**📌 Data Cleaning Case Study: Employee Performance Data**

**📖 Scenario:**

A company has collected employee performance data but has noticed several **data quality issues**. Your task is to clean the dataset and prepare it for analysis using **Pandas**.

**📂 Dataset: employee\_data\_dirty.csv**

**📌 Sample Data (Before Cleaning)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Employee\_ID | Name | Age | Department | Salary | Working\_Hours | Projects\_Completed | Performance\_Rating |
| 101 | Alice | 29 | HR | 50000 | 40 | 5 | 4.2 |
| 102 | Bob | -35 | IT | 70000 | 50 | 8 | NaN |
| 103 | Charlie | 28 | Sales | 45000 | ? | 4 | 3.9 |
| 104 | NaN | 40 | IT | NaN | 55 | 10 | 4.9 |
| 105 | Eve | 32 | Marketing | 60000 | 42 | NaN | 4.3 |
| 106 | Alice | 29 | HR | 50000 | 40 | 5 | 4.2 |

**🔎 Data Cleaning Tasks**

Using **Pandas**, perform the following cleaning tasks:

**Step 1: Handling Missing Values**

1. Identify missing values in the dataset.
2. Fill missing **Performance\_Rating** with the department’s average rating.
3. Replace missing **Salary** with the department’s median salary.
4. If an employee's **Projects\_Completed** is missing, fill it with 0.

**Step 2: Fixing Incorrect & Inconsistent Data**

1. Convert **Working\_Hours** column values like ? to NaN and fill them with the mean working hours.
2. Fix negative values in the **Age** column by taking absolute values.
3. Standardize **Department Names** to lowercase (e.g., "HR" → "hr").
4. Remove duplicate records (e.g., **Alice appears twice**).

**Step 3: Formatting & Final Checks**

1. Convert **Salary** into integer values (if stored as float).
2. Save the cleaned dataset as cleaned\_employee\_data.csv.

**📌 Case Study 2: Cleaning Customer Transaction Data**

**📖 Scenario:**

A retail store has collected customer transaction data, but it contains errors such as **missing values, incorrect formats, and duplicate entries**. Your task is to clean the dataset using **Pandas** to prepare it for further analysis.

**📂 Dataset: customer\_transactions\_dirty.csv**

**📌 Sample Data (Before Cleaning)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Transaction\_ID | Customer\_ID | Name | Age | Gender | Purchase\_Amount | Payment\_Method | Purchase\_Date |
| TXN001 | C001 | John Doe | 25 | Male | 120.50 | Credit Card | 2024-02-10 |
| TXN002 | C002 | Jane Doe | ? | Female | -200.00 | Cash | 10/02/2024 |
| TXN003 | C003 | NaN | 35 | Male | 500.75 | PayPal | 2024-02-12 |
| TXN004 | C001 | John Doe | 25 | Male | 120.50 | Credit Card | 2024-02-10 |
| TXN005 | C004 | Sam Smith | 28 | ? | NaN | Debit Card | 2024-02-13 |
| TXN006 | | C005 | Alice | 22 | Female | 99.99 | ? | 2024-02-14 |

**🔎 Data Cleaning Tasks**

Using **Pandas**, perform the following steps:

**Step 1: Handling Missing & Incorrect Data**

1. Replace ? in **Age and Gender** with appropriate values (mean age, most common gender).
2. Fill missing **Purchase\_Amount** with the median purchase amount.
3. Convert **negative Purchase\_Amount** values to positive.
4. Replace missing **Payment\_Method** values with "Unknown".

**Step 2: Standardizing Formats**

1. Convert all **Purchase\_Date** values into a proper date format (YYYY-MM-DD).
2. Convert **Payment\_Method** values to lowercase for consistency (e.g., "Credit Card" → "credit card").

**Step 3: Removing Duplicates & Final Adjustments**

1. Remove duplicate **Transaction\_ID** entries.
2. Ensure **Purchase\_Amount** is stored as a float with two decimal places.
3. Save the cleaned dataset as cleaned\_customer\_transactions.csv.

**📌 Case Study 3: Cleaning Online Course Enrollment Data**

**📖 Scenario:**

An online education platform tracks student enrollments but has **inconsistent data entries**. Your task is to clean the dataset using **Pandas** and make it suitable for analysis.

**📂 Dataset: course\_enrollment\_dirty.csv**

**📌 Sample Data (Before Cleaning)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enrollment\_ID | Student\_ID | Name | Age | Gender | Course\_Name | Enrollment\_Date | Course\_Fee | Status |
| ENR001 | S001 | John Doe | 21 | Male | Python Basics | 2024-01-10 | 200 | Enrolled |
| ENR002 | S002 | Jane Doe | ? | Female | Data Science | 10/01/2024 | -500 | Completed |
| ENR003 | S003 | NaN | 23 | Male | Web Development | 2024-01-12 | 300 | Pending |
| ENR004 | S004 | Sam Smith | 20 | ? | Python Basics | 2024-01-10 | NaN | Enrolled |
| ENR005 | S001 | John Doe | 21 | Male | Python Basics | 2024-01-10 | 200 | Enrolled |
| ENR006 | S005 | Alice | 22 | Female | Data Science | 2024-01-14 | ? | Completed |

**🔎 Data Cleaning Tasks**

**Step 1: Handling Missing & Incorrect Data**

1. Replace ? in **Age and Gender** with appropriate values (mean age, most common gender).
2. Fill missing **Course\_Fee** values with the course's average fee.
3. Convert **negative Course\_Fee** values to positive.

**Step 2: Standardizing Formats**

1. Convert all **Enrollment\_Date** values into a proper date format (YYYY-MM-DD).
2. Ensure **Status** values follow a uniform format ("Enrolled", "Completed", "Pending").

**Step 3: Removing Duplicates & Final Adjustments**

1. Remove duplicate **Enrollment\_ID** entries.
2. Ensure **Course\_Fee** is stored as an integer.
3. Save the cleaned dataset as cleaned\_course\_enrollment.csv.