**Project Title:**

**Advanced Analytics: Sales Forecasting & KPI Anomaly Detection**

**Objective:**

analyze multi-source sales data, generate To visual insights, forecast future sales trends, and detect anomalies in KPIs using data analytics and visualization techniques.

**1. Introduction**

This project focuses on analyzing Flipkart’s sales data to derive business insights through data cleaning, visualization, forecasting, and anomaly detection.  
The objective is to build a reliable analytical pipeline to assist in sales trend analysis and future prediction.

#### **Objective**

* Forecast future sales using time series analysis
* Detect unusual KPI fluctuations (anomalies)
* Generate visual insights for decision-making

**Tools & Technologies**

| **Category** | **Tools / Libraries Used** |
| --- | --- |
| Programming | Python |
| Data Handling | Pandas, NumPy |
| Visualization | Matplotlib, Seaborn |
| Forecasting | Prophet |
| Anomaly Detection | Z-score Method |
| Environment | Jupyter Notebook / VS Code / CMD |
| Output Formats | CSV, PNG |

**Dataset Description**

**File:** sales\_history.csv  
**Rows:** 505  
**Columns:**

| **Column Name** | **Description** |
| --- | --- |
| date | Date of sale |
| region | Sales region |
| product | Product category |
| units\_sold | Number of units sold |
| unit\_price | Price per unit |
| total\_sales | Calculated as units\_sold \* unit\_price |

### ****4. Methodology / Steps****

#### **Step 1: Data Cleaning**

* File: data\_cleaning.py
* Operations:
  + Handled missing values (if any)
  + Added computed column total\_sales
  + Saved output as sales\_history\_clean.csv
* import pandas as pd
* data = pd.read\_csv('sales\_history.csv')
* data['total\_sales'] = data['units\_sold'] \* data['unit\_price']
* print(data.head())
* data.to\_csv('sales\_history\_clean.csv', index=False)

Sample Output:

