Walmart Sales Forecasting

1. Introduction

Sales forecasting is a critical task in retail analytics. Accurate forecasts allow companies like Walmart to manage inventory, plan promotions, optimize staffing, and improve customer satisfaction.

In this project, we aim to **predict future sales** using historical sales data from the Walmart Store Sales Forecast dataset. The task involves:

- Creating **time-based features** (date, month, lags, rolling averages).
- Applying various **regression models** to forecast sales.
- Comparing models using standard evaluation metrics.
- Visualizing the actual vs predicted sales performance.

---

2. Dataset Description

- **Dataset Source**: Walmart Sales Forecasting (Kaggle)
- **Files Used**:
  - `train.csv` – historical sales data for multiple stores and departments.
  - `test.csv` – data for prediction (without target column).

Columns in the dataset:

- **Store**: Store ID
- **Dept**: Department ID
- **Date**: Weekly date (time series index)
- **Weekly_Sales**: Weekly sales (Target variable)
- **Other features**: Holiday indicator, etc.

For this project, we filtered **Store 1, Dept 1** as an example and built forecasting models on that subset.

---

3. Methodology

3.1 Preprocessing

- Converted `Date` column to datetime.
- Sorted data by date for time series consistency.

- Created weekly frequency (`asfreq('W')`).
- Filled missing values using forward fill.

3.2 Feature Engineering

- **Time-based features**: `year`, `month`, `weekofyear`, `dayofweek`, `is_month_start`, `is_month_end`.
- **Lag features**: `lag_1, lag_2, lag_3, lag_4, lag_12, lag_26, lag_52` to capture past sales trends.
- **Rolling statistics**: Rolling mean (4, 12 weeks) and rolling std to capture seasonality.

3.3 Train/Test Split

- Training: all weeks except last `n` weeks.
- Testing: last `n` weeks (same horizon as test dataset).
- Time-aware splitting was applied (not random split).

---

4. Models Implemented

1. **Naive Baseline (Lag-1)**
    - Predicted sales as previous week's sales.
2. **Linear Regression (LR)**
    - Simple linear regression on all engineered features.
3. **Random Forest Regressor (RF)**
    - Non-linear model capturing complex interactions.
4. **LightGBM (Gradient Boosting)**
    - Time-series cross-validation with boosting trees.
5. **XGBoost**
    - Another gradient boosting approach with strong performance on structured data.

---

5. Evaluation Metrics

Models were compared using:

- **MAE (Mean Absolute Error)**
- **RMSE (Root Mean Squared Error)**
- **R^2 Score (Goodness of fit)**