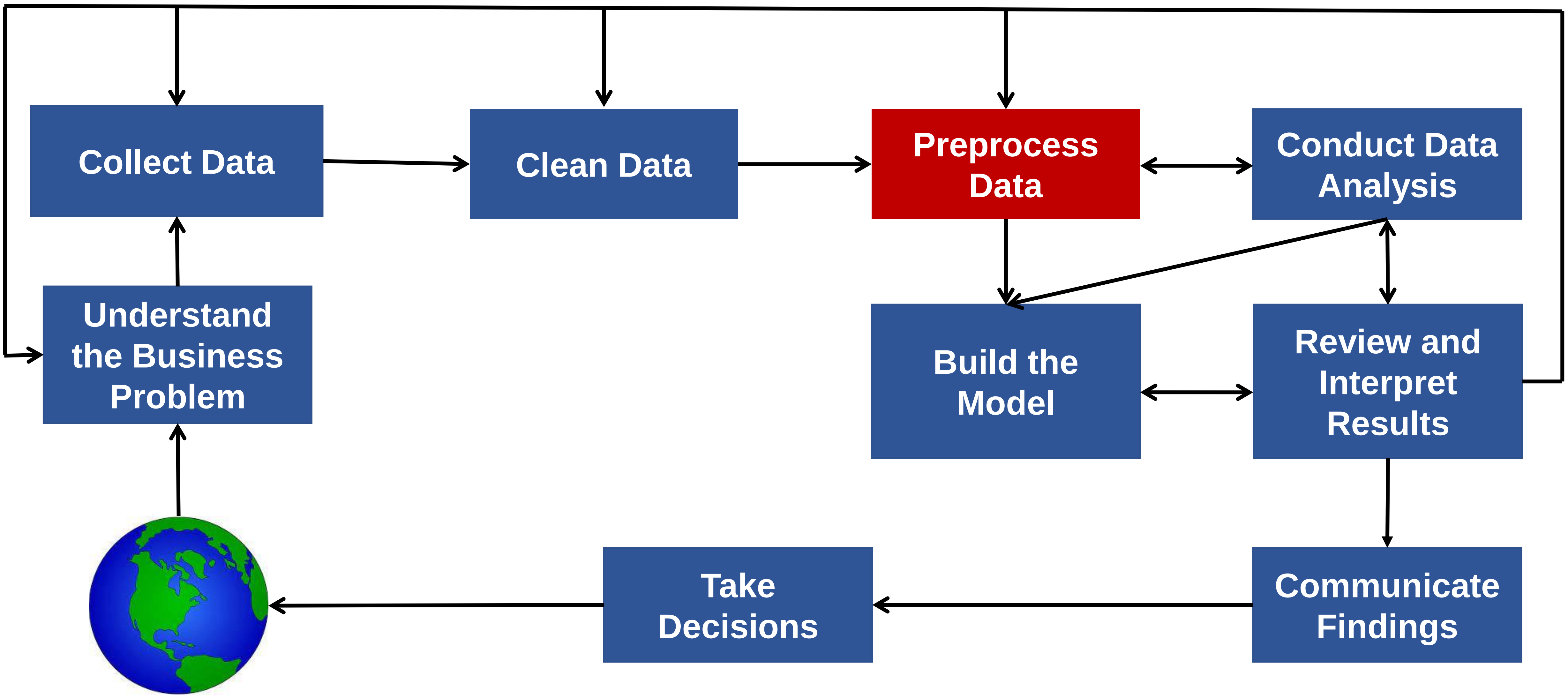
Data Pre-processing

Dr. Sandareka Wickramanayake

sandarekaw@cse.mrt.ac.lk

Data Science Process



Collect Data

Clean Data

Preprocess Data

Conduct Data Analysis

Understand the Business Problem

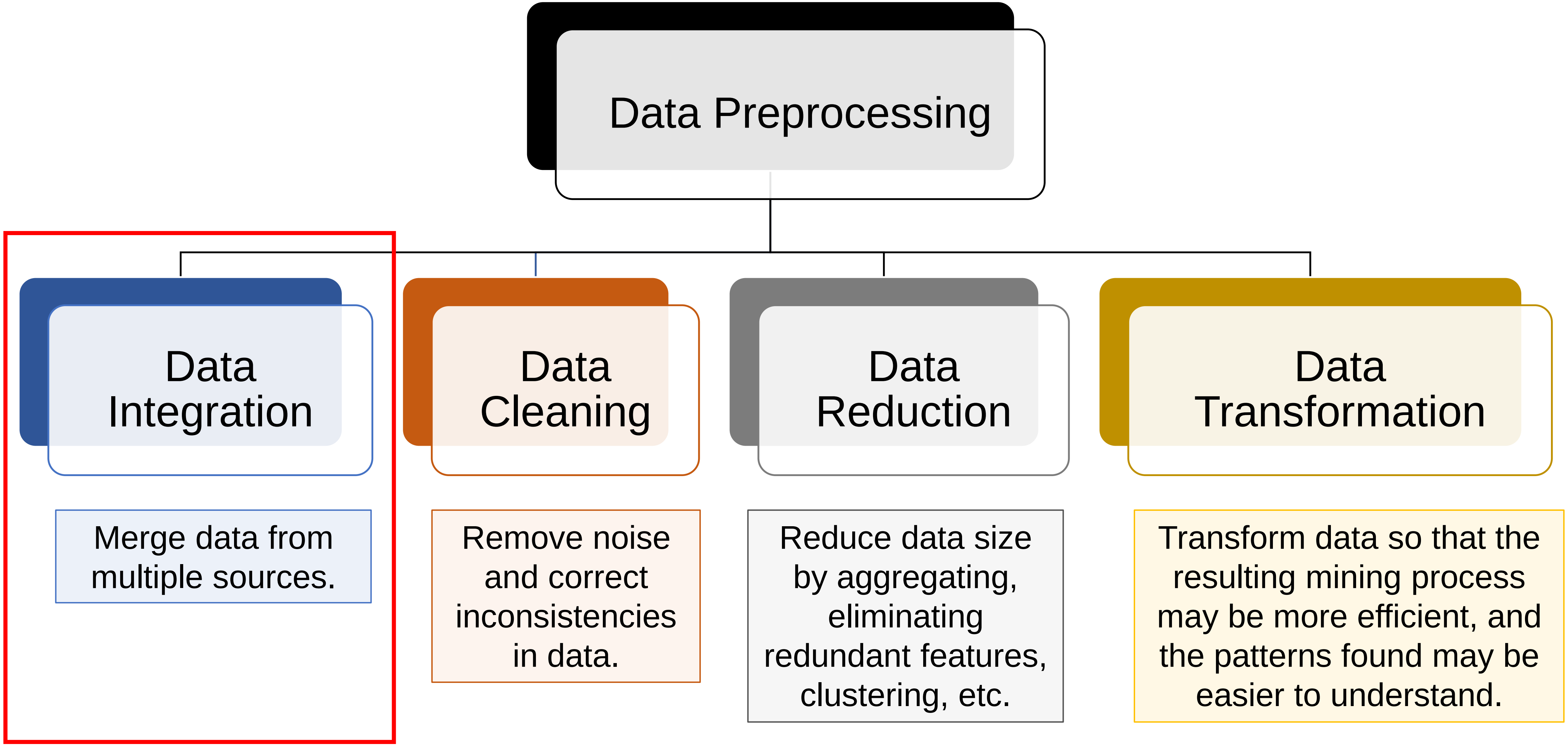
Build the Model

Review and Interpret Results

Take Decisions

Communicate Findings

Major Tasks in Data Preprocessing



Data Preprocessing

Data Integration

Merge data from multiple sources.

