Data Visualization Dashboard

Objective:

The objective of this project is to create an interactive Data Visualization Dashboard using Python libraries such as Matplotlib and Pandas. The dashboard helps visualize datasets, clean data, and analyze insights through different types of charts and summary statistics.

Tools Used:

• Python
• Pandas
• Matplotlib
• CSV Dataset

Project Explanation:

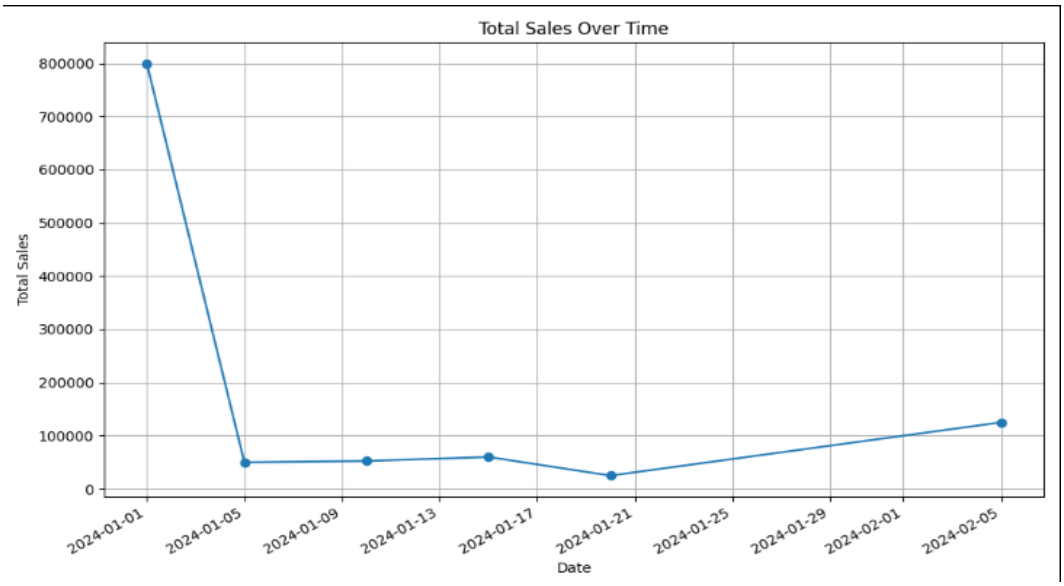
1. The dataset was loaded using Pandas from a CSV file.
2. Data was cleaned by handling missing values and formatting dates.
3. Matplotlib was used to generate various charts:
• Line Chart – to show sales trends over time.
• Bar Chart – to compare total sales by category.
• Pie Chart – to display product-wise sales distribution.
4. Summary statistics were generated to show overall sales performance.
5. The charts and processed data were exported as image and CSV files.

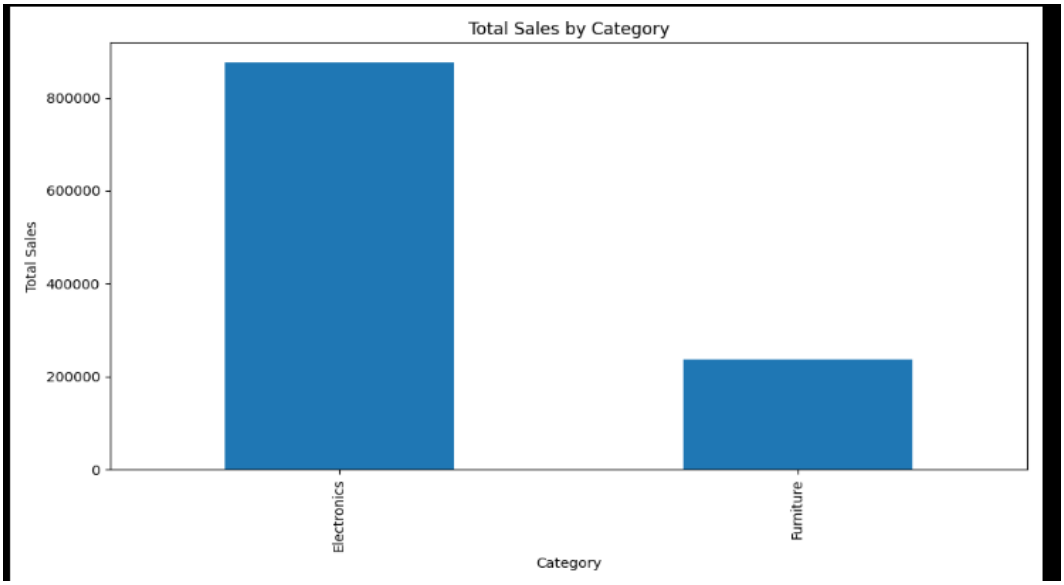
Source Code:

