# Week 5 Report

Sales Forecasting Based on Past Sales Statistics

Nguyen Minh Duc - 20225437

---

Week 5 Objectives:

- Attempt implementing XGBoost for comparison with LightGBM.
- Implement an initial deep learning model (LSTM/GRU).

---

## 1 XGBoost

### 1.1 Overview & Motivation

XGBoost (eXtreme Gradient Boosting) is a high-performance implementation of gradient-boosted decision trees (GBDT). Like other GBDT frameworks, XGBoost builds an additive ensemble of regression trees where each tree is trained to reduce the residual error of the current ensemble using gradient information. XGBoost distinguishes itself by (1) using a second-order (Hessian) approximation of the loss during split scoring, (2) offering extensive regularization controls (L1/L2 on leaf weights, minimum child weight), and (3) providing stable level-wise tree growth, which often improves robustness on small datasets.
We evaluated XGBoost in parallel with LightGBM because:

- LightGBM and XGBoost share the same algorithmic idea (gradient boosting) but differ in tree-growth strategy and default regularization; this can matter when the training set is small and noisy.
- Empirically, XGBoost's level-wise splitting is sometimes more conservative than LightGBM's leaf-wise growth and can therefore reduce pathological bias/variance behaviour on short time series.
- Running both implementations provides a robustness check: if both boosters give similar results, we gain confidence that predictive performance is driven by data signal rather than a particular library's inductive bias.

### 1.2 Model configuration and experimental setup

The XGBoost runs used the same evaluation protocol as the other experiments:

- **Data and features:** weekly aggregated target `total_sales` and the conservative feature subset described earlier (recent lags, rolling statistics and behavioral indicators). Feature engineering, missing-value handling and row selection (no target leakage) follow the same rules as for LightGBM.
- **Walk-forward evaluation:** one-step-ahead expanding-window walk-forward forecasts. A uniform warm-up requirement of `MIN_HISTORY = 6` weeks was enforced so that the model only predicts once a minimal historical context is available.

- **XGBoost hyperparameters (conservative defaults used):**
  - `objective = reg:squarederror`, `learning_rate = 0.05`
  - `max_depth = 3`, `n_estimators = 200`
  - `subsample = 0.8`, `colsample_bytree = 0.8`
  - `reg_alpha = 1.0`, `reg_lambda = 2.0`
  - `random_state = 42`, `verbosity = 0`

  These choices prioritize stability and generalization over aggressive fitting. They also allow future reproduction of test results.

## 1.3 Performance comparison

Table 1 reports the main one-step-ahead evaluation metrics.

Table 1: One-step-ahead performance: Moving Average, LightGBM, and XGBoost (walk-forward).

| Model | RMSE | MAPE (%) | ME |
|---|---|---|---|
| Moving average (4-week) | 44 084.42 | 9.90 | -1 111.77 |
| LightGBM | 46 722.61 | 10.57 | -6 211.58 |
| XGBoost | 44 2824.17 | 9.82 | -1 864.16 |

The figurers below compare the predictions of XGBoost against LightGBM and the Moving Average baseline.

