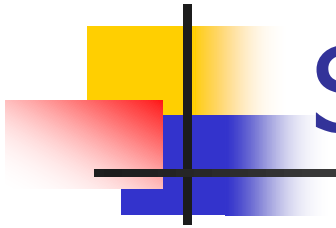
Chapter 4

# Data quality and preprocessing

# Data quality and preprocessing

- The third phase of the CRISP-DM methodology is data preparation
- Need to prepare the data in before numerical models can be applied

| Food | Age | Distance | Company |
|---|---|---|---|
| Chinese | 51 | close | good |
| Italian | 43 | very close | good |
| Italian | 82 | close | good |
| Burgers | 23 | far | bad |
| Chinese | 46 | very far | good |
| Chinese | 29 | too far | bad |
| Burgers | 42 | very far | good |
| Chinese | 38 | close | bad |
| Italian | 31 | far | good |



# Summary

- Data quality
- Converting to a diferente scale type
- Converting to a diferente scale
- Data transformation
- Dimensionality reduction

# Data quality

- It is estimated that the noise ratio can represent 5% or more of the total data set
- Data quality is important and can be affected by internal and external factors:
  - Internal factors can be linked to the measurement process and the collection of information through the attributes chosen
  - External factors are related to faults in the data collection process

# Data quality: missing values

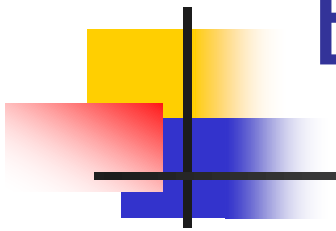
- Causes for missing values
  - An attribute was not considered when the data collection started
  - Some attributes were unknown when some object values were collected
  - Distraction, misunderstanding or refusal when the values of some attribute were collected
  - Lack of necessity to supply the value of some attribute for some objects
  - Non-existence of a value for some attributes for particular objects
  - Fault in the data collection device
  - Cost/difficulty to assign class labels to objects in classification problems

# Data quality: missing values

- Approaches to deal with missing data
  - Ignore missing values
    - Remove columns with missing data
    - Modify a learning algorithm to allow it to accept and work with missing values
  - Remove rows with missing values
  - Fill the missing values estimating their value from the values for this attribute in the other objects

# Data quality: missing values

- The filling of missing data is the most common approach
  1) Create a new value for the attribute, meaning that it was a missing value before
  2) Create a new, related, attribute, with Boolean values, whose value will be true if there was a missing value in the related attribute, and false otherwise
  3) Estimate the value
     - Fill with a location value derived from the attribute values, like mean or median, for quantitative and ordinal, and mode for nominal values
     - The previous method by considering, for classification tasks, only the objects from the same class
     - A learning algorithm to induce a prediction model to predict the value to be filled in a particular attribute



# Example

- Suppose that, due to a data transmission problem, part of our friends data sent to a colleague had missing data
- That data set was filled:
  - Using the mode, for qualitative values
  - Rounded average for quantitative values
  - Both considering only the objects from the same class

# Example

- *Fill the gaps and compare with the original values*
  - *It is possible to see many differences …*

| Data with missing values | | | | Data without missing values | | | |
|---|---|---|---|---|---|---|---|
| **Food** | **Age** | **Distance** | **Company** | **Food** | **Age** | **Distance** | **Company** |
| Chinese | 51 | close | good | Chinese | 51 | close | good |
| | | | good | Italian | 43 | very close | good |
| Italian | 82 | | good | Italian | 82 | close | good |
| Burgers | 23 | far | bad | Burgers | 23 | far | bad |
| Chinese | 46 | | good | Chinese | 46 | very far | good |
| Chinese | | | bad | Chinese | 29 | too far | bad |
| Burgers | | very far | good | Burgers | 42 | very far | good |
| Chinese | 38 | close | bad | Chinese | 38 | close | bad |
| Italian | 31 | far | good | Italian | 31 | far | good |

# Data quality: redundant data

- Redundant data is the excess of data
  - Redundant objects are those that do not bring any new information to a data set
  - They are irrelevant data
  - In the extreme, redundant data can be duplicated data
- Redundancy occurs mainly in the whole set of attributes
- Redundancy can also occur in the predictive attributes, when the values for a predictive attribute can be derived from the values of other predictive attributes