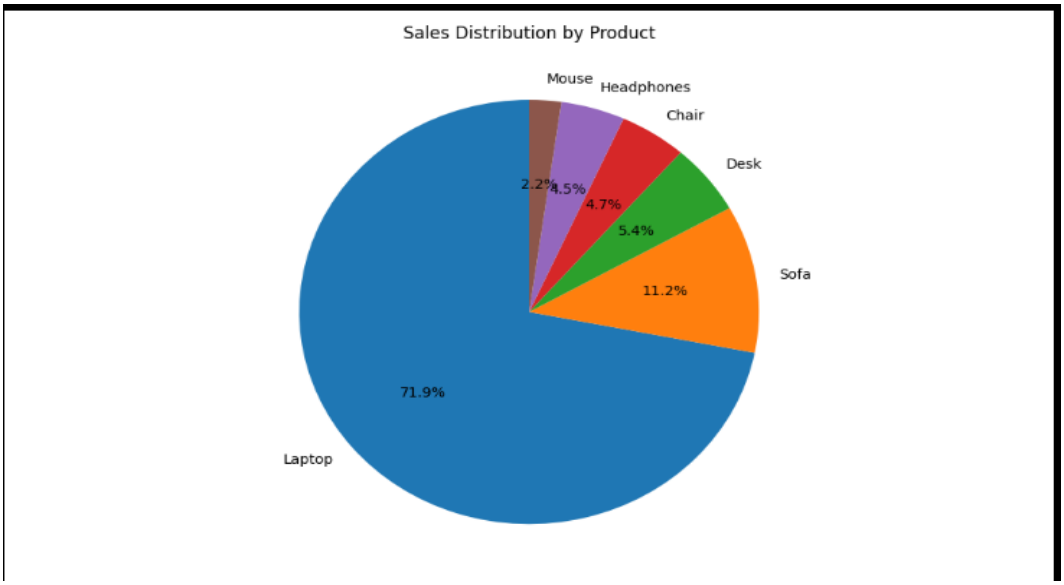
```
import pandas as pd
import matplotlib.pyplot as plt

# Load dataset
df = pd.read_csv('sales_data.csv')

# Data cleaning
df['Date'] = pd.to_datetime(df['Date'])
df.fillna(0, inplace=True)

# Line chart - Sales over time
plt.figure(figsize=(8,4))
plt.plot(df['Date'], df['Total Sales'], marker='o')
plt.title('Total Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('total_sales_over_time.png')
plt.close()

# Bar chart - Sales by category
```

```
category_sales = df.groupby('Category')['Total
Sales'].sum().reset_index()
plt.figure(figsize=(8,4))
plt.bar(category_sales['Category'], category_sales['Total Sales'])
plt.title('Total Sales by Category')
plt.xlabel('Category')
plt.ylabel('Total Sales')
plt.tight_layout()
plt.savefig('sales_by_category.png')
plt.close()

# Pie chart - Sales distribution by product
product_sales = df.groupby('Product')['Total Sales'].sum()
plt.figure(figsize=(6,6))
plt.pie(product_sales, labels=product_sales.index, autopct='%1.1f%%',
startangle=90)
plt.title('Sales Distribution by Product')
plt.tight_layout()
plt.savefig('sales_by_product_pie.png')
plt.close()

# Summary statistics
summary = df.describe()
print(summary)
```

Output Screenshots:







Conclusion:

The Data Visualization Dashboard successfully analyzes and visualizes data using Pandas and Matplotlib. It demonstrates the ability to load, process, and visually interpret data trends and summaries effectively.