Data science Toolbox Python



(**project report**)

COMPUTER SCIENCE AND ENGINEERING

Name – Tilak Patel

Submitted to: MADHU BALA MA’AM

***Under the Guidance of***

***MADHU BALA MA’AM***

***UID NO: 31770***

### (Name of faculty coordinator with U.Id and designation)

I, Tilak Patel, a student of Bachelor of Technology under CSE discipline at Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own work and is genuine

Registration Number: 12419708

**CERTIFICATE**

This is to certify that **RAHUL KUMAR PANDEY** bearing Registration no:12306664 has completed INT375 project titled, **MADHU BALA** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

**Signature and Name of the Supervisor Designation of the Supervisor**

**School of COMPUTER SCIENCE AND ENGINERRING**

Lovely Professional University Phagwara, Punjab.

Date:12-05-25  
  
  
  
  
 GitHub link - <https://github.com/Tilak-Patel-22/Python-project-.git>

LinkedIn link - <https://www.linkedin.com/posts/tilak-patel-30228329b_python-datascience-> eda-activity-7316802971252486144-VwL7?utm\_source=share&utm\_medium=member\_desktop&rcm=ACoAAEh6JfIBoafSRMW3FrRfaOzjyTh30-AzMnA

TABLE OF CONTENTS

1. Introduction
2. Source Of Data Set
3. EDA Process
4. Analysis of Data Set
   1. Introduction
   2. General Description
   3. Specific Requirements, Functions and Formula
   4. Analysis Results
   5. Visualization
5. Conclusion
6. Future Scope
7. Reference

Introduction

This data-driven project analyzes retail sales data to extract actionable insights into product performance, customer behavior, and regional sales patterns. The key focus areas of analysis include:

* **Product Performance**: Evaluation of total and trend-based sales across various products to identify top-performing and underperforming items, aiding in inventory and marketing strategies.
* **Buyer Behavior**: Study of buyer-product relationships to uncover purchasing patterns, preferences, and concentration of sales across key clients.
* **Geographic Distribution**: Regional analysis based on city and pincode-level data to determine high-revenue zones, underserved areas, and opportunities for market expansion.
* **Sales Outliers and Anomalies**: Application of IQR-based outlier detection methods to identify irregularities in invoice amounts and product weights, supporting data integrity and fraud detection efforts.
* **Correlation Analysis**: Examination of numerical relationships—such as the link between product weight and sales amount—to surface cost-driving product characteristics.

By leveraging this structured dataset, the project delivers insights relevant to business analysts, sales managers, and strategic planners aiming to enhance operational efficiency, forecast demand, and optimize retail performance through data-informed decisions.

Source of Data Set

The data set used in this project has been sourced from a retail and distribution company, containing comprehensive transactional records of product sales. It captures key attributes such as invoice date, product name, net weight, invoice amount, buyer details, consignee city and address, and delivery pin code.

This structured dataset enables detailed exploration of product-wise performance, regional sales trends, and buyer purchasing behavior. It also supports statistical analysis of sales volume and pricing patterns, offering valuable insights into commercial performance, customer segmentation, and operational opportunities within the retail space.

**Dataset Source**: Internal company-provided transactional sales data.

Exploratory Data Analysis (EDA)

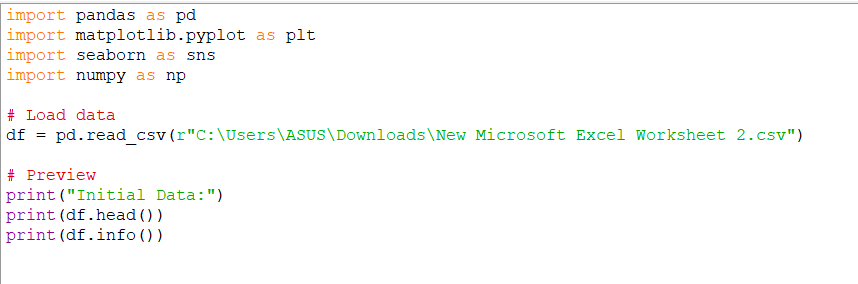
Exploratory Data Analysis (EDA) was conducted to gain a comprehensive understanding of the sales dataset before progressing to advanced analysis. The dataset was initially imported into the Python environment using the **panda’s** library and read directly from a .csv file.