# Data quality: redundant data

There is a data pre-processing technique, named **deduplication**, whose goal is to identify and remove copies of objects in a data set

| Original Data | | | | Deduplicated | | | |
|---|---|---|---|---|---|---|---|
| **Food** | **Age** | **Distance** | **Company** | **Food** | **Age** | **Distance** | **Company** |
| Chinese | 51 | close | good | Chinese | 51 | close | good |
| Italian | 43 | very close | good | Italian | 43 | very close | good |
| Italian | 43 | very close | good | --- | --- | --- | --- |
| Italian | 82 | close | good | Italian | 82 | close | good |
| Burgers | 23 | far | bad | Burgers | 23 | far | bad |
| Chinese | 46 | very far | good | Chinese | 46 | very far | good |
| Chinese | 29 | too far | bad | Chinese | 29 | too far | bad |
| Chinese | 29 | too far | bad | --- | --- | --- | --- |
| Burgers | 42 | very far | good | Burgers | 42 | very far | good |
| Chinese | 38 | close | bad | Chinese | 38 | close | bad |
| Italian | 31 | far | good | Italian | 31 | far | good |

# Data quality: inconsistent data

- Inconsistent values can be found in the predictive and/or target attributes
- An example of an inconsistent value in a predictive attribute is a zip code that does not match with the city name

- Some inconsistencies are easily detected, like when an attribute value violates a known relation between attributes
  - the value of attribute A is larger than the value of attribute B, or
  - the attribute has an invalid value, like a negative value in a predictive attribute that should have only positive values

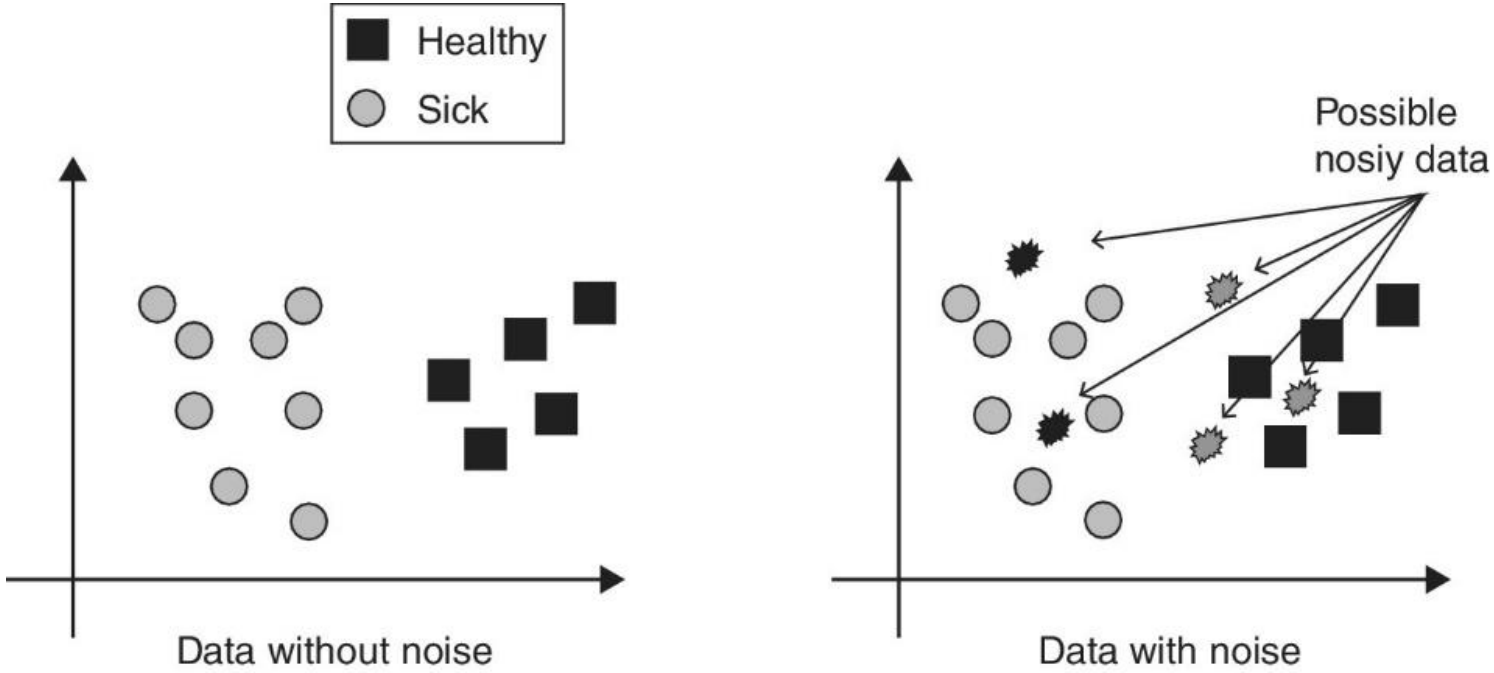
# Data quality: inconsistent data

- In this example we show a data set with the inconsistent values highlighted
- Inconsistent values can be treated as missing values

