

import pandas as pd

* To load the **pandas** library and gives it the short name **pd**

data = pd.read\_csv("C:\\Users\\Dell\\Downloads\\Superstore Sales Dataset.csv")

data.head()

* To load a CSV file into a DataFrame named data and displays the first 5 rows of the dataset.

data.info()

data.isnull().sum()

* data.info() :Overview of dataset structure.

We discovered that order date and ship date are string data type, And postal code is float.

* data.isnull().sum() :Count of missing values per column.

We discovered 11 nulls in postal code

data["Postal Code"] = data["Postal Code"].fillna("53401")

* To **fill missing values in the "Postal Code" column with "53401"** to ensure there are no blanks in that column.

This postal code is for Burlington, Vermont

data.isnull().sum()

* To make sure that count of nulls is 0 after filling it

data["Postal Code"] = data["Postal Code"].astype(str)

* This line **converts all postal codes to strings**, which is a common and safe practice for working with non-numeric identifiers

data["Order Date"] = data["Order Date"].astype("datetime64[ns]")

data["Ship Date"] = data["Ship Date"].astype("datetime64[ns]")

* These lines **convert "Order Date" and "Ship Date" to datetime format**, enabling accurate date-based operations

data["Order Date"] = data["Order Date"].dt.strftime("%Y-%m-%d")

data["Ship Date"] = data["Ship Date"].dt.strftime("%Y-%m-%d")

* These lines **format the dates as strings like "2025-02-15"**, making them cleaner for viewing.

data["Order Date"] = data["Order Date"].astype("datetime64[ns]")

data["Ship Date"] = data["Ship Date"].astype("datetime64[ns]")

* These lines **convert "Order Date" and "Ship Date" to datetime format**, enabling accurate date-based operations.

duplicate\_count = data.duplicated().sum()

print(f"Number of duplicate rows: {duplicate\_count}")

* To get the total count of duplicate rows

data.info()

* To make sure everything is adjusted

cleaned\_file\_path = "D:\Superstore\_Sales\_Cleaned.csv"

data.to\_csv(cleaned\_file\_path, index=False)

print(f"Cleaned dataset saved to: {cleaned\_file\_path}")

* Ensures that the cleaned data is **saved** in a **reliable format** (CSV) and provides **feedback to the user** about the save location. It is essential for **data persistence**, **sharing**, and **future use**, making it a critical step in the data processing workflow.