Data Cleaning Guide

Step 1: Load the Data

```
import pandas as pd

# Load the dataset
df = pd.read_csv("your_dataset.csv")  # Replace with your file path
print(df.head())
```

Step 2: Understand the Data

```
# Basic info
print(df.info())
print(df.describe())
print(df.columns)
```

Step 3: Handle Missing Values

```
# Check for missing values
print(df.isnull().sum())

# Drop rows or columns
df = df.dropna()  # drops rows with any missing value
# OR fill missing values
```

```
df['column_name'].fillna(df['column_name'].mean(), inplace=True)
```

Step 4: Handle Duplicates

```
# Check for duplicates
print(df.duplicated().sum())

# Remove duplicates
df = df.drop_duplicates()
```

Step 5: Handle Data Types

```
# Convert data types if needed
df['date_column'] = pd.to_datetime(df['date_column'])
df['numeric_column'] = df['numeric_column'].astype(float)
```

Step 6: Encode Categorical Variables

```
# One-hot encoding
df = pd.get_dummies(df, columns=['categorical_column'])

# Label encoding
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['label_column'] = le.fit_transform(df['label_column'])
```

Step 7: Normalize/Scale Data

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
df[['feature1', 'feature2']] = scaler.fit_transform(df[['feature1', 'feature2']])
```

Step 8: Visualize the Data

```
import matplotlib.pyplot as plt
import seaborn as sns

# Correlation heatmap
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.show()

# Boxplot to see outliers
sns.boxplot(x=df['feature_column'])
plt.show()
```

1. Data Cleaning

Definition:

The process of detecting and correcting (or removing) inaccurate, inconsistent, or incomplete data from a dataset to improve its quality.

Why it matters:

Raw data often has errors like typos, duplicates, or missing values. Cleaning makes the data suitable for analysis or modeling.

Examples:

Removing duplicate rows

Fixing incorrect or inconsistent values

Handling missing data

---

2. Handling Nulls (Missing Values)

Definition:

Dealing with empty or missing values in your dataset (shown as NaN in Python).

Methods:

Remove: Drop rows/columns with missing values.

Impute: Fill missing values with:

Mean/Median (for numerical data)

Mode (for categorical data)

A specific value (like 0 or "Unknown")

Example:

```
df['age'].fillna(df['age'].mean(), inplace=True)
```

---

3. Encoding (Categorical Encoding)

Definition:

Converting categorical (non-numeric) data into numeric form so that machine learning models can process them.

Common types:

Label Encoding: Assigns a unique number to each category.

One-Hot Encoding: Creates separate binary (0/1) columns for each category.

Example (One-Hot):

pd.get_dummies(df['gender'])

---

4. Feature Scaling

Definition:

Adjusting the values of numeric features to a common scale without distorting differences in the ranges.

Why it's important:

Some algorithms (like KNN, SVM, Gradient Descent) are sensitive to the scale of input data.

Common methods:

Min-Max Scaling: Scales data to a [0,1] range.

Standardization (Z-score): Mean = 0, Std = 1

Example (Min-Max Scaling):

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df[['age', 'income']] = scaler.fit_transform(df[['age', 'income']])
```