

# **Data Science**



**By:**  
**Dr. Shikha Deep**

# Data Preparation

- Handling Missing Data (`dropna()`, `fillna()`)
- Handling Duplicates (`drop_duplicates()`)
- Handling Outliers (IQR method, Z-score)
- Data Scaling & Normalization (`StandardScaler`, `MinMaxScaler`)
- Handling Imbalanced Data (basic intro to oversampling/undersampling/SMOTE)

# 1. Handling Missing Data

- Detecting missing values (isnull, info)
- Removing missing values (dropna)
- Filling missing values (fillna with mean, median, mode, forward/backward fill)

## 2. Methods to Handle Missing Data

### (a) Detect Missing Data

- To check where values are missing.
- Method: **isnull()** or **info()**

### (b) Drop Missing Data

- Use when missing values are **very few** and won't affect results.
- Method: **dropna()**

## (c) Fill Missing Data (Imputation)

- Replace missing values with some logic:
  - **Mean** (good for numeric data, balanced values)
  - **Median** (good for numeric data with outliers)
  - **Mode** (good for categorical data)
  - **Forward Fill / Backward Fill** (use previous/next value)

CODE 1:

Name	Maths	Science	English
Amit	80	75	82
Neha	90	NaN	78
Raj	NaN	88	85
Simran	70	60	NaN
Ali	85	NaN	89

```
import pandas as pd
import numpy as np

data = {
    "Name": ["Amit", "Neha", "Raj", "Simran", "Ali"],
    "Maths": [80, 90, np.nan, 70, 85],
    "Science": [75, np.nan, 88, 60, np.nan],
    "English": [82, 78, 85, np.nan, 89]
}

df = pd.DataFrame(data)
print("Original DataFrame:\n", df)
```

## **# Detect Missing Data**

```
print("\nCheck missing values:\n", df.isnull().sum())  
print("\nData info:")  
print(df.info())
```

## **# Drop Missing Data**

```
df_drop = df.dropna()  
print("\nAfter dropping missing values:\n", df_drop)
```



## # Fill Missing Data

### (1) Fill with mean

```
df_mean = df.fillna(df.mean(numeric_only=True))  
print("\nFill with Mean:\n", df_mean)
```

### (2) Fill with median

```
df_median = df.fillna(df.median(numeric_only=True))  
print("\nFill with Median:\n", df_median)
```

### (3) Fill with mode (for categorical or repeated values)

```
df_mode = df.fillna(df.mode().iloc[0])  
print("\nFill with Mode:\n", df_mode)
```

#### **(4) Forward Fill**

```
df_ffill = df.fillna(method="ffill")  
print("\nForward Fill:\n", df_ffill)
```

#### **(5) Backward Fill**

```
df_bfill = df.fillna(method="bfill")  
print("\nBackward Fill:\n", df_bfill)
```

## NOTE:

(1) Categorical fields (e.g., City, Gender) or numeric with clear repeated values.

- **Important:** If all values are unique, `Series.mode()` returns all values; in a DataFrame, `df.mode().iloc[0]` will pick the smallest among them. That's why this can look odd for numeric columns with no repeats.

(2)

Original DataFrame → Shows some NaN values.

`isnull().sum()` → Count of missing values in each column.

`dropna()` → Removes rows with any missing values.

`fillna(mean/median/mode)` → Replaces NaN with computed values.

`forward/backward fill` → Copies nearest values.

### Flowchart: Choosing Missing Value Imputation Method



