COMP4131: Data Modelling and Analysis

Lecture 2: Data Wrangling and Pre-processing

Kian Ming Lim

University of Nottingham Ningbo China

kian-ming.lim@nottingham.edu.cn

February 20, 2025

Outline

1. Introduction to Data Wrangling and Pre-processing
2. The Data
3. Data Cleaning
4. Data Transformation
5. Data Integration and Reduction
6. The Complete ML Workflow

Introduction to Data Wrangling and Pre-processing

Data Wrangling and Pre-processing

Data wrangling and data pre-processing are interconnected processes in the data preparation pipeline that involve cleaning, transforming, and organizing raw data into a structured and optimized format suitable for analysis, modeling, or decision-making.

Data Wrangling and Pre-processing

- Data wrangling is the process of cleaning, transforming, and organizing raw data into a usable format.
- It involves handling inconsistencies, missing values, and errors, as well as integrating data from multiple sources to prepare it for analysis or further processing.

Workflow of the Data Wrangling Process

Data Wrangling and Pre-processing

- Data pre-processing is the process of preparing data for analysis or machine learning after it has been wrangled.
- It ensures the data is optimized for specific analytical or modeling purposes via techniques such as:
  - Scaling / normalization
  - Encoding categorical variables
  - Handling outliers
  - Splitting data into training and testing sets

Why is Data Wrangling and Pre-processing Important?

Real-World Data Challenges:

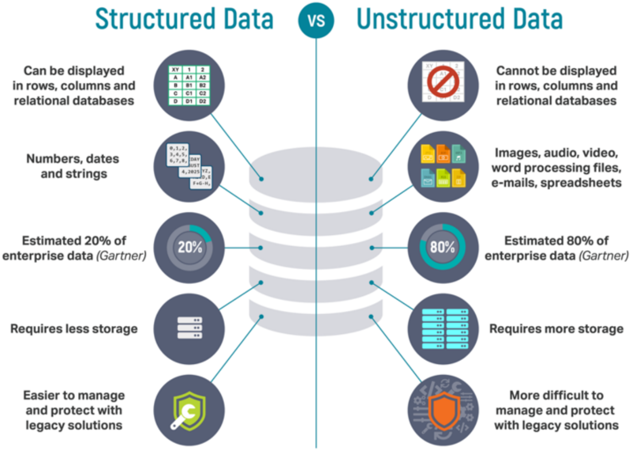
- **Missing values:** Incomplete data that can lead to biased results.
- **Duplicates:** Repeated records that inflate or skew analysis.
- **Inconsistent formats:** Non-standardized data, such as mixed date formats or varying units.
- **Outliers:** Extreme values that can distort statistical calculations.

Impact on Analysis:

- Poor-quality data leads to **inaccurate insights** and **unreliable models**.
- **Garbage in, garbage out (GIGO):** Results are only as good as the data used.

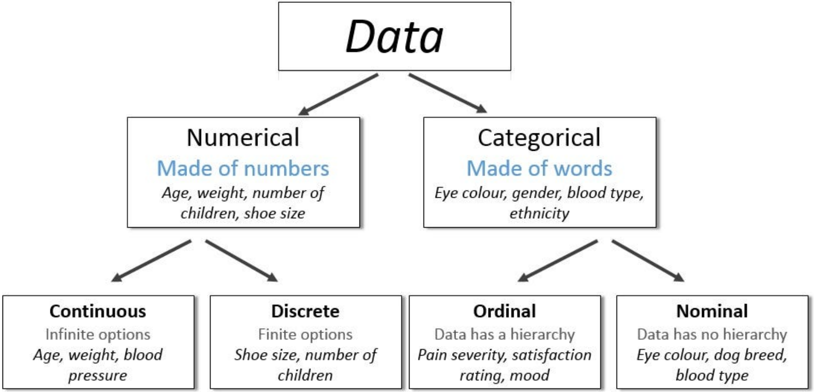
The Data

Types of Data



Source: Lawtomated, Structured vs. Unstructured Data: What are they and why care?

Types of Data



Source: Medium: Data Science Basics.

Common Notations

Feature (Attribute)

Label (Class, Output, Target)

Examples (Samples, Instances, Observations)

housingMedianAge (feature)	totalRooms (feature)	totalBedrooms (feature)	medianHouseValue (label)
15	5612	1283	66900
19	7650	1901	80100
17	720	174	85700
14	1501	337	73400
20	1454	326	65500

Feature and Label

Features:

- A feature is an input variable—the x variable in machine learning.
- A simple machine learning project might use a single feature, while a more sophisticated machine learning project could use millions of features, specified as:

$$x_1, x_2, \ldots, x_N$$

Labels:

- A label is the thing we're predicting—the y variable in machine learning.

