***** Advanced Machine Learning ***** Data Preprocessing

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Winter term 2020/2021

DHBW
Duale Hochschule
Baden-Württemberg

Agenda for this Unit

Introduction

Why Data Preprocessing?

- Data preprocessing is an important step in data mining and machine learning.

- **'Garbage in, garbage out'** holds true for all data mining and machine learning algorithms (you always get back a result, but is it sensible or useful?).

- There are lots of problems which impede effective learning:

  - Out-of-range values (e. g.: `income` = -100)

  - impossible data combinations (e. g.: `sex` = male $\land$ `pregnant` = yes)

  - Missing values

  - Anomalies and outliers (values which deviate drastically from the other ones)

- Several data mining processes were introduced in order to ensure high-quality data:

  – KDD (Knowledge Discovery in Databases) process $\Rightarrow$ fig. 1

  – CRISP-DM (Cross Industry Standard Process for Data Mining) $\Rightarrow$ fig. 2

- Key steps in any data mining process:

  – Data cleaning

  – Data transformation

  – Data integration

  – Data reduction

Proper data preprocessing is necessary to learn effectively from the data!

Data Mining Processes
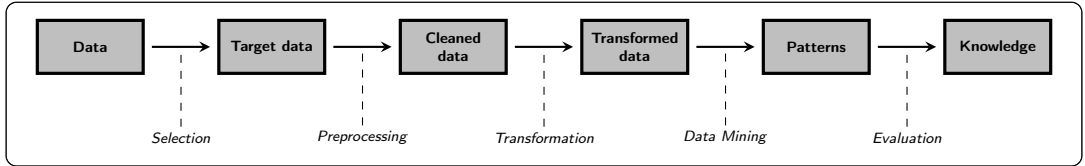
Knowledge Discovery in Databases (KDD)

| Data | → | Target data | → | Cleaned data | → | Transformed data | → | Patterns | → | Knowledge |

Selection   *Preprocessing*   *Transformation*   *Data Mining*   *Evaluation*

Figure 1: KDD process

⚠ **The terms are not used consistently throughout the literature.**

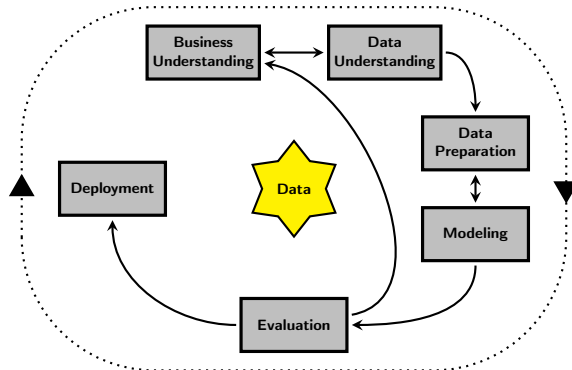# Cross Industry Standard Process for Data Mining (CRISP-DM)

Figure 2: CRISP-DM process

Phases of CRISP-DM

- **Business understanding**

  – Determine what you want to accomplish from a business perspective.
    (*What goals do we want to achieve?*, *Why is the project necessary?*)

  – Assess the current business situation w. r. t. risks, resources, constraints, assumptions, etc.

  – Results: Project plan, business success criteria (*how to measure success?*)

- **Data understanding:**

  – Acquire the data needed to achieve the goals specified in the project plan.

  – Use tools for data exploration (*e. g. compute distributions of key attributes, perform simple aggregations and statistical analyses*).

  – Results: Data description report and data quality report.

- **Data preparation:**

  – Integrate, select and clean the data based on the data description report / data quality report.

  – Construct new features if needed (**feature engineering**).

- **Modeling**
  - Choose a machine learning / data mining technique and find good hyper-parameters.
  - Train the model and test it on a separate test set.