To ensure data quality, missing values in important address fields such as CONSI\_ADD1 and CONSI\_ADD2 were replaced with "not known". Irrelevant columns like Driver Name and Mobile Number were dropped to streamline the dataset. Duplicate records, if any, were identified and removed to maintain data consistency and prevent analytical bias.

Outlier detection was performed on numerical columns such as INV\_AMT and NET\_WT using the **IQR (Interquartile Range)** method. This helped in identifying unusually high or low values that could potentially distort summary statistics and visual patterns. These outliers were also visualized using boxplots for better interpretability.

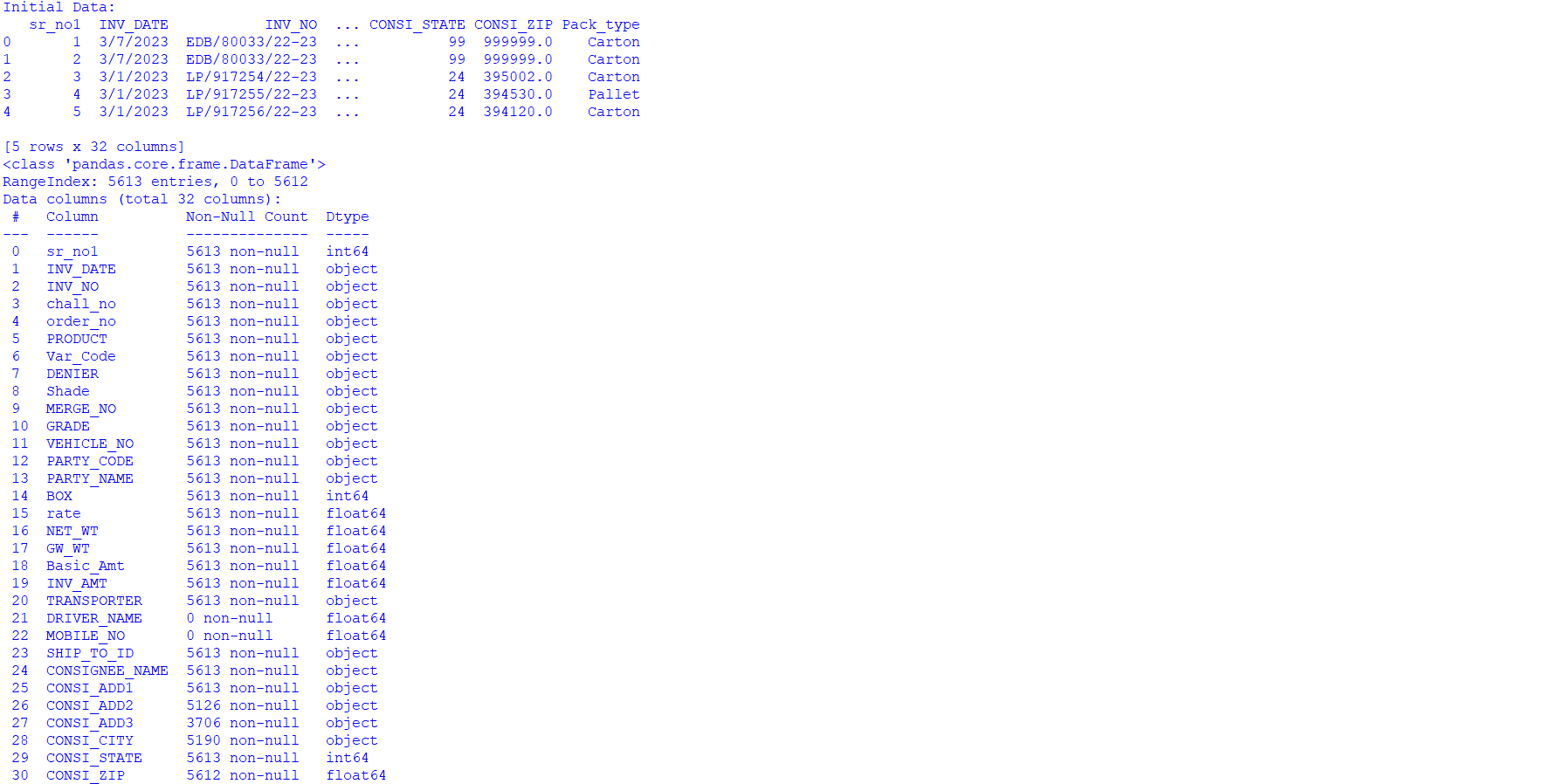
To analyze overall trends and key patterns, visualizations were created using **Seaborn** and **Matplotlib**. Bar plots were used to showcase top-selling products, high-performing cities, and revenue-leading pin codes. A stacked bar chart was implemented to explore buyer-product purchase relationships. Scatter plots were utilized to visualize the correlation between net weight and invoice amount. Furthermore, a correlation heatmap helped assess the relationships between numerical features.