Examples

Examples:

- An **example** is a particular instance of data, $\mathbf{x}$. (We put $\mathbf{x}$ in boldface to indicate that it is a vector.)
- Examples can be categorized into:
  - **Labeled examples** - Includes both feature(s) and the label. That is:

$$\text{Labeled examples: } \{\text{features, label}\} : (\mathbf{x}, y)$$

  - **Unlabeled examples** - Contains features but not the label. That is:

$$\text{Unlabeled examples: } \{\text{features}\} : (\mathbf{x})$$

Data Cleaning

What are Missing Values?

- Missing data can occur due to errors in data collection, storage, or processing.

Impact:

- Missing data can lead to biase or incorrect analysis.

	A	B	C	D
0	1.0	2.0	3.0	4.0
1	5.0	6.0	NaN	8.0
2	10.0	11.0	12.0	NaN

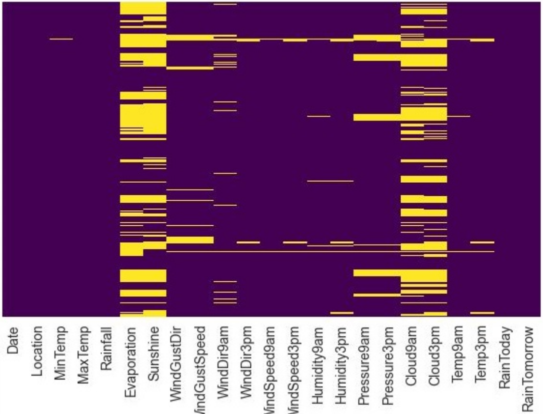
Identifying Missing Data

Methods to Identify Missing Data:

- Visual inspection (e.g., heatmaps).
- Programmatic methods: `isnull()` or `isna()` in Pandas.

Example:

- A dataset with missing values in columns.



Handling Missing Values

Common Techniques:

- Remove missing values (rows/columns).
- Impute missing values (mean, median, or mode).
- Forward/backward fill: use previous or next values to fill missing data.

	A	B	C	D
0	1.0	2.0	3.0	4.0
1	5.0	6.0	NaN	8.0
2	10.0	11.0	12.0	NaN

Remove rows

	A	B	C	D
0	1.0	2.0	3.0	4.0

Remove cols

	A	B
0	1.0	2.0
1	5.0	6.0
2	10.0	11.0

Mean Imputation:

$$x_{\text{new}} = \frac{\sum x_i}{n}$$

	A	B	C	D
0	1.0	2.0	3.0	4.0
1	5.0	6.0	NaN	8.0
2	10.0	11.0	12.0	NaN

Imputation

	A	B	C	D
0	1.0	2.0	3.0	4.0
1	5.0	6.0	7.5	8.0
2	10.0	11.0	12.0	6.0

Outlier Detection and Treatment

What are Outliers?

- A data point that significantly deviates from other observations.

Impact:

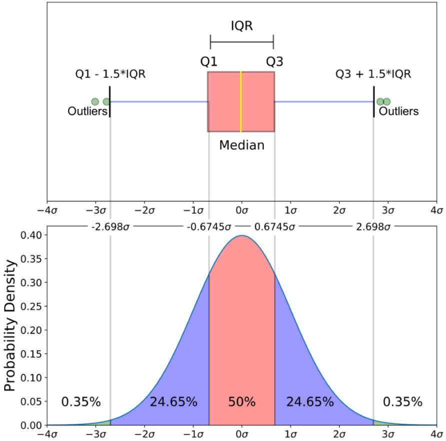
- Outliers can distort statistical analysis and model performance.

Methods to Identify Outliers:

- Z-score.
- IQR (Interquartile Range).