- **Evaluation**
  - Evaluate the model(s) w. r. t. the business objectives. (*In how far does it meet the business goals?*)
  - Review the entire process. (*e. g. highlight activities that have been missed or should be repeated*)

- **Deployment**
  - Deploy the model into a productive environment.
  - Determine the maintenance strategy and monitor the model.

Data Preparation

Data Cleaning

- Bad data quality can (and will most probably) lead to impoverished downstream task results.

- Therefore, it is necessary to remove erroneous data, inconsistencies and outliers.

- The detection of such anomalies often requires a great extent of domain knowledge and is therefore not easy.
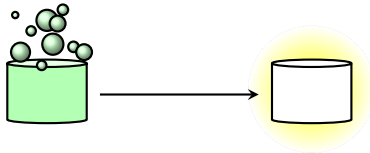
Figure 3:   Data cleaning

- Another problem to be handled is given by missing data (e. g. some feature values are not known for some of the data examples).

- Possible strategies include:

    – Deletion of the affected data example

    – Imputation of the missing value(s), e. g.:

        ▷ Further data collection.

        ▷ Use the mean / median / mode as a substitute (**What is the difference between these three?**)

        ▷ Fill in the most probable value (learn a model, e. g. decision trees to impute the missing value)

    – Replace unknown values by a global placeholder, e. g. *'unknown'* or *'?'*

> **Which technique is used depends on the number of missing values.**

Data Transformation

- Most algorithms require the data to be in a certain form.

- If the form of the data is not as required, it has to be transformed accordingly:

    – Data smoothing (*removal of noise and peaks in the data*)

    – Aggregation (*e. g. computation of sum or average values*)

    – Normalization (*force the data to be in a certain range*)

    – Discretization (*numeric data $\rightarrow$ discrete data*)

    – Numerization (*discrete data $\rightarrow$ numeric data*)

- We will have a closer look at normalization and discretization.

Normalization

- Observation: Features which can take large values dominate features with a small range of values.

- Possible transformations:

  - **Min-max normalization** (left: resulting range [0, 1], right: resulting range [a, b]):

  $$z = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad\qquad z = a + \frac{(x - x_{min}) \cdot (b - a)}{x_{max} - x_{min}} \tag{1}$$

  - **Mean normalization**:

  $$z = \frac{x - \overline{x}}{x_{max} - x_{min}} \tag{2}$$

  - **Standardization** ($\overline{x} = 0$ and $\sigma = 1$):

  $$z = \frac{x - \overline{x}}{\sigma} \tag{3}$$

- Scaling can be harmful, if the data contains many outliers. Libraries like `scikit-learn` also offer robust scaling which can be used in such cases.

Unsupervised Discretization

- **Domain dependent**
    - Suitable discretizations are often known.
    - E. g. age [0–18] $\longrightarrow$ baby [0–3], child (3–6], school child (6–10], teenager (10–18]

- **Equal-width**
    - Divide the range into a number of intervals with equal width.
    - E. g. age [0–18] $\longrightarrow$ [0–3], [4–7], [8–11], [12–15], [16–18]

- **Equal-frequency**
    - Create the intervals such that they comprise roughly the same number of data points.
    - E. g. if the number of bins is set to 5, each will comprise 20 % of the data.

Supervised Discretization – χ-Merge

- **Initialization:**
  1. Sort the examples by feature value.
  2. Construct one interval for each value.

- **Interval merging:**
  1. Compute the χ^2 value for each pair of adjacent intervals:

  $$\chi^2 = \sum_{j=1}^{2} \sum_{k=1}^{\kappa} \frac{(a_{jk} - e_{jk})^2}{e_{jk}} \qquad \text{where} \qquad e_{jk} = n_j \cdot \frac{a_{1k} + a_{2k}}{n_1 + n_2} \tag{4}$$

  Legend:

  $a_{jk} \equiv$ number of examples in j-th interval which have class k

  $e_{jk} \equiv$ expected number of examples in j-th interval which have class k

  $n_j \equiv$ total number of examples in j-th interval

  2. Merge the intervals with the lowest χ^2 value

- **Stopping criterion:** χ^2 values of all pairs exceed a significance threshold.

