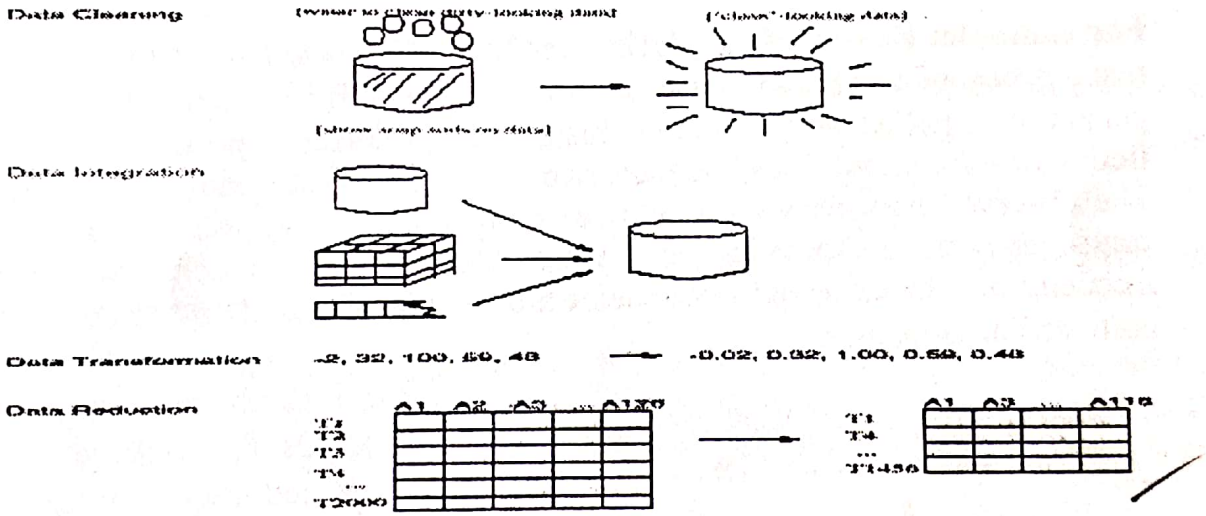
UNIT-II

Data Pre-processing: Data Preprocessing: An Overview of Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization

2.1. Data Preprocessing: The real-world databases are highly susceptible to noisy, missing data due to their huge size and merging the different data from multiple sources. These data need to process for usage. To do, different data processing techniques were developed.

Data Preprocess Techniques:

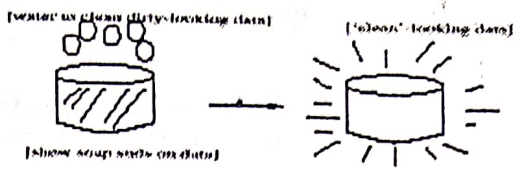
- **Data Cleaning:** It can be applied to remove noise and correct the inconsistence data in the database.
- **Data integration:** It merges the data from multiple sources into coherent data store such a Data Warehouse.
- **Data Transformation:** Normalize the data for accuracy. It means, transfer the data from one format into required format.
- **Data reduction:** It reduces the data size by aggregating, eliminating redundant features, or clustering for instance.



Need for preprocessing: Incomplete, noisy and inconsistent data are common place properties of large real world databases and data warehouses.

- Incomplete data can occur for a number of reasons. For example,
  - ✓ Customer information for sales transaction data may not be available all attributes.
  - ✓ Relevant data may not be recorded due to a misunderstanding.
  - ✓ Missing data for some attributes.
  - ✓ Modification to the data may have been overlooked.
  - ✓ Data that were inconsistent with other recorded data may have been deleted.
- Noisy data can occur for a number of reasons.
  - ✓ The data collection instruments used may be faulty.
  - ✓ Data entry errors occurring by human or some data errors are occurring by computer.
  - ✓ Errors in data transmission can also occur.
- Inconsistent data can occur only when file processing is in usage.

