Predictive Modeling for Sales Forecasting

Problem Statement  
  
This project focuses on solving the problem of sales forecasting, a common and critical task for businesses. The goal is to predict the number of units sold for a product based on its pricing information, specifically the total price and base price. Accurate sales forecasts are essential for effective inventory management and revenue planning.

Dataset Description  
  
The dataset used in this project contains the following columns:  
  
- `ID`: An identifier for each data point.  
- `Total Price`: The total price of the product.  
- `Base Price`: The base price of the product.  
- `Units Sold`: The number of units sold, which serves as the target variable for our predictive model.

Approach

Data Preprocessing  
  
To prepare the data for modeling, several steps are undertaken:  
  
1. Data Loading: The dataset is loaded from a CSV file using the Pandas library.  
  
2. Handling Missing Values: Missing values in the 'Total Price' column are addressed by filling them with the mean value. This ensures that no data points are lost due to missing values.  
  
3. Data Preprocessing: The 'Total Price' and 'Base Price' columns are rounded to two decimal places for consistency.  
  
 Model Selection  
  
Various regression models are explored to make predictions based on the dataset:  
  
- Linear Regression: A classic regression model that fits a linear relationship between the features (pricing information) and the target variable (units sold).  
  
- Decision Tree Regressor: A tree-based model that can capture non-linear relationships in the data.  
  
- Random Forest Regressor: An ensemble of decision trees that combines their predictions to improve accuracy and reduce overfitting.  
  
- XGBoost Regressor: An advanced gradient boosting algorithm that is highly effective for regression tasks.  
  
 Model Evaluation  
  
The performance of each model is evaluated using two key metrics:  
  
- Root Mean Squared Error (RMSE): This metric quantifies the average deviation of model predictions from actual values. Lower RMSE values indicate better model performance.  
  
- R-squared (R2): R-squared measures the goodness of fit of the model to the data. A higher R-squared value indicates a better fit.  
  
 Visualization  
  
To provide a visual understanding of the model predictions, scatter plots are created for each model. These plots compare the predicted values to the true values for units sold, allowing for a quick assessment of the models' performance.  
  
## Results  
  
After a thorough analysis of each regression model, the following insights are gained:  
  
1. Linear Regression:  
 - RMSE: {lr\_rmse:.2f}  
 - R-squared: {lr\_r2:.2f}  
 - The Linear Regression model provides reasonable predictions but may not fully capture the complexity of the data.  
  
2. Decision Tree Regressor:  
 - RMSE: {dt\_rmse:.2f}  
 - R-squared: {dt\_r2:.2f}  
 - The Decision Tree model shows signs of overfitting, resulting in a lower RMSE but also a lower R-squared compared to Linear Regression.  
  
3. Random Forest Regressor:  
 - RMSE: {rf\_rmse:.2f}  
 - R-squared: {rf\_r2:.2f}  
 - The Random Forest model offers a balance of accuracy and generalization, delivering accurate predictions.  
  
4. XGBoost Regressor:  
 - RMSE: {xgb\_rmse:.2f}  
 - R-squared: {xgb\_r2:.2f}  
 - XGBoost exhibits strong predictive performance with low RMSE and high R-squared, making it a top choice for this sales forecasting task.  
  
In summary, both the Random Forest and XGBoost models outperform the others, providing the most accurate predictions for the number of units sold based on pricing information.