Machine Learning Model Development on Online Retail Dataset

Dataset Overview & Exploratory Data Analysis (EDA)

The dataset consists of online retail transaction data. We first load and concatenate all sheets into a single DataFrame.

Key EDA Steps:
- Checked dataset structure using df.info() and df.describe().
- Handled missing values (Customer ID rows removed).
- Created histograms and scatter plots to visualize feature relationships.

Preprocessing & Feature Engineering

- Created a new column 'TotalSales' as Quantity * Price.
- Selected 'Quantity' and 'Price' as input features (X).
- Target variable: 'TotalSales' (y).
- Split data into training (80%) and testing (20%) sets using train_test_split().

Model Selection & Training

Implemented four regression models:
1. Linear Regression (Baseline)
2. Random Forest Regressor (Ensemble Learning)
3. Gradient Boosting Regressor (Boosting)
4. Decision Tree Regressor (Tree-based)

Performance Metrics: RMSE, R^2 Score, Cross-Validation Score.

Hyperparameter Tuning

Used GridSearchCV to optimize the best-performing models.

Best Parameters:

- Random Forest: n_estimators=50, Best CV R^2 = 0.5287
- Gradient Boosting: learning_rate=0.2, n_estimators=200, Best CV R^2 = 0.7527
- Decision Tree: max_depth=None, min_samples_split=5, Best CV R^2 = 0.3652

After tuning, Gradient Boosting had the best performance.

Final Model Evaluation

Best Model: Gradient Boosting Regressor (Tuned)

Performance:
- RMSE: 11.02
- R^2 Score: 0.9879

Analysis:
- Residuals were closer to zero, indicating better predictions.
- Feature Importance: Quantity had the most impact.

Conclusion

Final Best Model: Gradient Boosting Regressor (Tuned)

Key Findings:
- Linear Regression performed poorly due to non-linearity.
- Decision Trees overfitted the training set.
- Random Forest and Gradient Boosting improved accuracy significantly.

The project successfully:
- Cleaned and prepared real-world data.
- Implemented and compared multiple ML models.
- Improved performance via hyperparameter tuning.