```
import pandas as pd
import numpy as np

# Load the dataset
data = pd.read_csv('your_dataset.csv')  # Replace with your filename
print("Initial shape:", data.shape)

# ----- STEP 1: Check and handle missing values -----
print("\nMissing values per column:")
print(data.isnull().sum())

# Fill missing values (Example: numeric with median, object with mode)
for col in data.columns:
    if data[col].dtype == 'object':
        data[col].fillna(data[col].mode()[0], inplace=True)
    else:
        data[col].fillna(data[col].median(), inplace=True)

# ----- STEP 2: Remove duplicates -----
data.drop_duplicates(inplace=True)
print("\nShape after removing duplicates:", data.shape)

# ----- STEP 3: Standardize categorical text values -----
# Example: Cleaning gender column
if 'Gender' in data.columns:
```

```
    data['Gender'] = data['Gender'].str.strip().str.lower()
    data['Gender'] = data['Gender'].replace({
        'm': 'male', 'male': 'male',
        'f': 'female', 'female': 'female'
    })

# ----- STEP 4: Convert date columns -----
# Replace 'date_column' with your actual column name
date_cols = ['date_of_birth', 'join_date', 'appointment_date']  # Example names
for col in date_cols:
    if col in data.columns:
        data[col] = pd.to_datetime(data[col], errors='coerce')

# ----- STEP 5: Fix column names -----
data.columns = [col.strip().lower().replace(' ', '_') for col in data.columns]

# ----- STEP 6: Correct data types -----
if 'age' in data.columns:
    data['age'] = pd.to_numeric(data['age'], errors='coerce').astype('Int64')

# ----- STEP 7: Handle outliers (e.g., using IQR method) -----
# Example for 'age' column
if 'age' in data.columns:
    Q1 = data['age'].quantile(0.25)
    Q3 = data['age'].quantile(0.75)
```

```
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    data = data[(data['age'] >= lower_bound) & (data['age'] <= upper_bound)]
    print("\nShape after outlier removal:", data.shape)


# ----- STEP 8: Final summary -----
print("\nFinal dataset info:")
print(data.info())


# Save cleaned data
data.to_csv('cleaned_output.csv', index=False)
print("\n Cleaned data saved to 'cleaned_output.csv'")
```

explanation :

1. Load the messy file

```
python
CopyEdit
data = pd.read_csv('your_dataset.csv')
```

 It opens your Excel or CSV file so Python can work with it.

---

2. Find missing values

```
python
CopyEdit
data.isnull().sum()
```

 It checks if any data is **missing**, like empty cells (e.g., someone didn't write their age or gender).

3. Fill in missing data

```
python
CopyEdit
fillna() using mode or median
```

 If something's missing:

- If it's **text** (like gender), it fills it with the most common word (like "Male" or "Female").
- If it's a **number** (like age), it fills it with the middle value (called median).

4. Remove duplicate rows

```
python
CopyEdit
data.drop_duplicates()
```

 If someone's name or data was copied twice, it removes the extra copy.

5. Fix text like Gender

```
python
CopyEdit
data['Gender'] = ...
```

 If Gender is written like `M`, `Male`, `male`, it makes all of them say just `male`. Same for `F`, `Female`, etc. It keeps things **consistent**.

6. Fix dates

```
python
CopyEdit
pd.to_datetime()
```

 If dates are written differently (e.g., `01-02-2022` or `2022/02/01`), it makes all dates follow one standard format (like `2022-02-01`).

7. Rename column titles

```
python
CopyEdit
data.columns = ...
```

 It changes column names like `Customer Name` → `customer_name` — all lowercase, no spaces.

8. Fix number formats

```
python
CopyEdit
data['age'] = ...
```

 Makes sure that numbers like age are treated as numbers (not as text).

9 . Remove strange or extreme values

```
python
CopyEdit
IQR method
```

 If someone wrote age as `200`, it probably is a mistake. The code removes those weird entries by checking for **outliers** — values that are too far from the usual range.

10. Save the cleaned file

```
python
CopyEdit
data.to_csv('cleaned_output.csv')
```

 After all cleaning is done, it saves the new file with a new name like **`cleaned_output.csv`**, ready for analysis or graphs!