Data Preprocessing

Tourism & AI

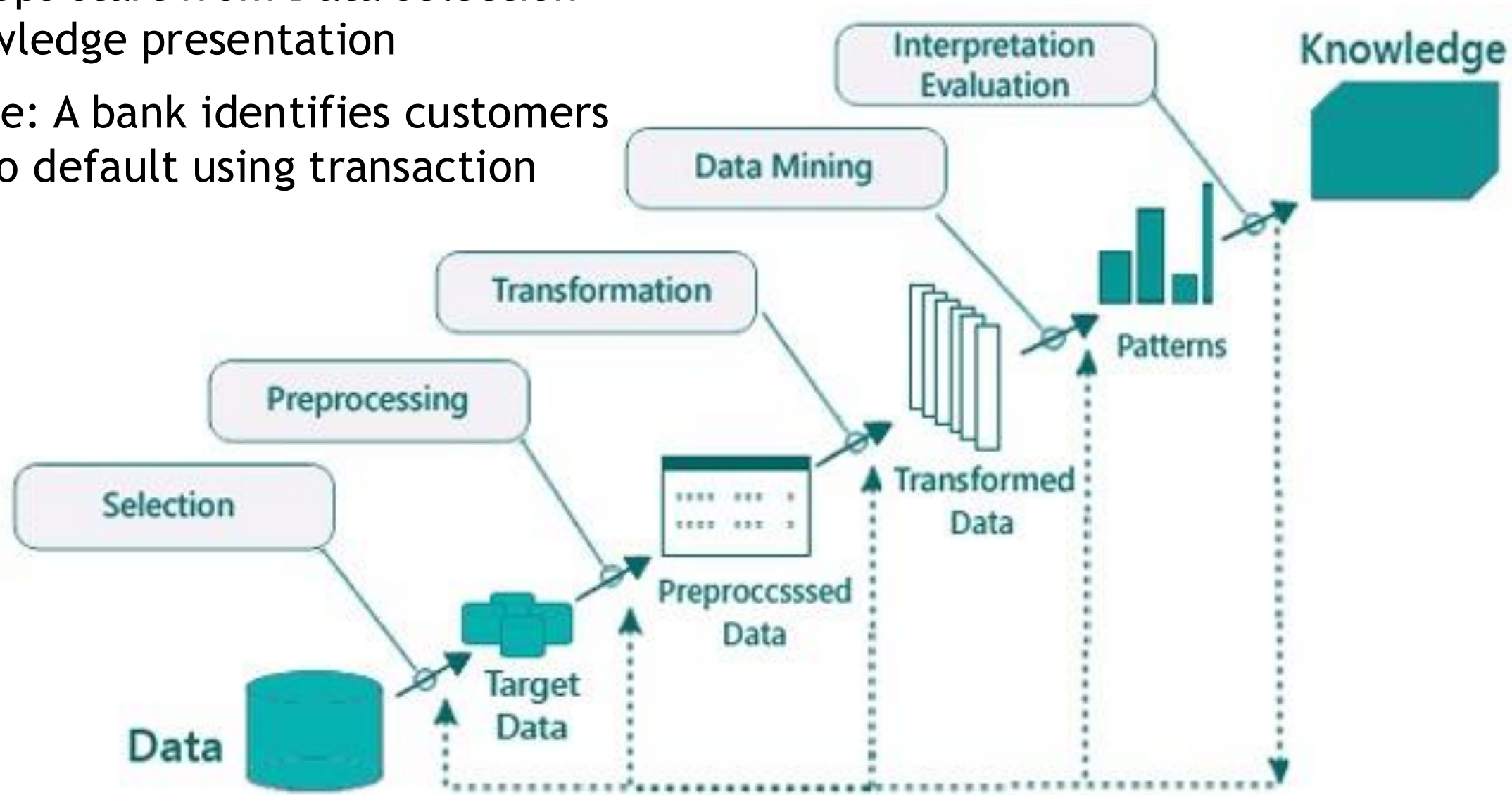
Bidur Devkota, PhD (GCES)

