Module No : 02

Introduction to data mining

Data Pre-processing

Major Tasks involved in Data Preprocessing

- Following are the major steps involved in data preprocessing
  - Data Integration
  - Data Cleaning
  - Data Reduction
  - Data Transformation.

Major Tasks involved in Data Preprocessing

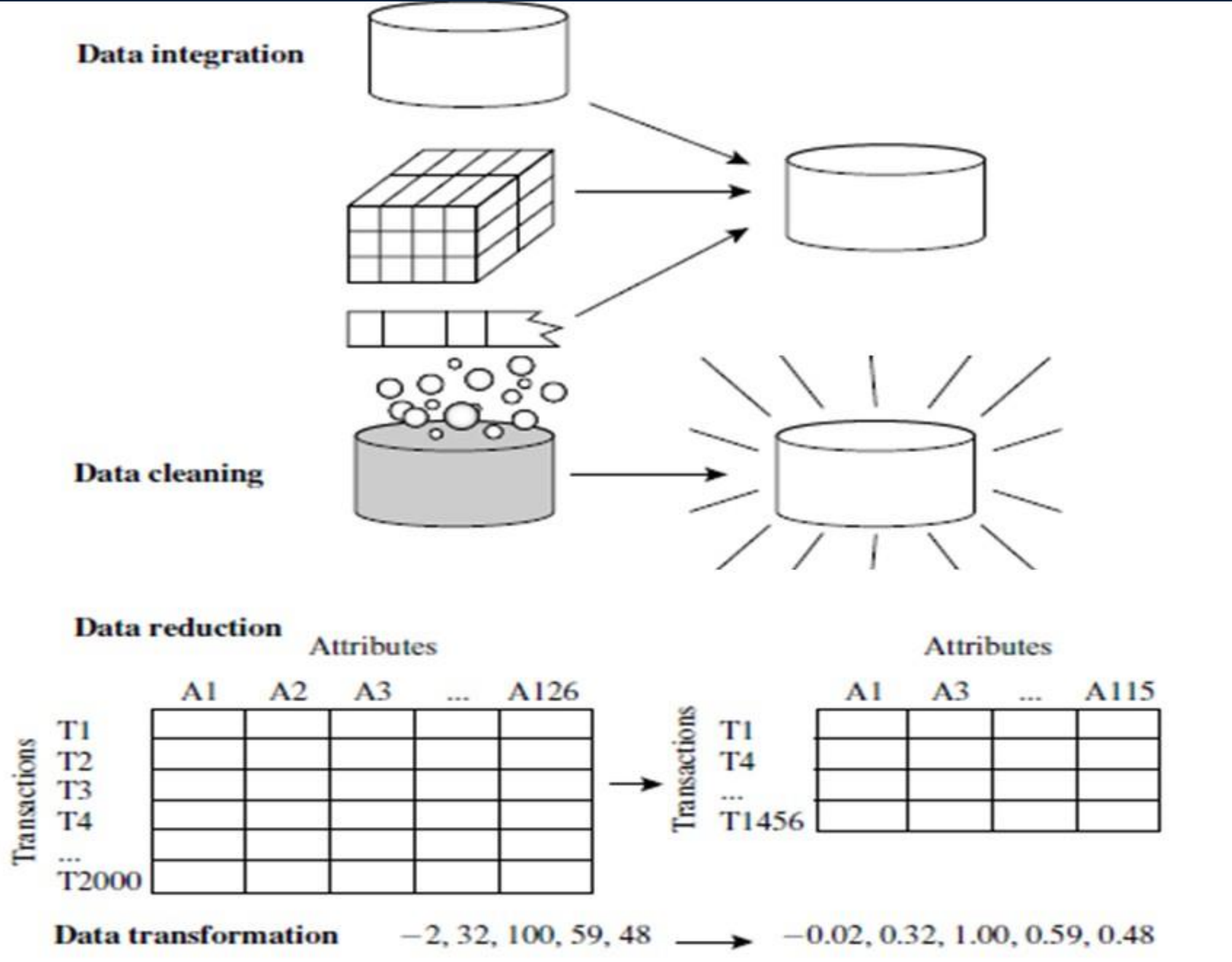
C	1	1
D	3	1
E	1.5	0.5

Q3 A) Define Metadata. Discuss the types of Metadata stored in a data warehouse. Illustrate with an example. [10]

B) Discuss different steps involved in Data Pre-processing [10]

Q4 A) Discuss various OLAP Models and their architecture [10]

B) Define Classification. Discuss the issues in Classification. A simple example from the stock market involving only discrete ranges has profit as categorical attribute, with values { Up, Down} and the training data is: [10]



Forms of data preprocessing.