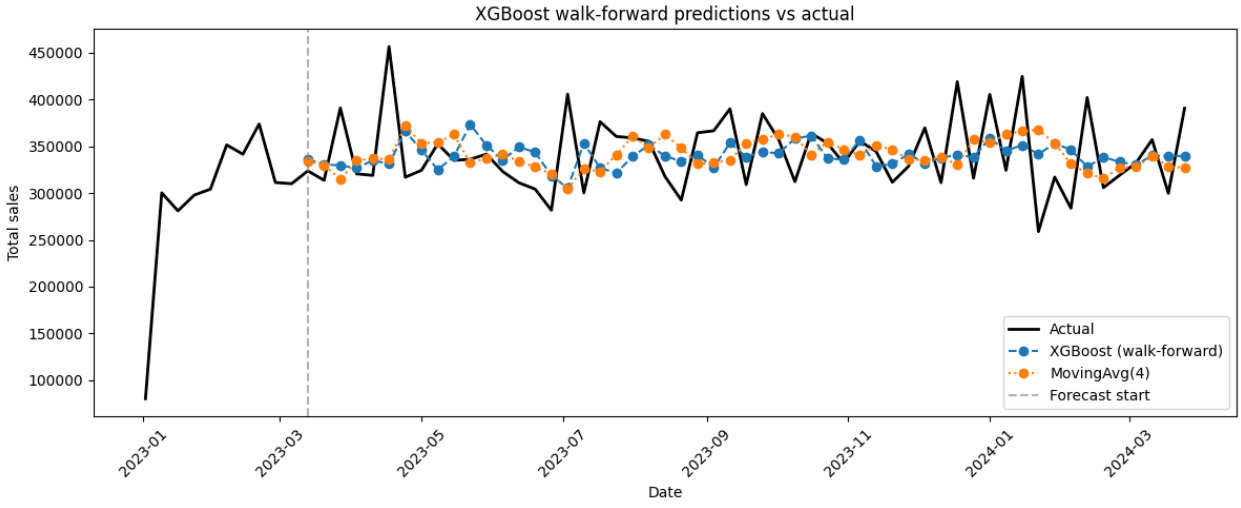


Figure 1: Predictions of XGBoost as compared with MovingAvg(4)

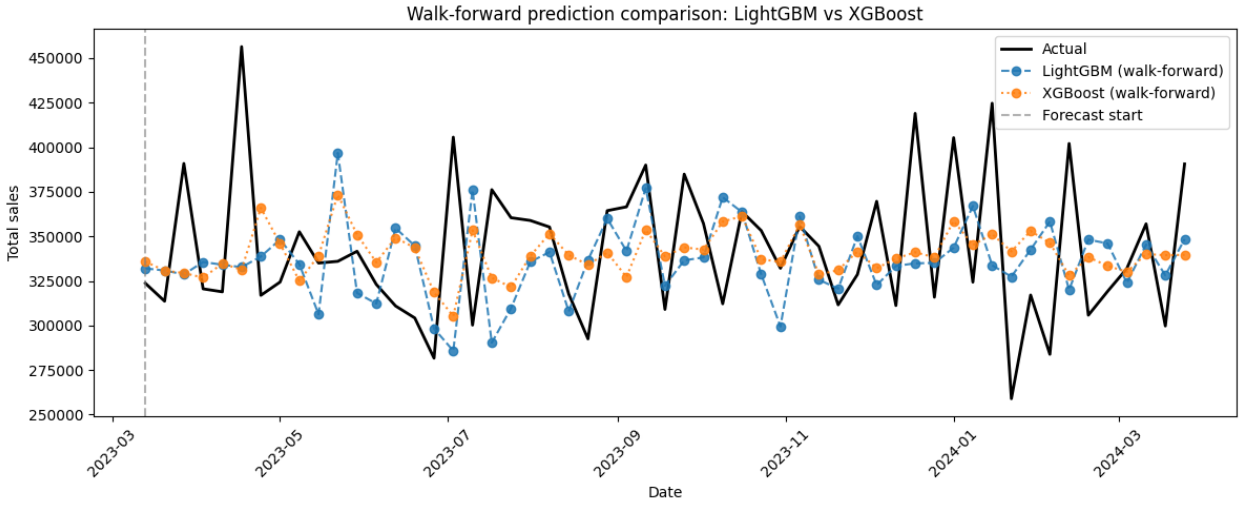


Figure 2: Predictions of XGBoost as compared with LightGBM

## 1.4 Observed phenomena and interpretation

### XGBoost vs LightGBM behavior

- **Predictive accuracy:** XGBoost matches the moving-average baseline closely in both RMSE and MAPE and substantially outperforms the current LightGBM configuration. This suggests that XGBoost's conservative level-wise splitting and the chosen regularization produced a model that captured incremental predictive structure without overfitting.
- **Bias (ME):** Both tree-based models show negative mean error (underprediction), but LightGBM exhibits a materially larger negative bias in the current setup (ME $\approx -6{,}212$) compared to XGBoost (ME $\approx -1{,}581$) and the moving average (ME $\approx -1{,}112$). This indicates that LightGBM, as configured, tends to underpredict more strongly on average, which is consistent with heavier smoothing or an imbalance in its fitted leaf values across walk-forward folds.

