Lecture 5: Data Preparation

Data Preparation

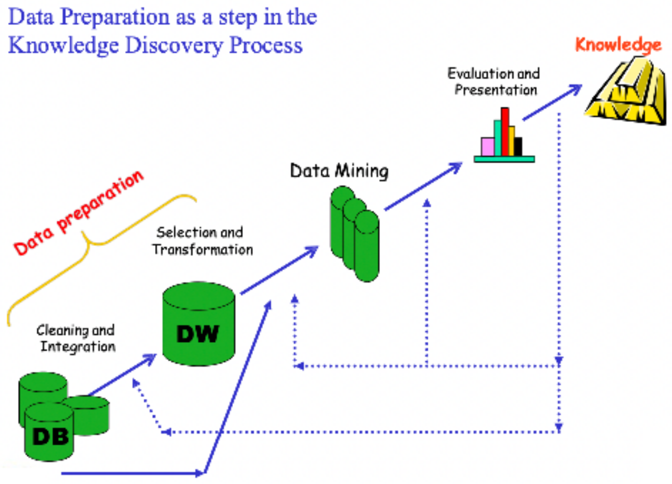
- Introduction to Data Preparation.
- Types of Data.
- Outliers.
- Data Transformation.
- Missing Data.

Why Prepare Data

- Some data preparation is needed for all mining tools.
- The purpose of preparation is to transform datasets so that their information content is best exposed to the mining tool.
- Error prediction rate should be lower (or the same) after the preparation as before it.
- Preparing data also prepares the miner so that when using prepared data the miner produces better models, faster.
- Good data is a prerequisite for producing effective models of any type.
- Data need to be formatted for a given software tool.
- Data need to be made adequate for a given method.
- Data in the real world is dirty.
    - **incomplete**: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data.
        - e.g., occupation="".
    - **noisy**: containing errors or outliers.
        - e.g., Salary = "-10", Age = "222".
    - **inconsistent**: containing discrepancies in codes or names.
        - e.g., Age= "42", Birthday = "03/07/1997".
        - e.g., Was rating "1,2,3", now rating "A, B, C".
        - e.g., discrepancy between duplicate records.

Major Tasks in Data Preparation

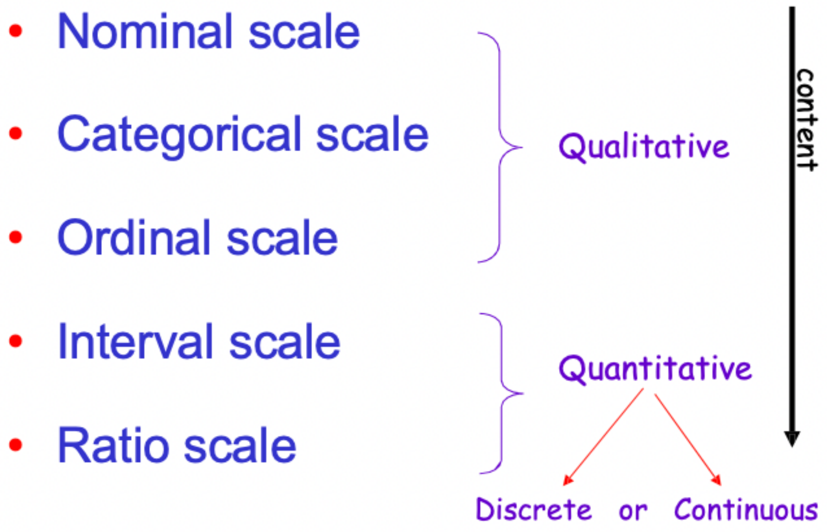
- Data discretization:
    - Part of data reduction but with particular importance, especially for numerical data.
- Data cleaning:
    - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
- Data integration:
    - Integration of multiple databases, data cubes, or files.
- Data transformation:
    - Normalization and aggregation.
- Data reduction:
    - Obtains reduced representation in volume but produces the same or similar analytical results.