Gandaki University

Pokhara, Nepal

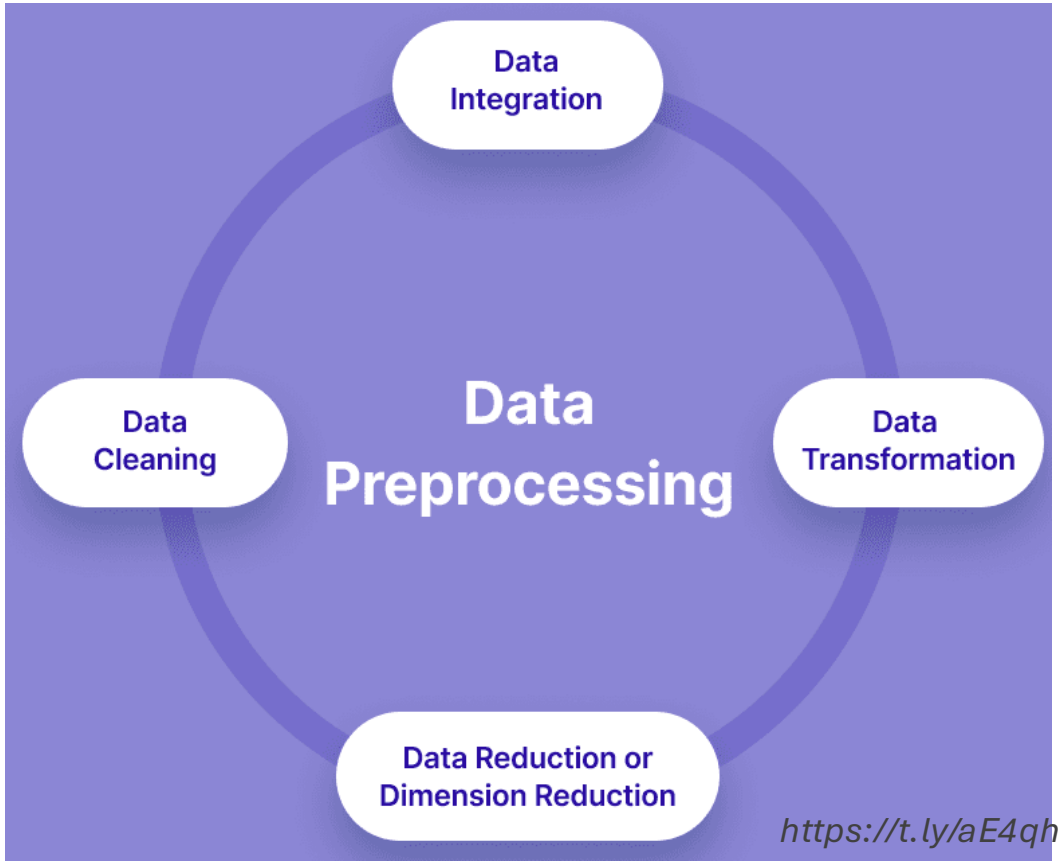
KDD Process

- KDD Steps start from Data selection to Knowledge presentation
- Example: A bank identifies customers likely to default using transaction data



Data Preprocessing

- **Prepares raw data for analysis** by improving quality and efficiency.
- Raw data is often **incomplete**, **noisy**, **redundant**, or **inconsistent**, leading to "garbage in, garbage out."
- Steps:
  - **cleaning**: removing errors
  - **Integration**: merging sources
  - **Transformation**: normalizing
  - **Reduction**: simplifying
  - **Discretization**: binning



Data Preprocessing

Preprocessing may take 50-80% of analysis time but ensures accurate mining results

- **Benefits:**
  - Enhances model accuracy,
  - reduces computational cost,
  - handles real-world data issues.
- **Common tools:**
  - Python (pandas, scikit-learn)
  - R
  - ETL tools: RapidMiner, Talend
  - Etc.

