Data Preparation

Data Preparation is the process of **transforming raw data** into a **clean** and **organized** format that is **ready for analysis**.

- Also known as **data cleaning**, **data cleansing**, or **data pre-processing**;
- Improves the **quality** of the data;
- Adjusts the data to **better fit** the **requirements** of the **data mining** algorithm;
- One of the most important steps in the KDD process, occupying 60% to 80% of the time;

Data preparation is composed of several steps:

- **Selection** - selecting the relevant data;
- **Transformation** - transforming the data into a suitable format;
  - **Integration** - combining data from multiple sources;
  - **Cleansing** - removing noise and inconsistencies;
  - **Feature engineering** - creating new features from existing ones.

Integration

- **Goal**: **merge** data from **multiple sources**;
- **Issues**:
  - **Heterogeneous representations** - different data types, units, scales, and structures;
  - **Redundancy** - same data represented multiple times.

Cleansing

- **Goal**: **improve** the **quality** of the data;
- **Issues**:
  - **Missing values** - missing data;
  - **Noisy data** - data with errors or outliers;
  - **Inconsistent data** - data that does not match the expected format.

Feature Engineering

- **Feature engineering** is the process of **creating new features** from **existing ones**;
- **Goal**: **reduce** the **complexity** of the data, creating **simpler** and **more informative** features, **without information loss**;
- **Issues**:
  - **Large dimensionality** - too many features;
  - **High complexity** - too complex features;
  - **Low expressiveness** - features that do not represent the data well.

Missing Values

- **Missing values** are values that are **not present** in the data;
- The **reasons** for missing values can be:
  - Equipment malfunction;
  - Data not collected, since it was not considered important;
- Most data mining algorithms **cannot handle missing values**;
- Solutions:
  - **Ignore** records with missing values;
    - Can be bad if the number of records with missing values is high;
  - **Fill** missing values;
    - **Constant** value - value `NA` or `0` for example - value to describe the absence of a value;
    - **Mean/median/mode** value - usually, the variable becomes less relevant;
      - If the mean is used, the **distribution** of the variable is **preserved**;
    - **Conditional mean** value - mean value of the records with the same class;
    - **Most probable value** - value with the highest probability of occurring, using a probability distribution, or a model.
    - However, filling missing values can **bias** the data.

Discretization - Dealing with Noise

- **Noise** corresponds to **unexpected values** in the data; they express some level of **corruption/distortion**;
- **Discretization** is the process of **transforming continuous values** into **discrete values** - transforming **numerical** variables into **symbolic** variables;
  - **Equal-width discretization** - divides the range of variable A into k intervals of equal size;
  - **Equal-frequency discretization** - divides the range of variable A into k intervals, each containing **approximately** the same number of samples.

Variables Encoding - Dealing with Inconsistencies

- **Inconsistent data** occurs when there is some value that is **incoherent** with the rest of the data, but from a **record perspective - noise is from a variable perspective**;
- **Dummification**, also known as **one-hot encoding**, is the process of **transforming symbolic variables** into **binary variables** - transforming **symbolic** variables into **numerical** variables - **dummy variables**;
  - **Binary** - each value is represented by a **single** binary variable;
  - **One-hot** - each value is represented by a **set** of binary variables;
- Dummification should be used with **caution**:
  - **Curse of dimensionality** - the number of variables increases exponentially;
  - **Correlation** - the variables are highly correlated.

Scaling

- **Difference of magnitude between variables scales** can create **inconsistencies**;
- Solutions: **Scale all variables** to the **same range**;
  - **Normalization** - scale all variables to the **same range**; a common range is $[0, 1]$;

    * **Drawback** - out-of-bounds error can happen if a value cannot be mapped to the range;
  - **Standardization** - computes a transformation that **centers** the data around the **mean** and **scales** it to the **variance**.
    * $z = \frac{X-\mu}{\sigma}$ - **z-score**;
    * **negative** if $z < mean$;
    * **positive** if $z > mean$;
    * More **robust** to **outliers** than normalization;
  - Does decision trees, and random forests;
  - Does not affect Naive Bayes, since it is based on probabilities;

Balancing

- A dataset is **unbalanced** if the **number of samples** in each **class** is **not similar** - this can **bias** the **model**;
- To solve this situation there are two approaches:
  - **Weighing - increase** the **weight** of the **minority class**;
  - **Resampling - change** the **number of samples** in each **class**;
    * **Undersampling - remove** samples from the **majority class**;
      · Not recommended when the minority class is too small, since it can lead to **information loss**;
    * **Oversampling - duplicate** samples from the **minority class**.
      · **SMOTE - Synthetic Minority Oversampling Technique** - creates **synthetic samples** from the **minority class** for each sample in the minority class, it finds its **nearest neighbors** and **creates** a **new sample** that is a **linear combination** of the **original sample** and its **nearest neighbors**.
      · Should be applied when the minority samples are too similar, and we need to enlarge the space covered by them;
    * **Hybrid - remove** samples from the **majority class** and **duplicate** samples from the **minority class**.