Data Preparation as a step in the Knowledge Discovery Process

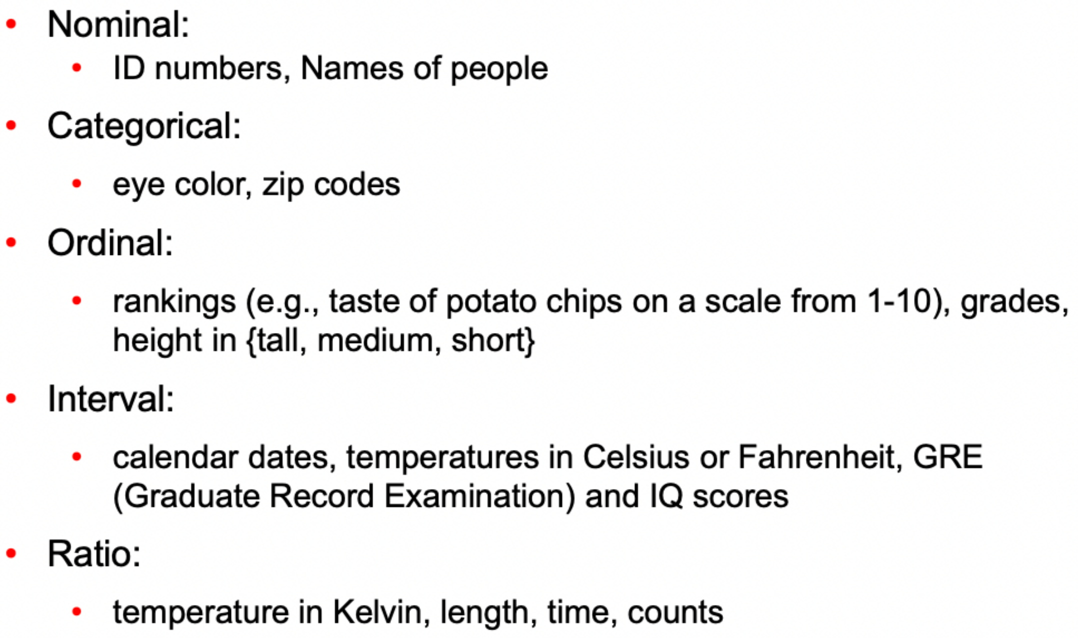
Types of Data

Types of Measurements



- Nominal scale
- Categorical scale
- Ordinal scale
- Interval scale
- Ratio scale

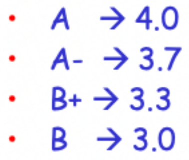
Types of Measurements: Example



- Nominal:
  - ID numbers, Names of people
- Categorical:
  - eye color, zip codes
- Ordinal:
  - rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in {tall, medium, short}
- Interval:
  - calendar dates, temperatures in Celsius or Fahrenheit, GRE (Graduate Record Examination) and IQ scores
- Ratio:
  - temperature in Kelvin, length, time, counts

Data Conversion

- Some tools can deal with nominal values but other need fields to be numeric.
- Convert ordinal fields to numeric to be able to use “>” and “<” comparisons on such fields.



- A → 4.0
- A- → 3.7
- B+ → 3.3
- B → 3.0

- Multi-valued, unordered attributes with small no. of values:
  - E.g., Color=Red, Orange, Yellow, ..., Violet

  - For each value v create a binary “flag” variable C_v , which is 1 if Color=v, 0 otherwise.

Conversion: Nominal, Many Values

- Examples:
  - US State Code (50 values).
  - Profession Code (7,000 values, but only few frequent).
- Ignore ID-like fields whose values are unique for each record.
- For other fields, group values “naturally”:
  - E.g., 50 US States => 3 or 5 regions.
  - Profession – select most frequent ones, group the rest.
- Create binary flag-fields for selected values.

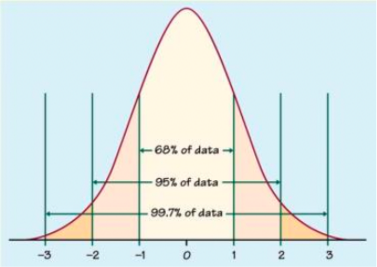
Outliers

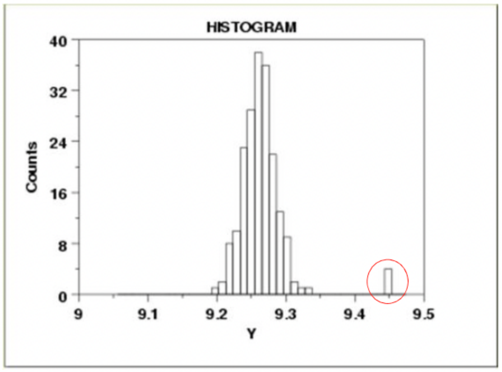
- Outliers are values thought to be out of range.
  - **“An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism”.**
  - Can be detected by standardizing observations and label the standardized values outside a predetermined bound as outliers.
  - Outlier detection can be used for fraud detection or data cleaning.
- Approaches:
  - Do nothing.
  - Enforce upper and lower bounds.
  - Let binning handle the problem.

