**Products sales analysis**

**Phase 4:** In this section continue building the project by performing different activities like feature engineering, model training, evaluation etc as per the instructions in the project

1. **Collection**: Make sure you have collected relevant data on product sales. This data can include information on sales figures, product attributes, customer information, and any other relevant data.
2. **Data Preprocessing**: Clean and preprocess the data. This involves handling missing values, removing duplicates, and converting data types as necessary.
3. **Feature Engineering**: Create new features or transform existing ones to better represent the underlying patterns in your data. Feature engineering can significantly impact the performance of your model.
4. **Data Splitting**: Split your dataset into a training set and a testing set to evaluate your model's performance.
5. **Model Selection**: Choose an appropriate machine learning model for your task. Popular choices for sales prediction include linear regression, decision trees, random forests, and more advanced models like gradient boosting or neural networks.
6. **Model Training**: Train your selected model on the training data. Be sure to tune hyperparameters for optimal performance.
7. **Model Evaluation**: Evaluate your model's performance on the testing dataset. Common evaluation metrics for regression tasks include Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.
8. **Model Interpretation**: Understand the importance of different features in your model. This can help identify which aspects of your products or marketing efforts are most influential on sales.
9. **Visualization**: Create visualizations to communicate your findings effectively. This can include sales trends, feature importance plots, and more.
10. **Documentation**: Document your entire process, from data collection to model evaluation. This will make it easier for others to understand your work.
11. **Deployment (optional)**: If your model performs well, you can deploy it to make real-time predictions on future

PROGRAM

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error

data = pd.DataFrame({

'Date': pd.date\_range(start='2023-01-01', periods=12, freq='M'),

'Price': [100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210],

'SalesVolume': [50, 48, 45, 42, 40, 38, 35, 32, 30, 28, 25, 22]

})

data['PrevMonthSales'] = data['SalesVolume'].shift(1)

X = data[['Price', 'PrevMonthSales']]

y = data['SalesVolume']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mae = mean\_absolute\_error(y\_test, y\_pred)

print(f"Mean Absolute Error: {mae:.2f}")

Sample output:

Mean Absolute Error: 1.28