2.2.Data Cleaning: Real-world data can be incomplete, noisy, and inconsistent. Data cleaning routines (algorithms) to fill missing values, noisy and inconsistent data.



2.2.1. Missing Values: Many tuples have no value for several attributes. For example: not entered customer income. This can be filled as the missing value. To remove the missing values, follow below methods.

- *Ignore the tuple:* In data mining, tuple contains several attributes with missing values. It is poor performance for data mining. So avoid the missing values problem.
- *Fill in the missing values manually:* This approach is time-consuming and may not be feasible for large data set with missing values.
- *Use a global constant to fill in the missing value:* Replace all missing attribute values by the same constant such as a label like "*unknown*" or ∞
- *Use the attribute to fill the missing value:* Replace the missing values based on attributes by computer command. For example, "missing the income of customers" as empty. This empty income will be replaced by a given data i.e. $28000.
- *Use the decision trees to fill the missing value:* By using decision tree algorithms, filling the missing values in the attributes.

2.2.2. What is Noisy Data? How to Handle Noisy Data?

Noisy is a random error or variance in a measured variable. Noise is removed by using data *smoothing* techniques. The smoothing techniques are Binning, clustering, combined computer, and regression.

- **Binning method:** first sort data and partition into (equi-depth) bins then one can
  - smooth by bin means
  - smooth by bin median
  - smooth by bin boundaries
- **Clustering:** Detect and remove outliers
- **Combined computer and human inspection:** Detect suspicious values and check by human.
- **Regression:** smooth by fitting the data into regression functions

1. **Binning method:** It can smooth sorted data values by consulting its "*neighborhood*" data. The sorted values are distributed into number of "buckets" or "bins". Because binning methods consult the neighborhood of values, they perform local smoothing. For example, sorted for price in dollars is shown in bin method. **For example, Sorted data for price : 3,7,14,19,23,24,31,33,38**

Step 1: *Partition into bins*: (The sorted data of price is partitioned into equal parts)

Bin 1: 3,7,14 **Bin 2:** 19,23,24 **Bin 3:** 31,33,38

Step 2: *Smoothing by bin means:* In this step each value in a bin is replaced by ***mean*** value of the bin.

For example, the ***mean*** of the values 3, 7, & 14 in bin 1 is 8 i.e.[(3+7+14)] /3=8]

Bin 1: 8,8,8 **Bin 2:** 22,22,22 **Bin 3:** 34,34,34

Step 3: *Smoothing by bin boundaries:* In smoothing by bin boundaries, the maximum & minimum values in give bin or identify as the bin boundaries. Each bin value is then replaced by the closest boundary value. In general, the larger the interval range of values in each bin is constant.

Bin 1: 3, 3, 14 **Bin 2:** 19, 24, 24 **Bin 3:** 31, 31, 38

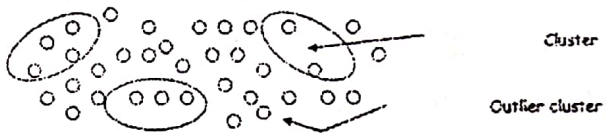
- Consider the another following example: **Sorted data for price (in dollars):**
  4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* 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 (4 + 8 + 9 + 15) /4 = 9
- Bin 2: 23, 23, 23, 23 (21+ 21+ 24 + 25)/4 = 23
- Bin 3: 29, 29, 29, 29 (26 + 28 + 29 + 34)/4 = 29

* Smoothing by bin boundaries:
- Bin 1: 4, 4, 4, 15 (minimum value is 4 and maximum value is 15)
- Bin 2: 21, 21, 25, 25 (minimum value is 21 and maximum value is 25)
- Bin 3: 26, 26, 26, 34 (minimum value is 26 and maximum value is 34)

2. **Clustering:** A cluster can detect the similar values which formed as group. The outlier group values are specified outside of the cluster, shown in figure.