Outlier Detection

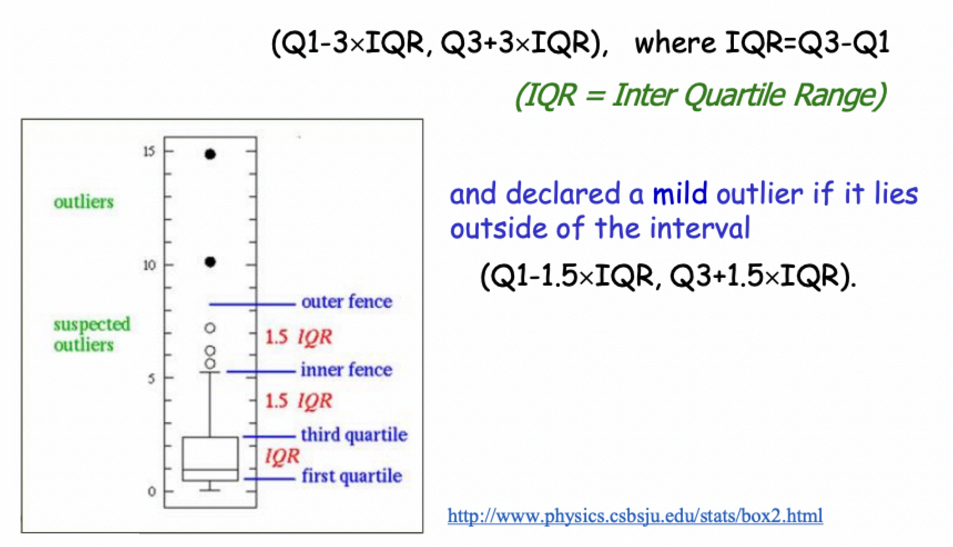
- Univariate:
  - Compute mean and std. deviation. For k=2, k=3, x is an outlier if outside limits (normal distribution assumed).

    $$(\bar{x} - ks, \bar{x} + ks)$$

    - 

    - 

  - Boxplot: An observation in an extreme outlier if:



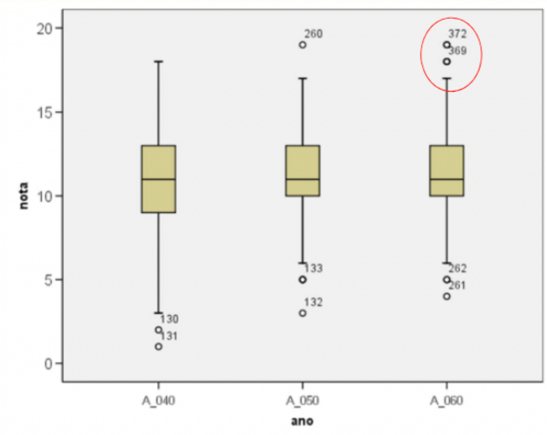
(Q1-3×IQR, Q3+3×IQR),   where IQR=Q3-Q1

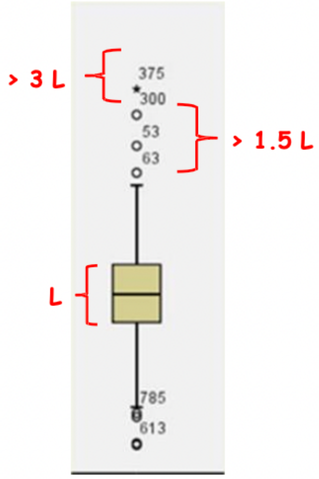
(IQR = Inter Quartile Range)