Z-score:

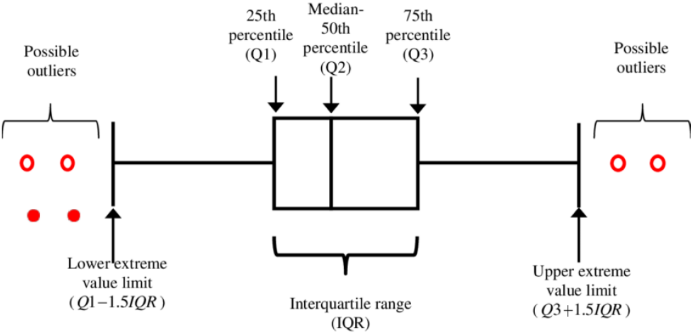
- Measures how far a data point is from the mean in terms of standard deviations.
- Formula: $Z = \frac{(x-\mu)}{\sigma}$



Methods to Identify Outliers

Interquartile Range (IQR):

- Identifies outliers as data points outside 1.5 $\times$ IQR.
- Formula: $\text{IQR} = Q3 - Q1$



Handling Outliers

Remove:

- Delete outlier rows.

Cap:

- Replace outliers with a threshold value (e.g., upper/lower bounds).

Transform:

- Apply log or square root transformations to reduce the impact of outliers.

Dealing with Duplicates

What are Duplicates?

- Duplicates can occur due to data entry errors or merging datasets.

Impact:

- Duplicates can lead to overcounting and biased results.

Methods:

- Identify and remove duplicates.

Data Transformation

Introduction to Data Transformation

What is Data Transformation?

- The process of converting data into a suitable format for analysis or modeling.

Why is it Important?

- Improves the performance of the models.
- Ensures data is on a consistent scale.
- Handles categorical and continuous data appropriately.

Key Techniques:

- Feature scaling.
- Encoding categorical data.
- Data binning.
- Feature engineering.
- Handling imbalanced data.

Feature Scaling

What is Feature Scaling?

- Rescaling features to a specific range (e.g., 0 to 1) or standardizing them to have a mean of 0 and a standard deviation of 1.

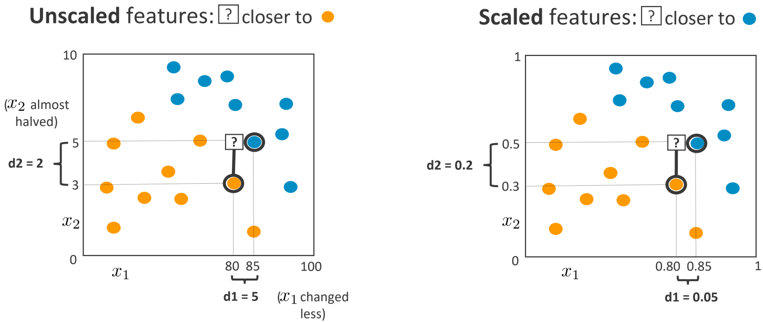
Why is it Important?

- Ensures all features contribute equally to the model.
- Prevents features with larger scales from dominating.

Common Methods:

- Normalization (Min-Max scaling).
- Standardization (Z-score scaling).

Feature Scaling



Source: AWS Machine Learning University (MLU)

Normalization vs. Standardization

Normalization (Min-Max Scaling):

- Rescales data to a range of [0, 1].
- Formula: $X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$

Standardization (Z-score Scaling):

- Rescales data to have a mean of 0 and a standard deviation of 1.
- Formula: $X_{\text{std}} = \frac{X - \mu}{\sigma}$

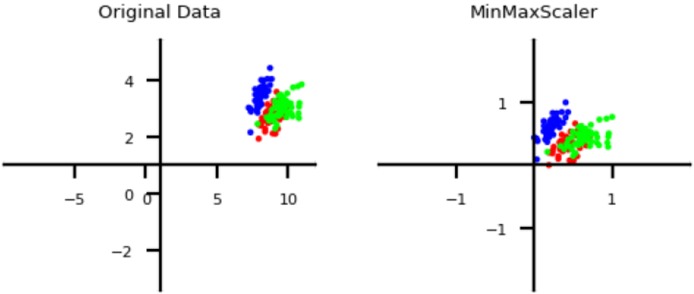
When to Use:

- Normalization: When data distribution is unknown or not Gaussian.
- Standardization: When data follows a Gaussian distribution.

Normalization

Normalization (Min-Max Scaling):

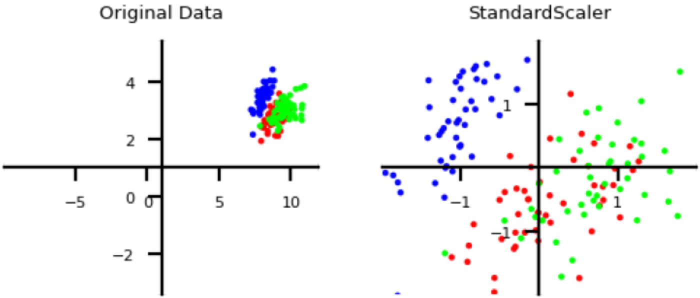
- Scales all features between a given min and max value (e.g. 0 and 1).
- Makes sense if min/max values have meaning in your data.
- Sensitive to outliers.