This structured EDA approach provided a solid foundation for uncovering insights into product performance, regional sales distribution, customer behavior, and numerical relationships within the sales data.



Understanding the Data Structure

We used basic functions like head (), info () and describe () to view the top rows, data types, column names, and summary statistics of the dataset



Cleaning of Data

Before conducting any visualizations or statistical analyses, the dataset underwent a thorough data cleaning process to ensure the quality and reliability of the results. The initial step involved identifying missing values using the isnull().sum() function. This helped highlight columns that contained incomplete information which could otherwise distort trends or lead to misleading insights.

**1. Handling Missing Categorical Data:**

* The columns CONSI\_ADD1 and CONSI\_ADD2, which represent consignee address information, had several missing entries. Since addresses are textual and do not directly affect numerical analysis, missing values in these columns were filled with the placeholder "not known". This approach preserved the records without introducing bias or altering the dataset structure.

**2. Dropping Irrelevant or Sensitive Columns:**

* Certain columns such as "Driver Name" and "Mobile Number" were considered irrelevant for the scope of sales performance analysis. Moreover, these fields could pose privacy concerns. Hence, they were removed from the dataset using the drop() method. This helped declutter the data and focus only on attributes relevant to business insights.

**3. Saving the Cleaned Data:**

* After completing the cleaning steps, the modified dataset was saved as a new CSV file (cleaned\_sales\_data.csv) to maintain a clean working version, separate from the original file. This ensured that all subsequent analyses were performed on consistent, preprocessed data.

**4. Visualizing Missing Data Patterns:**

* A heatmap was generated to visually inspect the distribution and pattern of missing values across the dataset. This graphical representation allowed for a quick assessment of data quality and helped verify that all significant null values were appropriately addressed.

By systematically addressing missing values and removing irrelevant information, the data cleaning process ensured that the dataset was both accurate and analysis-ready. These preprocessing steps were crucial for maintaining data integrity and enabling meaningful exploratory data analysis (EDA) across various business dimensions such as sales trends, product performance, and customer demographics.

