**Extra Mile Documentation: Advanced Insights & Machine Learning**

**Objective:**

To predict trends and automate analysis using SQL Server data and modern Python-based ML tools. This documentation expands upon the technical implementation, tools used, and insights derived

**4.1 Automated Data Cleaning**

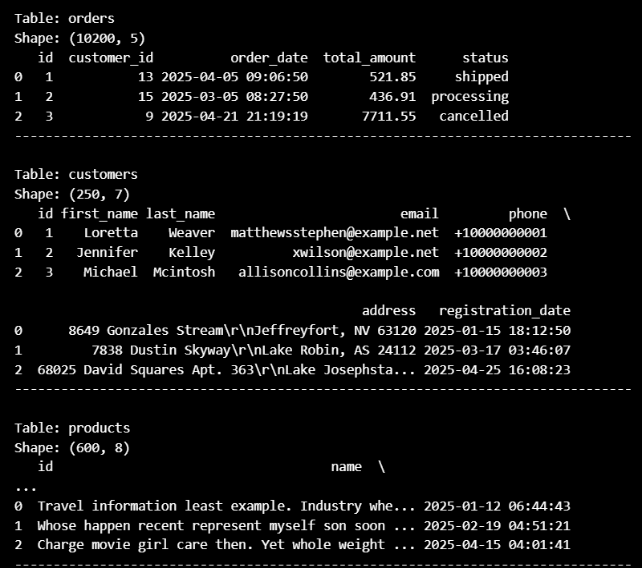
**Tools & Definitions:**

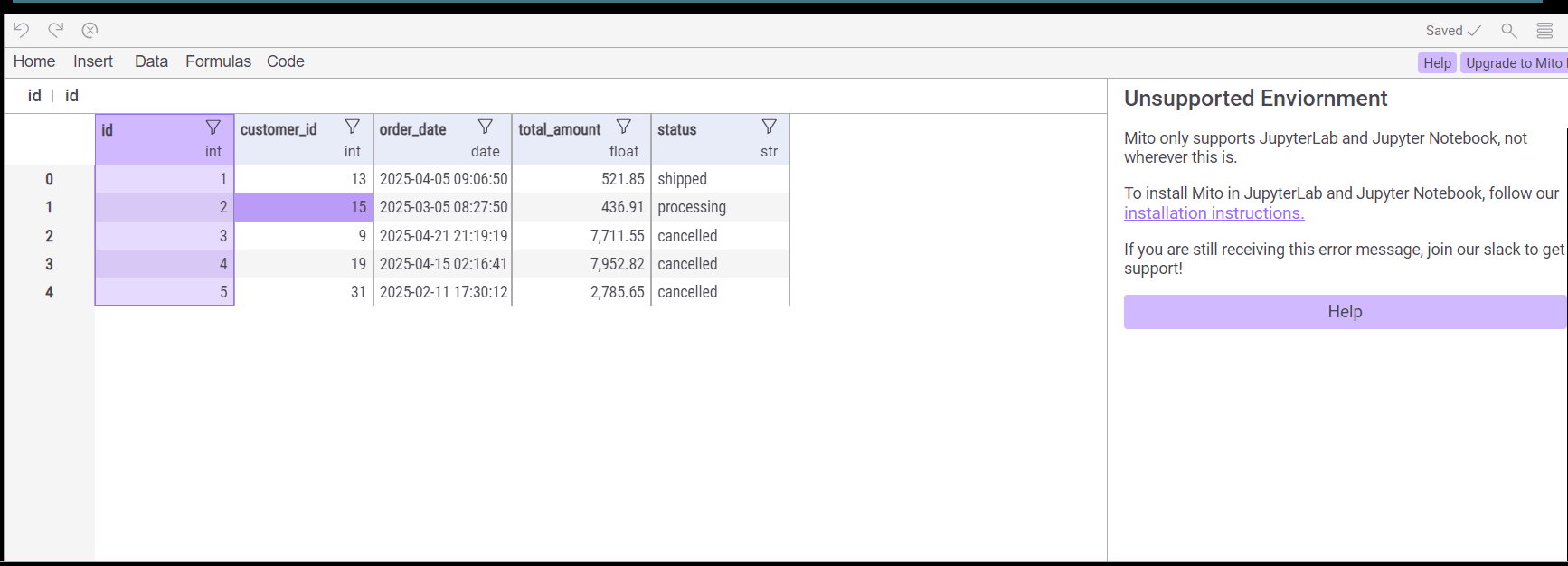
* **Mito: A low-code spreadsheet-like interface integrated with Jupyter that allows for rapid data wrangling and generates Python code behind the scenes.**
* **Pandas: A powerful Python data analysis library used for handling structured data (DataFrames).**
* **pyodbc: A Python module for connecting to ODBC-compliant databases, like SQL Server.**