and declared a **mild** outlier if it lies outside of the interval

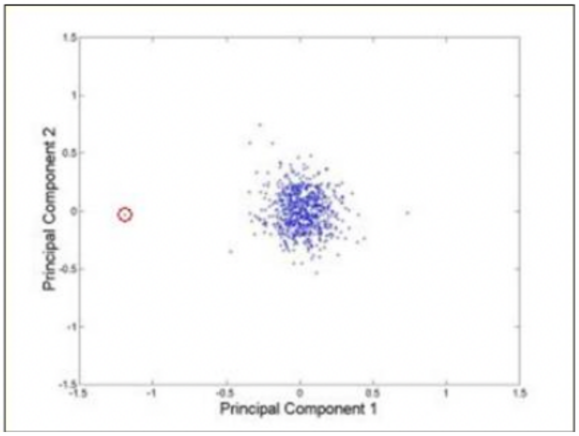
(Q1-1.5×IQR, Q3+1.5×IQR).

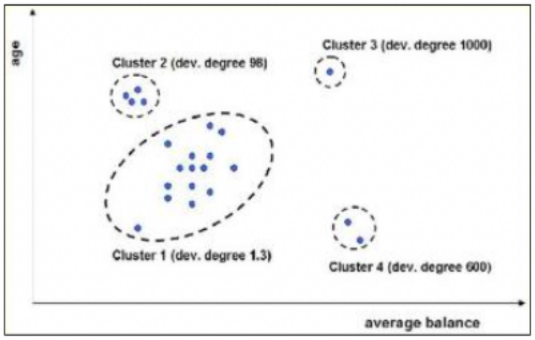
http://www.physics.csbsju.edu/stats/box2.html





- <span>Multivariate</span>:
  - Clustering:
    - Very small clusters are outliers.

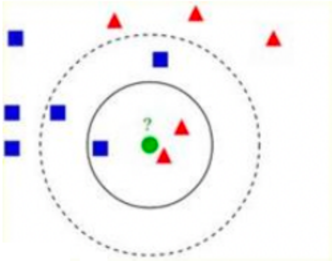


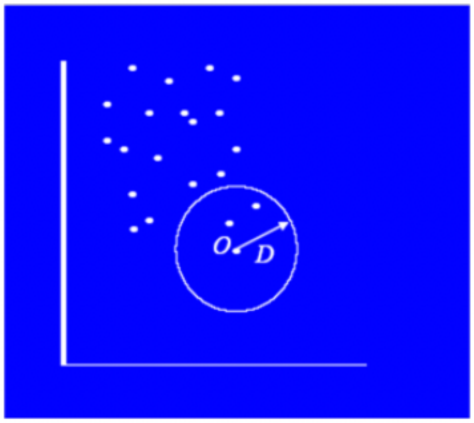


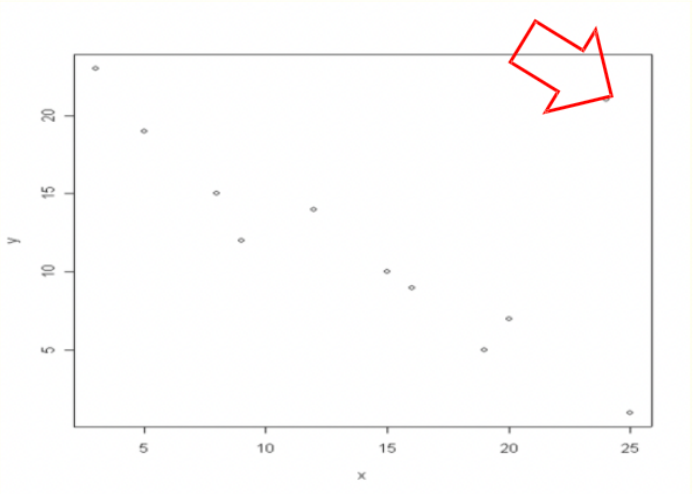
http://www.ibm.com/developerworks/data/library/techarticle/dm-0811wurst/

  - Distance based:
    - An instance with very few neighbors within D is regarded as an outlier.

Knn algorithm







A bi-dimensional outlier that is not an outlier in either of its projections.