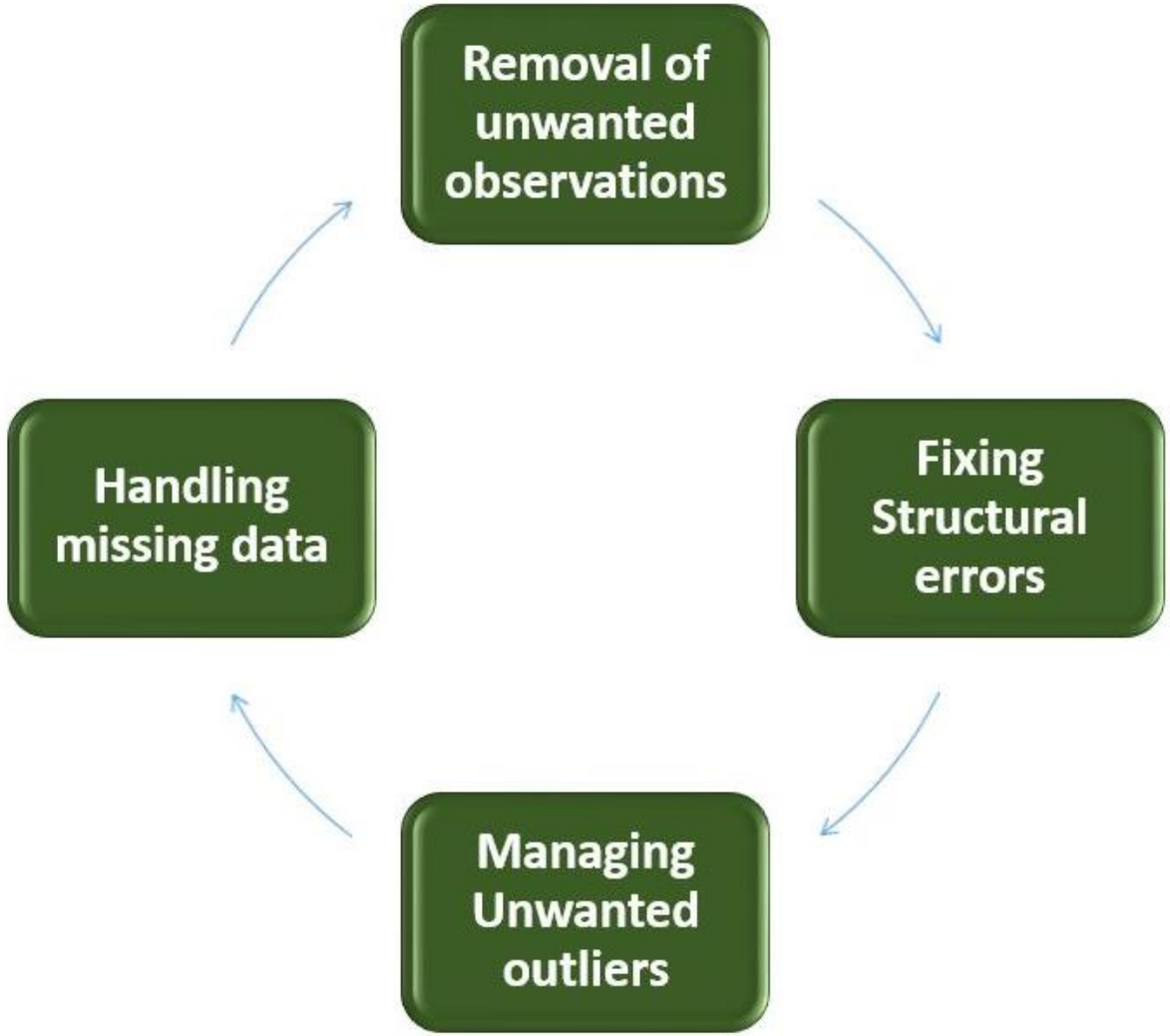
- **Examples:**
  - Survey dataset:
    - fixes missing ages or inconsistent formats (e.g., ”PKR" vs. ”Pokhara")
  - Weather dataset:
    - Updating units of data from multiple station’s sensor readings
      - Celsius to Fahrenheit