**Process:**

1. **Established a secure connection to SQL Server using pyodbc and extracted the following relational tables:**
   * **orders, customers, products, categories, order\_details, reviews**
2. **Tables were loaded into a Python dictionary for modular and programmatic access.**
3. **Cleaning operations:**
   * **Filled missing parent\_id fields in the categories table with the median.**
   * **Converted data types for date fields (order\_date, registration\_date) and price fields (unit\_price, total\_price).**
4. **Hierarchical enhancement:**
   * **Used recursive merging to represent parent-child relationships in categories.**
   * **Constructed a visual graph of category relationships using networkx.**

**Insight: Enhanced relational integrity and data quality, setting a strong foundation for downstream modeling and analytics.**

1. Stored tables into a dictionary structure for efficient access and manipulation.
2. Standardized missing values across tables:
   * Filled NaNs in parent\_id using median.
   * Ensured consistent data types for date and numeric fields.
3. Enhanced hierarchical relationships:
   * Merged categories with their parent categories.
   * Visualized the category hierarchy using a networkx directed graph.



**4.2 Predictive Modeling**

**Tools & Definitions:**

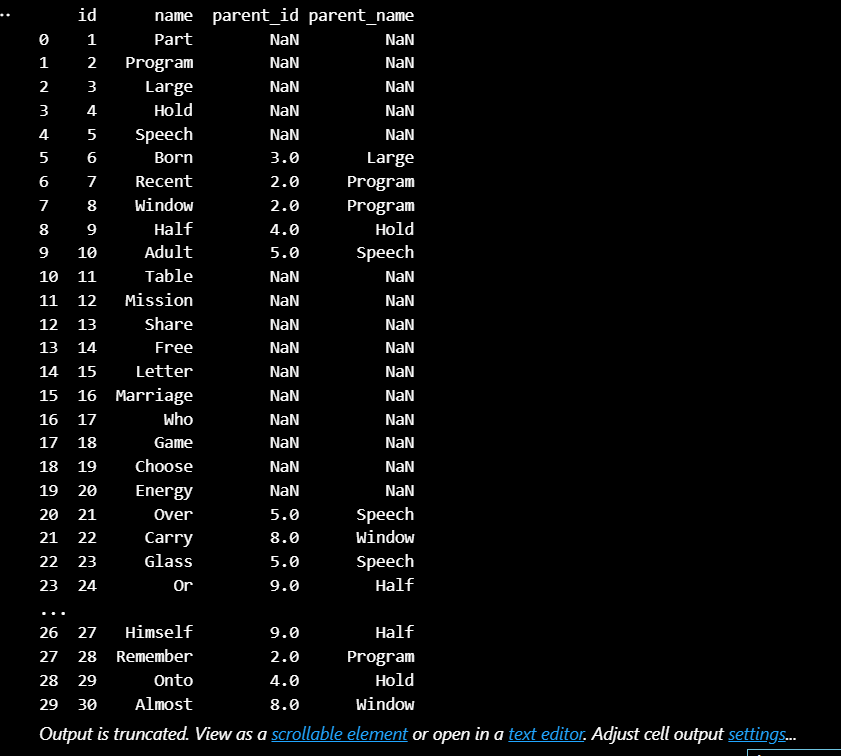
* NumPy: Library for numerical operations and matrix handling.
* scikit-learn: Machine learning library providing tools for preprocessing, modeling, and evaluation.
* XGBoost: An optimized gradient boosting framework for high-performance regression and classification.
* Matplotlib & Seaborn: Visualization libraries for creating plots, charts, and statistical graphs.
* Joblib: Serialization library used to save trained machine learning models.