Data Cleaning

Remove noise and correct inconsistencies in data.

Data Reduction

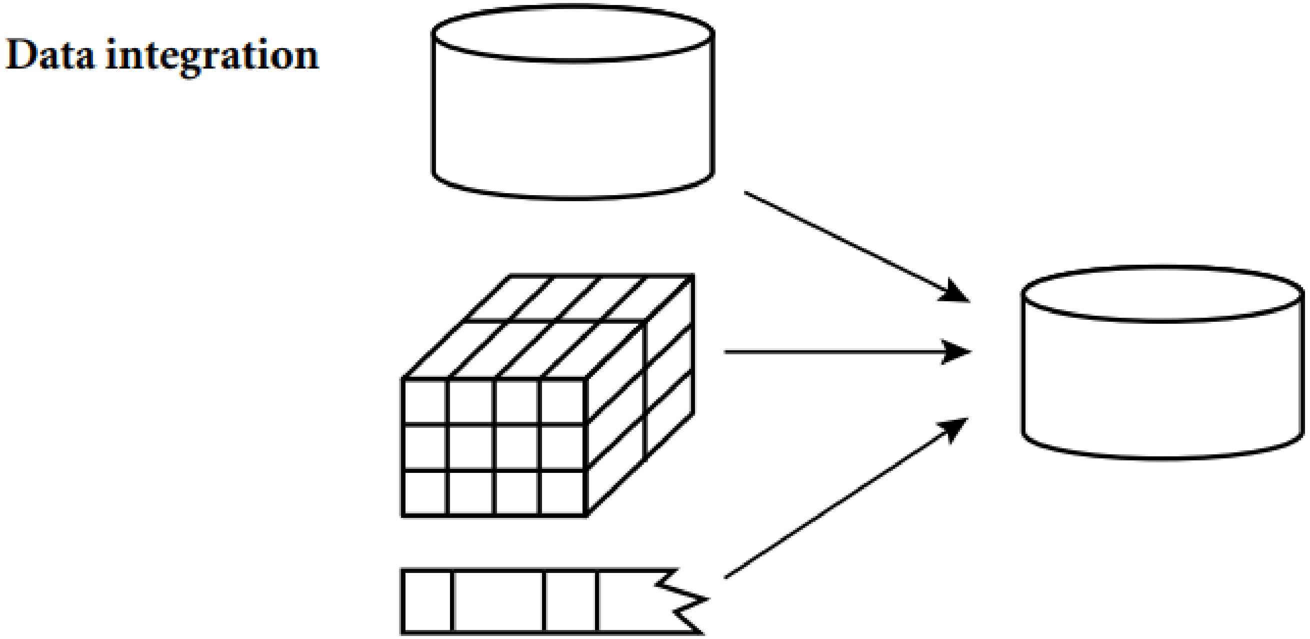
Reduce data size by aggregating, eliminating redundant features, clustering, etc.

Data Transformation

Transform data so that the resulting mining process may be more efficient, and the patterns found may be easier to understand.

Data Integration

- Data Integration – The merging of data from multiple sources.
- Allow access to more information ➔ Better data analysis, and model building



Data Integration

- Challenges in data integration
  - The entity identification problem
  - Attribute redundancy
  - Tuple duplication
  - Data value conflicts

- The entity identification problem
  - How can equivalent real-world entities from multiple data sources be matched up?
    - E.g., Database A – *Customer_ID*   Database_B – *cus_number*
  - Meta data about attributes can be used to avoid these problems.

Data Integration

- Challenges in data integration
  - The entity identification problem
  - Attribute redundancy
  - Tuple duplication
  - Data value conflicts

- Attribute redundancy
  - An attribute is redundant if it can be “derived” from another attribute or set of attributes.
  - Inconsistencies in attribute or dimension naming can also cause redundancies.
  - Correlation analysis can be used to identify redundancies.
    - Given two attributes, correlation analysis measures how strongly one attribute implies the other.
    - Nominal data – Chi-square, Metric data – Correlation coefficient

Data Integration

- Challenges in data integration
  - The entity identification problem
  - Attribute redundancy
  - Tuple duplication
  - Data value conflicts