| Friend | Max temp | Weight | Height | Years | Gender | Company |
|---|---|---|---|---|---|---|
| Andrew | 25 | 77 | 175 | 10 | M | Good |
| Bernhard | 31 | **1100** | 195 | 12 | M | Good |
| Carolina | 15 | 70 | 172 | 2 | F | Bad |
| Dennis | 20 | 85 | **210** | 16 | M | Good |
| Eve | 10 | 65 | 168 | 0 | F | Bad |
| Fred | 12 | 75 | 173 | 6 | M | Good |
| Gwyneth | 16 | 75 | **10** | 3 | F | Bad |
| Hayden | 26 | 63 | 165 | 2 | F | Bad |
| Irene | 15 | 55 | 158 | 5 | F | Bad |
| James | 21 | 66 | 163 | 14 | M | Good |
| Kevin | **300** | 95 | 190 | 1 | M | Bad |
| Lea | 13 | 72 | **1072** | 11 | F | Good |
| Marcus | 8 | 83 | 185 | 3 | F | Bad |
| Nigel | 12 | 115 | 192 | 15 | M | Good |

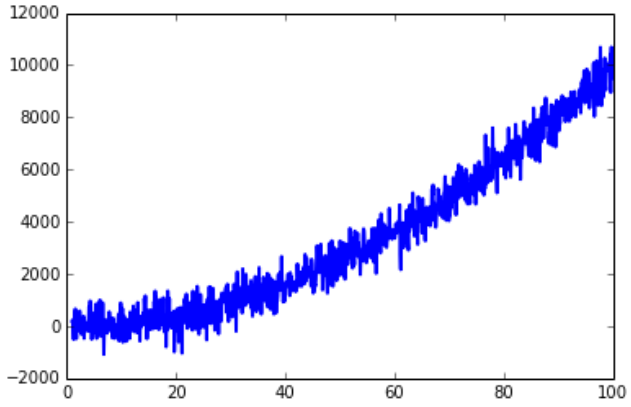
# Data quality: noisy data

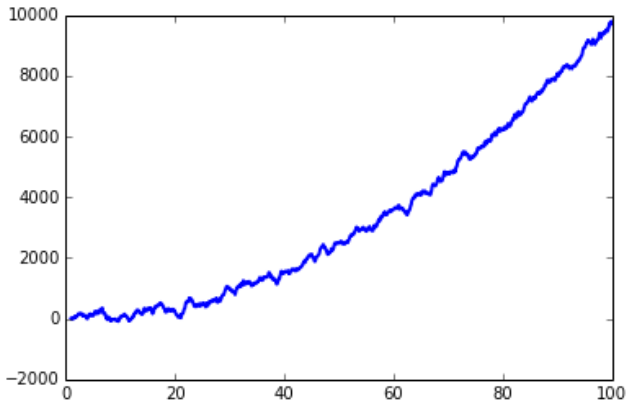
- Noisy data are data that do not fill in the set of standards expected from them
  - Noises can be caused by incorrect or distorted measurements, human error or even contamination of the samples



# Data quality: noisy data

- Noise detection can be performed by adaptation of classification algorithms or by the use of noise filters for data pre-processing
- It is not usually possible to be sure that an object is noisy