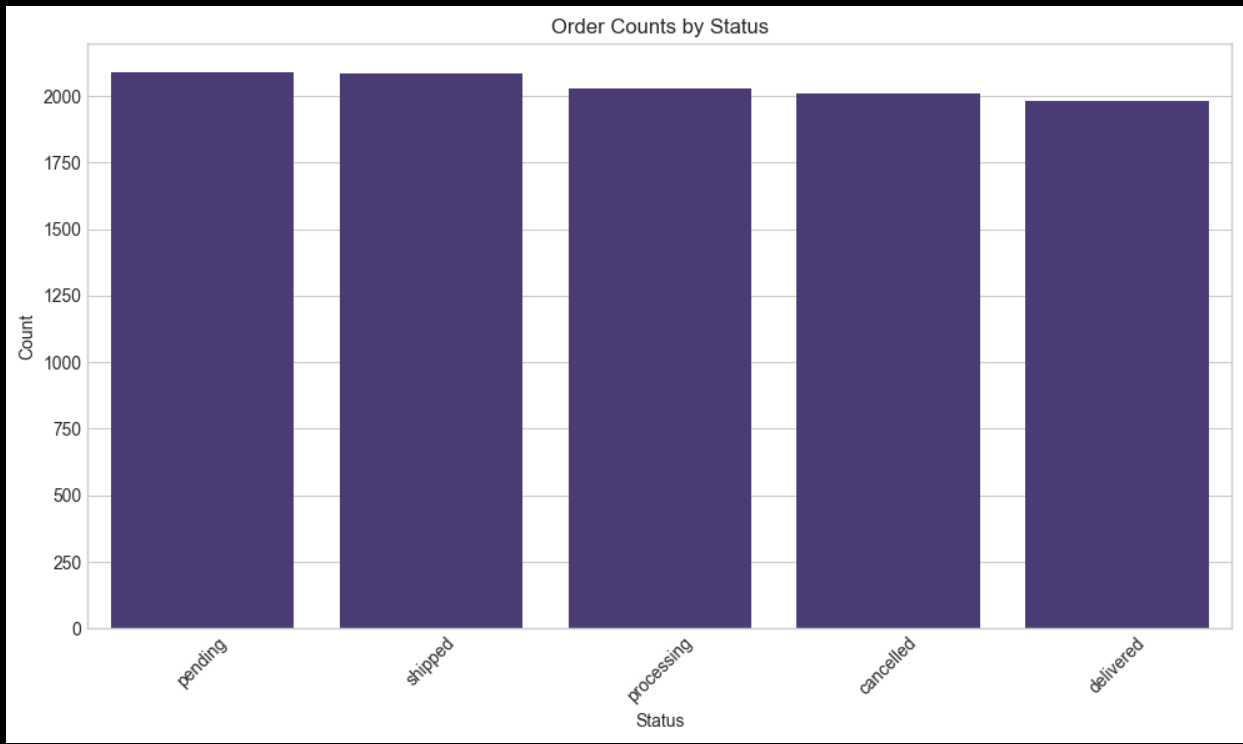
**Workflow:**

**Exploratory Data Analysis (EDA):**

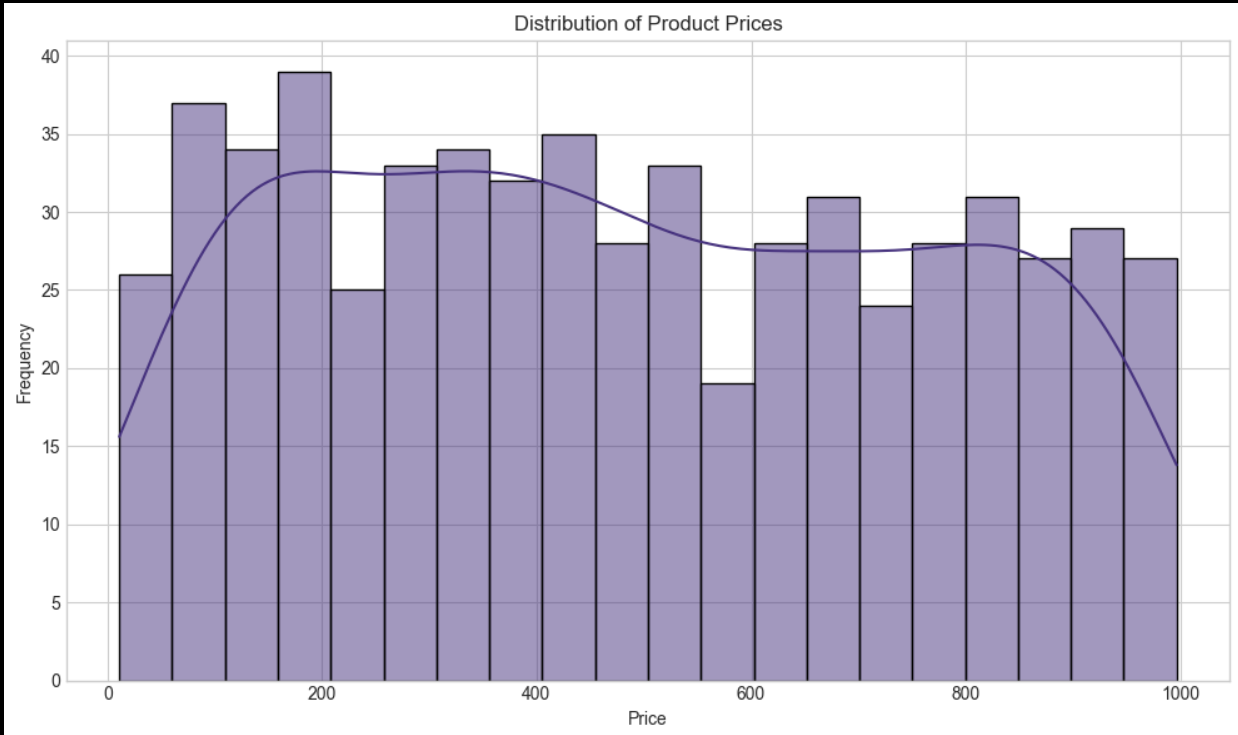
* Generated visualizations to analyze:
  + Order distribution by status and date
  + Product price and review rating histograms
  + Monthly revenue trends and customer acquisition timeline
  + Category-level sales and top products by revenue

