UNIT 4

GitHub

DATA PRE-PROCESSING BEFORE ML MODEL

A. Data

- Where does data come from, and why does it come? Behind data, there is always a science — an **aim or a question**. Based on that aim, we work and collect data, or obtain it from different sources. Basically, an aim or question is the reason why we collect data.
- **Metadata explains the details of data**, such as where the data came from, how it was collected, when it was created, its format, and how it should be used.
- **Data preprocessing** means making data capable of being used effectively by a machine learning model so that it performs in the best possible way. Even if the data is correct and safe, preprocessing is still required. Preprocessing is always necessary.

  If the data is already correct, there are still some steps needed to further improve it so that a machine learning model can work properly and efficiently.

B. Data Preprocessing

- Data preprocessing is the process of evaluating, filtering, manipulating, and encoding data so that a machine learning algorithm can understand it and use the resulting output. The major goal of data preprocessing is to eliminate data issues such as missing values, errors, noise, inconsistencies, improve data quality, and make the data useful for machine learning purposes.

  Data practitioners spend ~80% of their time on data preprocessing and management, as raw data is messy, coming from diverse sources.

  Structured sequence for preprocessing:

  1. Acquire the dataset
  2. Import libraries
  3. Load/import datasets
  4. Check for missing values
  5. Encode non-numerical data
  6. Scale the features
  7. Split into training, validation, evaluation sets.
- **Why is data preprocessing important?**

Data preprocessing is required for almost all types of data analysis, data science, and AI development to produce trustworthy, precise, and resilient findings for corporate applications.

Machine learning and deep learning algorithms perform best when data is presented in a way that streamlines the solution to a problem.

Data wrangling, data transformation, data reduction, feature selection, and feature scaling are all examples of data preprocessing approaches teams use to reorganize raw data into a format suitable for certain algorithms. This can significantly reduce the processing power and time necessary to train a new machine learning or AI system or perform an inference against it.

Most of the current data science packages and services now contain preprocessing libraries that automate many of these activities.

See More…

C. Data Preprocessing Key Steps and Techniques in Machine Learning

Free Course: Preprocessing for Machine Learning in Python

1. Data Cleaning

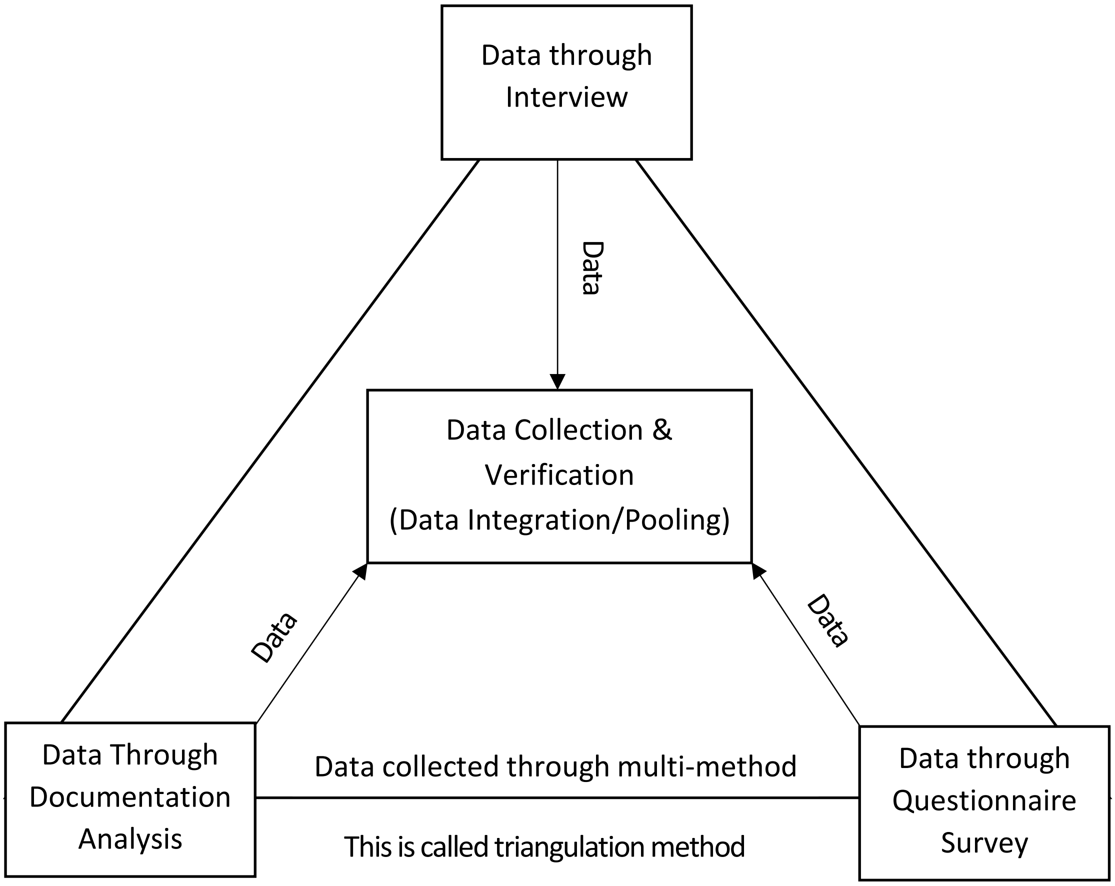
Data cleaning, also called data cleansing or data scrubbing, is the process of identifying and correcting errors and inconsistencies in raw data sets to improve data quality.

