Week 3 Report

Sales Forecasting Based on Past Sales Statistics

Nguyen Minh Duc - 20225437

Week 3 Objectives:

- Feature engineering (lags, rolling stats, seasonal indicators, behavioral features).
- Implement baseline models (naive, moving average, basic ARIMA/Prophet).

1 Feature Engineering

This section describes the data transformation and feature engineering steps applied to construct a modeling-ready weekly time series. This aims to encode temporal dynamics, behavioral patterns, and commercial signals for statistical forecasting models, while balancing feature richness against the limited length of the time series.

1.1 Aggregation and Preprocessing

The original transaction-level dataset was aggregated to a **weekly frequency**, chosen as a compromise between excessive noise at the daily level and insufficient data at coarser scales. Each observation corresponds to one calendar week and includes aggregated sales, customer behavior, and operational metrics. The weekly timestamp (`Date`) represents the start of each aggregation window.
Holiday indicators and yearly encodings were intentionally excluded, as prior exploratory analysis showed weak and inconsistent relationships between holidays and sales, and the dataset spans only approximately 15 months, insufficient for learning robust annual patterns.

1.2 Base Aggregated Features

At the weekly level, the following core metrics were computed:

- **Sales and volume**: total weekly revenue (`total_sales`), number of orders (`n_orders`), and total quantity sold (`total_quantity`).
- **Pricing**: average unit price (`avg_unit_price`) and quantity-weighted average unit price (`avg_unit_price_wt`).
- **Customer behavior**: average session duration, average pages viewed, proportion of returning customers, and number of unique active customers.
- **Promotions and operations**: average discount amount, proportion of orders receiving discounts, average delivery time, and average customer rating.

These features summarize weekly demand, pricing conditions, customer engagement, and service quality.

1.3 Lag Features

To capture short- and medium-term temporal dependencies, lagged versions of selected variables were created. Specifically, lags of 1, 2, 4, and 12 weeks were computed for key signals such as:

- total sales,
- total quantity,
- customer engagement metrics (session duration, pages viewed),
- discount-related measures,
- proportion of returning customers.

Lag features enable autoregressive models to leverage recent historical information without explicitly increasing model order.

1.4 Rolling Statistics

Rolling window features were introduced to summarize recent trends and volatility. For multiple window sizes (3, 4, and 12 weeks), the following statistics were computed:

- rolling mean and rolling standard deviation,
- rolling median,
- rolling interquartile range (25th and 75th percentiles).

These features provide smoothed representations of local trends, dispersion, and distributional shifts, which are particularly useful when the raw series exhibits high volatility.

1.5 Behavioral and Interaction Features

Additional derived features were designed to encode interpretable behavioral relationships, including:

- average quantity per order,
- orders per customer,
- interaction terms such as discount amount and returning-customer rate multiplied by session duration.

Such features aim to capture nonlinear effects between customer engagement, promotional intensity, and realized sales.

1.6 Category Composition Features

To reflect changes in product mix over time, weekly category shares were computed for major product groups (e.g., Electronics, Home & Garden, Sports, Fashion, Toys). These features encode demand composition rather than absolute volume and may help explain structural changes in sales dynamics.

1.7 Baseline Seasonal Features

For benchmarking purposes, a simple seasonal-naive estimate with a 4-week period and its residual were included. These features serve as references for evaluating whether more complex models can outperform basic seasonal persistence.

1.8 Final Remarks

The resulting feature set contains a large number of engineered variables relative to the number of available weekly observations. While this raises concerns about overfitting, the feature-rich representation is primarily intended to support baseline comparisons and exploratory modeling. Subsequent modeling stages will require careful feature selection or regularization to ensure robust generalization.

2 Baseline Models

This section documents the baseline forecasting experiments, the evaluation metrics used, the models implemented, and a concise discussion of the empirical results. All models were evaluated with a *walk-forward* (rolling-origin) one-step-ahead procedure using a uniform minimum history window of six weeks. This produces realistic, deployment-like error estimates because each forecast uses only information that would have been available at prediction time.

