Week 4 – Ecommerce-Sales-Analysis

Developers Arena – Python Basics Internship

1. Project Overview

This project is part of **Week 4: Data Visualization & Your First Complete Project** under the Developers Arena Data Science Internship.

The objective of this project is to perform a **complete data analysis and visualization workflow** using Python.In this project, an e-commerce sales dataset is loaded, cleaned, analyzed, and visualized using charts.By combining numerical analysis with visual representations, the project demonstrates how data can be transformed into meaningful insights that support better understanding and decision-making.

2. Project Objective

The main objectives of this project are:

- To load and explore a real-world sales dataset
- To clean and validate the dataset by handling missing values and duplicates
- To perform basic sales analysis using pandas
- To create at least two different types of visualizations
- To extract and interpret insights from charts
- To document the analysis process and findings clearly

3. Setup Instructions

Follow the steps below to run the project:

1. Install **Python 3.x** from the official website: https://www.python.org
2. Open **Command Prompt / Terminal** and install the required libraries:

```
pip install pandas matplotlib notebook
```

3.Ensure the project folder contains the following structure:

- analysis.ipynb
- data/sales_data.csv
- visualizations/
- report/

4.Navigate to the project folder and start Jupyter Notebook:

```
jupyter notebook
```

5.Open the file **analysis.ipynb** from the browser interface.

6.Run all cells from top to bottom to execute the analysis and generate visualizations.

4. Code Structure

CODE:

```
import pandas as pd

import matplotlib.pyplot as plt

df = pd.read_csv(r"C:\Users\DARSHAN\OneDrive\Desktop\INTERNSHIP\sales_data.csv")

df.info()

df.isnull().sum()

df.fillna(0, inplace=True)

df.drop_duplicates(inplace=True)

total_sales = df["Total_Sales"].sum()

average_sales = df["Total_Sales"].mean()

total_sales, average_sales

product_sales = df.groupby("Product")["Total_Sales"].sum()

region_sales = df.groupby("Region")["Total_Sales"].sum()

plt.figure(figsize=(8,5))

product_sales.plot(kind="bar")

plt.title("Total Sales by Product")
plt.xlabel("Product")

plt.ylabel("Sales Amount")

plt.tight_layout()

plt.show()

plt.figure(figsize=(6,6))

region_sales.plot(kind="pie", autopct="%1.1f%%")

plt.title("Sales Distribution by Region")

plt.ylabel("")

plt.tight_layout()

plt.show()
```

5. Program Flow

1. The program starts by importing the required Python libraries such as pandas and matplotlib.
2. The sales dataset is loaded from a CSV file into a pandas DataFrame.
3. The dataset is explored to understand its structure, columns, and data types.
4. Missing values in the dataset are identified and handled appropriately.
5. Duplicate records are removed to ensure clean data.
6. Basic sales metrics such as total sales and aggregated sales values are calculated.
7. The data is grouped by product to analyze product-wise sales performance.
8. The data is grouped by region to analyze region-wise sales distribution.
9. A bar chart is generated to visualize total sales by product.
10. A pie chart is created to show the distribution of sales across regions.
11. The visualizations are saved to the `visualizations` folder.
12. Key insights are derived from the analysis and visualizations.
13. The program completes execution successfully.

6. Technical Details

Programming Language: Python 3

Development Environment: Jupyter Notebook

Libraries Used:

- **pandas** – for data loading, cleaning, and analysis
- **matplotlib** – for creating visualizations

Data Structure:

- The dataset is stored and processed using a **pandas DataFrame**, which allows efficient data manipulation and aggregation.

Key Operations Performed:

- `read_csv()` to load the dataset
- `head()` and `info()` to explore the dataset
- `isnull()`, `fillna()`, and `drop_duplicates()` for data cleaning
- `groupby()` and `sum()` for aggregating sales data
- `mean()` for calculating average sales

Visualization Techniques:

- **Bar Chart** to compare total sales across products
- **Pie Chart** to visualize sales distribution by region

File Handling:

- Visualizations are saved as image files in the `visualizations/` directory

- Project reports are documented using Markdown files

Error Handling:

- Missing values are handled to prevent calculation errors
- Folder creation is managed to avoid file-saving issues

7.Testing Evidence

The project was tested at different stages to ensure correct execution and accurate results.

Test Case 1: Dataset Loading

Input: `sales_data.csv` file present in the `data/` folder
Expected Outcome: Dataset loads successfully into a pandas DataFrame
Result: Passed

Test Case 2: Dataset Exploration

Input: Loaded DataFrame
Expected Outcome:

- First few rows displayed correctly
- Dataset information (rows, columns, data types) displayed without errors
  Result: Passed

Test Case 3: Missing Values Handling

Input: Dataset containing missing values
Expected Outcome: Missing values replaced appropriately using `fillna()`
Result: Passed

Test Case 4: Duplicate Records Removal

Input: Dataset with duplicate rows
Expected Outcome: Duplicate records removed using `drop_duplicates()`
Result: Passed

Test Case 5: Sales Metrics Calculation

Input: Column `Total_Sales`
Expected Outcome:

- Total sales calculated correctly
- Average sales calculated correctly
  Result: Passed

Test Case 6: Product-wise Aggregation

Input: Grouped by `Product`
Expected Outcome: Correct total sales calculated for each product
Result: Passed