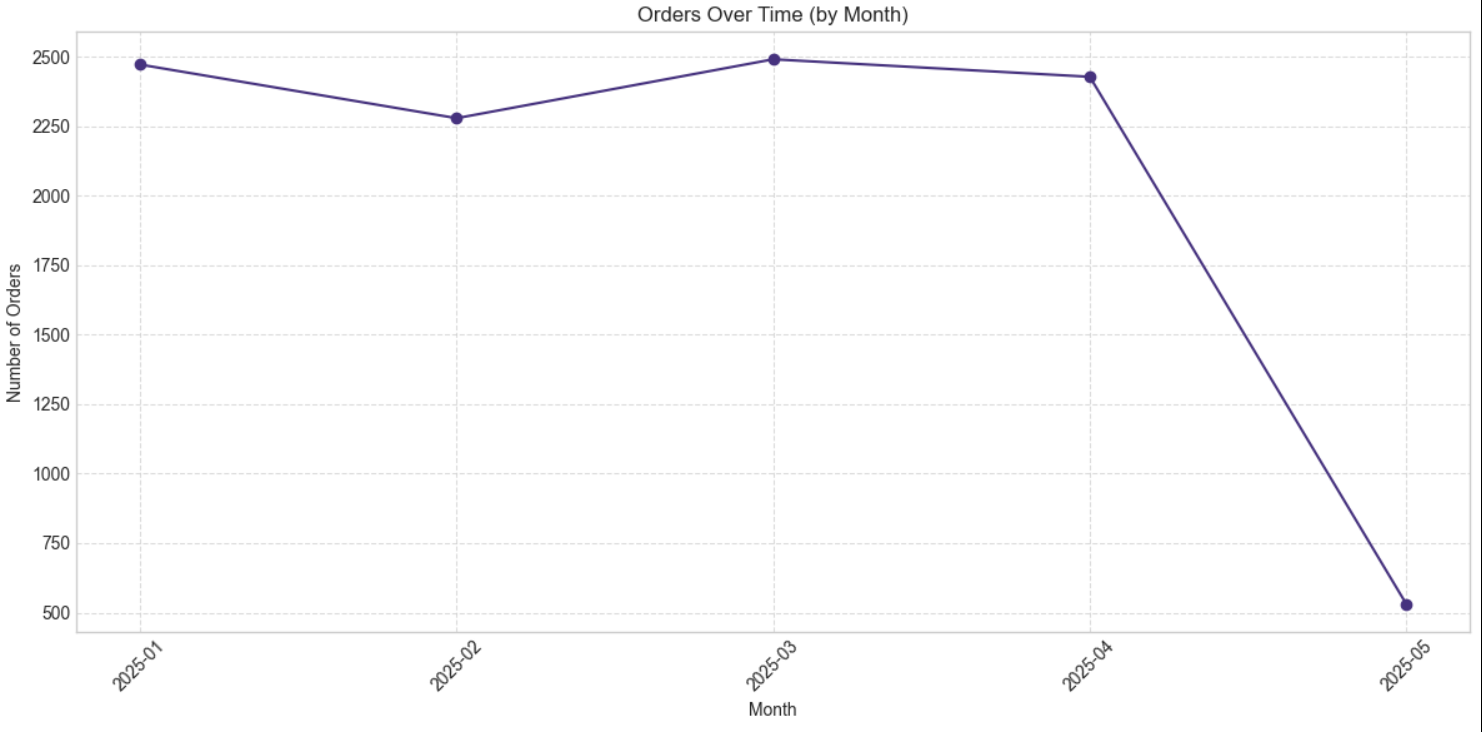
Insight: Identified high-performing products and peak sales periods. Detected imbalance in order statuses indicating potential operational bottlenecks**.**