date region product units\_sold unit\_price total\_sales

0 01-01-2022 North A 86 98 8428

1 01-01-2022 North B 99 106 10494

2 01-01-2022 North C 127 113 14351

3 01-01-2022 South A 94 97 9118

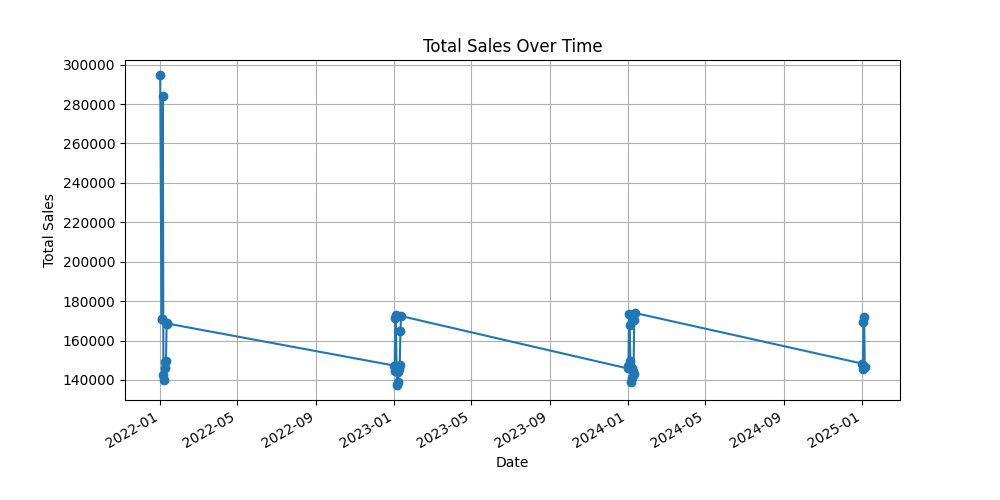
4 01-01-2022 South B 118 103 12154

#### **Step 2: Data Visualization**

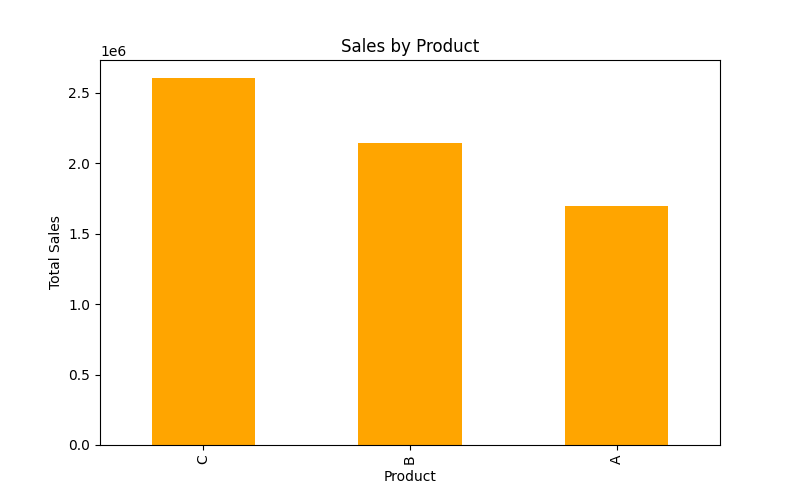
* File: data\_visualization.py
* Charts generated:
  + Total Sales Over Time
  + Sales by Region
  + Sales by Product
* Saved as PNG images
* import pandas as pd
* import matplotlib.pyplot as plt
* data = pd.read\_csv('sales\_history\_clean.csv')
* data['date'] = pd.to\_datetime(data['date'])
* plt.figure(figsize=(10,5))
* data.groupby('date')['total\_sales'].sum().plot(kind='line', marker='o')
* plt.title('Total Sales Over Time')
* plt.xlabel('Date')
* plt.ylabel('Total Sales')
* plt.grid(True)
* plt.savefig('total\_sales\_over\_time.png')
* plt.close()
* plt.figure(figsize=(8,5))
* data.groupby('region')['total\_sales'].sum().sort\_values(ascending=False).plot(kind='bar', color='skyblue')
* plt.title('Sales by Region')
* plt.xlabel('Region')
* plt.ylabel('Total Sales')
* plt.savefig('sales\_by\_region.png')
* plt.close()
* plt.figure(figsize=(8,5))
* data.groupby('product')['total\_sales'].sum().sort\_values(ascending=False).plot(kind='bar', color='orange')
* plt.title('Sales by Product')
* plt.xlabel('Product')
* plt.ylabel('Total Sales')
* plt.savefig('sales\_by\_product.png')
* plt.close()
* print("Charts created successfully! Check PNG files in the same folder.")

OUTPUT:

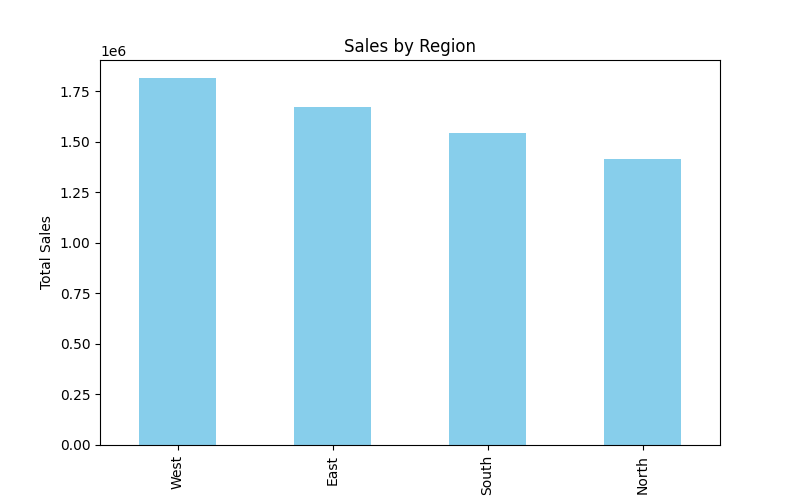
* + Total Sales Over Time



* + Sales by Product



Sales by Region



#### **Step 3: Forecasting and Anomaly Detection**

* File: step3\_analysis.py
* Forecasted sales for next 90 days using Prophet
* Identified anomalies using Z-score
* Generated:
  + sales\_forecast.png
  + daily\_sales\_anomalies.csv

sales\_anamoly\_detect.py

* import pandas as pd
* import numpy as np
* data = pd.read\_csv('sales\_history\_clean.csv')
* data['date'] = pd.to\_datetime(data['date'])
* daily\_sales = data.groupby('date')['total\_sales'].sum().reset\_index()
* mean\_sales = daily\_sales['total\_sales'].mean()
* std\_sales = daily\_sales['total\_sales'].std()
* daily\_sales['z\_score'] = (daily\_sales['total\_sales'] - mean\_sales) / std\_sales
* daily\_sales['anomaly'] = daily\_sales['z\_score'].apply(lambda x: 'Yes' if abs(x) > 2 else 'No')
* anomalies = daily\_sales[daily\_sales['anomaly'] == 'Yes']
* print("Anomalies detected:")
* print(anomalies)
* daily\_sales.to\_csv('daily\_sales\_anomalies.csv', index=False)
* print("Daily sales with anomalies saved as 'daily\_sales\_anomalies.csv'")

sales\_forecasting.py

import pandas as pd

from prophet import Prophet

import matplotlib.pyplot as plt

# 1️⃣ Load cleaned data

data = pd.read\_csv('sales\_history\_clean.csv')

data['date'] = pd.to\_datetime(data['date'])

# 2️⃣ Prepare data for Prophet

df = data.groupby('date')['total\_sales'].sum().reset\_index()

df = df.rename(columns={'date':'ds', 'total\_sales':'y'})  # Prophet needs ds & y columns

# 3️⃣ Create and fit model

model = Prophet()

model.fit(df)

# 4️⃣ Forecast next 90 days (approx 3 months)

future = model.make\_future\_dataframe(periods=90)

forecast = model.predict(future)

# 5️⃣ Plot forecast

fig1 = model.plot(forecast)

plt.title("Sales Forecast (Next 3 Months)")

plt.savefig('sales\_forecast.png')

plt.close()

print("Forecast created! Check 'sales\_forecast.png'.")