2.1 Evaluation metrics and rationale

We report three complementary metrics:

- **Root Mean Squared Error (RMSE)**: measures absolute error in the original units and penalizes large deviations. RMSE is useful to quantify the scale of forecasting errors.
- **Mean Absolute Percentage Error (MAPE %)**: a scale-free relative error expressed in percent, useful to compare performance across series with different magnitudes.
- **Mean Error (ME)**: the (signed) average error $\frac{1}{n}\sum(\hat{y}_t - y_t)$. ME indicates systematic bias (positive ME = overprediction, negative ME = underprediction).

These three metrics jointly provide (1) scale-sensitive accuracy (RMSE), (2) relative accuracy understandable to stakeholders (MAPE), and (3) bias diagnostics (ME). We selected them because they reflect both technical goodness-of-fit and operational concerns (e.g., persistent over- or under-forecasting).

2.2 Baseline models implemented

All baselines are one-step-ahead and evaluated using walk-forward updating (predictions are made sequentially and the true observation is revealed and used for the next step). The implemented models are:

Naive: the forecast is the most recent observed value (persistence).

Seasonal-naive (4-week): the forecast for week t equals the observation from week $t-4$ (4-week lag, approximating monthly seasonality).

Moving average (4-week): the forecast is the mean of the last 4 observed weeks (a simple smoothing baseline).

ARIMA(1,1,1): a small, standard autoregressive integrated moving-average model refit at each walk-forward step (order chosen conservatively to avoid over-parameterization given limited data).

Prophet: a structural time-series model (trend + seasonal components) refit at each step. Yearly seasonality was disabled due to the short series length; weekly seasonality was left at default settings but was weak in the EDA.

2.3 Prediction charts

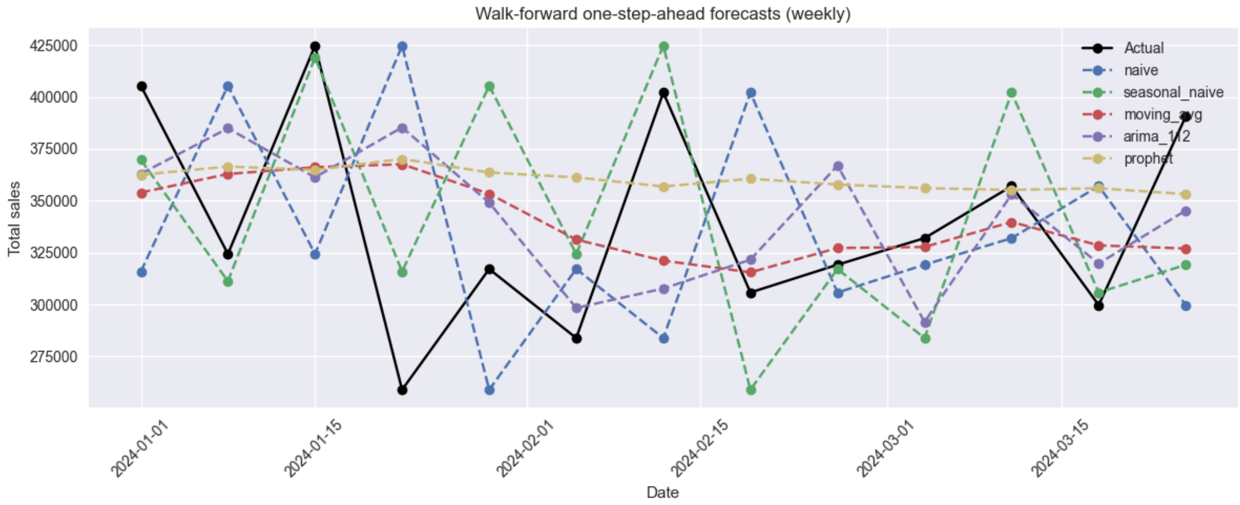

Figure 1: Predictions of all models (last 13 weeks)