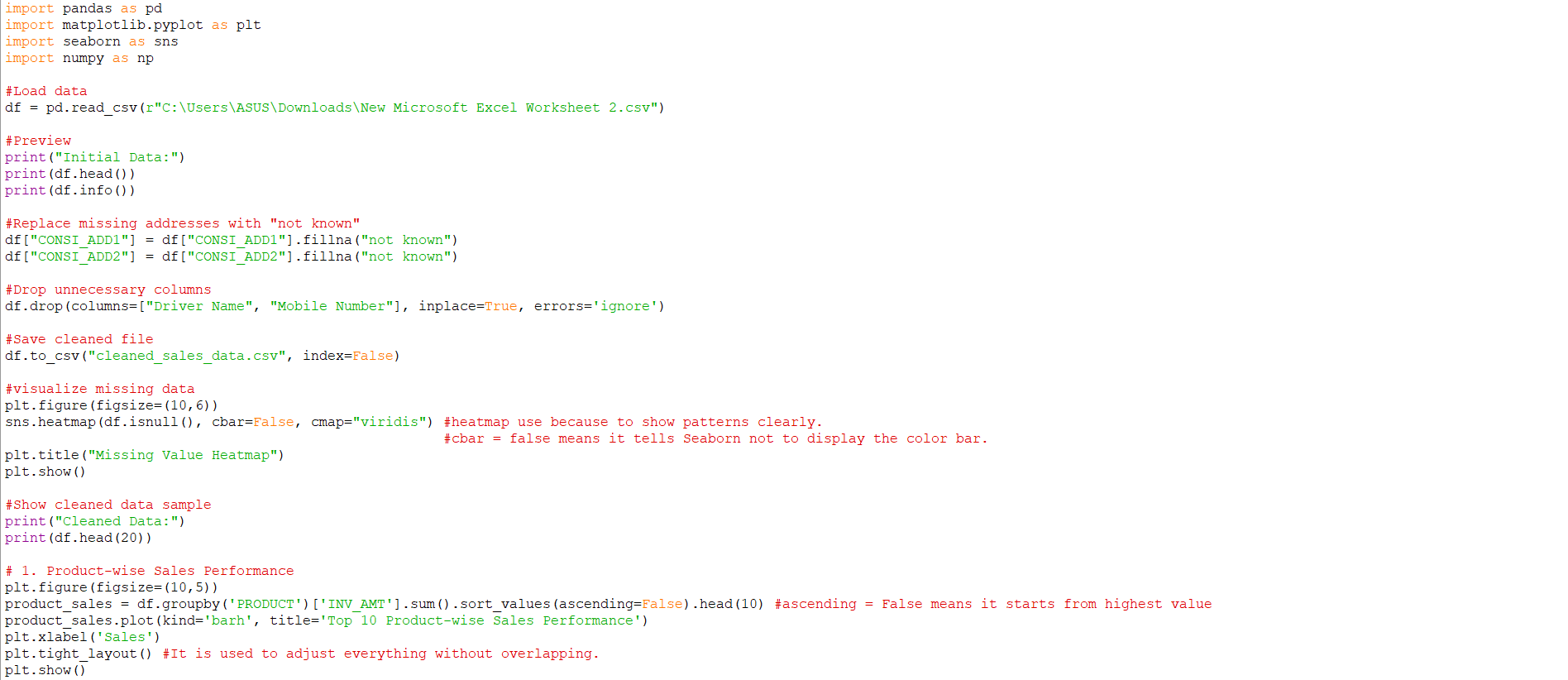
Extracting the Data

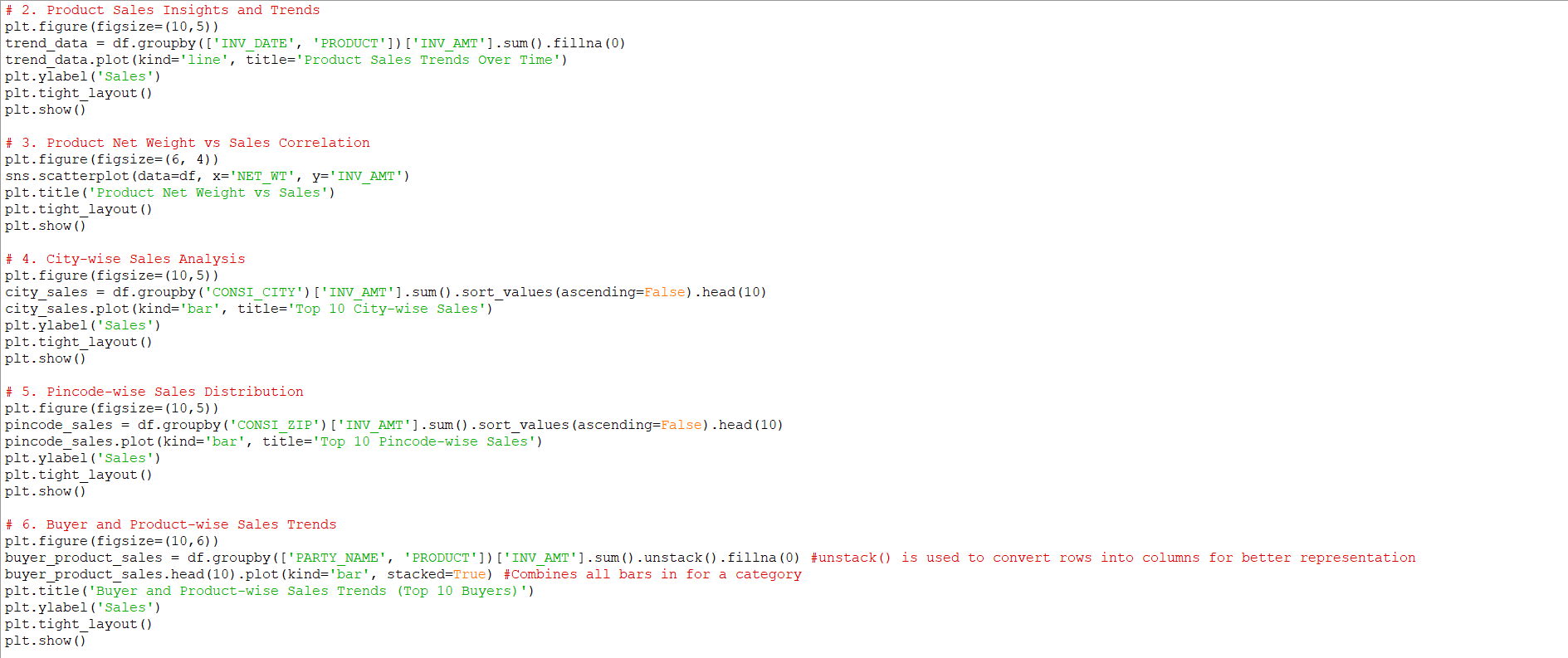
The EDA process enabled:

Discovery of actionable insights

Identification of trends and outliers

Preparation for deeper analysis like forecasting or customer profiling





A white background with text

AI-generated content may be incorrect.

A graph of a bar graph

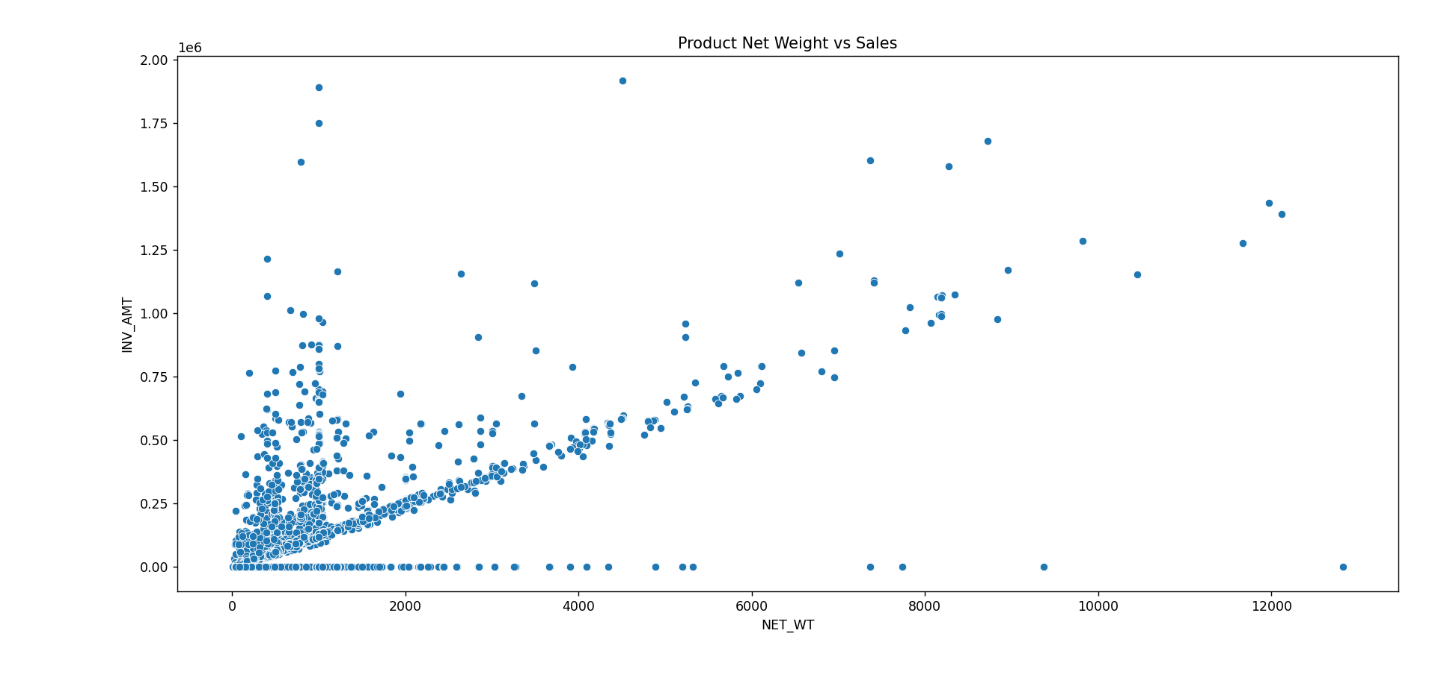
AI-generated content may be incorrect.A purple and yellow rectangle with white text