Data Reduction

- Databases are typically not collected with data mining / machine learning in mind.

- Many features may be:

  – irrelevant

  – uninteresting

  – redundant

- Removing such features might increase efficiency, improve accuracy and prevent overfitting.

- **Feature subset selection (FSS)** techniques try to determine appropriate features automatically.

> **Principal component analysis (PCA) can also be used to reduce the data. Since the algorithm was already covered, it is not presented here.**

Unsupervised FSS

- Use **domain knowledge**: An expert may know in advance that some features are irrelevant uninteresting or redundant.

- **Random sampling**

  - Select a random subset of the features.

  - Such an approach may be appropriate in the case of many weakly correlated features or in conjunction with ensemble methods (*remember random forests?*).

Supervised FSS

- **Filter approaches:**

  - Such techniques attempt to estimate the features' capabilities to discriminate between the classes.

  - The most discriminatory features are ultimately selected.

  - **Problems:**

    ▷ Redundant features will receive similar weights.

    ▷ Some features may only be important in combination with other features (e.g. XOR-problem).

- **Wrapper approaches:**

  - Search through the space of all possible feature subsets.

  - Each feature subset is tried in combination with the learning algorithm.

  - The subset which performs best is kept.

RELIEF algorithm (filter approach)

-

Wrapper approaches
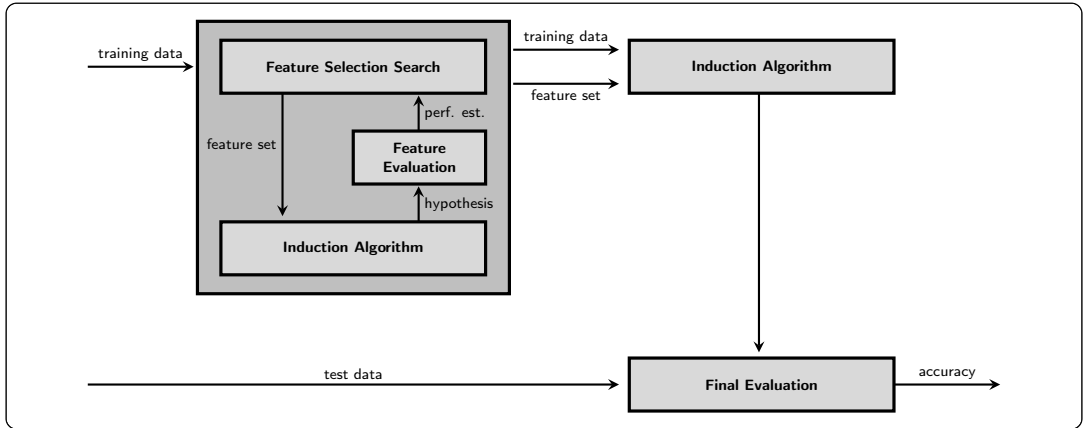
Figure 4:   Wrapper approach for feature subset selection

- **Forward selection:**

  1. Start with an empty feature set $\mathcal{F}$

  2. For each attribute A estimate accuracy of learning algorithm on $\mathcal{F} \cup \{A\}$

  3. $\mathcal{F} \longleftarrow \mathcal{F} \cup \{\text{attribute with highest accuracy}\}$

  4. go to 2 (until m features have been found

- **Backward selection:**

  – Start with a full feature set $\mathcal{F}$

  – Subsequently remove attributes from $\mathcal{F}$

Data Integration

-

Thank you very much for the attention!

Topic: ***** Advanced Machine Learning ***** Data Preprocessing
Term: Winter term 2020/2021

Contact:
Daniel Wehner, M.Sc.
SAP SE / DHBW Mannheim
daniel.wehner@sap.com

Do you have any questions?