See more…

- Handling/Impute Missing/Null Values
- Remove Noisy Data
- Outlier Detection and Removal
- Remove Duplicates
- Noise Reduction
- Fixing Inconsistencies
- Data Validation
- Data Standardization
- Data Normalization
- Data Type Correction
- Handling Invalid Values
- Data Smoothing
- Error Correction
- Removing Irrelevant Data

- Data Integrity Checks
- Handling Imbalanced Data

2. Data Integration



Data through Interview
Data
Data Collection & Verification (Data Integration/Pooling)
Data
Data
Data Through Documentation Analysis
Data collected through multi-method
This is called triangulation method
Data through Questionnaire Survey

Figure 2.1 Data Integration via Triangulation.

Data integration refers to the process of combining and harmonizing data from multiple sources into a unified, coherent format that can be put to use for various analytical, operational and decision-making purposes.

See more...

- Remove Duplicates/Data Redundancy
- Fixing Inconsistencies
- Data Cleaning (**Data cleaning** is necessary because after collecting data through multiple methods, new data may arrive, and therefore it must be cleaned.)
- Merge data
- Data consolidation

3. Data Transformation

Data transformation is at the heart of data analysis and machine learning. In a machine learning pipeline, which includes modifying the raw data and converting raw data into a more suitable format or structure for analysis and model training purpose, to improve its quality, performance and make it compatible with the requirements of a particular task or system. In data transformation, we usually deal with issues such as noise, missing values, outliers, and non-normality.

See more Link 1 | Link 2 | link 3

- Scaling
- Normalization
- Aggregation (combining two or more variable/features)
- Generalization
- Data Transformation Techniques are:
    - Smoothing
    - Aggregation
    - Discretization
    - Attribute Construction
    - Generalization
    - Normalization
        - Min-Max Normalization
        - Z-Score Normalization
        - Decimal Scaling
    - Data Reduction
    - Encoding Techniques
        - One-Hot Encoding
        - Label Encoding
        - Binary Encoding
        - Frequency Encoding
    - Feature Scaling
    - Data Integration
        - Handling schema mismatch
        - Resolving naming conflicts
        - Merging tables
    - Data Encoding for Text (Text Transformation)
        - Tokenization
        - Stemming / Lemmatization

    - TF-IDF
    - Word embeddings
  - Data Binarization
  - Data Scaling and Standardization for Outliers
    - Log transformation
    - Square root transformation
    - Reciprocal transformation
  - Higher Level Concepts

4. Data Reduction

Data reduction is the process of reducing the size of a dataset while still preserving the most important information, to improve the efficiency and performance of machine learning algorithms and other data-driven processes.

This can be beneficial in situations where the dataset is too large to be processed efficiently, or where the dataset contains a large amount of irrelevant or redundant information.

Table 4.1 Input Feature Matrix for Col51 Prediction.

Col1	Col2	Col3	Col4	….	Col50	Predict the Col51 based on Col1 to Col50
1						
2						
…						
100						

50 (feature) x 100 (rows) = 5000

25 (feature—top priority) x 100 (rows) = 2500

See more…

  - Dimensionality Reduction | Link1, Link2
  - Numerosity Reduction | Link1, Link2

Table 4.2 Categorical to Numeric Data Encoding

Sex				
Male	to →	4	to →	1
female	to →	6	to →	0
Male	to →	4	to →	1

female	to →	6	to →	0

This technique is also called data encoding technique.

Convert categorical to numeric.

- Data Compression Link
- Encoding

Why do we reduce data? Link

a. When we use large datasets on personal computers, they require very high resources such as RAM, large storage capacity, and GPU support. Data reduction helps lower these resource requirements.

b. During a **pilot project** (a small or initial phase of a larger project), data reduction techniques are used more heavily to work efficiently with limited data and resources.

c. Even when we have high computational power, we may still prefer **dimensionality reduction** because we want to keep only those features that have a significant impact on the output and remove less important ones.

5. Data Discretization | Link1, link2

When we can convert Numeric variable to Nominal variable i.e., data discretization. Discretization is the process of converting continuous data/values or numerical data/values into discrete categories/features or bins.

- Numerical to Categorical conversion of data
- Binning
    - Clusters analysis