Data Transformation

Normalization

- For distance-based methods, normalization helps to prevent that attribute with large ranges out-weight attributes with small ranges.
  - min-max normalization

$$v' = \frac{v - \min_v}{\max_v - \min_v}(new\_max_v - new\_min_v) + new\_min_v$$

  - z-score normalization

$$v' = \frac{v - \bar{v}}{\sigma_v}$$

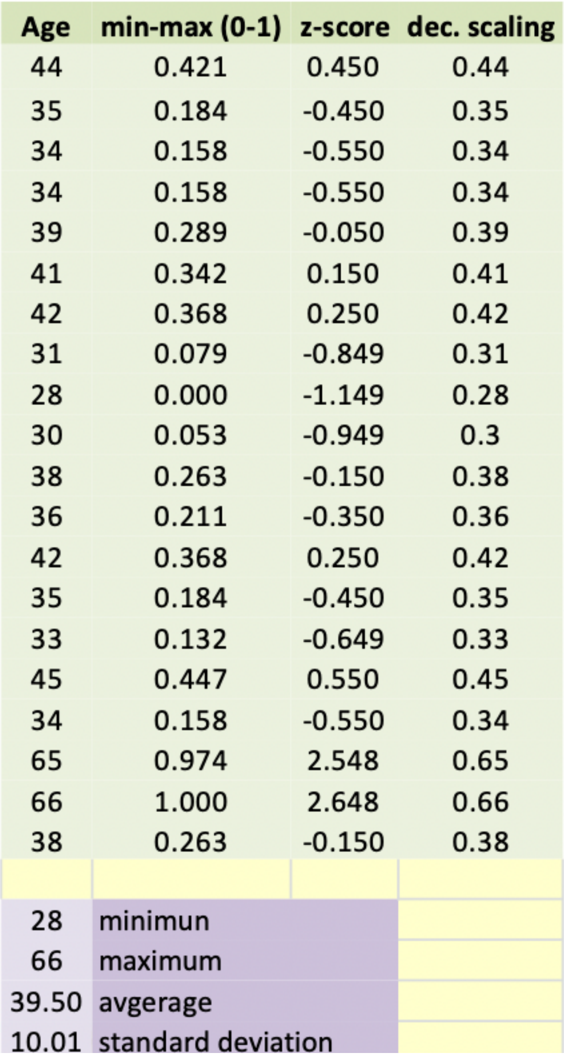
does not eliminate outliers

  - normalization by decimal scaling

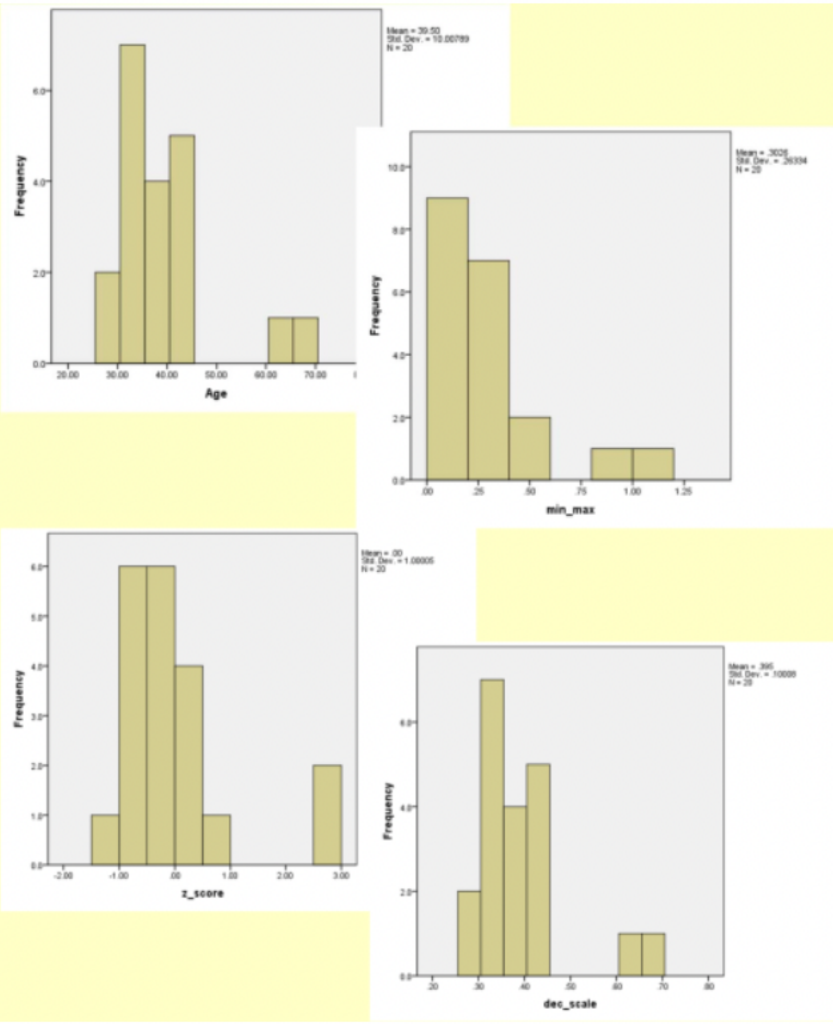
$$v' = \frac{v}{10^j}$$

Where j is the smallest integer such that Max(| v' |)<1

range: -986 to 917 => j=3   -986 -> -0.986   917 -> 0.917



Age	min-max (0-1)	z-score	dec. scaling
44	0.421	0.450	0.44
35	0.184	-0.450	0.35
34	0.158	-0.550	0.34
34	0.158	-0.550	0.34
39	0.289	-0.050	0.39
41	0.342	0.150	0.41
42	0.368	0.250	0.42
31	0.079	-0.849	0.31
28	0.000	-1.149	0.28
30	0.053	-0.949	0.3
38	0.263	-0.150	0.38
36	0.211	-0.350	0.36
42	0.368	0.250	0.42
35	0.184	-0.450	0.35
33	0.132	-0.649	0.33
45	0.447	0.550	0.45
34	0.158	-0.550	0.34
65	0.974	2.548	0.65
66	1.000	2.648	0.66
38	0.263	-0.150	0.38
28	minimun		
66	maximum		
39.50	avgerage		
10.01	standard deviation		



Data Transformation

- It is the process to create new attributes:
  - Often called transforming the attributes or the attribute set.
- Data transformation usually combines the original raw attributes using different mathematical formulas originated in business models or pure mathematical formulas.
- Linear Transformations:
  - Normalizations may not be enough to adapt the data to improve the generated model.
  - Aggregating the information contained in various attributes might be beneficial.
  - If B is an attribute subset of the complete set A, A new attribute Z can be obtained by a linear combination:

$$Z = r_1 B_1 + r_2 B2 + \cdots + r_m B_M$$

- Quadratic Transformations:
  - In Quadratic Transformations, a new attribute is built as follows:

$$Z = r_{1,1} B_1^2 + r_{1,2} B_1 B_2 + \cdots + r_{m-1,m} B_{m-1} B_m + r_{m,m} B_m^2,$$

  - These kinds of transformations have been thoroughly studied and can help to transform data to make it separable.
- Non-polynomial Approximations of Transformations:
  - Sometimes polynomial transformations are not enough.
  - For examples, guessing whether a set of triangles are congruent is not possible by simply observing their coordinates.