Data Preprocessing: Data Cleaning

Missing Values: the values/data that is not present for some variables in the given dataset.

Noisy data: Data containing errors, outliers, or random variations that distort true patterns.

Inconsistent Data: Data that conflicts or disagrees across sources or records (e.g., mismatched values for the same entity)



https://t.ly/t9W2v

Data Preprocessing: Data Cleaning

Handle Missing Values

- **Deletion method-**Ignore the tuple: Not effective if the dataset is small.
- **Imputation Methods:**
  - **Fill manually**: Tedious and infeasible for large datasets.
  - **Use a global constant**: Fill with a value like "Unknown."
  - **Use a measure of central tendency**: Fill with the attribute mean or median.
  - **Use the most probable value**: Infer using regression, KNN, decision trees, etc.
- **Challenge**: Deciding imputation methods without biasing data; handling large-scale noise.
- **Real Example**:
  - Titanic dataset (Kaggle): Clean missing 'Age' values by imputing based on passenger class; remove duplicate entries.

Data Preprocessing: Data Cleaning

Handling Noisy Data (Smoothing)

- **Binning:** Sort data and smooth by consulting neighboring values (e.g., smooth by bin mean, median, or boundaries)
- **Regression:** Fit data to a regression function to smooth out noise.
- **Moving Average Smoothing:** for time-series data to reduce variance.
- **Clustering:** Detect and remove outliers that fall outside of clusters.

Data Preprocessing: Data Cleaning

Correcting Inconsistencies

- Use domain knowledge or expert input to correct inconsistencies.
- Example:
  - Resolve name conflict by Standardizing ("St.", "Street", "Str." -> "St.")
  - Standardizing units (e.g. converting measurement units)
  - Harmonizing formats (e.g., DD-MM-YYYY to YYYY-MM-DD)

Outlier Detection Methods

- Statistical tests (e.g. Z-score, Interquartile Range)
- Clustering techniques (e.g. DBSCAN)
- Distance-based methods (e.g. K-nearest neighbors)

