
📘 Day 4: Data Cleaning and Preprocessing

🎯 Objective

Before building any Machine Learning model, it's essential to **clean** and **prepare** your data properly. Most real-world datasets are messy — they contain:

- Missing values
- Duplicates
- Categorical text data
- Skewed numerical values

This day will teach you how to **detect and fix** these problems step-by-step using Python libraries like `pandas` and `sklearn`.

🧠 Topics Explained

1. 🔍 Missing Data

Sometimes, data might be **missing** in some columns (like someone not reporting their age). Common strategies to handle this:

- **Remove** the rows or columns with missing values
- **Fill (impute)** missing values with:
    - Mean/Median (for numeric data)
    - Mode (for categorical data)

2. 📑 Duplicates

- Some rows in the dataset may be **exact copies** of others — these are usually errors and should be **removed**.

3. 🏷️ Encoding Categorical Variables

- Machine Learning algorithms **can't handle text labels** (e.g., 'male', 'female', 'yes', 'no'). So, you need to **convert** them into numbers.
- Methods:
    - **Label Encoding** – Assigns a number to each label (e.g., male=1, female=0)
    - **One-Hot Encoding** – Creates a new column for each category with 1/0

4. ⚖️ Feature Scaling

- Your features (like income, age, height) might have **very different scales**.
- Scaling helps make the training process **faster** and **more accurate**.
- Methods:
    - **StandardScaler**: converts values to mean 0 and standard deviation 1
    - **MinMaxScaler**: scales values between 0 and 1

💼 Real-World Dataset: Titanic

The Titanic dataset is a classic beginner dataset. It includes passenger details like:

- Age
- Sex
- Fare paid
- Survival status
- Cabin, class, and more

We'll use it to demonstrate cleaning techniques.

🎮 Game-Based Learning Activities

1. 🎲 **Find the Odd Passenger**

- After scaling the data, check which passengers have **unusually high or low values** for `fare` or `age`.
- Objective: Detect "outlier" passengers and try to understand why they're different.

2. 🔍 **Missing Data Detective**
   - Identify which columns have missing values.
   - Fill them using different strategies (e.g., median, mode).
   - Try different approaches and see how the dataset changes.
3. 🕹️ **Encode the Survivor**
   - Convert the `sex`, `embarked`, and other categorical columns to numbers.
   - Create a simple guessing game to predict survival based on encoded features.

✅ 1. From the `seaborn` library (recommended for beginners)

You can directly load the Titanic dataset with **1 line of code** using `seaborn`:

```
import seaborn as sns
df = sns.load_dataset('titanic')
df.head()
```

Pros:

- No file download needed
- Clean and beginner-friendly format

✅ 2. From Kaggle

Kaggle is a great place for real ML competitions and datasets.

📦 **Dataset link**:
👉 https://www.kaggle.com/c/titanic/data

You'll get:

- `train.csv`
- `test.csv`
- `gender_submission.csv`

How to use:

1. Sign in to Kaggle
2. Go to the Titanic dataset page
3. Click **Download All**
4. Use this code to read:

```
import pandas as pd
df = pd.read_csv("train.csv")
df.head()
```

✅ 3. From GitHub (Raw CSV URL)

You can load it directly using a raw URL from GitHub:

```
url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"
df = pd.read_csv(url)
df.head()
```

No need to download manually, works well in notebooks.

Summary of Options

Source	Code Example	Notes
Seaborn	`sns.load_dataset('titanic')`	Easiest for practice
Kaggle	`pd.read_csv("train.csv")`	Good for full ML pipeline
GitHub (raw)	`pd.read_csv("raw_url")`	Easy for online notebooks