Standardization

Standardization (Z-score Scaling):

- Generally most useful, assumes data is more or less normally distributed.
- Per feature, subtract the mean value μ, scale by standard deviation σ



Encoding Categorical Data

What is Categorical Data?

- Data that represents categories (e.g., gender, color, country).

Why Encode Categorical Data?

- Most machine learning algorithms require numerical input.

Common Encoding Techniques:

- One-hot encoding.
- Label encoding.
- Binary encoding.

One-hot Encoding, Label Encoding, Binary Encoding

One-hot Encoding:

- Converts each category into a binary vector.
- Example: Red = [1, 0, 0], Green = [0, 1, 0], Blue = [0, 0, 1].

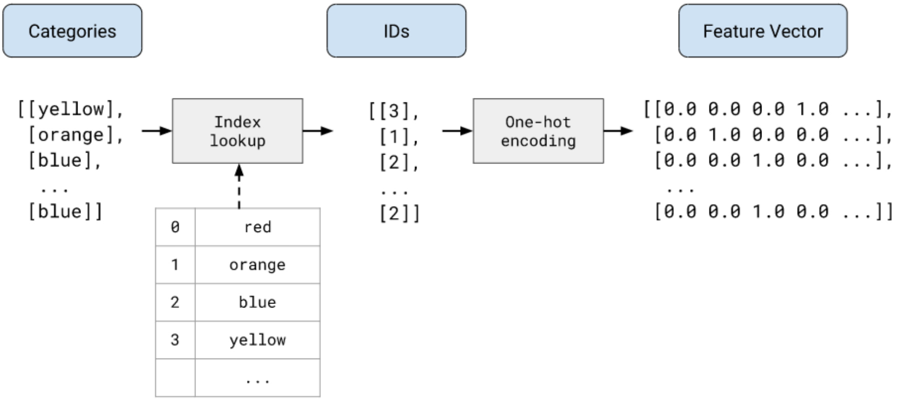
Label Encoding:

- Assigns a unique integer to each category.
- Example: Red = 0, Green = 1, Blue = 2.

Binary Encoding:

- Combines label encoding with binary representation.
- Example: Red = 00, Green = 01, Blue = 10.

One-hot Encoding



Data Binning

What is Data Binning?

- Converting continuous data into discrete categories (bins).

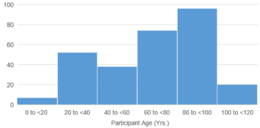
Why Use Data Binning?

- Simplifies complex data.
- Handles outliers.
- Improves model performance for certain algorithms.

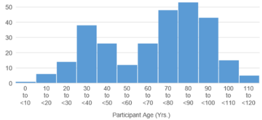
Example (Age):

- 0-18 = "Child",
- 19-35 = "Young Adult",
- 36-60 = "Adult",
- 60+ = "Senior".

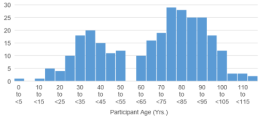
Data Binning



6 bins



12 bins



24 bins

Feature Engineering

What is Feature Engineering?

- The process of creating new features or modifying existing ones to improve model performance.

Techniques:

- Adding new features (e.g., calculating ratios or differences).
- Modifying features (e.g., log transformation).
- Dropping irrelevant or redundant features.

Example:

- Creating a "BMI" feature from height and weight.

Handling Imbalanced Data

What is Imbalanced Data?

- When one class significantly outnumbers the other(s) in a classification problem.

Why is it a Problem?

- Models may become biased toward the majority class.

Techniques to Handle Imbalanced Data:

- Resampling: Oversampling the minority class or undersampling the majority class.
- Synthetic Data Generation: Using techniques like SMOTE, GAN.
- Algorithmic Approaches: Using class-weighted models.

ACTIVITY . THINK-PAIR-SHARE

- Pair up with the person next to you and identify what are the issues in the following heart disease classification data set. What are the solutions?