****

\

**Explore product price distribution**



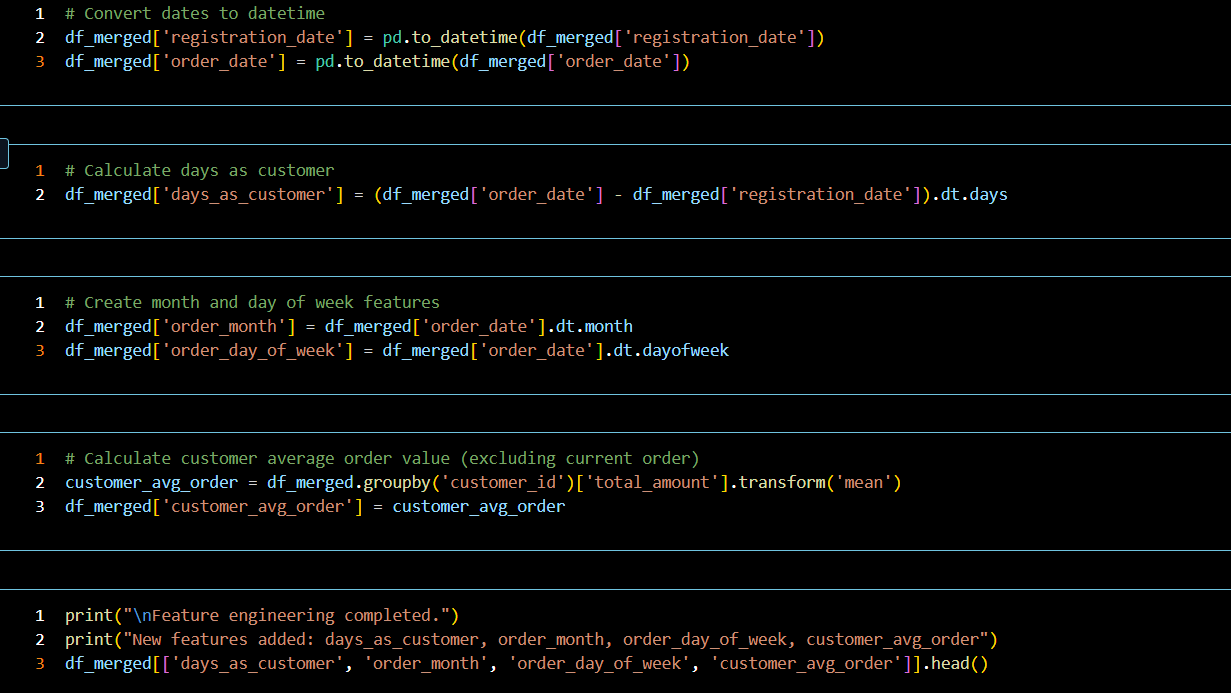


**Explore order trends over time**

**Feature Engineering:**

* Created new variables such as:
  + days\_as\_customer: Days since a customer joined
  + order\_month, order\_day\_of\_week: Useful for identifying seasonality
  + customer\_avg\_order: Calculated per-customer order value trends
  + Repeat buyer flags and historical purchase metrics

Insight: Time-based and customer-centric features significantly improved model accuracy and personalization capabilities.



**Data Preprocessing:**

* Addressed missing values using SimpleImputer
* Scaled numerical features with StandardScaler
* Encoded categorical variables with OneHotEncoder
* Combined all steps into a Pipeline for reproducibility