3. **Regression:** Data can be smoothed by fitting the data to a function is called regression. *Linear regression* involves finding the "*best*" line to fit two variables. So that one variable can be used to predict the other.

4. **Inconsistent Data:** Inconsistent data occur due to during data entry, functional dependencies b/w attributes and missing values. The Inconsistent data can be detected and corrected either by manually or knowledge engineering tools.

2.2.3. Data Cleaning as a Process:

1. Data inconsistency detection: For data consistency, follow the below steps.

- Check metadata of database.
- Check field overloading
- Check uniqueness rule, repeated rule and null rule
- Use commercial tools
  - Data scrubbing (cleaning): For example, (postal code, spell-check) to detect errors and make corrections.
  - Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers).
- Data migration and integration:
  - Data migration tools: allow transformations to be specified

- ETL (Extraction/Transformation/Loading) tools: allow users to spec transformations through a graphical user interface
- Integration of the two processes:
  - Iterative and interactive (e.g., Potter's Wheels)

2.3. Data Integration: In data mining, merging the data from multiple sources and store it in Data warehousing. The multiple sources may be multiple databases, data cubes, or flat files.

There are number of issues to consider during data integration process. They are

2.3.1. Schema integration and object matching: In schema integration, multiple data sources must be matched. In this case *entity identification problem will be raised.* For example, observe that the *customer_id* in one database and *customer_id* in another database must be same entity. Otherwise "entity identification problem will be raised".

2.3.2. Redundancy : It is another important issue. It redundant the attributes when preparing annual report from multiple quarters sales of AllElectronics company.

1. **Correlation Analysis:** Given two attributes can measure how strongly one attribute implies the other, based on the available data. For nominal data, we use the x^2 *(chi-square)* test. For numeric attributes, we can use the *correlation coefficient* and *covariance*, both of which access how one attribute's values vary from those of another.

For example: ExampleCorrelation analysis of nominal attributes using x^2. Suppose that a group of 1500 people was surveyed. The gender of each person was noted. Each person was polled as to whether his or her preferred type of reading material was fiction or nonfiction. Thus, we have two attributes, *gender* and *preferred reading*.
The observed frequency (or count) of each possible joint event is summarized in the contingency table shown in Table, where the numbers in parentheses are the expected frequencies. The expected frequencies are calculated based on the data distribution for both attributes using Eq.

$$e_{11} = \frac{count(male) \times count(fiction)}{n} = \frac{300 \times 450}{1500} = 90,$$

Using Eq., we can verify the expected frequencies for each cell. For example, the expected frequency for the cell (*male, fiction*) is and so on.

Example 2.1's 2 × 2 Contingency Table Data

	male	female	Total
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

Using Eq. (3.1) for χ^2 computation, we get

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$
$$= 284.44 + 121.90 + 71.11 + 30.48 = 507.93.$$

2. *Note:* Are *gender* and *preferred_reading* correlated?
3.

Covariance (Numeric Data): In probability theory and statistics, correlation and covariance are two similar measures for assessing how much two attributes change together.

Consider two numeric attributes A and B, and a set of n observations $\{(a1,b1), \ldots (an,bn)\}$. The mean values of A and B, respectively, are also known as the **expected values** on A and B, that is,

$$E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n} \qquad E(B) = \bar{B} = \frac{\sum_{i=1}^{n} b_i}{n}.$$

The covariance between A and B is defined as

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}.$$

Table Stock Prices for *AllElectronics* and *HighTech*

Time point	AllElectronics	HighTech
t1	6	20
t2	5	10
t3	4	14
t4	3	5
t5	2	5

$$E(AllElectronics) = \frac{6+5+4+3+2}{5} = \frac{20}{5} = \$4$$

$$E(HighTech) = \frac{20+10+14+5+5}{5} = \frac{54}{5} = \$10.80.$$

Thus, using Eq. we compute

$$Cov(AllElectroncis, HighTech) = \frac{6\times 20+5\times 10+4\times 14+3\times 5+2\times 5}{5} - 4\times 10.80$$

$$= 50.2 - 43.2 = 7.$$

Therefore, given the positive covariance we can say that stock prices for both companies rise together.