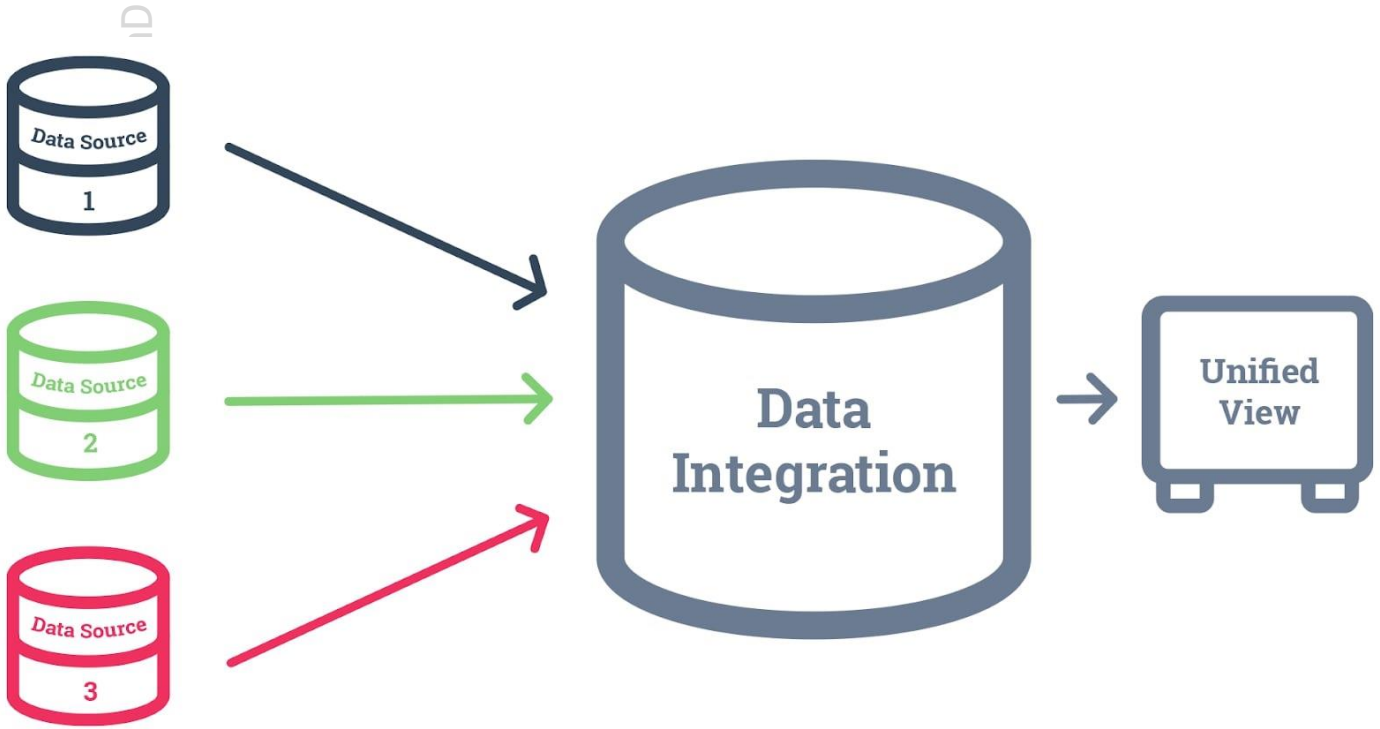
Data Preprocessing: Data Cleaning

Example: A customer dataset has missing/duplicate/unrealistic Income values, and typos in the City field.

- **Missing Income**: Fill with the mean or median income of customers from the same Profession.
- **Noisy Income**: An Income value recorded as "**-50**", fill it with mean or median value.
- **Inconsistent data**: Entries like "Kthmandu", "kathmandu", "Ktm" are all mapped to a standardized value: "Kathmandu".
- **Duplicate data**: remove the duplicate entry

Data Preprocessing: Data Integration

Data Integration: Combining data from multiple sources into a coherent data store, like data warehouse.



Issues: Schema mismatch, redundancy, data conflicts.

Example: Combining customer data from web + mobile + in-store.

- Resolve conflicting customer IDs and eliminate duplicate records.
- Handles schema conflicts (e.g., different attribute names) via metadata

Data Preprocessing: Data Integration Tasks

- **Schema Integration:**
  - Combining metadata from different sources.
  - **Challenge - Entity identification problem:**
    - The same attribute may have different names in different databases
    - e.g. cust_id in one, customer_id in another
- **Redundancy and Correlation Analysis:**
  - An attribute might be redundant if it can be derived from another attribute or set of attributes.
  - If a dataset has both '*age*' and 'dob', then 'age' is redundant because it can be derived as (**current date - dob**)
  - Correlation analysis (e.g.Chi-square test, covariance) can help identify redundant attributes.