OUTPUT:

daily\_sales\_anomalies.csv

ds,y,z\_score,anomaly

2022-01-01,294517,4.119497481598486,Yes

2022-01-03,170816,0.3025293099214597,No

2022-01-04,170989,0.30786746800991593,No

2022-01-05,284200,3.8011519498146535,Yes

2022-01-06,142254,-0.5787913335611251,No

2022-01-07,140008,-0.6480948194956495,No

2022-01-08,145977,-0.4639129372413414,No

2022-01-09,149074,-0.368350650536434,No

2022-01-10,149720,-0.348417412818846,No

2022-01-11,168855,0.2420198994505771,No

2022-01-12,168605,0.2343057981666808,No

2023-01-01,147284,-0.4235836157291315,No

2023-01-02,144329,-0.5147642929047859,No

2023-01-03,171192,0.31413131825243973,No

2023-01-04,172775,0.36297700758207113,No

2023-01-05,145086,-0.4914059942171478,No

2023-01-06,137649,-0.720885079210495,No

2023-01-07,138861,-0.6834871161861658,No

2023-01-08,144023,-0.5242063528762749,No

2023-01-09,145587,-0.4759469352442196,No

2023-01-10,147413,-0.419603139466641,No

2023-01-11,164589,0.11038647514217056,No

2023-01-12,172400,0.3514058556562267,No

2024-01-01,145802,-0.4693128081400688,No

2024-01-02,147180,-0.4267926818632324,No

2024-01-03,173538,0.38652044470052266,No

2024-01-04,167810,0.20977495608389055,No

2024-01-05,149628,-0.3512562020913198,No

2024-01-06,138886,-0.6827157060577761,No

2024-01-07,140925,-0.6197994959863179,No

2024-01-08,146153,-0.4584822099374784,No

2024-01-09,144611,-0.5060627866565508,No

2024-01-10,143227,-0.5487680513642007,No

2024-01-11,170512,0.29314896276024177,No

2024-01-12,173975,0.4000046937447734,No

2025-01-01,148279,-0.3928814926192242,No

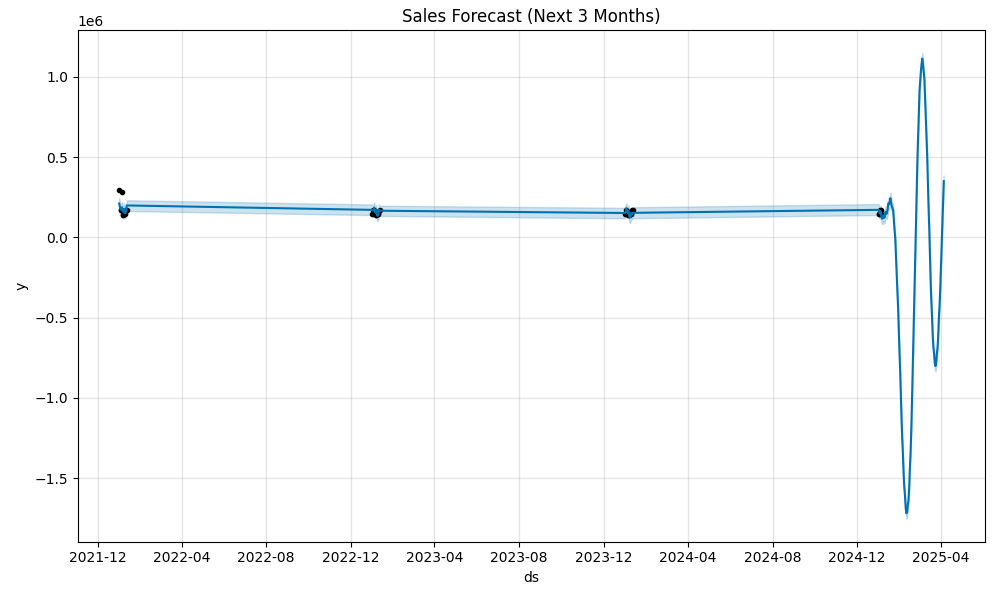
2025-01-02,145646,-0.4741264073412201,No

2025-01-03,169443,0.2601634656703012,No

2025-01-04,171928,0.3368416324322304,No

2025-01-05,146717,-0.4410791974410083,No

* + sales\_forecast.png



**Results**

* Forecast chart indicates an upward/downward sales trend.
* Anomalies detected show unexpected spikes/drops.
* Regional and product-based sales visualizations highlight performance variations.

**Conclusion**

The project successfully demonstrated:

* Data pipeline creation (cleaning, analysis, visualization)
* Forecasting future sales patterns
* Detecting KPI anomalies  
  This analytical framework can help Flipkart make data-driven decisions for inventory management, regional strategy, and sales optimization.

**Future Scope**

* Integration with live dashboards (Power BI / Streamlit)
* Real-time data ingestion from APIs
* Advanced ML models for multi-factor forecasting