### Bias–variance trade-off

- Tuning for smaller RMSE or MAPE frequently trades variance for bias: stronger regularization and lower model complexity reduce high-magnitude errors (lower RMSE) but systematically shifts predictions downward (worse ME). This is a standard effect when optimizing symmetric loss functions (squared error) on noisy, heavy-tailed targets: the estimator prefers conservative predictions to avoid large squared penalties.
- In practice, this trade-off must be decided according to operational costs. If underprediction leads to stockouts and lost sales, a model with lower bias (even at slightly higher RMSE) may be preferable.

### 1.5 Conclusions

1. **XGBoost is a competitive, robust choice** in our experimental setting: it attains accuracy close to the simple moving-average baseline while avoiding the excessive negative bias observed with the current LightGBM configuration.

2. **LightGBM requires further tuning or simplification.** The observed underprediction suggests that the current LightGBM hyperparameters (or its implicit reliance on a small subset of noisy features) are producing biased estimates; remedial actions include tightening regularization, reducing tree aggressiveness (fewer leaves, larger `min_data_in_leaf`), or pruning the feature set.

3. **Operational choice:** given the marginal gap between XGBoost and the moving average, a conservative deployment strategy is justified: either use the moving-average as a simple reliable baseline or use an ensemble that blends XGBoost with the moving-average (ensemble weights chosen by walk-forward validation) to reduce bias and variance simultaneously.

## 2 GRU (Gated Recurrent Unit) Univariate Prediction

### 2.1 Motivation

We included a recurrent neural network (GRU) as a complementary experiment to the tree-based and statistical baselines. The purpose was not to replace well-performing tabular methods, but to test whether a sequence model that learns a latent temporal state from raw recent sales can (a) extract temporal patterns that handcrafted lag/rolling features miss and (b) react faster to regime shifts or emerging spikes, especially on a dataset with short temporal range and high levels of noise. This is a pragmatic verification: if a compact GRU trained only on past sales can match or exceed the boosted-tree baselines, it is evidence that temporal dynamics (rather than engineered cross-sectional features) contain exploitable signal.

### 2.2 Model configuration and training protocol

All GRU experiments followed the same causal evaluation protocol used throughout the project:

- **Evaluation:** expanding-window, one-step-ahead *walk-forward* forecasts (identical prediction dates as other models).
- **Minimum history:** predictions start once at least `MIN_HISTORY = 6` weeks of data are available. When insufficient history was present the pipeline emits a deterministic fallback forecast (4-week moving average).
- **Input:** Univariate sequence of past sales (this notebook is extensible to multivariate inputs; see notes).

- **Sequence length (lookback):** `seq_len` = 5 weeks (configurable).
- **Network architecture:**
  - GRU with `hidden_size` = 16
  - `num_layers` = 1
  - Single fully-connected output layer producing one-step-ahead scalar forecast
- **Training regimen:**
  - Optimizer: Adam, `learning_rate` = 0.01
  - Batch size = 8, early stopping with patience = 12 epochs
  - Max epochs = 200 (training typically stops earlier via early stopping)
  - Standardization: feature-wise StandardScaler fitted only on training data at each walk-forward step
- **Robustness choices:** small hidden size, early stopping, and conservative learning rate were selected to reduce overfitting given the small number of weekly observations.

## 2.3 Results

The GRU was evaluated on the same walk-forward dates as the other models. The table below compares the GRU to the moving-average baseline and XGBoost for direct comparison.