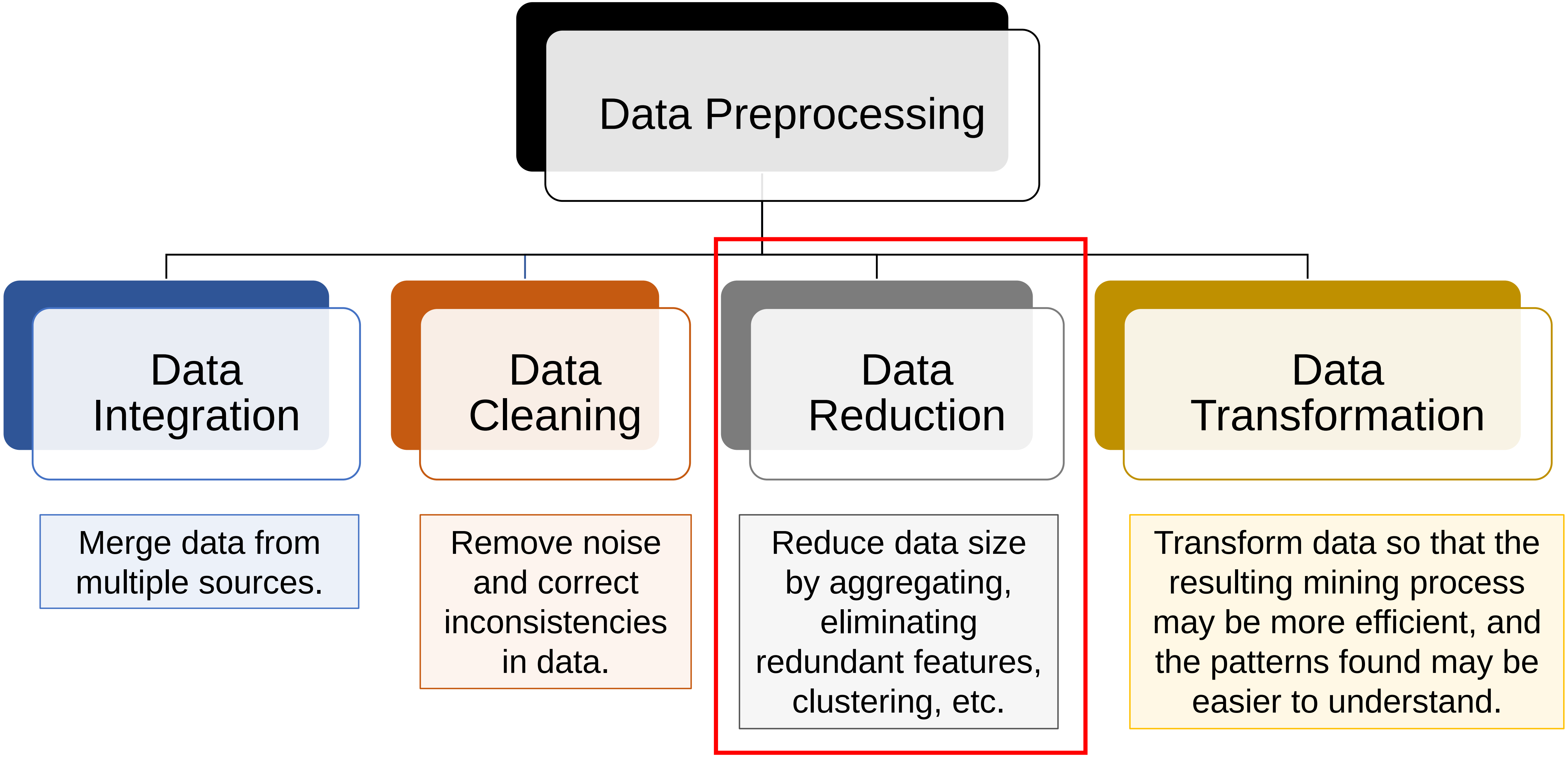
- Tuple duplication
  - Situations where two or more identical tuples exist for a unique data entry case.

Data Integration

- Challenges in data integration
  - The entity identification problem
  - Attribute redundancy
  - Tuple duplication
  - Data value conflicts

- Data value conflicts
  - For the same real-world entity, attribute values from different sources are different.
  - Can be caused by differences in representation, scaling, or encoding.

Major Tasks in Data Preprocessing



Data Preprocessing

Data Integration

Merge data from multiple sources.

Data Cleaning

Remove noise and correct inconsistencies in data.

Data Reduction

Reduce data size by aggregating, eliminating redundant features, clustering, etc.

Data Transformation

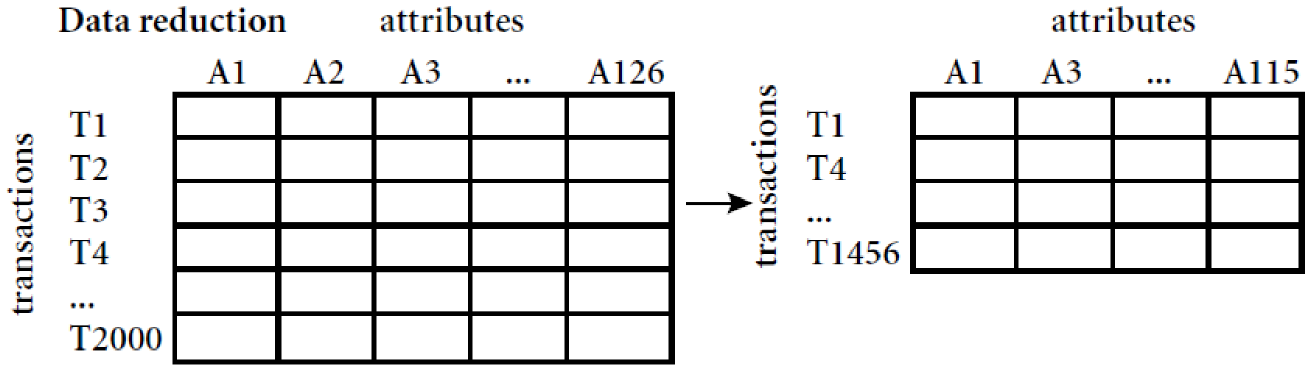
Transform data so that the resulting mining process may be more efficient, and the patterns found may be easier to understand.

Data Reduction

- Reduce the number of attributes or objects
- Data reduction methods
  - Aggregation
  - Sampling
  - Dimensionality reduction
    - Attribute selection
    - Principal Component Analysis (PCA)

Data Reduction

- Produce a reduced representation of the dataset.
  - Reduce the number of attributes or objects
- Analysing the reduced dataset is more efficient yet produces the same (almost same) results.



- Data reduction methods
  - Aggregation
  - Sampling
  - Dimensionality reduction
    - Attribute selection
    - Principal Component Analysis

Data Reduction - Aggregation

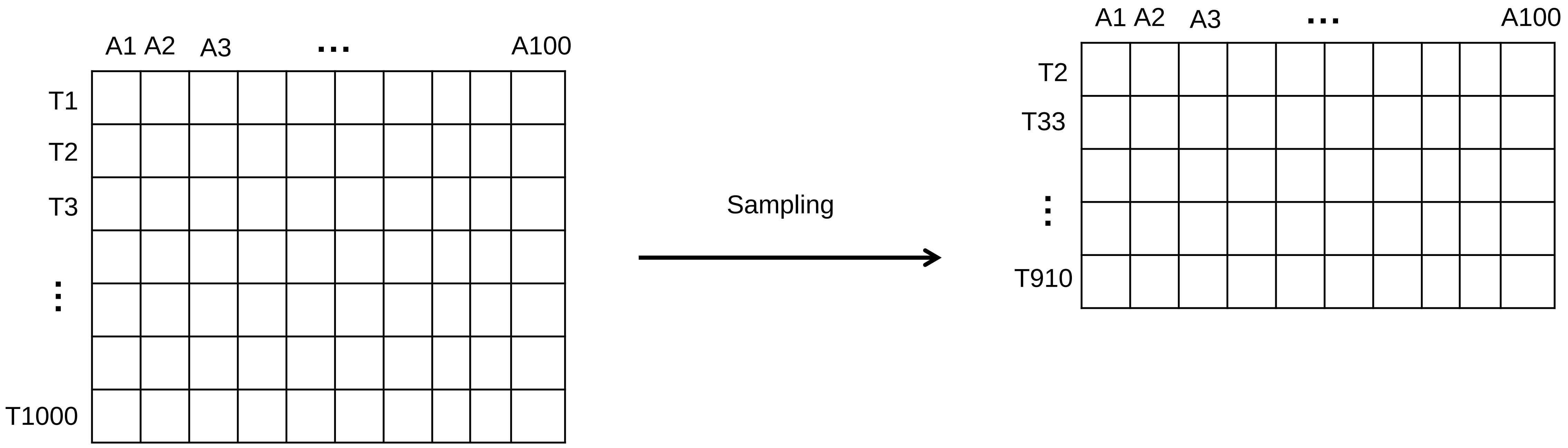
- Combining two or more attributes (or objects) into a single attribute (or object).
- Change of scale
  - Cities aggregated into regions, states, countries, etc.
- Provides more “stable” data
  - Aggregated data tends to have less variability
- Can use clustering or histograms