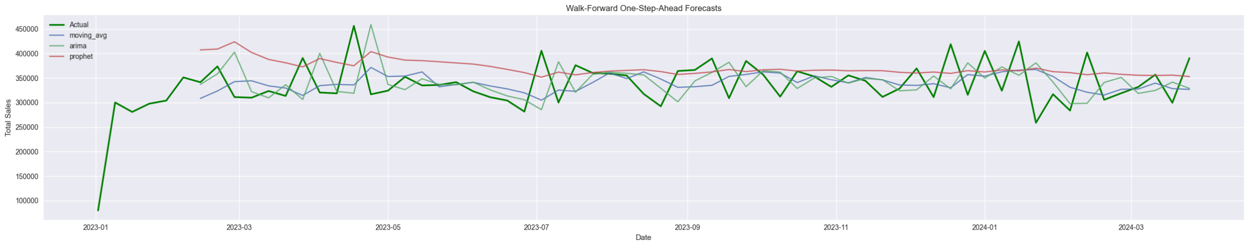

Figure 2: Predictions of Seasonal Naive, ARIMA, and Prophet models (59 weeks)

2.4 Results

Table 1 summarizes the evaluation metrics for each model (computed on every valid one-step prediction produced under the walk-forward evaluation described above).

Table 1: Baseline model performance (walk-forward one-step-ahead).

Model	RMSE	MAPE (%)	ME
naive	63 428.41	13.90	-664.72
moving_avg	43 698.10	9.99	-1 325.87
seasonal_naive	52 395.30	12.14	-2 456.74
arima	54 110.44	11.81	1 239.32
prophet	52 868.51	13.96	28 891.20

2.5 Remarks on model behavior

The empirical results and the prediction chart indicate the following patterns:

- **Moving average (4-week) performs best overall** in both absolute and relative terms (lowest RMSE and lowest MAPE). This indicates that a simple short-window smoothing baseline captures most of the predictable component in the weekly series. Given the high volatility and limited autocorrelation found in the EDA, this result is consistent with the notion that short-term local averages are strong, robust predictors.

- **Naive and seasonal-naive** behave like persistence baselines; the seasonal-naive (lag-4) performs worse than the 4-week moving average, meaning that a pure copy-from-4-weeks-ago rule is less accurate than the smoothed recent average.

- **ARIMA(1,1,1) provides smoothed, autoregressive forecasts** and displays a modest over-prediction bias (positive ME). Early in the series, with minimal training history, ARIMA tends to produce forecasts that resemble a lagged version of the observed series (persistence-like behavior). This arises because autoregressive components dominate and parameter estimates are conservative with small sample sizes, so the model effectively extrapolates recent levels.

- **Prophet produced a large positive ME** (substantial overprediction). Examination of the forecast chart shows that Prophet produces gentle trend behavior and, particularly in early steps, slightly exaggerated level estimates. Prophet's trend assumptions apparently caused a strong positive bias on average for this dataset. This aligns with the EDA finding of weak seasonality and noisy week-to-week variation.

- **All models are relatively smooth and fail to capture sharp spikes.** The series contains sizable idiosyncratic deviations (promotions, one-off demand surges) that are difficult for simple endogenous time-series models to predict. This explains why smoothing-based baselines (moving average) perform particularly well in aggregate.

2.6 Conclusions

1. **A simple moving-average (4-week) baseline is difficult to beat.** For this dataset (weekly, short span, high volatility), the moving average achieves the best RMSE and MAPE; therefore any more complex model must demonstrably outperform this baseline to be considered useful in practice.

2. **Bias diagnostics matter.** ARIMA and Prophet exhibit positive systematic bias (especially Prophet). If forecasts are used for inventory or replenishment, a positively biased model could lead to unnecessary overstocking; conversely, negative bias would risk stockouts. ME should therefore be used alongside RMSE and MAPE in model selection and calibration.

3. **Next steps.** Given the limited autocorrelation and strong residual variability:

    - pursue feature-driven models (e.g., tree-based regressors using the engineered behavioral, discount and category-share features) that can incorporate cross-sectional signals not available to pure univariate time-series models;

    - apply careful feature selection or regularization (low feature-to-sample ratio) to avoid overfitting given the small number of weekly observations.