**Sales Data Analysis Document**

**Title:** Sales Data Decoding and Analysis  
**Author:** Thamarai Selvan B  
**Date:** 01-02-2025

**Table of Contents**

1. Introduction
2. Data Cleaning Steps
3. Key Findings from the Analysis
4. Visualizations
5. Actionable Recommendations
6. Code/Workbook

**1. Introduction**

This document outlines the process of designing and analyzing a Sales Data Analysis Dashboard using Python and visualization libraries such as Matplotlib and Seaborn. The goal is to provide actionable insights through interactive visualizations that highlight key sales metrics, trends, and patterns. The analysis focuses on improving sales performance, understanding customer behavior, and identifying growth opportunities.

**2. Data Cleaning Steps**

**Step 1: Import Data**

Imported the following datasets into Python using Pandas:

* **sales\_data.csv**: Contains transaction-level sales data.
* **customer\_data.csv**: Contains customer demographic information.
* **product\_data.csv**: Contains product details.

**Step 2: Handle Missing Values**

* Checked for missing values using isnull().sum().
* Filled missing values using appropriate methods:
  + **Numerical columns**: Filled with the mean or median.
  + **Categorical columns**: Filled with the mode or a placeholder (e.g., "Unknown").

**Step 3: Remove Duplicates**

* Identified and removed duplicate rows using drop\_duplicates().

**Step 4: Merge Data**

* Merged sales\_data, customer\_data, and product\_data using common keys (customer\_id and product\_id).

**Step 5: Create Derived Columns**

Added new columns for analysis:

* **Revenue per Customer**: total\_sales / number of unique customers.
* **Month**: Extracted from the date column using pd.to\_datetime().
* **Customer Segment**: Categorized customers into four segments (Low, Medium, High, Very High) based on their total spending.

**3. Key Findings from the Analysis**

**Top-Performing Products**

* Identified the top 10 products by total sales.
* Example: Product A generated the highest revenue, contributing **15% of total sales**.

**Top-Performing Regions**

* **North region** contributed the highest sales (**40% of total revenue**).
* **South region** showed the lowest performance.

**Seasonal Trends**

* Sales peaked during **Month 6** and dropped during **Month 12**.

**Customer Segmentation**

* **Very High-Value Customers**: Contributed **90% of total revenue**.
* **High-Value Customers**: Contributed **10% of total revenue**.
* **Low-Value Customers**: Contributed **0% of total revenue**.

**4. Visualizations**

**4.1 Monthly Sales Trend**

* **Chart Type:** Line Chart
* **Description:** Shows the monthly trend of total sales over the year.
* **Insight:** Sales peaked in **Month 6** and dropped in **Month 12**.

**4.2 Customer Segmentation**

* **Chart Type:** Bar Chart
* **Description:** Displays the distribution of customers across segments (Low, Medium, High, Very High).
* **Insight:** High-value customers make up a small percentage but contribute significantly to revenue.

**4.3 Top Products by Daily Sales Trends**

* **Chart Type:** Horizontal Bar Chart
* **Description:** Highlights the top-performing products by total sales.
* **Insight:** **Product 6 and Product 8** are the best sellers.

**5. Actionable Recommendations**

**Increase Sales During Low-Performing Months**

* Launch targeted promotions and discounts during **February** to boost sales.
* Introduce new products or bundles to attract customers.

**Retain High-Value Customers**

* Implement a **loyalty program** to reward high-value customers.
* Offer exclusive deals or early access to new products.

**Optimize Underperforming Regions**

* Conduct **market research** in the South region to identify customer preferences.
* Increase marketing efforts and improve product availability in this region.

**Focus on Top-Performing Products**

* Provide incentives for **preferred payment methods** to enhance sales.

**6. Code/Workbook**

**Code Overview**

The analysis was performed using Python in a Jupyter Notebook.  
**Key libraries used:**

* **Pandas**: Data cleaning and manipulation.
* **Matplotlib/Seaborn**: Data visualization.
* **Scikit-learn**: Forecasting model (Linear Regression).

**Python Code**

# Import libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load datasets

sales\_data = pd.read\_csv('sales\_data.csv')

customer\_data = pd.read\_csv('customer\_data.csv')

product\_data = pd.read\_csv('product\_data.csv')

# Merge datasets

merged\_data = pd.merge(sales\_data, customer\_data, on='customer\_id')

merged\_data = pd.merge(merged\_data, product\_data, on='product\_id')

# Create derived columns

merged\_data['revenue\_per\_customer'] = merged\_data['total\_sales'] / merged\_data['customer\_id'].nunique()

merged\_data['month'] = pd.to\_datetime(merged\_data['date']).dt.month

# Visualize monthly sales trend

plt.figure(figsize=(10, 6))

sns.lineplot(x='month', y='total\_sales', data=merged\_data)

plt.title('Monthly Sales Trend')

plt.xlabel('Month')

plt.ylabel('Total Sales')

plt.show()