3.3.3. *Detection and resolution of data value conflict*:

a) For example, given two attributes can measure how strongly one attribute implies the other based on the available data.

a) For example, for the same real-world entity, attribute values from different sources may differ. This may be due to differences in representation, scaling, or encoding. For instance, a *weight* attribute may be stored in metric units in one system and British imperial units in another.

This is done by using correlation method.

The correlation b/w attributes A and B can measure by the formula (**Correlation Coefficient**)

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{(n-1)\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_ib_i)-n\bar{A}\bar{B}}{(n-1)\sigma_A\sigma_B}$$

where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B, and $\Sigma(a_ib_i)$ is the sum of the AB cross-product.

- ✓ If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.
- ✓ $r_{A,B} = 0$: independent; $r_{AB} < 0$: negatively correlated

2.4. Data Reduction: Data reduction techniques can be applied to obtain a reduce representation of the data set that is much smaller in volume, and closely maintain the integrity of the original data. That is, mining on the reduced data set should be more efficient to produce the same results.

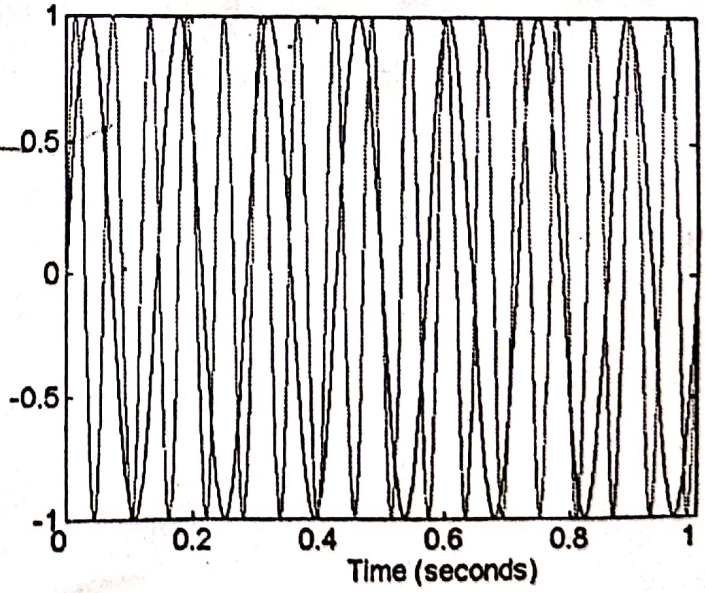
■ Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set. For example, to know one student result, it needs to extract from the entire result.

2.4.1. Data Reduction Strategies: Data reduction strategies include *dimensionality reduction*, *numerosity reduction*, and *data compression*.

1. **Dimensionality reduction** is the process of reducing the number of random variables or attributes under consideration. Dimensionality reduction methods include *wavelet transforms* and *principal components analysis*, which transform or project the original data onto a smaller space.
2. **Numerosity reduction** techniques replace the original data volume by alternative, smaller forms of data representation. These techniques may be parametric or nonparametric.
   - ✓ For ***parametric methods***, a model is used to estimate the data, so that typically only the data parameters need to be stored, instead of the actual data. (Outliers may also be stored.) Regression and log-linear models are examples.
   - ✓ ***Nonparametric methods*** for storing reduced representations of the data include *histograms*, *clustering*, *sampling*, and *data cube aggregation*.
3. **Data Compression:** In **data compression**, transformations are applied so as to obtain a reduced or "compressed" representation of the original data. If the original data can be *reconstructed* from the compressed data without any information loss, the data reduction is called **lossless**. If, instead, we can reconstruct only an approximation of the original data, then the data reduction is called **lossy**.