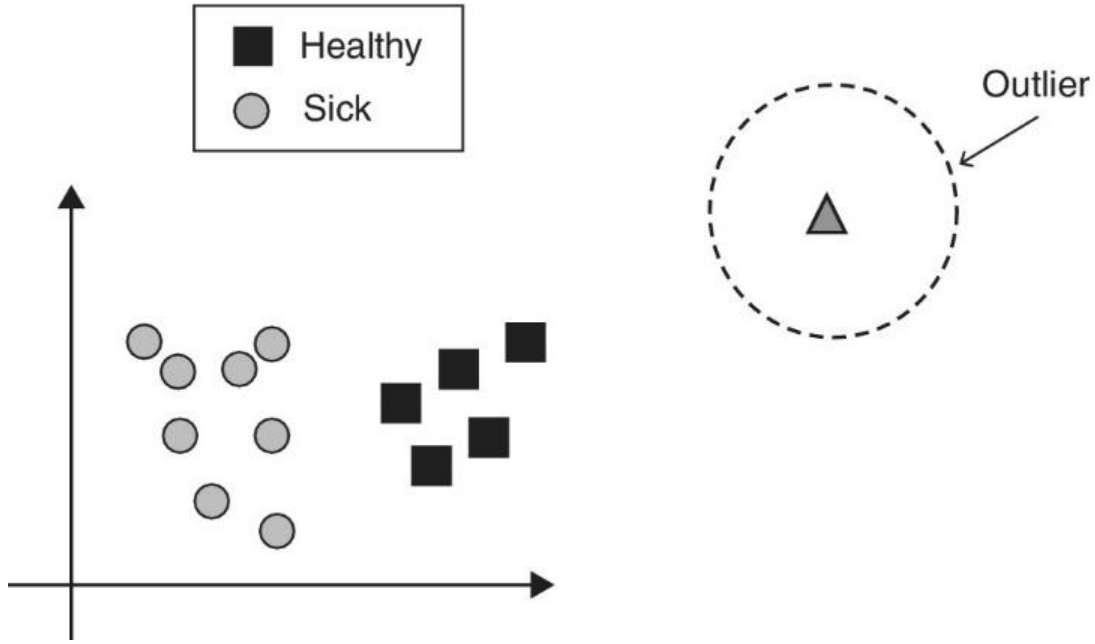




- It is usually performed by noise filters, which can look for noise in either the predictive attributes or in the target attribute
- Most filters were developed for target attributes
- Many label noise filters are based on the k-NN algorithm
- For sequential data we can use smoothing algorithms like Savitzky-Golay filter or Infinite Impulse Response filter

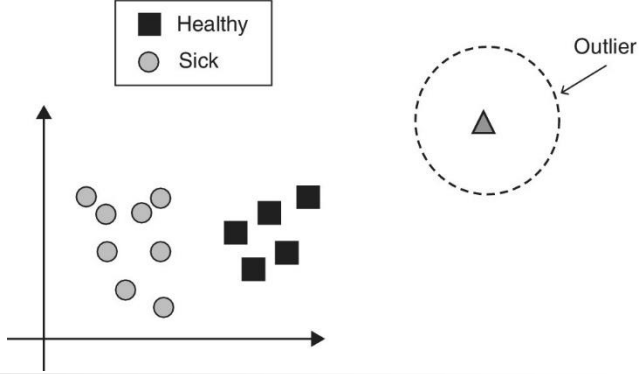
# Data quality: outliers

- In a data set, outliers are anomalous values or objects
- They can also be defined as unusual, despite correct, values of some attributes
  - Different from noisy, outliers can be legitimate values

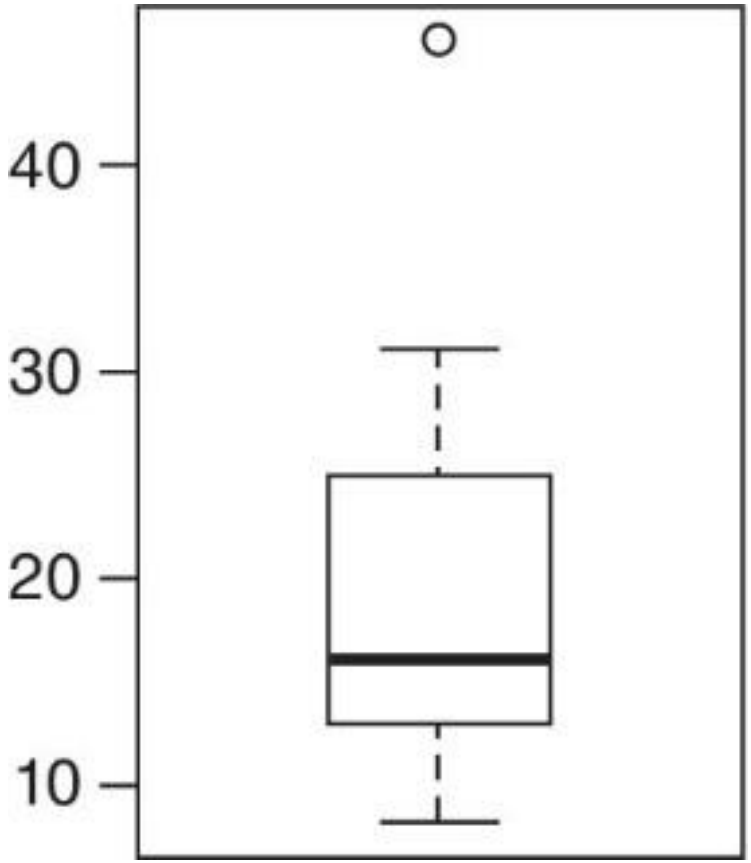
- There are several data analysis applications whose main goal is to find outliers in a data set



# Data quality: outliers




- Using the interquartile range to detect outliers
  - Let $Q_1$ and $Q_3$ be the 1st and 3rd quartile respectively
  - The interquartile range is given by $IQ = Q_3 - Q_1$
- Values below $Q_1 - 1.5 \times IQ$ or above $Q_3 + 1.5 \times IQ$ are considered too faraway from central values to be reasonable