Data Reduction - Sampling

- The main technique used for data selection.
  - For both preliminary and final analysis
- Used because processing the entire set of data is too expensive or time-consuming.
- *Sample should be representative*
  - Key principle for effective sampling.
  - Using a representative sample will work almost as well as using the entire dataset.
  - A representative sample has approximately the same property as the original dataset.
- Sample size vs representativeness

Data Reduction

- Sampling
  - Allows a large data set to be represented by a much smaller random sample



A1 A2 A3 … A100

T1

T2

T3

⋮

T1000

Sampling

A1 A2 A3 … A100

T2

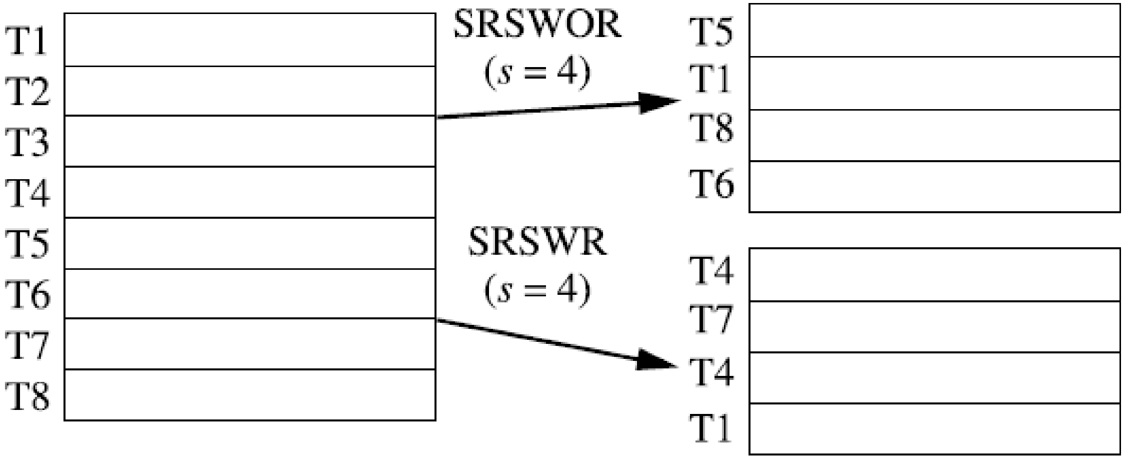
T33

⋮

T910

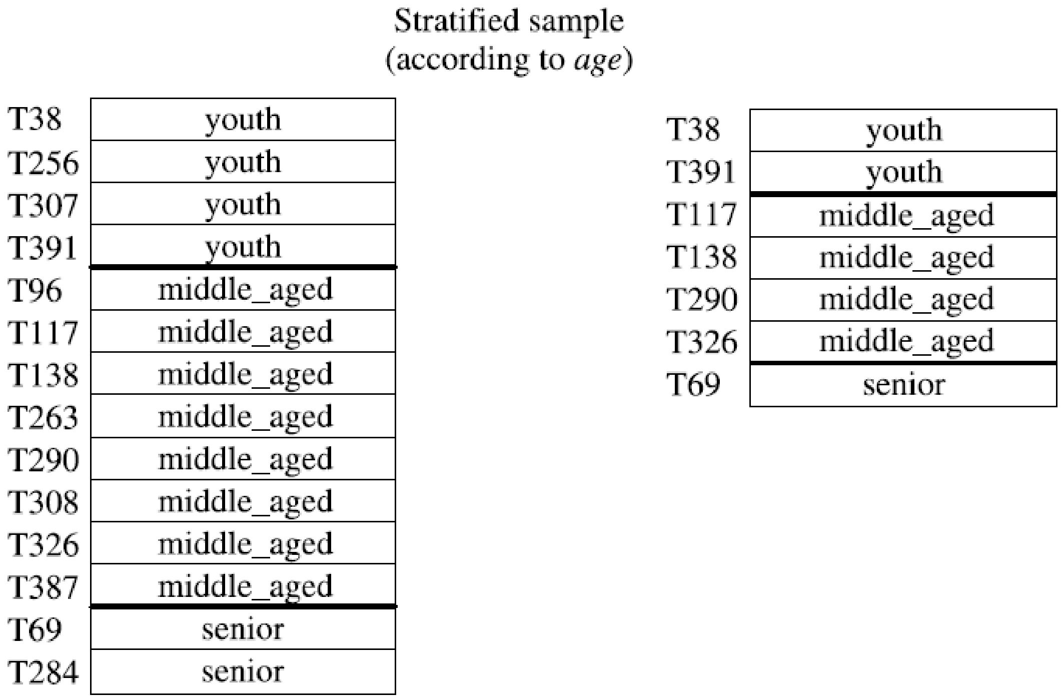
Data Reduction

- Sampling
  - Simple Random Sampling
    - There is an equal probability of selecting any particular item.
    - Simple Random Sampling With Out Replacement (SRSWOR)
      - Once an item is selected, it is removed from the population.
    - Simple Random Sampling With Replacement (SRSWR)
      - Items are not removed from the population as they are selected for the sample.
      - The same item can be picked up more than once



Data Reduction

- Sampling
  - Stratified sampling
    - Split the data into several partitions. Then draw random samples from each partition.