2.4.2. Wavelet Transforms:

The **discrete wavelet transform (DWT)** is a linear signal processing technique that, when applied to a data vector *X*, transforms it to a numerically different vector, *X'*, of **wavelet coefficients**. The two vectors are of the same length. When applying this technique to data reduction, we consider each tuple as an *n*-dimensional data vector, that is, $X = (x1, x2, ..., xn)$, depicting *n* measurements made on the tuple from *n* database attributes.



For example, in cricket play, the two team play is shown in Wavelet Transform. In this case the reduction technique can select only one based on the success of the team.

"How can this technique be useful for data reduction if the wavelet transformed data are of the same length as the original data?"

A discrete wavelet transform uses a hierarchical *pyramid algorithm* that divide the data at each iteration, resulting in fast computational speed. The method is as follows:

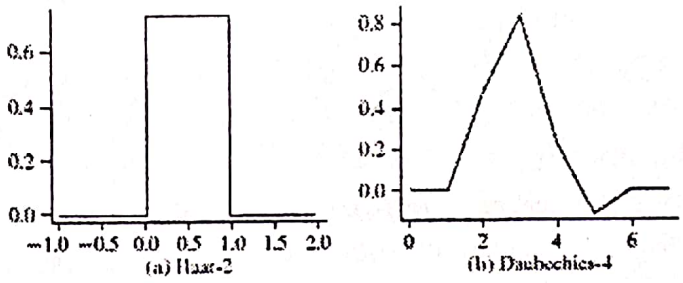
1. The length, L, of the input data vector must be an integer power of 2. This condition can be met by padding the data vector with zeros as necessary ($L >= n$).

2. Each transform involves applying two functions. The first applies some data smoothing, such as a sum or weighted average. The second performs a weighted difference, which acts to bring out the detailed features of the data.

3. The two functions are applied to pairs of data points in X, that is, to all pairs of measurements (x_{2i} ,x_{2i} +1). This results in two data sets of length $L/2$. In general, these represent a smoothed or low-frequency version of the input data and the high frequency content of it, respectively.

4. The two functions are recursively applied to the data sets obtained in the previous loop, until the resulting data sets obtained are of length 2.

5. Selected values from the data sets obtained in the previous iterations are designated the wavelet coefficients of the transformed data.



Examples of wavelet families. The number next to a wavelet name is the number of *vanishing moments* of the wavelet. This is a set of mathematical relationships that the coefficients must satisfy and is related to the number of coefficients.

2.4.3. Principal Components Analysis: Given N data vectors from n-dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data.

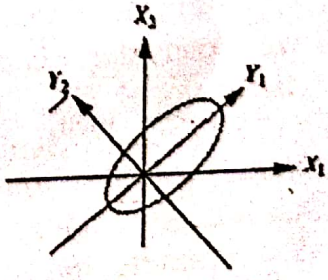
The basic procedure is as follows:

1. The input data are normalized, so that each attribute falls within the same range. This step helps ensure that attributes with large domains will not dominate attributes with smaller domains.

2. PCA computes k orthonormal vectors that provide a basis for the normalized input data. These are unit vectors that each point in a direction perpendicular to the others. These vectors are referred to as the *principal components*. The input data are a linear combination of the principal components.

3. The principal components are sorted in order of decreasing "significance" or strength. That is, the sorted axes are such that the first axis shows the most variance among the data, the second axis shows the next highest variance, and so on.

For example, *Figure* shows the first two principal components, $Y1$ and $Y2$, for the given set of data originally mapped to the axes $X1$ and $X2$. This information helps identify groups or patterns within the data.



4. Because the components are sorted in decreasing order of "significance," the data size can be reduced by eliminating the weaker components, that is, those with low variance.

2.4.4. Attribute Subset Selection: Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant to the mining task or redundant.