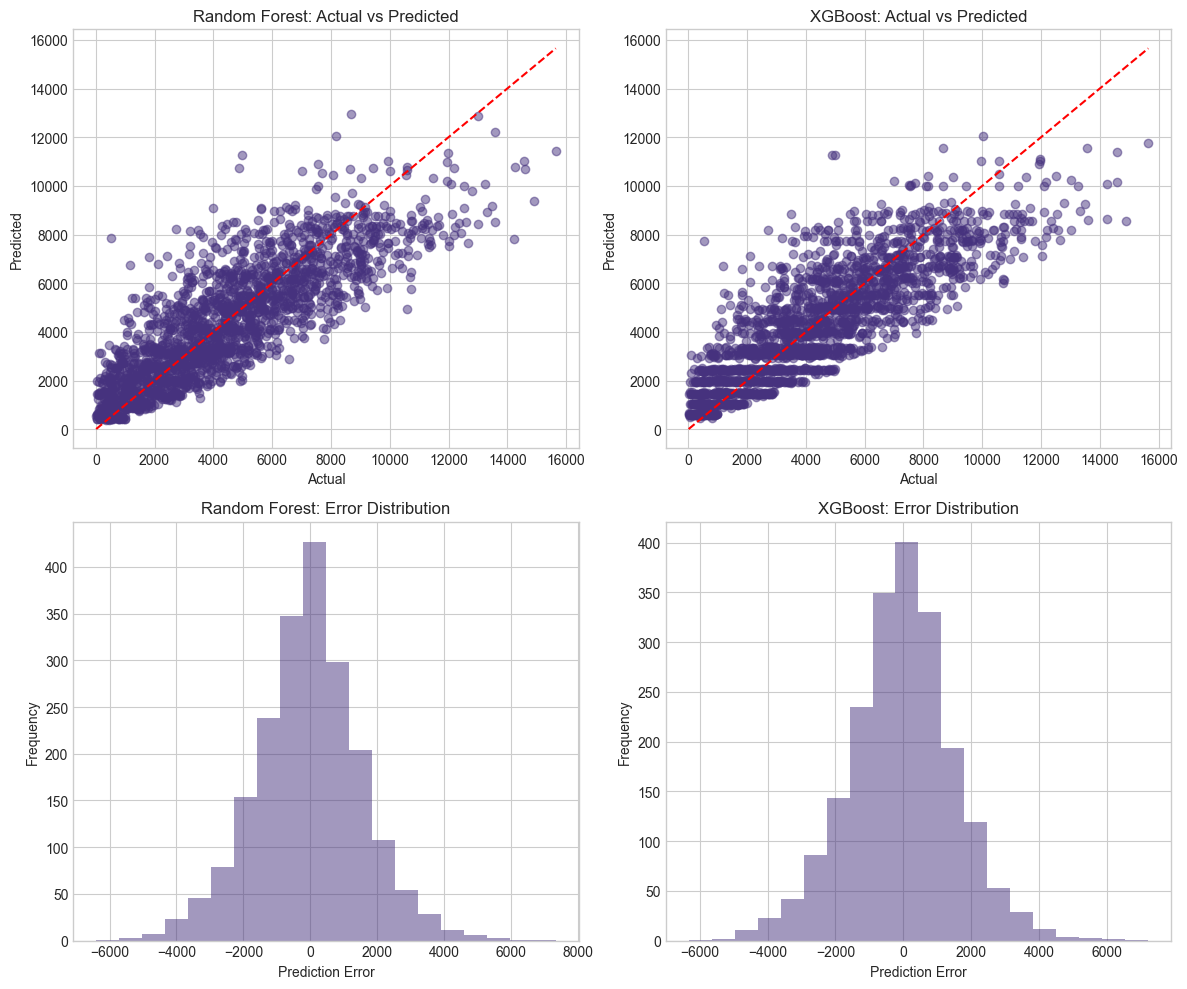
**Modeling:**

1. Random Forest Regressor:
   * A tree-based ensemble model known for robustness and ease of interpretation.
   * Baseline performance model with reliable accuracy.
2. XGBoost Regressor:
   * Used for its gradient boosting mechanism and superior handling of complex relationships.
   * Outperformed Random Forest in predictive performance.

**Evaluation Metrics:**

* MSE, RMSE, MAE, and R² scores were computed.

Insight: XGBoost showed lower RMSE and higher R², making it the preferred model for predicting customer order values.



**Summary**

This extra mile implementation significantly enhanced analytical capabilities by:

* Automating SQL data extraction, cleaning, and enrichment
* Engineering complex features for business-relevant prediction
* Visualizing KPIs to understand sales, orders, and product distribution
* Implementing two ML models (Random Forest & XGBoost) for value prediction
* Enabling NLP-driven data interaction for analysts and stakeholders

**Next Steps:**

* Deploy pipelines on Azure ML with automated retraining
* Schedule ETL and model updates via Azure Functions or Airflow
* Integrate model results into real-time dashboards (Power BI, Streamlit)
* Expand NLP querying capabilities with more domain-specific prompts