# Converting to a different scale type: nominal to relative

- The most common conversion from a nominal scale type to a relative one is the $1 - of - n$, also known as **One Hot Encoding**, which transforms $n$ values of a nominal attribute into $n$ binary attributes

| Nominal | Relative |
|---|---|
| Green | 001 |
| Yellow | 010 |
| Blue | 100 |

# Converting to a different scale type: nominal to relative

- By using the $1-of-n$ method we increase the number of attributes, each one of them with a large number of $0s$, i.e., the resulting data set can be quite sparse

- *A simple example for* $3$ *DNA sequences with* $5$ *nucleotides each,* $AATCA$, $TTACG$ *and* $GCAAC$
  - *We encode the nucleotides* $A$, $C$, $T$ *and* $G$ *by* $0001$, $0010$, $0100$ *and* $1000$, *respectively*

| Original DNA | | | | | Converted DNA | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | T | C | A | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| T | T | A | C | G | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| G | C | A | A | C | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

# Converting to a different scale type: nominal to relative

- To avoid the dimensionality problem of OHE, we can use **Label Encoding**, which assigns a different numerical value to each nominal value
- Careful: Many models can interpret these numerical values as ordered values, whereas there is no ordering in the original column

| Nominal | Relative |
|---|---|
| Green | 1 |
| Yellow | 2 |
| Blue | 3 |

# Converting to a different scale type: ordinal to relative or absolute

- Converting from ordinal to natural numbers:
  - starting with the value $0/1$ for the smallest value and, for each next value, add $1$ to the previous value
- Converting from ordinal to binary values:
  - the **gray code:** it keeps the distance between two consecutive values as a different value in one of the binary values
  - the **thermometer code**: it starts with a binary vector with only $0$ values and substitute one $0$ value by $1$, from right to left, as the ordinal value increases

| Nominal | Natural number | Gray code | Thermometer code |
|---|---|---|---|
| small | 0 | 000 | 000 |
| medium | 1 | 001 | 001 |
| large | 2 | 011 | 011 |
| very large | 3 | 010 | 111 |

# Converting to a different scale type: relative or absolute to ordinal or nominal

- Converting quantitative values to qualitative values is also named discretization:
  - nominal discretization, if the qualitative scale is nominal
  - ordinal discretization, if the qualitative scale is ordinal
- Infinite range to Finite range

- Discretization has two steps:
  1) The definition of the number of qualitative values, also named number of bins
  2) Given the number of bins, to define the interval of values to be associated with each bin:
     - by width: split the whole range of numbers in intervals with equal size
     - by frequency: use intervals containing equal number of values

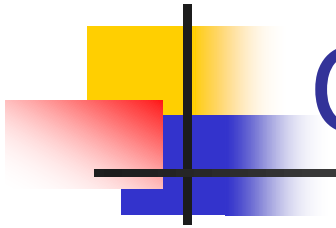
# Encoding Example

| Original data | | | | Converted data | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Food** | **Age** | **Distance** | **Company** | **f1** | **f2** | **f3** | **Age** | **Distance** | **Company** |
| Chinese | 51 | close | good | 0 | 0 | 1 | 51 | 2 | 1 |
| Italian | 43 | very close | good | 0 | 1 | 0 | 43 | 1 | 1 |
| Italian | 82 | close | good | 0 | 1 | 0 | 82 | 2 | 1 |
| Burgers | 23 | far | bad | 1 | 0 | 0 | 23 | 3 | 0 |
| Chinese | 46 | very far | good | 0 | 0 | 1 | 46 | 4 | 1 |
| Chinese | 29 | too far | bad | 0 | 0 | 1 | 29 | 5 | 0 |
| Burgers | 42 | very far | good | 1 | 0 | 0 | 42 | 4 | 1 |
| Chinese | 38 | close | bad | 0 | 0 | 1 | 38 | 2 | 0 |
| Italian | 31 | far | good | 1 | 1 | 0 | 31 | 3 | 1 |

# Discretization Example

- The chosen intervals for the association:
  - by width were $[(2,8), (9,15), (16,22)]$
  - by frequency were $[(2,5), (7,15), (16,20)]$

| | Conversion by | |
|---|---|---|
| **Quantitative** | **Width** | **Frequency** |
| 2 | A | A |
| 3 | A | A |
| 5 | A | A |
| 7 | A | B |
| 10 | B | B |
| 15 | B | B |
| 16 | C | C |
| 19 | C | C |
| 20 | C | C |

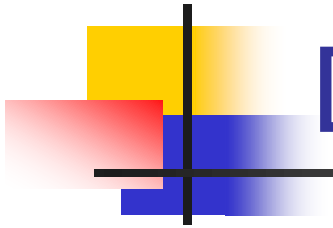


# Converting to a different scale

- Converting data in a scale to another scale of the same type is necessary specially when using distance measures
  - Assuring that all attributes are using the same scale guarantees that all attributes are equally used by the distance measure
- There are two ways to normalize the data:
  - **min-max rescaling**
    - For every value subtract the minimum value and divide the result by the amplitude
  - **standardization**
    - For every value subtract the average and divide by the standard deviation

