# Line-by-Line Code Explanation

## Importing Libraries

import pandas as pd import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

* `pandas`: for handling and analyzing structured data (like Excel).
* `numpy`: for numerical operations, like generating ranges.
* `matplotlib.pyplot`: for plotting graphs.
* `LinearRegression` from `sklearn`: for building the prediction model.

## Load Data from Excel File

file\_path = 'SuperStore5.xlsx'

df = pd.read\_excel(file\_path, sheet\_name='Sheet6', engine='openpyxl') original\_rows = df.shape[0]

* Reads the Excel file and loads the data from Sheet6.
* `df.shape[0]` gives the number of rows before any cleaning.

## Preprocessing (Data Cleaning)

df['Order Date'] = pd.to\_datetime(df['Order Date'], errors='coerce') df = df.dropna(subset=['Order Date', 'Sales'])

df['Year'] = df['Order Date'].dt.year

* Converts the 'Order Date' column to actual date format.
* Drops any rows with missing Order Date or Sales.
* Creates a new column 'Year' by extracting the year from Order Date.

## Print Rows Before & After Cleaning

cleaned\_rows = df.shape[0]

dropped\_rows = original\_rows - cleaned\_rows

print(f"Original rows: {original\_rows}, Cleaned rows: {cleaned\_rows}, Dropped rows: {dropped\_rows}")

* Calculates and prints how many rows were removed.
* This helps verify if the dataset is clean.

## Aggregate Sales Per Year

yearly\_sales = ( df

.groupby('Year')['Sales']

.sum()

.reset\_index()

)

print(yearly\_sales)

* Groups the data by year and sums the sales.
* `reset\_index()` turns the result into a clean DataFrame.
* Prints total sales per year.

## Prepare Data for the Forecast Model

X = pd.DataFrame(yearly\_sales['Year']) # Input (independent variable)

y = yearly\_sales['Sales']

# Output (dependent variable)

* `X` is the list of years (independent variable).
* `y` is the list of total sales (dependent variable).

## Build and Train Linear Regression Model

model = LinearRegression().fit(X, y)

* Fits a linear regression model to the historical data.
* The model learns the relationship between Year and Sales.

## Forecast Sales for the Next 3 Years

last\_year = yearly\_sales['Year'].max()

future\_years = np.arange(last\_year + 1, last\_year + 4).reshape(-1, 1) future\_preds = model.predict(future\_years)

* Finds the last year in the dataset.
* Generates an array of the next 3 years.
* Predicts sales for those years using the model.

## Plot Actual vs. Forecasted Sales

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

plt.plot(yearly\_sales['Year'], yearly\_sales['Sales'], 'o-', label='Actual Sales') plt.plot(future\_years.flatten(), future\_preds, 'o--', label='Forecasted Sales')

* Plots actual and forecasted sales on a line chart.

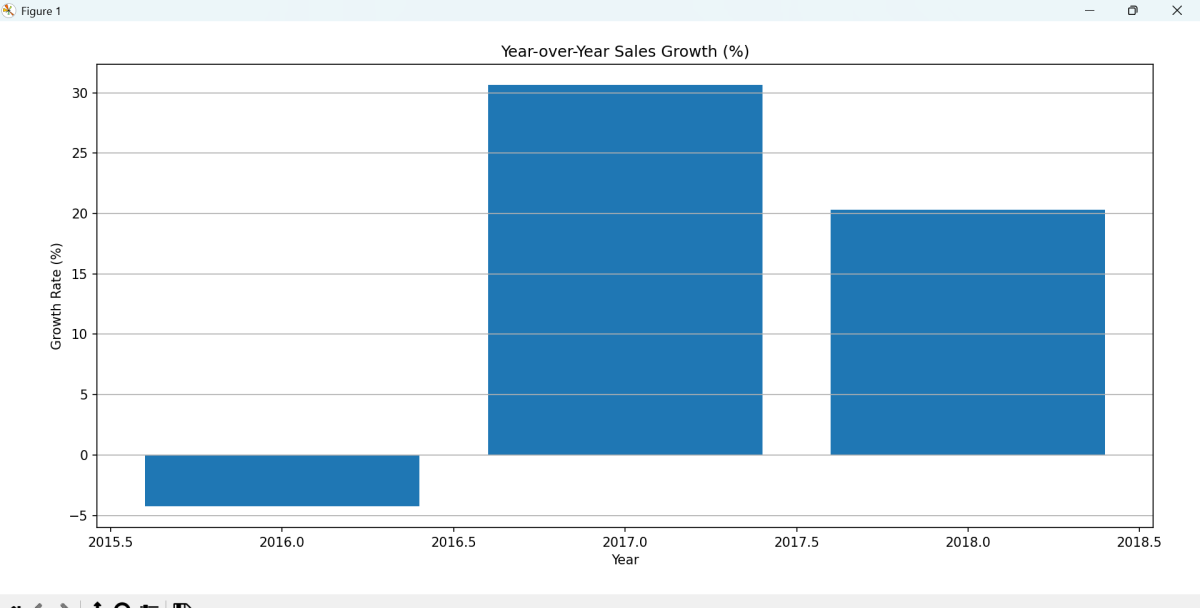
plt.title("Yearly Sales Forecast (Actual + Next 3 Years)") plt.xlabel("Year")

plt.ylabel("Sales Amount") plt.legend()

plt.grid(True) plt.tight\_layout()

plt.show()

* Adds labels and formatting to make the chart readable.