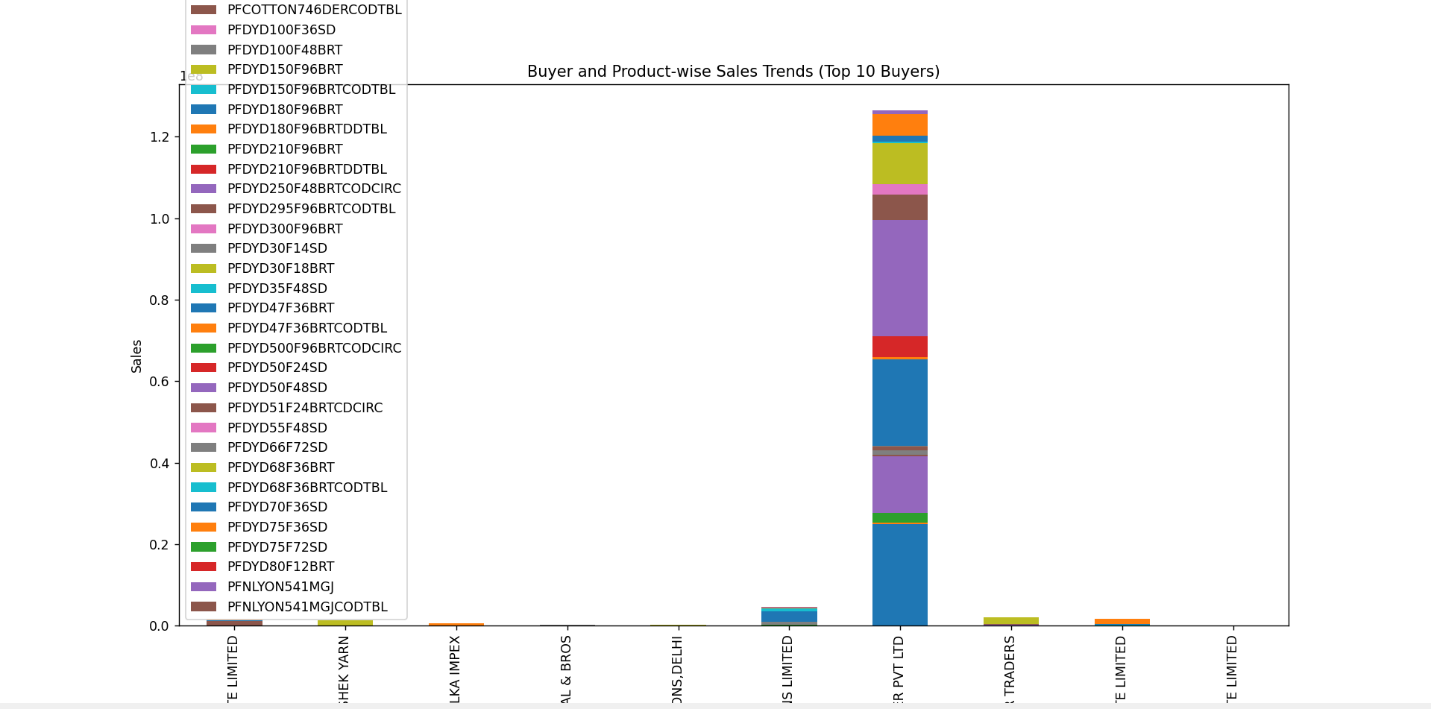
AI-generated content may be incorrect.