Data Preprocessing: Data Integration Tasks

- **Tuple Duplication:**
  - Detecting and merging duplicate records that refer to the same real-world entity
  - e.g. the same customer appearing twice with slightly different spellings of their name
- **Data Value Conflict Detection:**
  - The same real-world entity may have different values in different sources.
  - e.g. weight may be stored in kilograms in one source and pounds in another

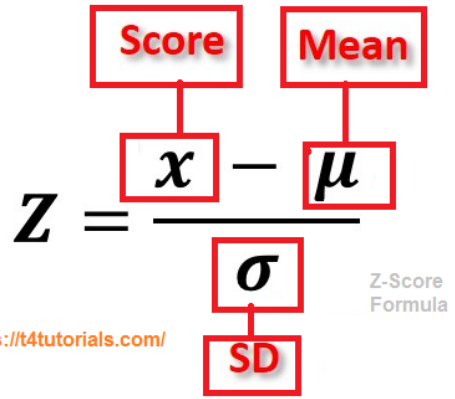
Data Preprocessing: Data Transformation & Discretization

Data Transformation: Convert data into suitable form for analysis.

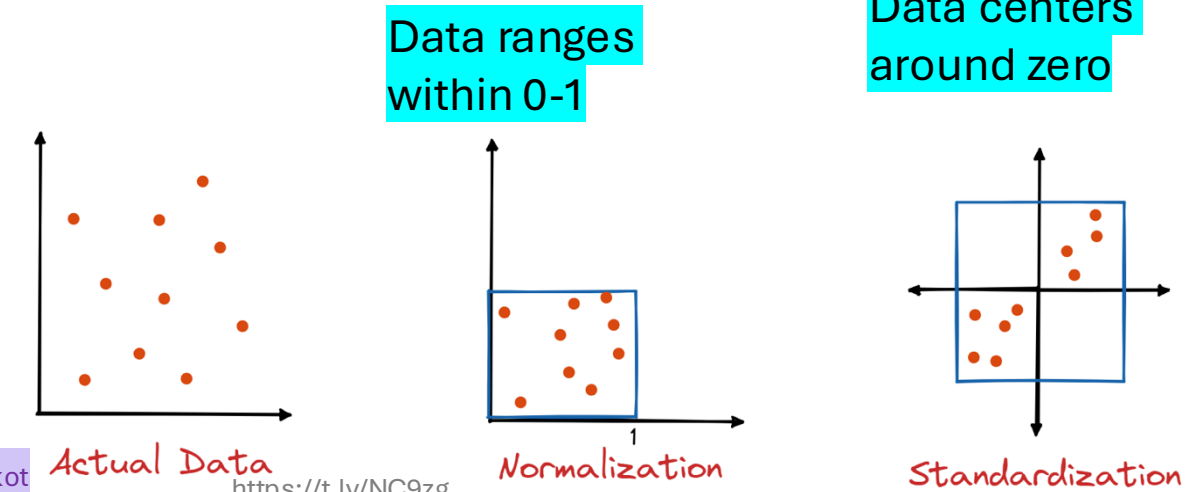
- **Normalization**: Scale values between 0 and 1
  - **Min-Max Normalization:**

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

  - **Z-Score Normalization:**



- **Standardization:**
  - Gaussian distribution: zero mean and unit variance
  - especially useful for algorithms sensitive to scale (e.g., SVM, PCA)



Data Preprocessing: Data Transformation & Discretization

Data Transformation: Convert data into suitable form for analysis.

- **Aggregation**: Summarize data
  - E.g. daily sales data can be aggregated into monthly totals
- **Encoding Categorical Variables:**
  - **One-hot encoding**: Creates binary variables for each category.
  - **Label encoding**: Assigns unique integers to categories (ordinal features).
  - **Binary /hash encoding**: maps categories to integers using a hash function to handle high cardinality variable.

City	Label Encoding	One-Hot Encoding	Hash Encoding
New York	0	[1, 0, 0, 0]	[1, 0, 1]
London	1	[0, 1, 0, 0]	[0, 1, 1]
Tokyo	2	[0, 0, 1, 0]	[1, 1, 0]
Paris	3	[0, 0, 0, 1]	[0, 0, 1]

Table: encoding 'city' column with different methods

Data Preprocessing: Data Transformation & Discretization

Data Transformation: Convert data into suitable form for analysis.