Stratified sample
(according to *age*)

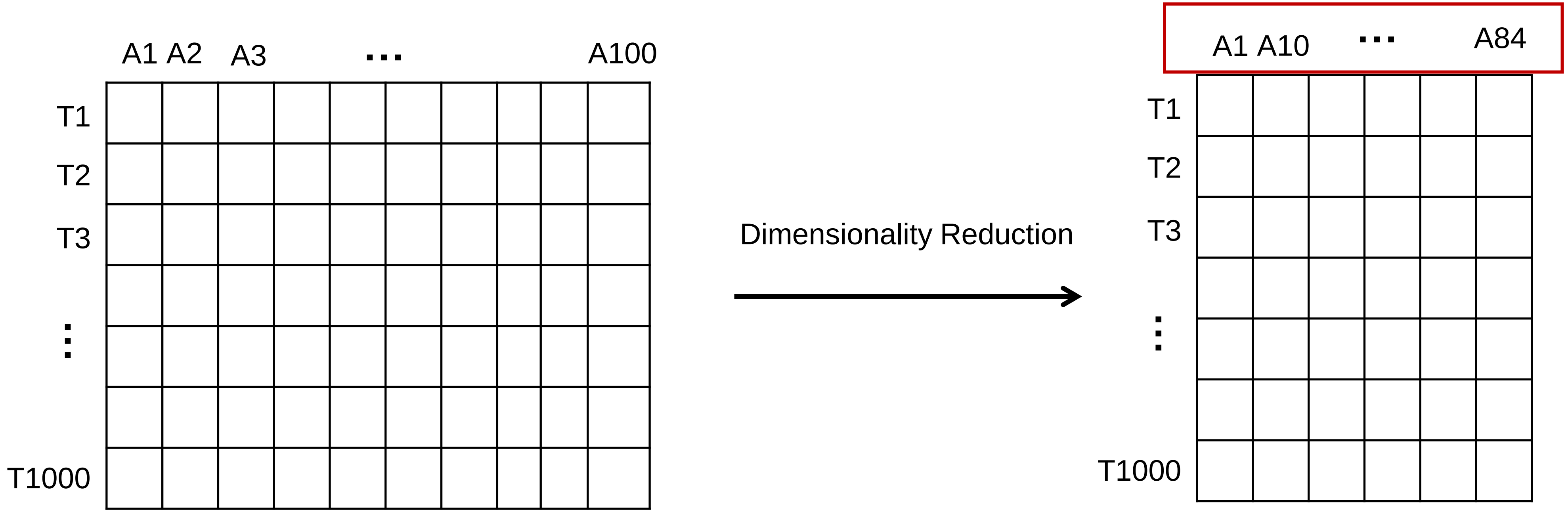
T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

A representative sample, especially when the data are skewed.

Data Reduction - Dimensionality Reduction

- Curse of dimensionality
  - When dimensionality (the number of features) increases, the data we need to generalize accurately grows exponentially.
  - When dimensionality (the number of features) increases, the data becomes increasingly sparse in the space it occupies.



Data Reduction - Dimensionality Reduction

- Purposes
  - Avoid the curse of dimensionality.
  - Reduce the time and memory required by the data mining algorithm.
  - Allow data to be more easily visualized.
  - May help to eliminate irrelevant attributes or reduce noise.
- Techniques
  - **Attribute selection**
  - **Principle Component Analysis (PCA)**
  - t-distributed stochastic neighbour embedding (t-sne)
  - Singular Value Decomposition (SVD)

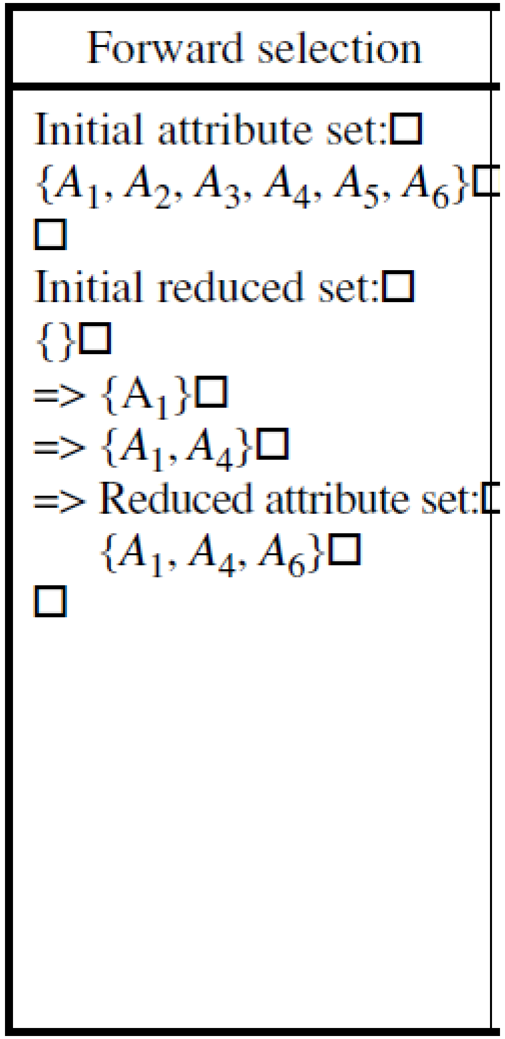
Data Reduction - Dimensionality Reduction

- Attribute selection
    - Identify and remove redundant or irrelevant attribute
        - Redundant attributes
            - Duplicate much or all of the information contained in one or more other attributes
            - E.g., the purchase price of a product and the amount of sales tax paid.
        - Irrelevant attributes
            - Contains no information that is useful for the task at hand.
            - E.g., a student’s ID is irrelevant to predicting student GPA.

    - Find a minimum set of representative attributes

Data Reduction - Dimensionality Reduction

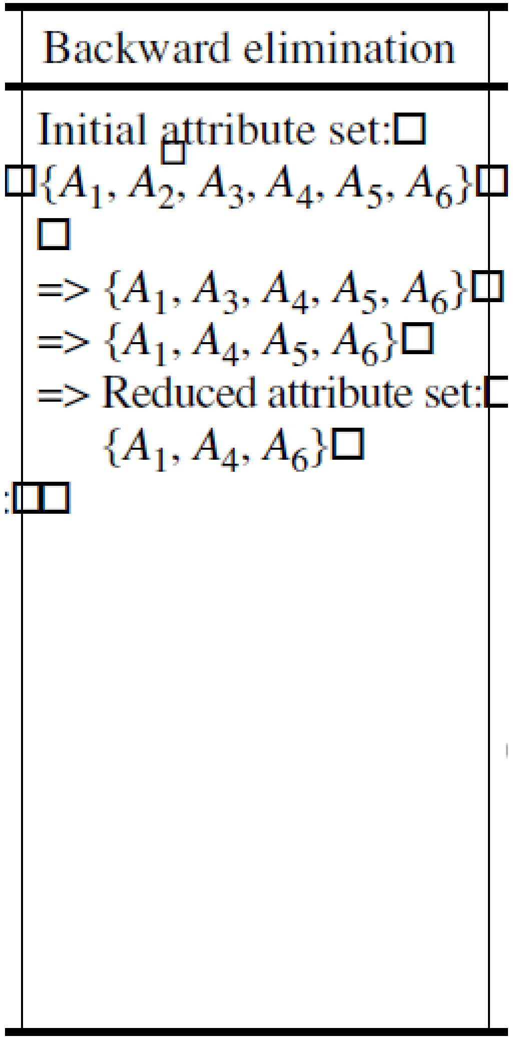
- Attribute selection - Attribute selection methods
  - Stepwise forward selection
    - Starts with an empty set of attributes as the reduced set.
    - At each subsequent iteration, the best of the remaining original attributes is added to the set.
    - The “best” (and “worst”) attributes are typically determined using tests of statistical significance .