Data Preprocessing:

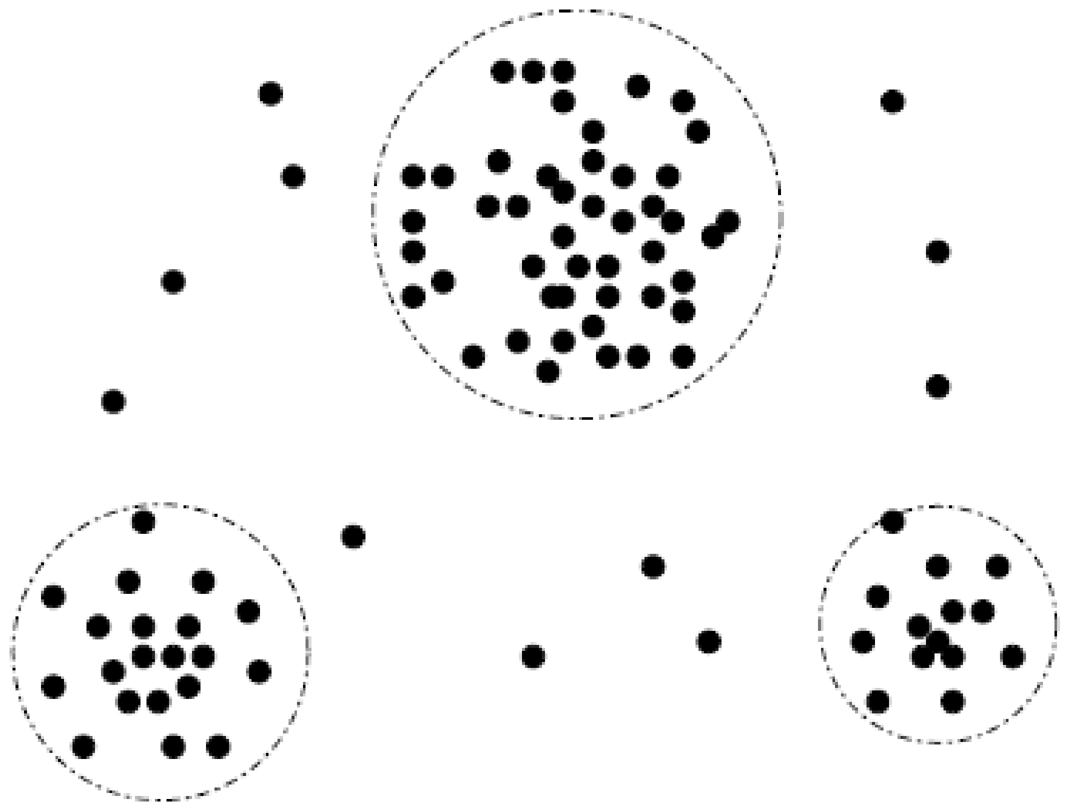
- **Data integration:** this involve integrating multiple databases, data cubes, or files
- combines data from multiple sources to form a coherent data store.
- following concepts contribute to smooth data integration.
  - The resolution of semantic heterogeneity
  - metadata
  - correlation analysis
  - tuple duplication detection
  - data conflict detection

Data Preprocessing: Data Cleaning

- **Data cleaning** routines work to ―clean‖ the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies
- Missing Values
  - Ignore the tuple
  - Fill in the missing value manually
  - Use a global constant to fill in the missing value
  - Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing value
  - Use the most probable value to fill in the missing value
- Noisy Data
- Outlier analysis

Data Preprocessing:

- **Data cleaning:** Outlier analysis



A 2-D customer data plot with respect to customer locations in a city, showing three data clusters. Outliers may be detected as values that fall outside of the cluster sets.