For example, if the task is to classify customers based on whether or not they are likely to purchase a popular new CD at *AllElectronics* of a sale, attributes such as the customer's telephone number are likely to be irrelevant, unlike attributes such as *age* or *music taste*.

This can be a difficult and time consuming task, especially when the data's behavior is not well known. Therefore, it is causing confusion for the mining algorithm. This can result in discovered patterns of poor quality and redundant attributes can slow down the mining process.

Other **Attribute subset selection:**

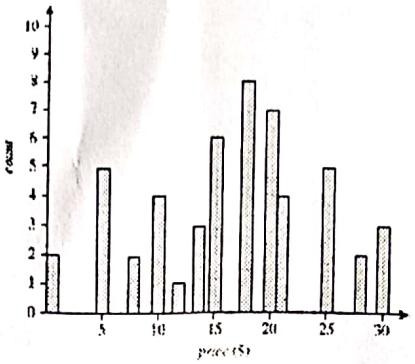
1. ***Stepwise forward selection:*** The procedure starts with an empty set of attributes. The best of the original attributes is determined and added to the set.
2. ***Stepwise backward elimination:*** The procedure starts with the full set of attributes. At each step it removes the worst attribute in the remaining set.
3. ***Combination of forward selection and backward elimination:*** The stepwise forward selection and backward elimination methods can be combined to produce the best attribute and removes the worst from among the remaining attributes.
4. **Decision tree induction**: Decision tree algorithms (e.g., ID3, C4.5, and CART) were originally intended for classification.

2.4.5. Regression and Log-Linear Models:

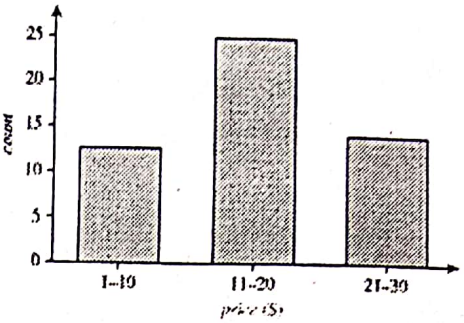
- ***Regression and Long-linear models:*** This model is used to approximate the given data. In this data are modeled to fit a straight line. For example, a random variable Y can be modeled as a linear function of another random variable X with equation $Y = \alpha + \beta X$.
- Linear regression: $Y = w X + b$
  - Two regression coefficients, w and b, specify the line and are to be estimated by using the data at hand
  - Using the least squares criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2, \ldots$.
- Multiple regression: $Y = b_0 + b_1 X_1 + b_2 X_2$
  - Many nonlinear functions can be transformed into the above
- Log-linear models:
  - Approximate discrete multidimensional probability distributions.
  - Estimate the probability of each point (tuple) in a multi-dimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations Useful for dimensionality reduction and data smoothing.

Histogram: An attribute data a partitions the data distribution of a into disjoint subsets, or buckets. If each bucket represents only a single attribute-value, the buckets are called *singleton buckets*.

For example, the following data are best of prices of commonly sold items at AllElectronics. The prices have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 15, 15, 15, 15, 15, 20, 20, 20, 20, 20, 20, 20, 25, 25, 25, 25, 30, 30, 30.