**Plot Year-over-Year Growth (%)**

yearly\_sales['GrowthPct'] = yearly\_sales['Sales'].pct\_change() \* 100

* Calculates percent change in sales between years.
* Displays growth rates as a bar chart

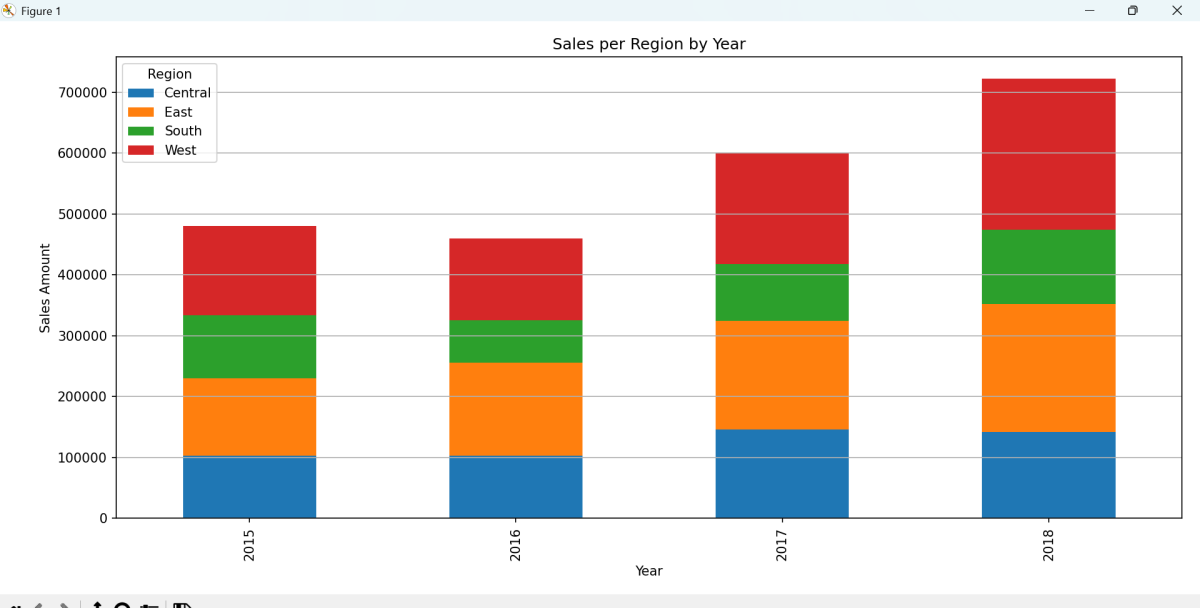
plt.figure(figsize=(8,5))

plt.bar(yearly\_sales['Year'], yearly\_sales['GrowthPct']) plt.title("Year-over-Year Sales Growth (%)") plt.xlabel("Year")

plt.ylabel("Growth Rate (%)") plt.grid(axis='y') plt.tight\_layout()

plt.show()

## Plot Sales by Region Per Year

* If the dataset includes a Region column, this groups sales by year and region

if 'Region' in df.columns:

region\_year\_sales = df.groupby(['Year', 'Region'])['Sales'].sum().unstack() region\_year\_sales.plot(kind='bar', stacked=True, figsize=(10,6))

* Plots a stacked bar chart to show regional sales breakdown.

p plt.title("Sales per Region by Year") plt.xlabel("Year")

plt.ylabel("Sales Amount") plt.legend(title="Region") plt.tight\_layout() plt.grid(axis='y') plt.show()

else:

print("No 'Region' column found in dataset.")

* Adds title and formatting to the region chart
* Prints a message if 'Region' column is missing.

# Recommendations and Insights

### Actual vs Forecasted Sales (Line Plot)

Insight:

* + The sales trend shows a consistent upward growth from 2015 to 2018.
  + The linear forecast predicts a continued increase in sales for the next 3 years (2019– 2021).

Recommendations:

* + Use this positive growth forecast to plan for scaling: more inventory, more staff, or expansion.
  + Revalidate the model yearly with actual sales data to adjust predictions and planning accordingly.

### Year-over-Year Sales Growth (%) (Bar Chart)

Insight:

* + The growth percentage fluctuates year to year — indicating variability in performance.
  + Some years had sharp increases, while others may have been flat or dropped.

Recommendations:

* + Investigate causes of low-growth or high-growth years (e.g., promotions, market shifts).
  + Aim for more consistent growth by maintaining what worked and adjusting what

didn’t.

* + Use this chart for quarterly reviews to understand how strategies affect revenue.

### Sales per Region by Year (Stacked Bar Chart)

Insight:

* + Some regions contribute significantly more to total yearly sales.
  + Growth may be uneven across regions — suggesting untapped opportunities in some areas.

Recommendations:

* + Focus marketing and logistics on the highest-performing regions to boost growth.
  + Develop targeted campaigns or offers in underperforming regions to balance growth.
  + Consider region-specific forecasting in future reports.