Forward selection
Initial attribute set:□ $\{A_1, A_2, A_3, A_4, A_5, A_6\}$□ □ Initial reduced set:□ {}□ => $\{A_1\}$□ => $\{A_1, A_4\}$□ => Reduced attribute set:□ $\{A_1, A_4, A_6\}$□ □

Data Reduction - Dimensionality Reduction

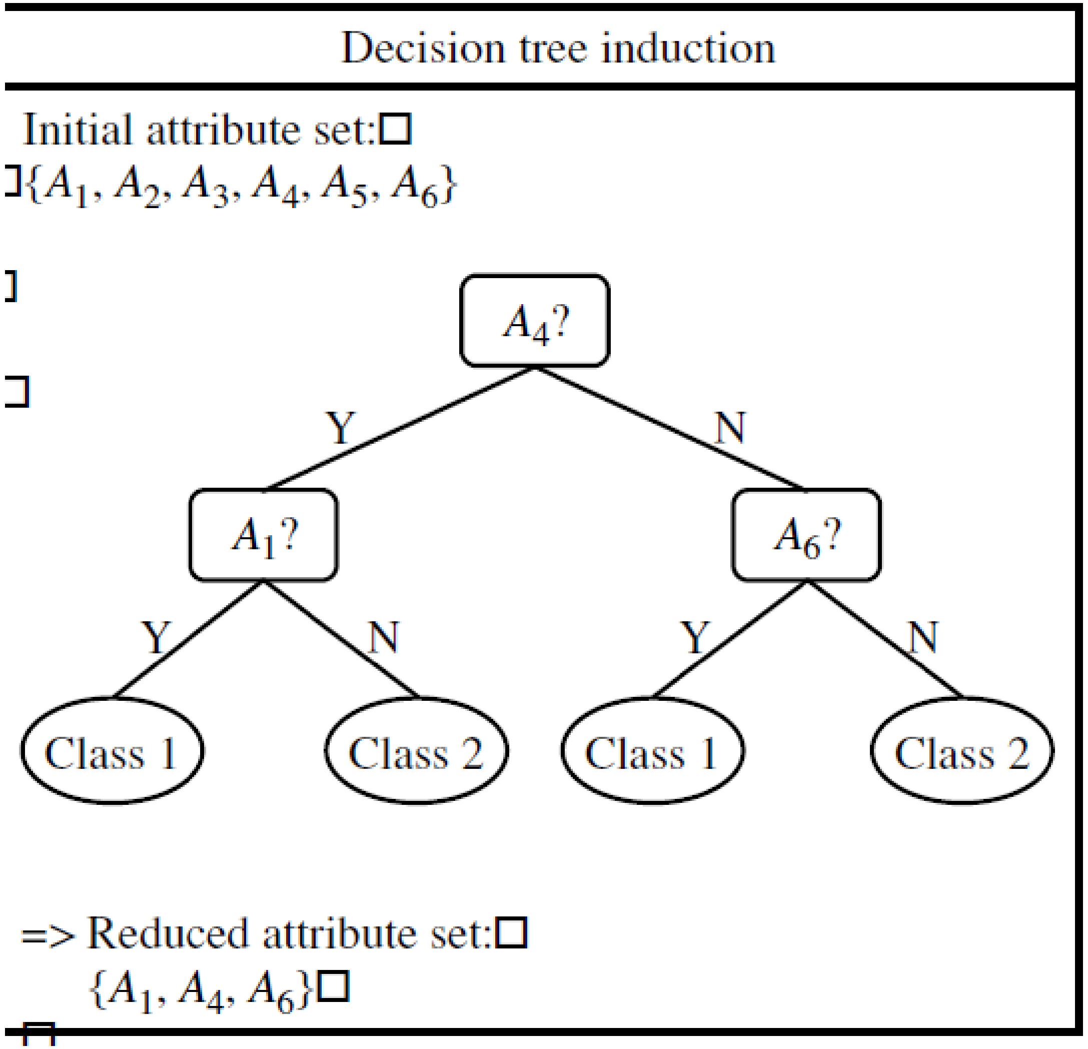
- Attribute selection - Attribute selection methods
  - Stepwise backward selection
    - Starts with the full set of attributes.
    - At each step, it removes the worst attribute remaining in the set.
    - The “best” (and “worst”) attributes are typically determined using tests of statistical significance.



Backward elimination
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$

Data Reduction - Dimensionality Reduction

- Attribute selection - Attribute selection methods
  - Decision tree induction
    - A decision tree is constructed from the given data.
    - All attributes that do not appear in the tree are assumed to be irrelevant.
    - The set of attributes appearing in the tree forms the reduced subset of attributes.



