🧮 Handling Missing Data | Numerical Data | Simple Imputer

When your dataset has **missing numbers** (like age, salary, etc.), you can't feed it into a model directly. Instead, we **fill in** those gaps using **imputation**.

➢ Introduction:

- Introduction to the importance of handling **missing values** in numerical data.
- Overview of **why we shouldn't drop rows** casually — we want to preserve data.

➢ Handling Missing Numerical Data

- You'll typically see NaN values in your numeric columns.
- Goal: fill missing values without introducing bias or damaging the dataset's integrity.

➢ Mean / Median Imputation

The most common methods:

Method	**When to Use**
Mean	Use when data is **normally distributed**
Median	Use when data has **outliers or is skewed**

- **Example:**
  1. Original: [20, NaN, 25, 30]
  2. After mean imputation: [20, 25, 25, 30]

Median is more robust to **extreme values (outliers)**.

➢ Arbitrary Value Imputation

- Instead of using mean/median, fill missing values with a **fixed number**, like 9999 or -1.
- Useful for models that can learn "something's missing" from that value.
- Example:
  1. [20, NaN, 25] → [20, -9999, 25]

Risk: Models might **misinterpret** this value as a real pattern.

➢ End of Distribution Imputation

- Replaces missing values with a number **at the extreme end** of the column's distribution.

- Example:
    1. Max value of column = 85
    2. Fill NaNs with 90 or 100
- Tells the model: “This value was missing — it’s special.”

Often used in tree-based models (like Random Forest or XGBoost) which can handle weird distributions.

➢ Summary:

- Missing data must be handled before modeling.
- Choose imputation strategy based on the **data’s distribution and model sensitivity**.
- Mean/Median is safe and simple.
- Use SimpleImputer in pipelines for production-ready ML.