Data Preprocessing: Data Reduction

- **Data reduction** obtains a reduced representation of the data set that is much smaller in volume, yet produces the same (or almost the same) analytical results.
- Data reduction strategies include
  - *data cube aggregation*
  - *dimensionality reduction*
  - *data compression*
  - *numerosity reduction.*

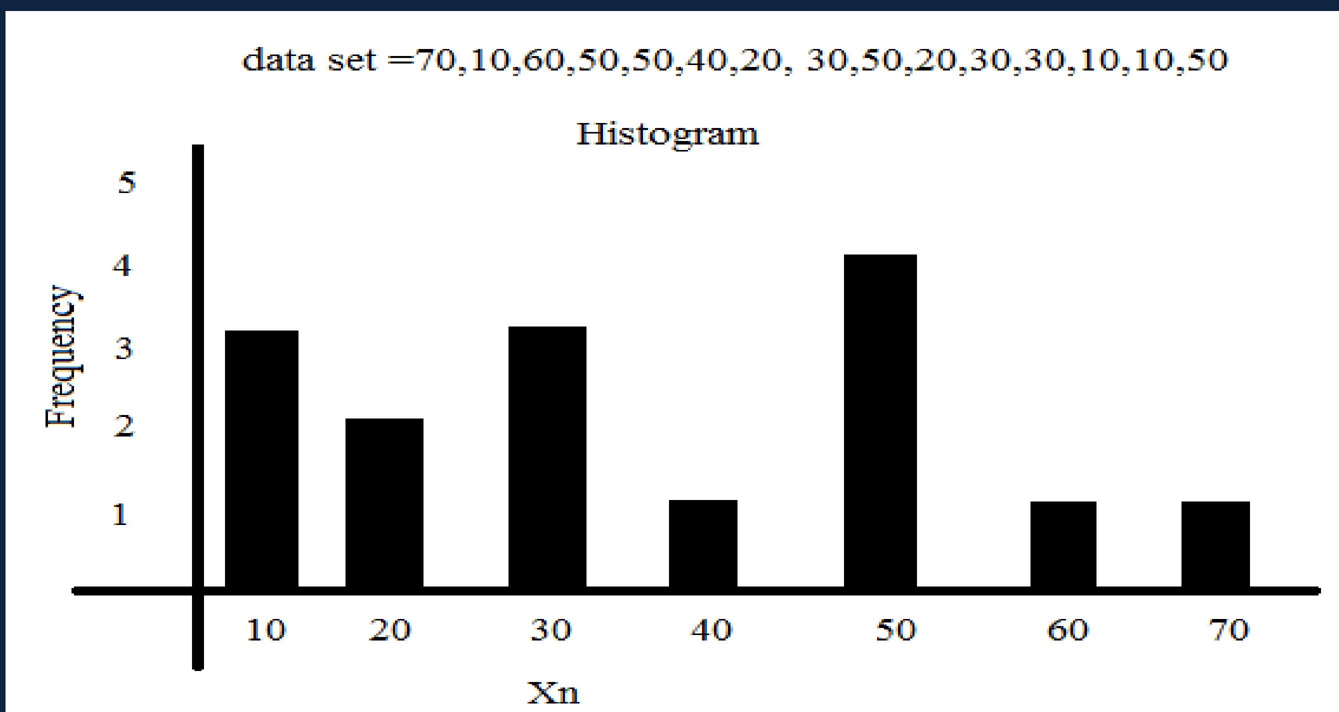
Data Preprocessing: Data Reduction

- Data cube aggregation, where aggregation operations are applied to the data in the construction of data cube.
- In **dimensionality reduction**, data encoding schemes are applied so as to obtain a reduced or ―compressed‖ representation of the original data (e.g., removing irrelevant attributes)
- Data compression, where encoding mechanisms are used to reduce the data set size
- In **numerosity reduction**, the data are replaced by alternative, smaller representations using *histograms*, *clusters*, *sampling*, or *data aggregation*

Data reduction: Histograms

- ―Histos‖ means pole, and ―gram‖ means chart, so a histogram is a chart of poles
- Plotting histograms is a graphical method for summarizing the distribution of a given attribute, X

data set =70,10,60,50,50,40,20, 30,50,20,30,30,10,10,50



Data Preprocessing: Data reduction

Clustering and sampling

- Clustering techniques consider data tuples as objects. They partition the objects into groups, or *clusters*, so that objects within a cluster are —similar‖ to one another and —dissimilar‖ to objects in other clusters.