- **Attribute construction:**
  Create new features
  - e.g. creating Area from Length and Width
- **Generalization:** Replacing low-level data (e.g., raw birth dates) with higher-level concepts
  - e.g. Age_Group: "Youth," "Adult," "Senior"

- **Log Transformations:**
  - Reduce skewness in data
    - e.g. income values

Original Income	Log(Income)
30,000	10.31
40,000	10.60
50,000	10.82
1,000,000	13.82

Here, the large gap between values shrinks i.e. skewness is reduced, and the data becomes more normally distributed.

Data Preprocessing: Data Transformation & Discretization

Data Transformation: Convert data into suitable form for analysis.

Examine Skewness?

- The original data has a **huge jump** from 50,000 to 1,000,000, causing **right skewness.**
- Log transformation **compresses large values** more (e.g., 1,000,000 → 13.82 vs 50,000 → 10.82), reducing skewness significantly.
- Square root transformation also reduces skewness by **shrinking large values** (e.g., 1,000,000 → 1000 vs 50,000 → 223.61), but **less aggressively than log.**

Original Income	Log(Income)	Sqrt(Income)
30,000	10.31	173.21
40,000	10.60	200.00
50,000	10.82	223.61
1,000,000	**13.82**	**1000.00**

*Here, the large gap between values (Original Income) shrinks i.e. skewness is reduced, and the data becomes more normally distributed (*but not always perfectly*).*

Data Preprocessing: Data Transformation & Discretization

Data Discretization: Reducing continuous attributes into finite categories/bins

- **Binning**: Top-level split (see Data Cleaning).
- **Histogram Analysis**: Using binning based on value distribution.
- **Clustering**: Grouping values into clusters.

- **Example**:
  - Age data (0-100): Discretize into bins like "Young" (0-30), "Middle" (31-60), "Senior" (61+).

Data Preprocessing: Data Transformation & Discretization

- **Binning** useful for:
  - Reduce model complexity
  - Improve interpretability
  - Building decision trees

- **Supervised Discretization**
  - Discretization guided by class labels of target variable
- **Unsupervised Discretization**
  - Based on the feature's distribution

Binning method	Bin 1	Bin 2	Bin 3	Comments
Equal-width	12, 15, 18, 21, 22, 25	28, 35	40, 50	Divide the range into intervals of equal width. May produce uneven counts if data is skewed.
Equal-frequency	12, 15, 18	21, 22, 25	28, 35, 40, 50	Each bin has (roughly) the same number of records. Maintains balance across bins.
Cluster-based	12, 15, 18	21, 22, 25, 28	35, 40, 50	Uses clustering (e.g., k-means) to group similar values. Flexible, data-driven.
Entropy-based	12, 15, 18, 21, 22	25, 28, 35, 40, 50	–	Splits to maximize information gain (minimize entropy) for a target variable.

Data Preprocessing: Data Transformation & Discretization

- **Example**: A dataset for a classification task has a continuous attribute Age (ranging from 18 to 80) and Income (ranging from 20,000 to 200,000).
  - **Normalization**:
    - Normalize Income using Min-Max to a [0,1] scale so it doesn't dominate over Age in a distance-based algorithm.
  - **Discretization**:
    - Age can be discretized into [18-30), [30-45), [45-60), [60+)].
    - Income can be discretized into ["Low", "Medium", "High"] based on percentile bins.

References

- Tan, P.N., Steinbach, M. and Kumar, V.,, 2006. Introduction to data mining. Pearson Education, Inc. 3.
- Han, J., Kamber, M. and Mining, D., 2006. Concepts and techniques. Morgan Kaufmann
- Acknowledge the assistance of LLMs like Gemini(Google),DeepSeek, ChatGPT(OpenAI) for helping to refine/generate some content for this work.