Table 2: Comparison: Moving Average, XGBoost, and GRU (walk-forward).

| Model | RMSE | MAPE (%) | ME |
|---|---|---|---|
| Moving average (4-week) | 44 084.42 | 9.90 | -1 111.77 |
| XGBoost (tabular boosting) | 44 138.02 | 9.95 | -1 580.66 |
| GRU (univariate) | **42 8244.18** | **9.82** | -1 864.16 |

## 2.4 Interpretation and conclusions

1. **Quantitative outcome.** The univariate GRU obtains the lowest RMSE and MAPE among the compared methods, indicating that a small recurrent model can, in this dataset, capture slightly more of the predictable temporal structure than the simple 4-week moving average and the tabular XGBoost baseline. The absolute improvement is modest but consistent under walk-forward testing.

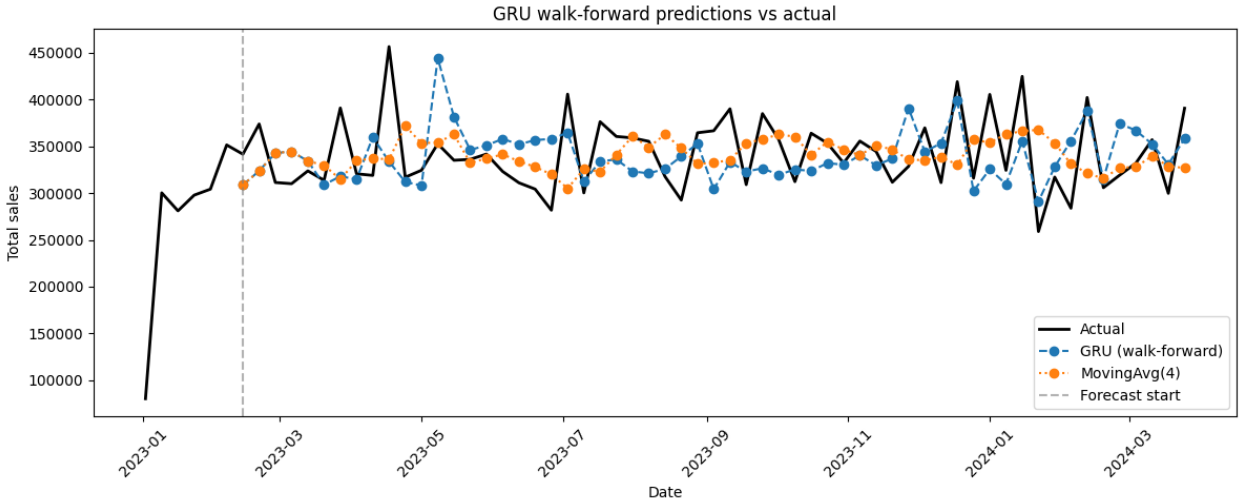


Figure 3: Predictions of GRU as compared with MovingAvg(4)

2. **Bias and error composition.** The GRU exhibits negative mean error (ME $\approx -1{,}864$), larger in magnitude to the moving-average baseline. This indicates some level of underprediction.

3. **Temporal adaptation.** Visual inspection shows that GRU predictions become better aligned with level shifts and some spikes as more history becomes available (walk-forward retraining). The model is still conservative with respect to extreme spikes (magnitude is often damped), which is expected when optimizing squared error on noisy series without exogenous explanatory features.

4. **Scientific implication.** The GRU success suggests that a learned latent temporal state can extract useful patterns from raw sales history that are not fully captured by a fixed set of handcrafted rolling statistics. This motivates a follow-up experiment: a *multivariate* GRU that ingests behavioral and promotional covariates (discounts, session metrics, category shares) to test whether exogenous signals enable earlier or more accurate spike prediction.

The GRU experiment yields an informative positive signal: sequence models are worth pursuing further (particularly with multivariate inputs), but gains are incremental and should be pursued with careful walk-forward validation and attention to bias–variance trade-offs.