# Scale conversion example

| | | | Min-max | | Standardization | |
|---|---|---|---|---|---|---|
| **Friend** | **Age** | **Education** | **Rescaled age** | **Rescaled educ.** | **Rescaled age** | **Rescaled educ.** |
| Bernhard | 43 | 2.0 | 1.0 | 0.0 | 0.76 | -1.15 |
| Gwyneth | 38 | 4.2 | 0.0 | 1.0 | -1.13 | 0.66 |
| James | 42 | 4.0 | 0.8 | 0.91 | 0.38 | 0.49 |



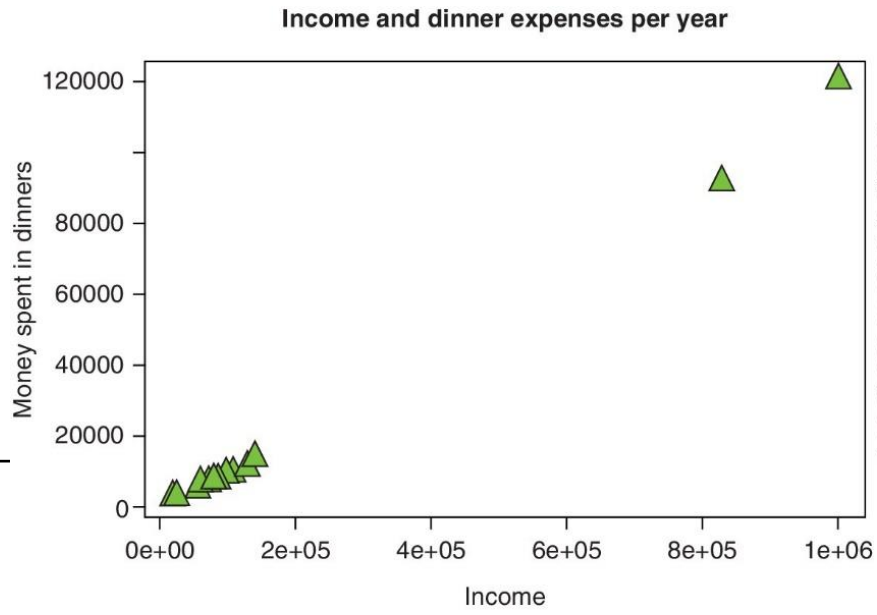
# Data transformation

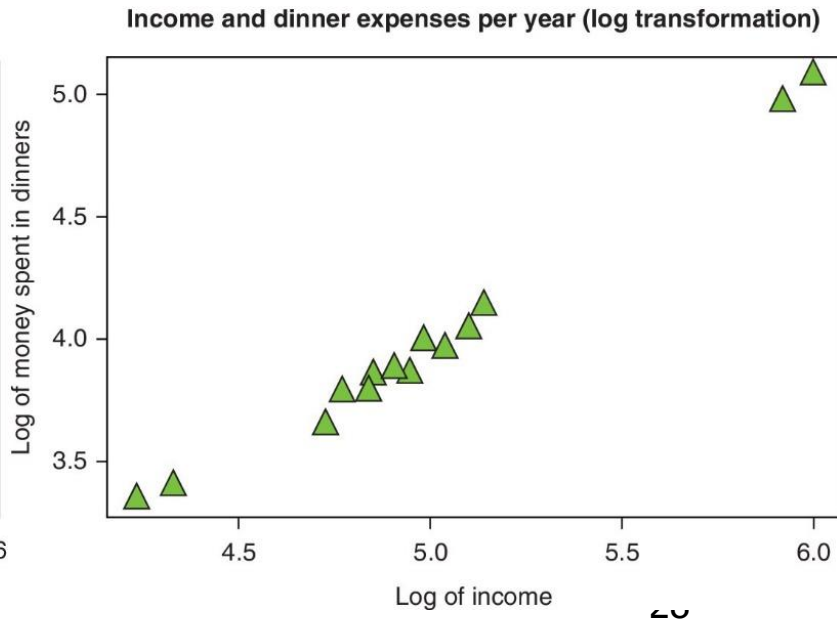
- Data transformation might be necessary to perform on a data set to simplify its analysis or to allow the use of some modeling techniques
- Some simple transformations used to improve data summarization are:
  - Apply a logarithmic function to the values of a predictive attribute
  - Conversion to the absolute value