- Sampling can be used as a data reduction technique because it allows a large data set to be represented by a much smaller random data sample (or subset)

Data Preprocessing: Data reduction

Attribute subset selection

- *Attribute subset selection* is a method of dimensionality reduction in which irrelevant, weakly relevant, or redundant attributes or dimensions are detected and removed
- It reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

Data transformation: **Normalization of data**

- Normalization is used to scale values so they fit in a specific range (adjusting the value range is important when dealing with attributes of different units and scales)
- E.g. when using the Euclidian distance all attributes should have the same scale for a fair comparison
- An attribute is normalized by scaling its value so that they fall within a small specific range such as 0.0 to 1.0
- Normalization is particularly useful for classification algorithms
- Methods for data normalization
  1. Min-max Normalization
  2. Z-score Normalization
  3. Decimal scaling

Data transformation: Normalization of data

1. Min-max Normalization:

Min-max normalization 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. Min-max normalization maps a value, v_i, of A to v_i' in the range $[new\_min_A, new\_max_A]$ by computing

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A.$$

Example

Suppose that the minimum and maximum values for the attribute *income* are \$12,000 and \$98,000, respectively

We would like to map *income* to the range $[0.0, 1.0]$. By min-max normalization, a value of \$73,600 for *income* is transformed to

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

☑ Data transformation: Normalization of data

1. Min-max Normalization:

Question:

The minimum and maximum values for the attribute *age* are 18 and 60 respectively, transform an age 41 to the range [0.0, 1.0] by using min-max normalization

Min-max normalization 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. Min-max normalization maps a value, v_i, of A to v_i' in the range $[new\_min_A, new\_max_A]$ by computing

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A.$$

Data transformation: Normalization of data

2. Z-Score Normalization: (zero-mean normalization)

In **z-score normalization** (or *zero-mean normalization*), the values for an attribute, A, are normalized based on the mean (i.e., average) and standard deviation of A. A value, v_i, of A is normalized to v_i' by computing

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A},$$

where $\bar{A}$ and σ_A are the mean and standard deviation, respectively, of attribute A. The mean and standard deviation were discussed in Section 2.2, where $\bar{A} = \frac{1}{n}(v_1 + v_2 + \cdots + v_n)$ and σ_A is computed as the square root of the variance of A

Data transformation: **Normalization of data**

2. Z-Score Normalization: (zero-mean normalization)

Example

Suppose that the mean and standard deviation of the values for the attribute *income* are \$54,000 and \$16,000, respectively.

With z-score normalization, a value of \$73,600 for *income* is transformed to

$$\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225.$$

Data transformation: Normalization of data

2. Z-Score Normalization: (zero-mean normalization)

Question:

Normalize the *age value 21* using z-score normalization for following observations of attribute *age*

Age
17
15
23
7
9
13

mean=14

standard deviation=5.76

ans=1.21

Data transformation: Normalization of data

3. Decimal Scaling:

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_i, of A is normalized to v'_i by computing

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

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

Example

Suppose that the recorded values of A range from -986 to 917. The maximum absolute value of A is 986. To normalize by decimal scaling,

we therefore divide each value by 1000 (i.e., $j = 3$) so that -986 normalizes to -0.986 and 917 normalizes to 0.917.

