Documentation for Project Contributions and Implementation

---

1. Data Preprocessing

Purpose:
Prepare raw data for exploratory analysis and modeling by ensuring it is clean, structured, and feature-rich.

Steps Taken:

- **Loading Data:**
    - Used `load_data.py` to read and validate the input data (CSV files).
    - Checked for duplicate rows and columns to eliminate redundancy.
- **Handling Missing Values:**
    - Missing entries in `CompetitionDistance` were filled with the median distance.
    - For categorical columns like `PromoInterval`, missing values were encoded as `0`.
- **Outlier Detection and Treatment:**
    - Used the Interquartile Range (IQR) method to identify outliers in `Sales` and `Customers`.
    - Applied capping for extreme values to maintain dataset integrity.
- **Feature Engineering:**
    - Extracted new date-based features, such as `DayOfWeek`, `IsWeekend`, `IsHoliday`, and `DaysToHoliday`.
    - Created boolean flags for special occasions and store reopening events.

---

2. Exploratory Data Analysis (EDA)

Purpose:
Gain insights into the data and uncover relationships between features.

Key Contributions:

- **Understanding Customer Behavior:**
    - Visualized trends in `Sales` before, during, and after holidays using line plots.
    - Assessed how promotions influenced sales across store types with grouped bar charts.
- **Correlation Analysis:**
    - Heatmaps revealed a strong correlation (0.85) between `Sales` and `Customers`.
- **Holiday Impact:**

    - Analyzed the impact of `StateHoliday` and `SchoolHoliday` on sales using boxplots.
- **Competitor Influence:**
    - Explored how `CompetitionDistance` and `CompetitionOpenSinceYear` affect sales trends.

Tools and Code:

- **`EDA.ipynb`:** Contains detailed analysis and plots.
- **`data_visualization.py`:** Encapsulates reusable functions for generating plots.

---

3. Modular Design and Reproducibility

Purpose:
Ensure the project can scale, adapt to new data, and be easily reproduced by other team members.

Key Features:

- **Sklearn Pipelines:**
    - Integrated feature scaling and model building into a unified pipeline.
    - Simplified preprocessing and regression tasks with modular steps.
- **Logging:**
    - Employed Python's `logging` library to capture each step in preprocessing, EDA, and modeling.
- **Version Control:**
    - Timestamped serialized models (`.pkl`) for traceability and comparison.

---

4. Modeling Setup

Purpose:
Build robust regression models and test deep learning architectures for sales prediction.

Current Progress:

- Explored tree-based algorithms like Random Forest for initial modeling.
- Conducted preliminary feature importance analysis to refine future modeling efforts.

---

5. Deployment

Purpose:
Serve models for real-time predictions via a REST API.

Implementation Plan:

- Use FastAPI for building lightweight and scalable endpoints.
- Serialize and load models dynamically to support multiple predictions per day.

Future Steps:

- Refine feature selection based on current insights.
- Train and evaluate a Long Short-Term Memory (LSTM) model for time series prediction.
- Finalize API deployment for serving predictions to stakeholders.