A graph with text on it

AI-generated content may be incorrect.A blue lines on a white background

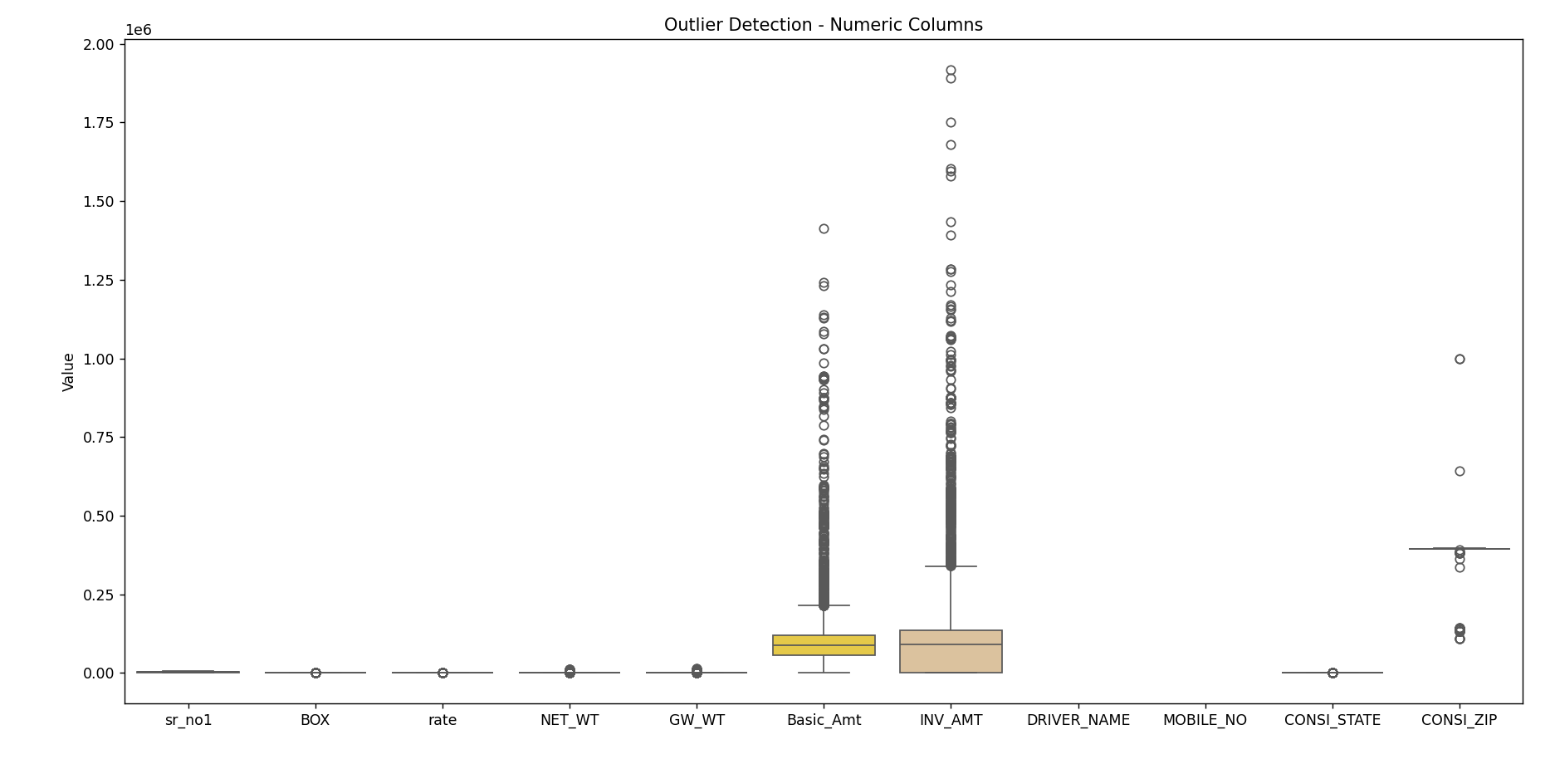
AI-generated content may be incorrect.





A screenshot of a graph

AI-generated content may be incorrect.



Code

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

# Load data

df = pd.read\_csv(r"C:\Users\ASUS\Downloads\New Microsoft Excel Worksheet 2.csv")

# Preview

print("Initial Data:")

print(df.head())

print(df.info())

# Replace missing addresses with "not known"

df["CONSI\_ADD1"] = df["CONSI\_ADD1"].fillna("not known")

df["CONSI\_ADD2"] = df["CONSI\_ADD2"].fillna("not known")

# Drop unnecessary columns

df.drop(columns=["Driver Name", "Mobile Number"], inplace=True, errors='ignore')

# Save cleaned file

df.to\_csv("cleaned\_sales\_data.csv", index=False)

# Visualize missing data

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

sns.heatmap(df.isnull(), cbar=False, cmap="viridis") #heatmap use because to show patterns clearly

#cbar = false means it tells seaborn not to display the color bar

plt.title("Missing Value Heatmap")

plt.show()

# Show cleaned data sample

print("Cleaned Data:")

print(df.head(20))

# 1. Product-wise Sales Performance

plt.figure(figsize=(10, 5))

product\_sales = df.groupby('PRODUCT')['INV\_AMT'].sum().sort\_values(ascending=False).head(10) #ascending = false means it starts from highest value

product\_sales.plot(kind='barh', title='Top 10 Product-wise Sales Performance')

plt.xlabel('Sales')

plt.tight\_layout() #It is used to adjust everything without overlapping

plt.show()