PatientID	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
1	52	M	TA	118	186	0	LVH	190	N	0	Flat	0
2	58	M	ASY	136	203	1	Normal	123	Y	1.2	Flat	1
3	44	M	ASY	110	197	0	LVH	177	N	0	Up	1
4	43	M	TA	120	291	0	ST	155	N	0	Flat	1
5	55	F	ATA	132	342	0	Normal	166	N	1.2	Up	0
6	66	M	ASY	112	212	0	LVH	132	Y	0.1	Up	1
7	55	M	NAP			0	Normal	155	N	1.5	Flat	1
8	53	M	ASY	123	282	0	Normal	95	Y	2	Flat	1
9	42	M	ASY	136	315	0	Normal	125	Y	1.8	Flat	1
10	43	M	ASY	110	211	0	Normal	161	N	0	Up	0
11	59	M	ASY	125		1	Normal	119	Y	0.9	Up	1
12	46	M	ASY	140	311	0	Normal	120	Y	1.8	Flat	1
13	42	M	ATA	120	198	0	Normal	155	N	0	Up	0
14	39	M	ASY	110	273	0	Normal	132	N	0	Up	0
15	50	F	ASY	110	254	0	LVH	159	N	0	Up	0
16	60	M	ASY	142	216	0	Normal	110	Y	2.5	Flat	1
17	56	M	ASY	115		1	ST	82	N	-1	Up	1
18	44	F	NAP	118	242	0	Normal	149	N	0.3	Flat	0
19	60	M	ASY	136	195	0	Normal	126	N	0.3	Up	0
20	32	M	ATA	125	254	0	Normal	155	N	0	Up	0
21	58	M	ATA	125	220	0	Normal	144	N	0.4	Flat	0
22	37	F	NAP	120	215	0	Normal	170	N	0	Up	0
23	57	M	ASY	140		1	Normal	100	Y	0	Flat	1
24	69	M	ASY	145	289	1	ST	110	Y	1.8	Flat	1
25	53	F	NAP	128	216	0	LVH	115	N	0	Up	0
26	44	M	ATA	120	184	0	Normal	142	N	1	Flat	0
27	34	M	ATA	98	220	0	Normal	150	N	0	Up	0
28	77	M	ASY	125	304	0	LVH	162	Y	0	Up	1
29	74	F	ATA	120	269	0	LVH	121	Y	0.2	Up	0
30	47	M	NAP	108	243	0	Normal	152	N	0	Up	1
31	63	M	ASY	96	305	0	ST	121	Y	1	Up	1
32	56	M	NAP	155		0	ST	99	N	0	Flat	1
33	32	M	TA	95		1	Normal	127	N	0.7	Up	1
34	65	F	ASY	150	225	0	LVH	114	N	1	Flat	1
35	65	M	ASY	144	312	0	LVH	113	Y	1.7	Flat	1

Data Integration and Reduction

Data Integration and Reduction

What is Data Integration?

- The process of combining data from different sources into a unified dataset.
- Ensures consistency and usability for analysis.

What is Data Reduction?

- The process of reducing the size or complexity of the dataset.
- Aims to retain important information while improving efficiency.

Key Techniques:

- Combining datasets (merging, concatenation, joining).
- Feature selection.
- Dimensionality reduction (PCA, t-SNE, UMAP).

Combining Datasets

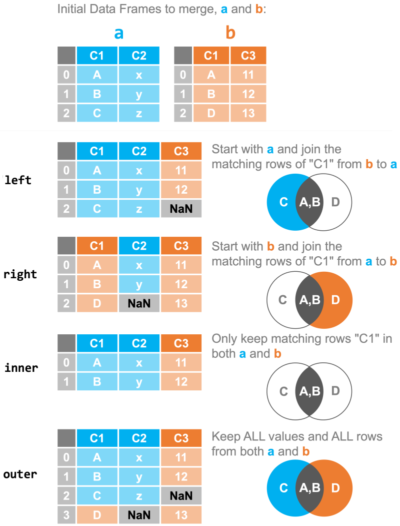
Why Combine Datasets?

- To enrich data by adding more features or samples.

Common Techniques:

- Merging: Combines datasets based on common keys (e.g., primary keys in databases).
- Concatenation: Stacks datasets either row-wise or column-wise.
- Joining: Combines datasets similar to SQL joins (e.g., inner, outer, left, right joins).

Combining Datasets



Source: Combining datasets, from Duke MIDS Practical Data Science course IDS 720 by Kyle Bradbury and Nick Eubank.

Feature Selection

What is Feature Selection?