☑ **Data transformation: Normalization of data**

3. Decimal Scaling:

Question:

Recorded values of an attribute temperature are in the range -30 to 45

Normalize it by using decimal scaling

Data transformation: Binning

- Data grouped together into bins
- Data binning or bucketing is a data preprocessing technique used to reduce the effect of minor observation errors
- Statistical data binning is a way to group a number of more/less continuous values into a smaller number of bins
- E.g. 1. if you have data about group of people you may arrange their ages into a smaller number of age intervals
- E.g. 2. Histograms are an example of data binning used in order to observe underlying distributions

Data transformation: Binning

- Binning methods smooth a sorted data value by consulting its ‗neighbourhood‘ (i.e. values around it)
- The sorted values are distributed into number of buckets or bins
- e.g. sorted data for price partitioned into depth 3

4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into equidepth bins:

(4-15) Bin 1 : 4, 8, 15

(16-24) Bin 2 : 21, 22, 24

(25-34) Bin 3: 25, 28, 34

☑ Data transformation: Binning

•Partition into equidepth bins:

(4-15)   Bin 1 : 4, 8, 15

(16-24) Bin 2 : 21, 22, 24

(25-34) Bin 3: 25, 28, 34

- Smoothing by bin means:

Bin 1: 9,9,9

Bin 2: 22, 22, 22

Bin 3: 29,29,29

- Smoothing by bin boundaries:

Bin 1: 4,4,15

Bin 2: 21, 21, 24

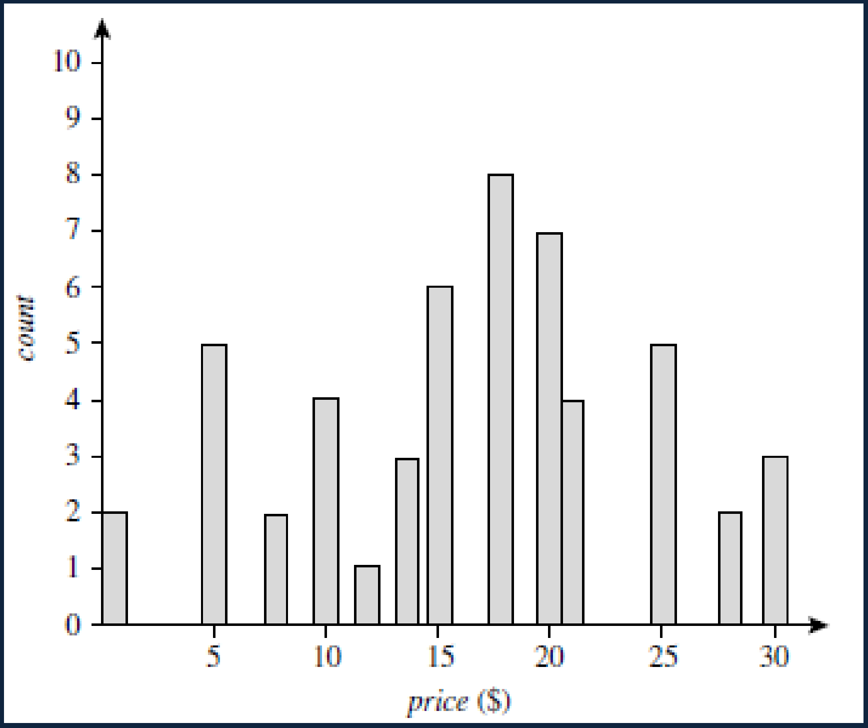
Bin 3: 25, 25, 34

Data Discretization

- Data discretization methods used to reduce the number of values for a given continuous attributes by dividing the range of the attribute into intervals.
- where the raw values of a numeric attribute (e.g., age) are replaced by interval labels (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., youth, adult, senior).
- Such methods can be used to automatically generate concept hierarchies for the data, which allows for mining at multiple levels of granularity.
  - Histogram analysis
  - Concept hierarchy generation

Data discretization: Histogram analysis

- A frequency distribution shows how often each different value in a set of data occurs.
- A **histogram** is the most commonly used graph to show frequency distributions. It looks very much like a bar chart



Data discretization: Concept hierarchy generation

- The concept hierarchies can be used to transform the data into multiple levels of granularity
- four methods for the generation of concept hierarchies for nominal data

1.Specification of a partial ordering of attributes explicitly at the schema level by users or experts