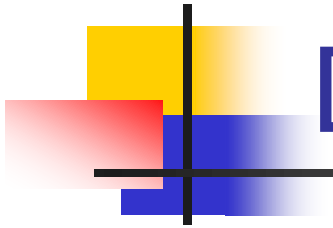
# Example

| friend | salary | dinner |
|---|---|---|
| Andrew | 17000 | 2200 |
| Bernhard | 53500 | 4500 |
| Carolina | 69000 | 6000 |
| Dennis | 72000 | 7100 |
| Eve | 125400 | 10800 |
| Fred | 89400 | 7100 |
| Gwyneth | 58750 | 6000 |
| Hayden | 108800 | 9000 |
| Irene | 97200 | 9600 |
| James | 81000 | 7400 |
| Kevin | 21300 | 2500 |
| Lea | 138400 | 13500 |
| Marcus | 830000 | 92000 |
| Nigel | 1000000 | 120500 |

- Suppose we have the following data illustrating, with two quantitative attributes: the financial income of our friends with how much they spend in dinners per year



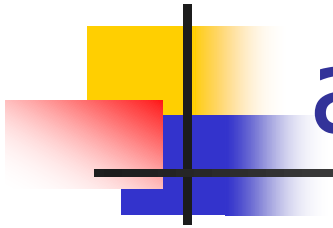





# Dimensionality reduction

- Benefits of dimensionality reduction:
  - Can reduce the training time, decrease memory needed and improve the performance of ML algorithms
  - Can eliminate irrelevant and/or noisy attributes
  - Allows the induction of simpler and more interpretable models
  - Makes the data visualization easier to understand
  - Can reduce the cost of feature extraction
- There are two alternatives to reduce the number of attributes:
  - Attribute aggregation: a group of attributes is substituted by a new attribute
  - Attribute selection: a subset of the initial attributes is selected

# Dimensionality reduction: attribute aggregation



- Attribute aggregation reduces the data to a given number of attributes allowing an easier visualization
  - The selected attributes are those that most differentiate the objects
  - Attribute aggregation techniques project the original data set to a new lower-dimensional space
- Several techniques have been proposed for attribute aggregation in the literature:
  - Principal Component Analysis
  - Independent Component Analysis
  - Multidimensional scaling

# Dimensionality reduction: attribute aggregation

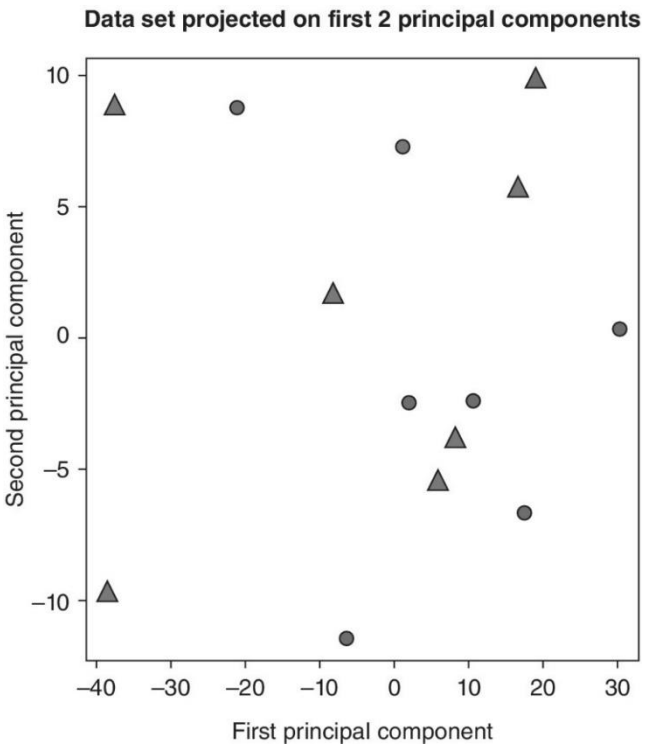
- **Principal Component Analysis**
  - Linearly projects a data set to another data set whose number of attributes is equal or smaller
  - PCA tries to reduce redundancy by combining the original attributes into new attributes
  - These new attributes are named principal components
  - The principal components are not linearly correlated
  - Each principal component is a linear combination of the original attributes
  - Columns need to be scaled before applying PCA

https://en.wikipedia.org/wiki/Principal_component_analysis

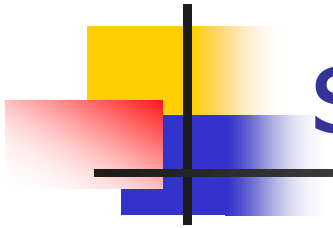
# Example

| Friend | Max temp | Weight | Height | Years | Gender |
|---|---|---|---|---|---|
| Andrew | 25 | 77 | 175 | 10 | M |
| Bernhard | 31 | 110 | 195 | 12 | M |
| Carolina | 15 | 70 | 172 | 2 | F |
| Dennis | 20 | 85 | 180 | 16 | M |
| Eve | 10 | 65 | 168 | 0 | F |
| Fred | 12 | 75 | 173 | 6 | M |
| Gwyneth | 16 | 75 | 180 | 3 | F |
| Hayden | 26 | 63 | 165 | 2 | F |
| Irene | 15 | 55 | 158 | 5 | F |
| James | 21 | 66 | 163 | 14 | M |
| Kevin | 30 | 95 | 190 | 1 | M |
| Lea | 13 | 72 | 172 | 11 | F |
| Marcus | 8 | 83 | 185 | 3 | F |
| Nigel | 12 | 115 | 192 | 15 | M |