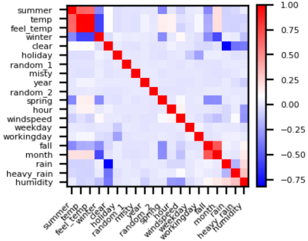
- The process of selecting the most relevant features for the analysis.
- Reduces dimensionality, improves interpretability, and enhances model performance.

Methods for Feature Selection

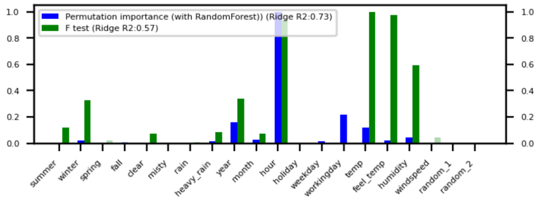
Common Methods:

- **Correlation:** Identifies features highly correlated with the target variable.
- **Variance Threshold:** Removes features with low variance.
- **Feature Importance:** Uses algorithms like Random Forest to rank feature importance.

A heatmap showing feature correlations



A bar chart ranking features by importance



Dimensionality Reduction

What is Dimensionality Reduction?

- Reducing the number of features while preserving important information.

Why is it Important?

- Reduces computational complexity.
- Improves model performance by removing noise.
- Visualizes high-dimensional data.

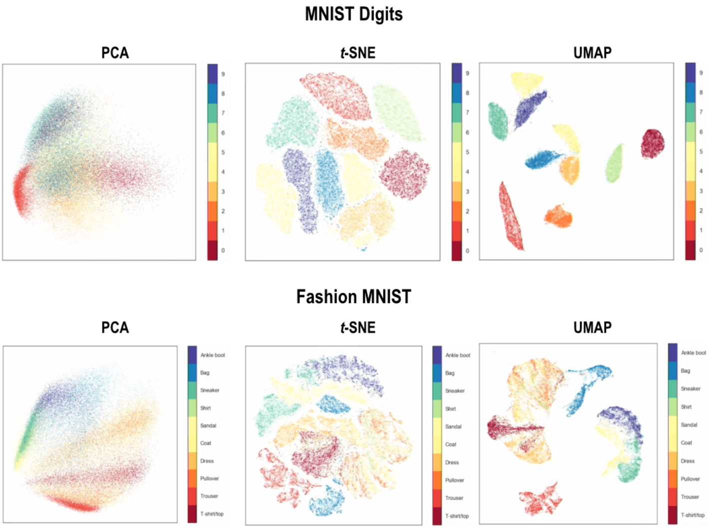
Common Techniques:

- PCA (Principal Component Analysis).
- t-SNE (t-Distributed Stochastic Neighbor Embedding).
- UMAP (Uniform Manifold Approximation and Projection).

Methods for Dimensionality Reduction

- **Principal Component Analysis (PCA):**
  - Projects data onto a lower-dimensional space.
  - Retains maximum variance.
- **t-SNE:**
  - Non-linear dimensionality reduction technique.
  - Focuses on preserving local relationships in data.
- **UMAP:**
  - Preserves local and global data structures.
  - Suitable for clustering and visualization.

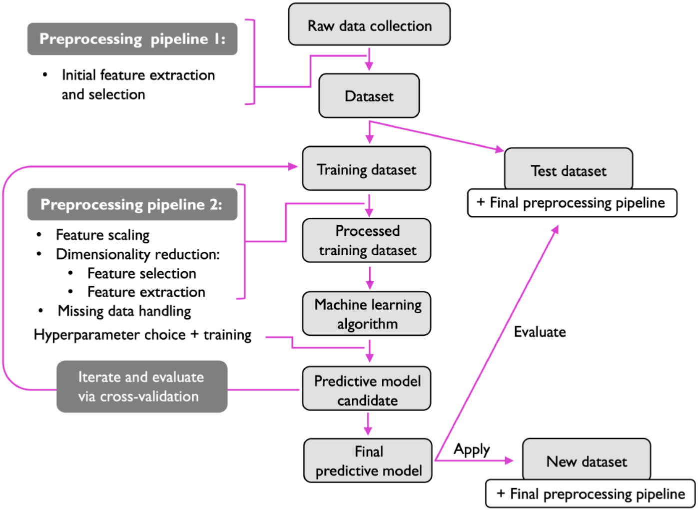
Dimensionality Reduction



Source: https://meta.caspershire.net/umap/

The Complete ML Workflow

The Complete ML Workflow



Source: STAT 451 – Introduction to Machine Learning and Statistical Pattern Classification by Sebastian Raschka