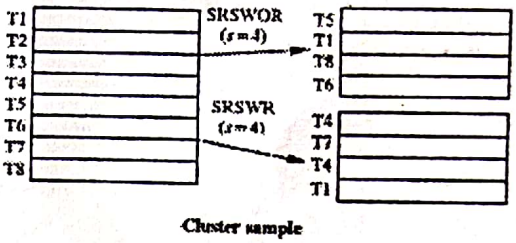
Before reduction of data



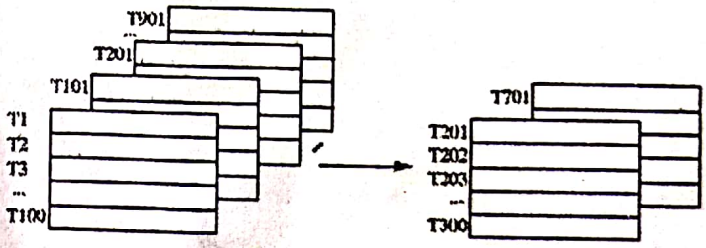
After reduction the data

Clustering: Clustering techniques consider data tuples as objects. They **partition the objects into groups or clusters.** So, the objects within a cluster are "similar" to one another and "dissimilar" to objects in the other clusters. In data reduction, the cluster representations of the data are used to replace the actual data. It is much more effective for data that can be organized into distinct clusters then for smeared data.

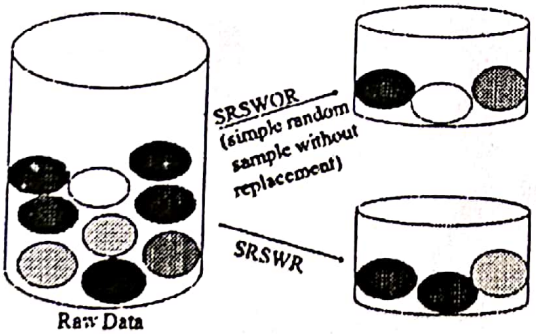
Sampling: Sampling can be used as a data reduction technique because it allows a large data set to be represented by a much smaller random sample of the data. Suppose the large set of D is divided into N possible samples. This is shown figure in.



Cluster sample



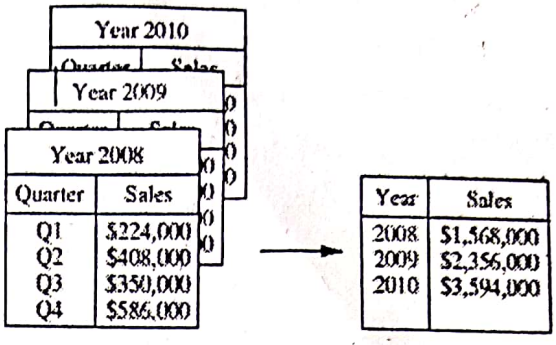
Stratified sample (according to age)



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

Data cube aggregation: Applying the aggregation operation on the construction of a data cube. For example, the data of AllElectronics sales per quarter for the year 1997 to 1999 is aggregate into one report, shown in figure.



Sales data for a given branch of *AllElectronics* for the years 2008 through 2010. On the *left*, the sales are shown per quarter. On the *right*, the data are aggregated to provide the annual sales.

2.5. Transformation:

In Data Transformation, the data are transformed or consolidated into forms for mining. Data Transformation can involve the following.

- ***Smoothing***: It removes the noise from data by using the techniques such as *binning, clustering,* and *regression.*
- ***Aggregation:*** Aggregation means summary of multiple data. For example, the daily or quarterly sales data may be aggregated as monthly or annual report.
- ***Generalization:*** Generalization means, *low level* or *"primitive"* (raw) data are replaced by *high-level* concept with the use of hierarchies.
- ***Normalization:*** Normalization means create the unstructured data as well structured data by using normalization methods. For example, convert the data from -1.0 to 1.0, 0.0 to 1.0 and so on. The normalization methods are

  1. min-max normalization
  2. z-score normalization and
  3. Normalization by decimal scaling.

1. **min-max normalization:** It performs a linear transformation on the original data. Suppose that $\min_A$ and $\max_A$ are the minimum and maximum values of an attribute A. These can be computed by using the formula.