Using PCA to extract 2 principal components from the left data set. (Gender was converted to numeric)

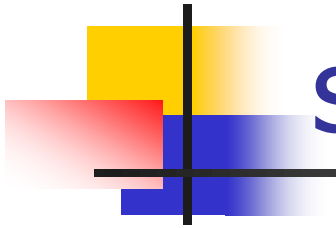


# Dimensionality reduction: attribute selection



- Instead of aggregating attributes, another approach to reduce dimensionality is by selecting a subset of the attributes
  - Among the benefits of attribute selection, it can speed up the learning process, since a smaller number of operations will need to be made
- Attribute selection techniques can be roughly divided into three categories:
  - Filters
  - Wrappers
  - Embedded

# Dimensionality reduction: attribute selection



- **Filters** look for simple, individual, relations between the predictive attribute values and the target attribute
- They rank the attributes according to this relation and select the first *n* attributes (*n* is arbitrary)
- An example of such relation is the Pearson correlation
- Filters disregard the relations between predictive attributes

- *Using the Pearson correlation on the excerpt of our list of friends*
- *If we want to select 3 attributes, years, gender and weight would be selected*

| Predictive attribute | Correlation |
|---|---|
| Years | 0.89 |
| Gender | 0.58 |
| Weight | 0.40 |
| Height | 0.21 |
| Max temp | 0.14 |

# Dimensionality reduction: attribute selection

- **Wrappers** explicitly use an algorithm to guide the attribute selection process
- At each iteration a subset of the attributes is selected and trained on the given model
- As a result, it selects the subset of predictive attributes that provides the highest predictive performance for the classifier
  - Wrappers usually present a much higher computational cost than filters

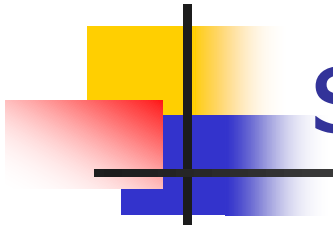
- Wrappers are good at detecting interactions between variables

| Predictive attribute | Predictive performance |
|---|---|
| Years | 0.78 |
| Height | 0.46 |
| Gender | 0.42 |
| Weight | 0.38 |
| Max temp | 0.14 |

# Dimensionality reduction: attribute selection

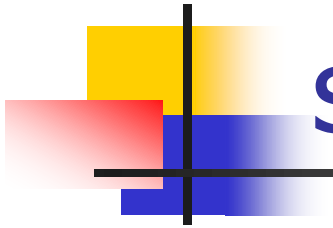
- **Search strategies**:
  - Exhaustive search: it tests with a predictive algorithm all possible subsets
  - Forward selection: it tests all available predictive attributes in order to add to the already selected predictive attributes until the addition does not improve the predictive performance anymore
  - Backward selection: it tests the removal of each predictive attribute from the full set of predictive attributes until the removal of more attributes does not improve the predictive performance anymore
  - There are other search strategies that use more sophisticated and complex mechanisms and, as a result, produce better attribute subsets. For example Genetic Algorithms, Hill Climbing

# Dimensionality reduction: attribute selection

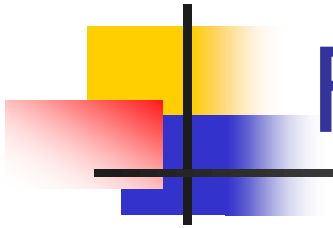


- In the **embedded** category, the attribute selection is performed as an internal procedure of a predictive algorithm
  - An example of a predictive technique able to perform embedded attribute selection is the decision tree technique

# Dimensionality reduction: attribute selection



- The **curse of dimensionality**, or dimensionality curse, relates the proportion between number of objects and number of attributes in a data set
- The number of objects required for efficient training increases exponentially with the increase of the number of attributes
- Increasing the number of attributes without increasing the number of objects, increases the sparsity of the data
- In a very sparse dataset it is very hard to find relations between attributes
- Objects that are sparse differ in many ways, grouping them becomes very hard



# Final remarks

- Data quality: missing values, redundant, inconsistent and noisy data, outliers
- Converting to a different scale type and  to a different scale
- Data transformation
- Attribute aggregation and selection

# Questions?