# 2. Product Sales Insights and Trends

plt.figure(figsize=(10, 5))

trend\_data = df.groupby(['INV\_DATE', 'PRODUCT'])['INV\_AMT'].sum().fillna(0)

trend\_data.plot(kind='line', title='Product Sales Trends Over Time')

plt.ylabel('Sales')

plt.tight\_layout()

plt.show()

# 3. Product Net Weight vs Sales Correlation

plt.figure(figsize=(6, 4))

sns.scatterplot(data=df, x='NET\_WT', y='INV\_AMT')

plt.title('Product Net Weight vs Sales')

plt.tight\_layout()

plt.show()

# 4. City-wise Sales Analysis

plt.figure(figsize=(10, 5))

city\_sales = df.groupby('CONSI\_CITY')['INV\_AMT'].sum().sort\_values(ascending=False).head(10)

city\_sales.plot(kind='bar', title='Top 10 City-wise Sales')

plt.ylabel('Sales')

plt.tight\_layout()

plt.show()

# 5. Pincode-wise Sales Distribution

plt.figure(figsize=(10, 5))

pincode\_sales = df.groupby('CONSI\_ZIP')['INV\_AMT'].sum().sort\_values(ascending=False).head(10)

pincode\_sales.plot(kind='bar', title='Top 10 Pincode-wise Sales')

plt.ylabel('Sales')

plt.tight\_layout()

plt.show()

# 6. Buyer and Product-wise Sales Trends

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

buyer\_product\_sales = df.groupby(['PARTY\_NAME', 'PRODUCT'])['INV\_AMT'].sum().unstack().fillna(0) #unstack() is used to convert rows into columns for better representation

buyer\_product\_sales.head(10).plot(kind='bar', stacked=True) #combines all bars in for a category

plt.title('Buyer and Product-wise Sales Trends (Top 10 Buyers)')

plt.ylabel('Sales')

plt.tight\_layout()

plt.show()

# Select numeric columns

numerical\_columns = df.select\_dtypes(include=['float64', 'int64']).columns

# Calculate IQR and bounds

Q1 = df[numerical\_columns].quantile(0.25)

Q3 = df[numerical\_columns].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

# Boolean mask of where outliers exist

outlier\_mask = (df[numerical\_columns] < lower\_bound) | (df[numerical\_columns] > upper\_bound)

# Create a combined boxplot

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

sns.boxplot(data=df[numerical\_columns], palette='Set2')

plt.title("Outlier Detection - Numeric Columns")

plt.ylabel("Value")

plt.tight\_layout()

plt.show()

# Print count of outliers per numeric column

print("Outlier counts per column:")

print(outlier\_mask.sum())

# Correlation

print("\nCorrelation Matrix:")

corr\_matrix = df[numerical\_columns].corr()

print(corr\_matrix)

# Heatmap for correlation

plt.figure(figsize=(5, 4))

sns.heatmap(corr\_matrix, annot=True, cmap="coolwarm", fmt=".2f")

plt.title("Correlation Heatmap")

plt.tight\_layout()

plt.show()

Conclusion

In conclusion, the structured approach to data preprocessing and analysis ensured a high level of data integrity and analytical reliability. Through meticulous cleaning, missing values were handled thoughtfully, preserving the underlying distribution and preventing biased interpretations. This laid a solid foundation for accurate and meaningful exploratory data analysis.

The subsequent EDA uncovered critical business insights—highlighting top-performing products, sales trends across time, and regional performance variations. Visualizations and correlation analyses provided a deeper understanding of customer behavior and operational efficiency, which can guide strategic decision-making.

Overall, the combination of robust data cleaning and insightful analysis not only enhanced the quality of the dataset but also transformed raw data into actionable intelligence. This process demonstrates the power of data-driven thinking in optimizing business outcomes and supporting informed decisions.

Future Scope

Automating recurring performance dashboards

Building predictive models for sales forecasting using machine learning

Creating interactive dashboards using Power BI or Stream lit for real-time insights

Implementing anomaly detection techniques for fraud or error spotting

References

Wes McKinney, "Python for Data Analysis", O'Reilly Media, 2nd Edition

https://pandas.pydata.org/

https://seaborn.pydata.org/

https://matplotlib.org/