**Project Title: SuperStore Sales Analysis**

**Tools Used:**

* **NumPy** – for numerical computations
* **Pandas** – for data manipulation
* **Matplotlib** – for data visualization

**Dataset Description:**

Assume you're working with a dataset like SuperStore\_Sales.csv, with the following columns:

Order\_ID, Order\_Date, Ship\_Date, Customer\_Name, Segment, Country, City, State,

Region, Product\_ID, Category, Sub\_Category, Product\_Name, Sales, Quantity,

Discount, Profit

I can generate this file for you if needed.

**Step-by-Step Breakdown**

**STEP 1: Load Libraries and Dataset**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

# Load Dataset

df = pd.read\_csv('SuperStore\_Sales.csv')

pd.set\_option('display.max\_columns', None) # Show all columns

print(df.head())

**STEP 2: Data Inspection**

print(df.info())

print("\nMissing values:\n", df.isnull().sum())

print("\nSummary Statistics:\n", df.describe(include='all'))

**STEP 3: Data Cleaning**

# Convert date columns

df['Order\_Date'] = pd.to\_datetime(df['Order\_Date'])

df['Ship\_Date'] = pd.to\_datetime(df['Ship\_Date'])

# Fill missing values if any

df['Profit'] = df['Profit'].fillna(df['Profit'].mean())

# Remove duplicates

df.drop\_duplicates(inplace=True)

# Add new features

df['Year'] = df['Order\_Date'].dt.year

df['Month'] = df['Order\_Date'].dt.month

df['Order\_Month'] = df['Order\_Date'].dt.to\_period('M')

**STEP 4: NumPy-based Calculations**

# Total Revenue

total\_sales = np.sum(df['Sales'])

# Average Discount

avg\_discount = np.mean(df['Discount'])

# Total Orders

total\_orders = df['Order\_ID'].nunique()

print(f"Total Revenue: ₹{total\_sales}")

print(f"Average Discount: {avg\_discount \* 100:.2f}%")

print(f"Total Orders: {total\_orders}")

**STEP 5: Data Analysis with Pandas**

**Top 10 Products by Sales**

top\_products = df.groupby('Product\_Name')['Sales'].sum().sort\_values(ascending=False).head(10)

**Sales by Region**

region\_sales = df.groupby('Region')['Sales'].sum()

**Monthly Sales Trend**

monthly\_sales = df.groupby('Order\_Month')['Sales'].sum()

**Category-Wise Profit & Quantity**

category\_analysis = df.groupby('Category')[['Profit', 'Quantity']].sum()

**Shipping Delay Analysis**

df['Shipping\_Days'] = (df['Ship\_Date'] - df['Order\_Date']).dt.days

shipping\_delay = df.groupby('Region')['Shipping\_Days'].mean()

**STEP 6: Data Visualization (Matplotlib)**

**1. Monthly Sales Line Plot**

monthly\_sales.plot(kind='line', marker='o', title='Monthly Sales Trend')

plt.xlabel('Month')

plt.ylabel('Sales (INR)')

plt.grid(True)

plt.tight\_layout()

plt.show()

**2. Top Products Bar Plot**

top\_products.plot(kind='barh', color='orange', title='Top 10 Products by Sales')

plt.xlabel('Sales (INR)')

plt.tight\_layout()

plt.show()

**3. Region-wise Pie Chart**

region\_sales.plot(kind='pie', autopct='%1.1f%%', startangle=140, title='Sales by Region')

plt.ylabel('')

plt.tight\_layout()

plt.show()

**4. Category-wise Profit and Quantity**

category\_analysis.plot(kind='bar', title='Category-wise Profit & Quantity')

plt.ylabel('Value')

plt.tight\_layout()

plt.show()

**5. Shipping Delay by Region**

shipping\_delay.plot(kind='bar', title='Average Shipping Days by Region')

plt.ylabel('Days')

plt.tight\_layout()

plt.show()

**STEP 7: Export Final Report**

df.to\_csv('Cleaned\_SuperStore\_Sales.csv', index=False)

# Summary Report

summary = {

"Total Revenue": total\_sales,

"Average Discount": avg\_discount,

"Total Orders": total\_orders

}

summary\_df = pd.DataFrame([summary])

summary\_df.to\_excel("SuperStore\_Summary\_Report.xlsx", index=False)