    - Computing the length of their segments will easily solve the problem → non polynomial transformation.

$$A = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$

- Polynomial Approximations of Transformations:
  - We have observed that specific transformations may be needed to extract knowledge.
    - But help from an expert is not always available.
  - When no knowledge is available, a transformation f can be approximated via a polynomial transformation using a brute search with one degree at a time.
    - Using the Weistrass approximation, there is a polynomial function f that takes the value Yi for each instance Xi.

$$Y = f(X_1, X_2, \ldots, X_n)$$

  - There are many polynomials verifying Y=f(X) as we want.
  - As the number of instances in the data set increases, the approximations will be better.
  - We can use computer assistance to approximate the intrinsic information.
  - When the intrinsic transformation is polynomial, we need to add the cartesian product of the attributes needed for the polynomial degree approximation.
  - Sometimes the approximation obtained must be rounded to avoid the limitations of the computer digital precision.
- Rank transformation:
  - A change in an attribute distribution can result in a change of model performance.
  - The simplest transformation to accomplish this in numerical attributes is to replace the value of an attribute with its rank.
  - The attribute will be transformed into a new attribute containing integer values ranging from 1 to m, being m the number of instances in the data set.
  - Next, we can transform the ranks to normal scores representing their probabilities in the normal distribution by spreading these values on the Gaussian curve using a simple transformation given by:

$$y = \Phi^{-1}\left(\frac{r_i - \frac{3}{8}}{m + \frac{1}{4}}\right)$$

  - Being ri the rank of the observation i and Φ the cumulative normal function.
  - Note: This transformation cannot be applied separately to the training and test partitions.
- Box-Cox Transformations:
  - When selecting the optimal transformation for an attribute is that we do not know in advance which transformation will be the best.
  - The Box-Cox transformation aims to transform a continuous variable into an almost normal distribution.
  - This can be achieved by mapping the values using following set of transformations:

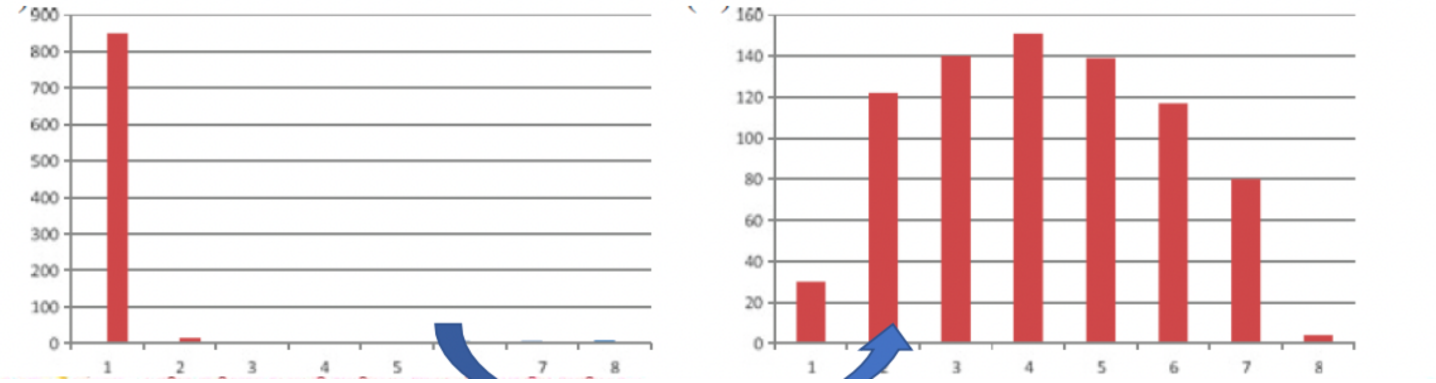
$$y = \begin{cases} x^{\lambda-1}/\lambda, & \lambda \neq 0 \\ log(x), & \lambda = 0 \end{cases}$$

- ○ All linear, inverse, quadratic and similar transformations are special cases of the Box-Cox transformations.
- ○ Please note that all the values of variable x in the previous slide must be positive. If we have negative values in the attribute, we must add a parameter c to offset such negative values:

$$y = \begin{cases} (x+c)^{\lambda-1}/g\lambda, & \lambda \neq 0 \\ log(x+c)/g, & \lambda = 0 \end{cases}$$

- ○ The parameter g is used to scale the resulting values, and it is often considered as the geometric mean of the data.
- ○ The value of λ is iteratively found by testing different values in the range from −3.0 to 3.0 in small steps until the resulting attribute is as close as possible to the normal distribution.

- Spreading the Histogram:
  - ○ Spreading the histogram is a special case of Box-Cox transformations.
  - ○ As Box-Cox transforms the data to resemble a normal distribution, the histogram is thus spread as shown here.



  - ○ When the user is not interested in converting the distribution to a normal one, but just spreading it, we can use two special cases of Box-Cox transformations.
    - ▪ 1. Using the logarithm (with an offset if necessary) can be used to spread the right side of the histogram: y = log(x).
    - ▪ 2. If we are interested in spreading the left side of the histogram, we can simply use the power transformation $y = x^g$.

- Nominal to Binary Transformation:
  - ○ The presence of nominal attributes in the data set can be problematic, especially if the Data Mining (DM) algorithm used cannot correctly handle them.
  - ○ The first option is to transform the nominal variable to a numeric one.
  - ○ Although simple, this approach has two big drawbacks that discourage it:
    - ▪ With this transformation we assume an ordering of the attribute values.
    - ▪ The integer values can be used in operations as numbers, whereas the nominal values cannot.
  - ○ To avoid the problems, a very typical transformation used for DM methods is to map each nominal attribute to a set of newly generated attributes.
  - ○ If N is the number of different values the nominal attribute has, we will substitute the nominal variable with a new set of binary attributes, each one representing one of the N possible values.