2. Specification of a portion of a hierarchy by explicit data grouping

3. Specification of a set of attributes, but not of their partial ordering

4. Specification of only a partial set of attributes

Data discretization: Concept hierarchy generation

1. Specification of a partial ordering of attributes explicitly at the schema level by users or experts

- user or expert can easily define a concept hierarchy by specifying a partial or total ordering of the attributes at the schema level.
- e.g. location dimension may contain the attributes (specifying the ordering) such as street < city < state < country

2. Specification of a portion of a hierarchy by explicit data grouping

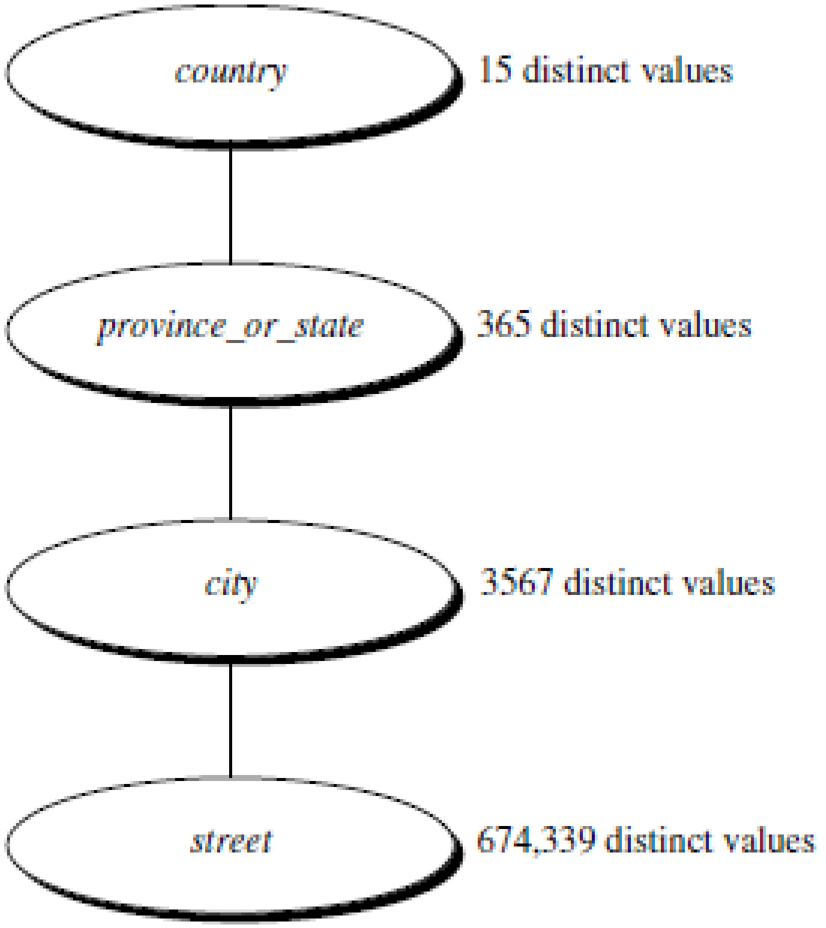
- a user could define some intermediate levels manually
- E.g. {ABC road, Hyderabad, A.P, India} subset of South India
- {XYZ road, Amritsar, Punjab, India} subset of North India

☑ **Data discretization: Concept hierarchy generation**

3. Specification of a set of attributes, but not of their partial ordering

- The system automatically generate the attribute ordering so as to construct a meaningful concept hierarchy.
- a concept hierarchy can be automatically generated based on the number of distinct values per attribute in the given attribute set.
- The attribute with the most distinct values is placed at the lowest hierarchy level.
- The lower the number of distinct values an attribute has, the higher it is in the generated concept hierarchy.
- E.g. see the diagram on next slide

Data discretization: Concept hierarchy generation



Automatic generation of a schema concept hierarchy based on the number of distinct attribute values.

Data discretization: Concept hierarchy generation

4. Specification of only a partial set of attributes

- Sometimes a user only have a vague idea about what should be included in a hierarchy.
- Consequently, the user may have included only a small subset of the relevant attributes in the hierarchy specification