Test Case 7: Region-wise Aggregation

Input: Grouped by `Region`
Expected Outcome: Correct sales distribution calculated for each region
Result: Passed

Test Case 8: Bar Chart Generation

Input: Product-wise sales data
Expected Outcome:

- Bar chart generated and displayed in the notebook

- Chart saved successfully in the `visualizations/` folder
  Result: Passed

Test Case 9: Pie Chart Generation

Input: Region-wise sales data
Expected Outcome:

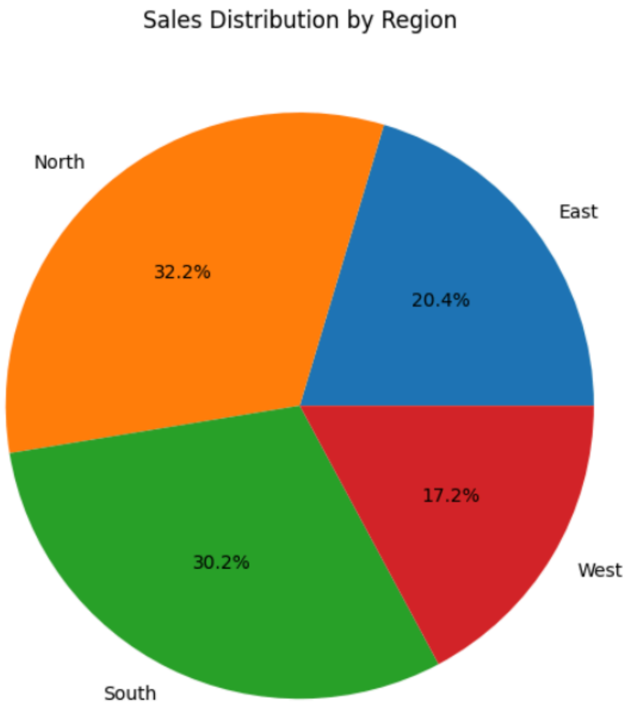
- Pie chart generated and displayed in the notebook
- Chart saved successfully in the `visualizations/` folder
  Result: Passed

Test Case 10: Notebook Execution

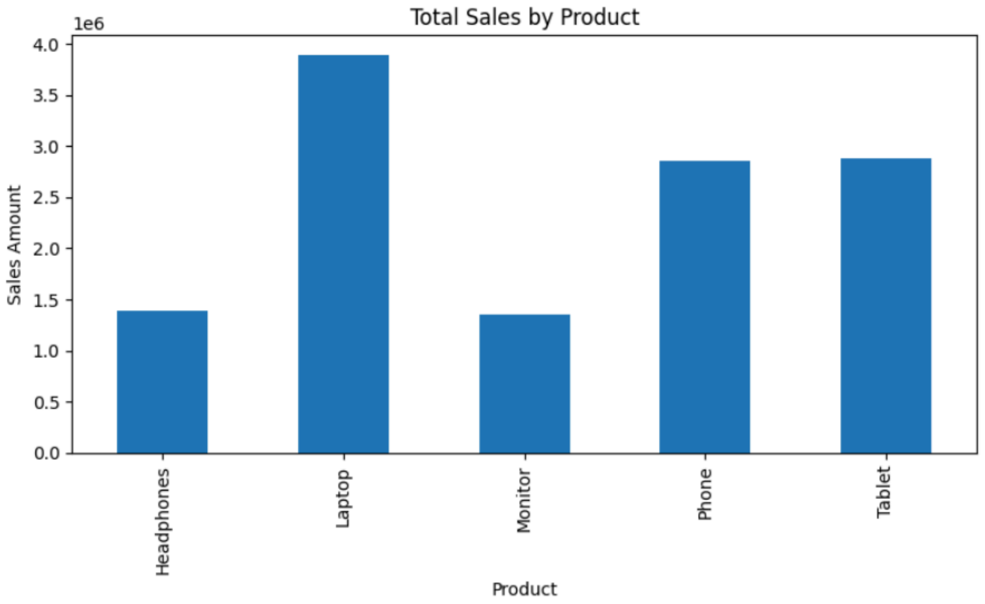
Input: Run all notebook cells from start to end
Expected Outcome: No runtime errors and complete execution
Result: Passed

8. Visual Documentation

OUTPUTSCREENSHOT 1 :



OUTPUTSCREENSHOT 2 :



9.Key Insights

1. The analysis shows the overall sales performance by calculating total revenue from all transactions.
2. The bar chart highlights that certain products contribute significantly more to total sales compared to others.
3. Product-wise analysis helps identify top-performing products, which are key revenue drivers.
4. The pie chart reveals that sales are not evenly distributed across regions, with some regions contributing a larger share of total sales.
5. Visualizing sales data makes it easier to identify patterns and trends that are not immediately obvious from raw numbers.
6. Combining numerical analysis with visualizations provides a clearer understanding of business performance and customer demand.

10.Conclusion

This project successfully demonstrates a complete data analysis and visualization workflow using Python.

By loading, cleaning, analyzing, and visualizing e-commerce sales data, meaningful insights were extracted from raw data.

The use of bar and pie charts helped present complex sales information in a clear and understandable manner.

This project strengthened practical skills in data handling, analysis, and visualization using pandas and matplotlib.

Overall, the project provides a strong foundation in data analysis and prepares for more advanced data science and visualization tasks in the future.