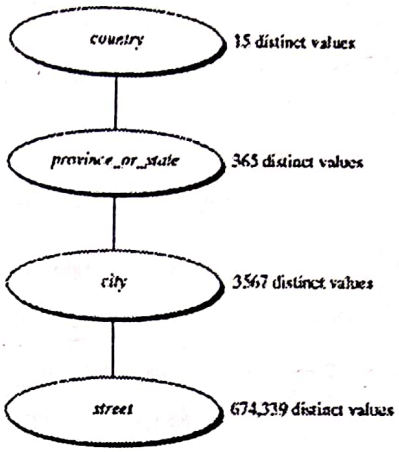
$$v = \frac{v - \min_A}{\max_A - \min_A} (new_max_A - new_min_A) + new_min_A$$

- Example: Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,000 is mapped to $\frac{73,600-12,000}{98,000-12,000}(1.0-0)+0=0.716$

2. **z-score normalization:** In this method, the values for an attribute A are normalized based on the *mean* and *standard deviation* of A. Value of V or A is normalized to V by computing. $v' = \frac{v - \mu_A}{\sigma_A}$
   Example: Let $\mu = 54,000$, $\sigma = 16,000$. Then $\frac{73,600-54,000}{16,000} = 1.225$

3. **Normalization by decimal scaling:** Normalizes by moving the decimal point of value of attribute A. The number of decimal points depends on the maximum absolute value of A. A value V of A is normalized to V^1 by computing. $V = V / 1.0^j$ Where j is smallest integer. For example, in the range -986 to 917, the maximum absolute value of A is 986. To normalize by decimal scaling by dividing each value 1000. i.e -0.986 and 0.917.

4. **Concept hierarchy generation for nominal data,** where attributes such as *street* can be generalized to higher-level concepts, like *city* or *country*. Many hierarchies for nominal attributes are implicit within the database schema and can be automatically defined at the schema definition level.



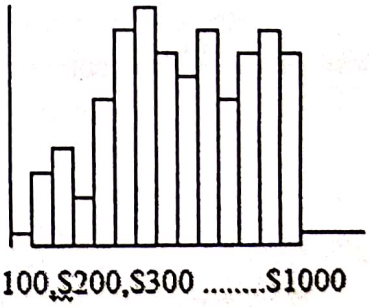
For example, a time dimension in a database may contain 20 distinct years, 12 distinct months, and 7 distinct days of the week. However, this does not suggest that the time hierarchy should be "*year <month < days of the week,*" with *days of the week* at the top of the hierarchy.

2.6.Discretization Technique: It can be used to reduce the number of values for a given continues attribute by dividing the range of attribute into intervals. Interval labels can then be use to replace actual data values.

→ *Features of Data Discretization:*
- → It leads to a concise
- → Easy to use
- → Knowledge-level representation of mining results.

→*Categories of Data Discretization:*
- → Supervised discretization
- → Unsupervised discretization or splitting histogram
- → Top-down discretization or splitting. Binning
- → Bottom-up discretization or merging.

→**Concept hierarchy Methods:** A concept hierarchies for numeric attribute can be constructed automatically based on data distribution analysis. It has five numeric concept hierarchy generations, namely *binning, histogram analysis, cluster analysis, entropy-based discretization, and data segmentation by 'natural partitioning'.*



1.*Binning*: Binning is a top-down splitting technique based on a specified number of bins. Binning method can smooth sorted data value by consulting its "*neighborhood*" data. These methods are also used as discritization method for numeric reduction and concept hierarchy generation. This method can be applied recursively to partition in order to generate the hierarchies.

2.Histogram analysis: Like binning, histogram analysis is an unsupervised discretization technique because it does not use class information. In the figure, a histogram showing the data distribution of the attribute price for a given data set. For example, the most frequent price range is roughly partitioned as $0 -$100, $101 - $200, and so on.

3. Cluster Analysis: Clustering is a method of grouping data into different groups, so that in each group share similar trends and it is concept of hierarchy.

4. Entropy-based discretization: An information-based measure called entropy. It can be used to recursively partition the values of a numeric attribute A, resulting in a hierarchical discretization. Such a discretization forms a numerical concept hierarchy for the attribute.