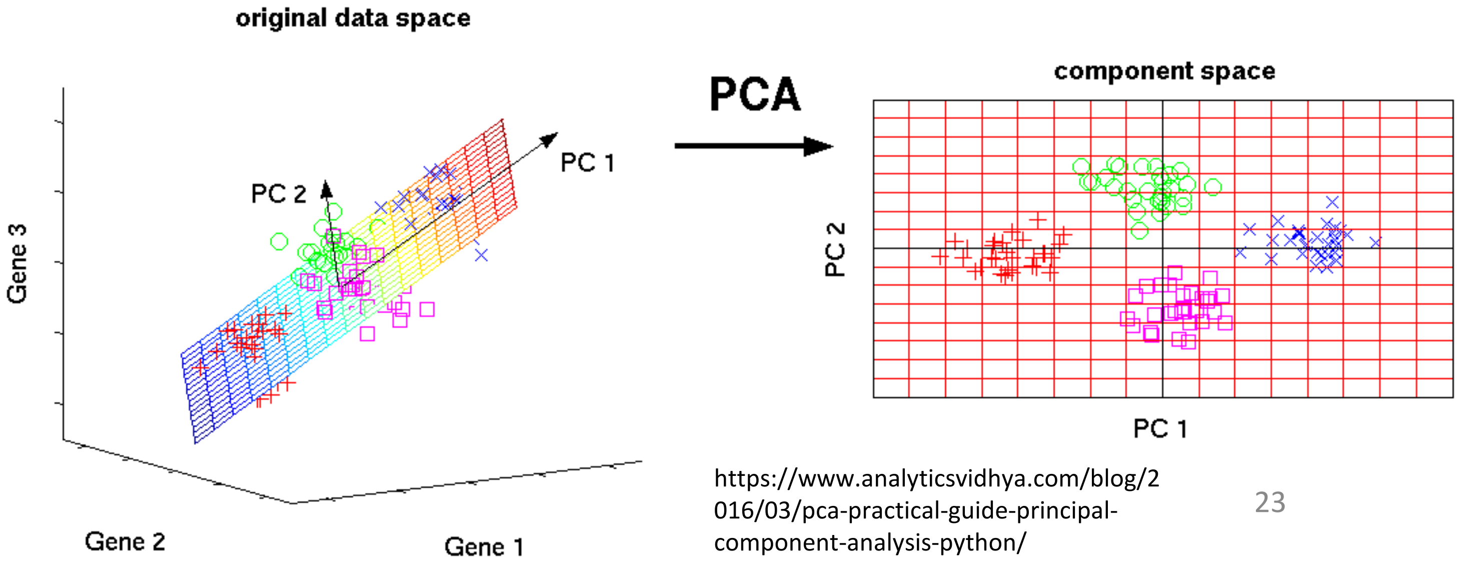
Decision tree induction

Initial attribute set:□
$\{A_1, A_2, A_3, A_4, A_5, A_6\}$

=> Reduced attribute set:□
$\{A_1, A_4, A_6\}$□

Data Reduction - Dimensionality Reduction

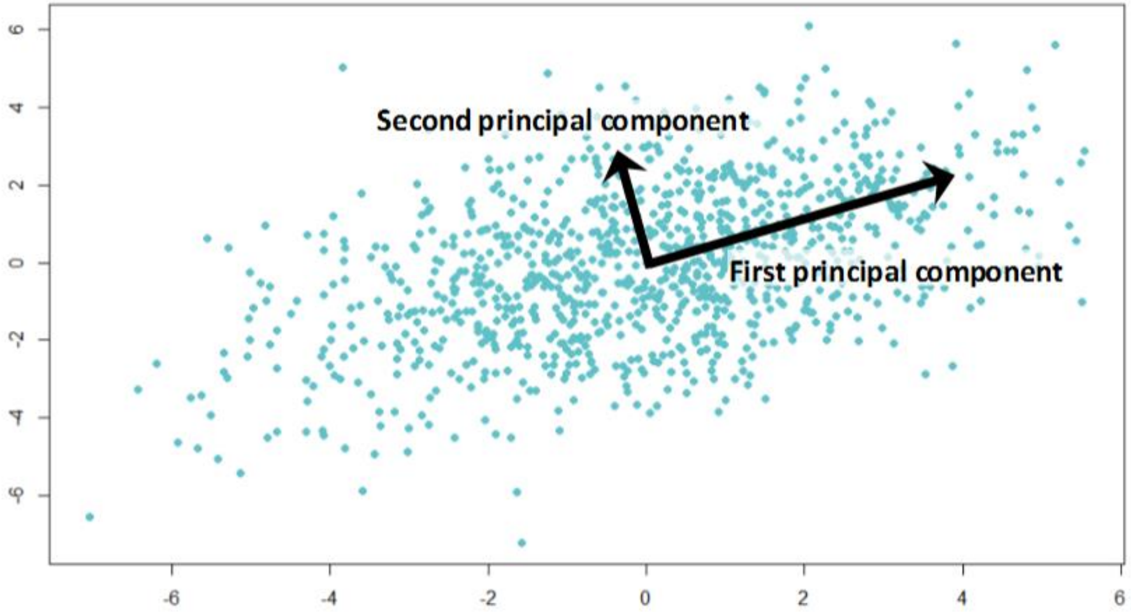
- Principal Component Analysis (PCA)
  - There may be **correlated attributes.**
  - A **few uncorrelated attributes** are desirable.
  - Suppose we have n attributes.
  - PCA searches for k n-dimensional orthogonal vector units that can best be used to represent the data, where $k \leq n$.
  - Original data are now projected to a much smaller space.
  - PCA “combines” the essence of attributes by creating an alternative, smaller set of variables**.**



https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/

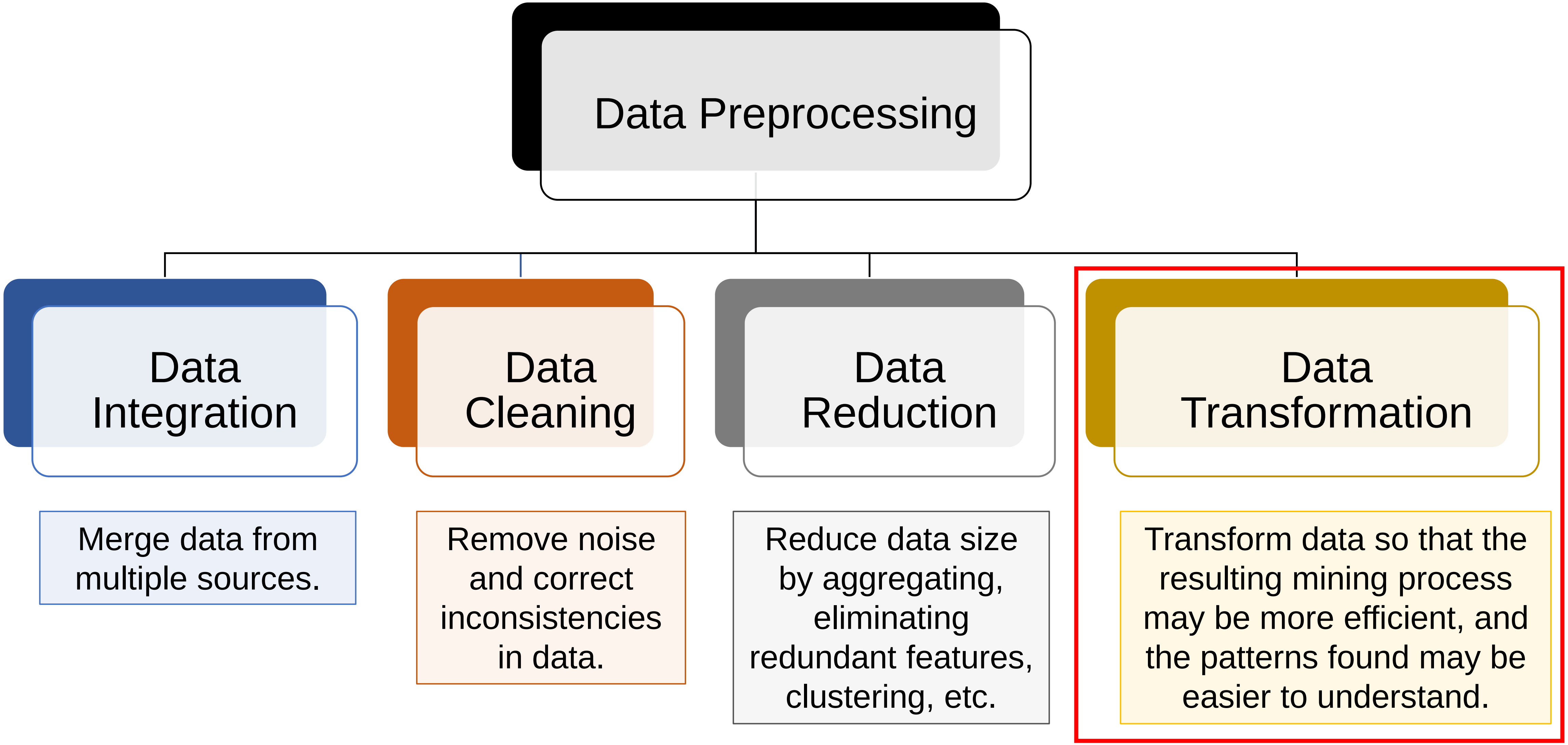
Data Reduction - Dimensionality Reduction

