1. **Missing values** are simply gaps in the data (e.g., `NaN` in Pandas). I usually start by checking how many there are with `df.isnull().sum()`. Depending on the situation, I'll either drop rows or columns with too many nulls (`df.dropna()`), or I'll impute missing entries—using the column's mean/median/mode or a forward-fill method (`df.fillna()`).

2. **Duplicate records** often sneak in when data is merged or collected twice. I spot them with `df.duplicated()` and remove the extras via `df.drop_duplicates()`. If only certain fields define a duplicate, I'll pass those as a subset to `drop_duplicates()` so I keep the first (or last) valid entry.

3. **`dropna()` vs. `fillna()`**

    - `dropna()` straight-up removes rows or columns that contain nulls.

    - `fillna()` replaces those nulls with a value or method (mean, median, forward-fill, etc.).
      I choose `dropna()` when losing data is acceptable, and `fillna()` when I need to preserve as much information as possible.

4. **Outlier treatment** is about handling extreme values that could skew analyses or models. I first detect outliers via the IQR method or z-scores, then either remove them, cap them at a percentile (winsorization), or apply transformations (like log). Cleaning outliers helps keep my summary stats and models robust.

5. **Standardizing data** means re-scaling numeric features so they have zero mean and unit variance. I typically use Scikit-learn's `StandardScaler()`—fit it on my training data, transform it, and then my K-Means or KNN algorithms perform much better because all features live on the same scale.

6. **Inconsistent data formats** (dates, strings) get messy fast. For dates, I convert columns with `pd.to_datetime()`, specifying the exact format if needed. For text fields (like gender), I `.str.lower().str.strip()` everything so "Male", "male " and " MALE" all become "male."

7. **Common data cleaning challenges** I run into:

    - Mixed data types in a column (numbers stored as strings)

    - Typos and inconsistent labeling in categorical fields

- High cardinality or sparse categories
- Hidden missing values represented by placeholders (e.g., “N/A,” “?”)
- Memory limits on huge datasets

8. **Checking data quality** is ongoing. I look at `df.info()`, `df.describe()`, and value counts for each column. I visualize distributions with histograms or boxplots to spot oddities. And for a quick, automated overview, I often run a profiling report (e.g., with `pandas_profiling`) to highlight missingness, correlations, and potential issues.

Interview Questions Related To Above Task:

1.What are missing values and how do you handle them?

2.How do you treat duplicate records?

3.Difference between dropna() and fillna() in Pandas?

4.What is outlier treatment and why is it important?

5.Explain the process of standardizing data.

6.How do you handle inconsistent data formats (e.g., date/time)?

7.What are common data cleaning challenges? 8.How can you check data quality?