    - For each instance, only one of the N newly created attributes will have a value of 1, while the rest will have the value of 0.
    - This transformation is also referred in the literature as 1-to-N transformation.
    - A problem with this kind of transformation appears when the original nominal attribute has a large cardinality:
        - The number of attributes generated will be large as well.
        - Resulting in a very sparse data set which will lead to numerical and performance problems.
- Transformations via Data Reduction:
    - When the data set is very large, performing complex analysis and DM can take a long computing time.
    - Data reduction techniques are applied in these domains to reduce the size of the data set while trying to maintain the integrity and the information of the original data set as much as possible.
    - Mining on the reduced data set will be much more efficient and it will also resemble the results that would have been obtained using the original data set.
    - The main strategies to perform data reduction are Dimensionality Reduction (DR) techniques.
    - They aim to reduce the number of attributes or instances available in the data set.
    - Well known attribute reduction techniques are Wavelet transforms or Principal Component Analysis (PCA).
    - Many techniques can be found for reducing the dimensionality in the number of instances, like the use of clustering techniques, parametric methods and so on.
    - The use of binning and discretization techniques is also useful to reduce the dimensionality and complexity of the data set.
    - They convert numerical attributes into nominal ones, thus drastically reducing the cardinality of the attributes involved.

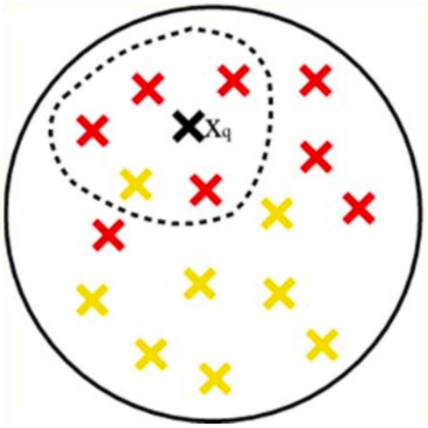
Missing Data

- Data is not always available.
    - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data.
- Missing data may be due to:
    - equipment malfunction.
    - inconsistent with other recorded data and thus deleted.
    - data not entered due to misunderstanding.
    - certain data may not be considered important at the time of entry.
    - not register history or changes of the data.
- Missing data may need to be inferred.
- Missing values may carry some information content.
    - E.g., a credit application may carry information by noting which field the applicant did not complete.
- There are always MVs in a real dataset.
    - MVs may have an impactiom modelling, in fact, they can destroy it!
    - Some tools ignore missing values, others use some metric to fill in replacements.
    - The modeler should avoid default automated replacement techniques.
    - Difficult to know limitations, problems and introduced bias.
    - Replacing missing values without elsewhere capturing that information removes information from the dataset.

How to Handle Missing Data?

- Ignore records (use only cases with all values)
    - Usually done when class label is missing as most prediction methods do not handle missing data well.
    - Not effective when the percentage of missing values per attribute varies considerably as it can lead to insufficient and/or biased sample sizes.
- Ignore attributes with missing values.
- Use only features (attributes) with all values (may leave out important features).
- Fill in the missing value manually.
    - tedious + infeasible?
- Use a global constant to fill in the missing value.
    - E.g., “unknown”. (May create a new class).
- Use the attribute mean to fill in the missing value.
    - It will do the least harm to the mean of existing data.
    - If the mean is to be unbiased.
    - What if the standard deviation is to be unbiased?
- Use the attribute mean for all samples belonging to the same class to fill in the missing value.
- Use the most probable value to fill in the missing value.
    - Inference-based such as Bayesian formula or decision tree.
    - Identify relationships among variables.
        - Linear regression, Multiple linear regression, Non-linear regression.
    - Nearest-Neighbor estimator
        - Finding the k neighbors nearest to the point and fill in the most frequent value or the average value.
        - Finding neighbors in a large dataset may be slow.

Nearest-neighbor



How to Handle Missing Data?

- Note that, it is as important to avoid adding bias and distortion to the data as it is to make the information available.
    - Bias is added when a wrong value is filled in.
- No matter what techniques you use to conquer the problem, it comes at a price. The more guessing you must do, the further away from the real data the database becomes. Thus, in turn, it can affect the accuracy and validation of the mining results.

Summary

- Every real-world data set needs some kind of data pre-processing.
- Deal with missing values.
- Correct erroneous values.
- Select relevant attributes.
- Adapt data set format to the software tool to be used.
- In general, data pre-processing consumes more than 60% of a data mining project effort.

Table of Contents