- Principal Component Analysis (PCA) – Basic Procedure
  1. The input data are normalized.
  2. PCA computes k orthonormal vectors (***Principal components***)
     - Each component is a linear combination of original attributes.
  3. The principal components are sorted in order of decreasing "significance".
  4. The size of the data can be reduced by eliminating the weaker components.



https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/

Major Tasks in Data Preprocessing



Data Preprocessing

Data Integration

Data Cleaning

Data Reduction

Data Transformation

Merge data from multiple sources.

Remove noise and correct inconsistencies in data.

Reduce data size by aggregating, eliminating redundant features, clustering, etc.

Transform data so that the resulting mining process may be more efficient, and the patterns found may be easier to understand.

Data Transformation

- Map the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values.
- To make the resulting mining process more efficient, and the patterns found may be easier to understand.
- Data transformation methods
  - Discretization of Continuous Attributes
  - Binarization
  - Normalization

Data Transformation - Discretization of Continuous Attributes

- Transform a continuous attribute (e.g., age, blood pressure) into a categorical attribute
- **Binning**
  - Two steps
    - Decide how many categories.
      - After the values are sorted, they are dived into n intervals by specifying $n-1$ **split points.**
    - Determine how to map the continuous attribute values to these categories.
      - All the values in one interval are mapped to the same categorical value.
  - The critical issue is how many split points to choose and where to place them (e.g., equal intervals, equal frequency, etc.)
- Unsupervised (e.g., binning, clustering) and supervised (e.g., decision trees).

Data Transformation - Binarization

- Transform either a continuous attribute or a categorical attribute into one or more binary attributes
- A simple method
  - Given m categorical values, assign each original value to an *integer* in the interval $[0, m-1]$.
  - Convert each of these m integers into a binary number
  - Require $n = ceiling(\log_2 m)$ binary digits to represent these integers.

Using fewer Digits

Categorical value	Int value	X1	X2	X3
Very Poor	0	0	0	0
Poor	1	0	0	1
Ok	2	0	1	0
Good	3	0	1	1
Great	4	1	0	0

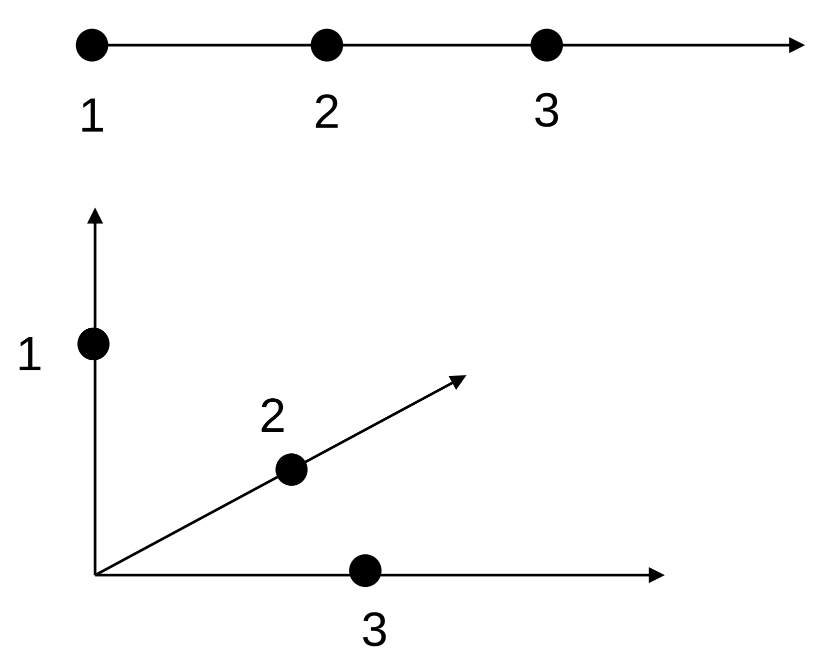
Categorical value	Int value	X1	X2	X3	X4	X5
Very Poor	0	1	0	0	0	0
Poor	1	0	1	0	0	0
Ok	2	0	0	1	0	0
Good	3	0	0	0	1	0
Great	4	0	0	0	0	1

Data Transformation - Binarization

- Transform either a continuous attribute or a categorical attribute into one or more binary attributes

Categorical value	Int value	X1	X2	X3
Doctor	0	1	0	0
Teacher	1	0	1	0
Engineer	2	0	0	1

Euclidian Space



Data Transformation - Normalization

- To help avoid dependence on the choice of measurement units.
- Attempts to give all attributes an equal weight.
- Useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbor classification and clustering.

- **Min-max Normalization (Scaling)**
- **Z-score Normalization (Standardization)**
- **Normalization by Decimal Scaling**

Data Transformation - Normalization

- **Min-max Normalization**

$$[min, max] \rightarrow [newMin, newMax\ ]$$

$$v' = \frac{v\ \ - min}{\max - min}(newMax - newMin) + newMin$$

- Example : Annual income range [$12,000, $300,000] normalized to [0.0, 1.0]

Then $73,000 is mapped to 0.21

$$\frac{73000\ \ - 12000}{300000\ \ - 12000}\ (1.0\ \ - 0.0) + 0.0 = 0.21$$

Data Transformation - Normalization

- **Z-score Normalization (Standardization, Zero-mean Normalization)**
  - The values for an attribute are normalized based on the mean and standard deviation.

μ – Mean σ – Standard deviation

$$v' = \frac{v - \mu}{\sigma}$$

  - Example: Annual income range [$12,000, $300,000]. Suppose μ = 54,000 and σ = 16,000. Then $73,000 is mapped to 1.225
  - Useful
    - when the actual minimum and maximum of the attribute are unknown.
    - When there are outliers.

Data Transformation - Normalization

- **Normalization by Decimal Scaling**
  - normalizes by moving the decimal point of values of the attribute.
  - The number of decimal points moved depends on the maximum absolute value of the attribute.

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

where j is the smallest integer such that $\max(|v'|) \leq 1$

- Example : $1, 10, 100, 1000 \rightarrow \frac{1}{10^3}, \frac{10}{10^3}, \frac{100}{10^3}, \frac{1000}{10^3}$ Here $j = 3$

Thank You!
Questions?