Sales Prediction Framework

Overview

This write-up outlines a straightforward approach to improving sales forecasts for retailers who may still be using older merchandise planning systems. Instead of jumping straight into a full system overhaul. The data used is simulated to mirror real-world behavior—and uses tools like LightGBM and Facebook Prophet to make better predictions. The predictions are more accurate and flexible than just comparing to last year's numbers, which is what most legacy systems rely on. The goal here is to help retailers make smarter inventory decisions, cut down on costs, and react faster to market shifts—without having to rip and replace everything right away. It's a practical way to move from legacy systems to more modern, data-driven planning, one step at a time.

I built a framework to predict future sales through the following steps:

1. Historical sales data was cleaned and organized.
2. Features (calculated or derived fields) were created to drive better predictions.
3. Off-the-shelf machine learning models—LightGBM and Facebook Prophet—were tested. Both are well-suited for time series predictions. I evaluated which model performed better in my scenario.
4. The framework was developed entirely in Python and included dynamic clustering to group similar products and improve forecast accuracy.

Data

The dataset was simulated to reflect real-world item-level sales behavior and included:

- 5,080 items with weekly sales data spanning three years
- A variety of product types, including seasonal, basic, discontinued, and newly introduced items
- Items with year-over-year sales increases and decreases, as well as holiday-influenced sales patterns
- Built-in data noise to mimic real-world variability

Simulated data allows transparent, reproducible research while protecting sensitive business information. Although the models were validated against real data, those comparisons are omitted here to maintain confidentiality. The simulated data is presented for open discussion.

Model Frameworks Used

Light Gradient Boosting Machine (LightGBM)

LightGBM is a fast, powerful machine learning algorithm effective at modeling complex data relationships. It is well-suited for time series forecasting.

Key features of LightGBM:

- **Efficiency:** Processes large datasets quickly, ideal for retail-scale problems
- **Flexibility:** Captures non-linear relationships critical to sales trends
- **Feature Importance:** Highlights the factors that most influence predictions

LightGBM builds a series of decision trees using gradient boosting to iteratively improve predictions.

Facebook Prophet

Facebook Prophet is a user-friendly tool for forecasting time series data with strong seasonality and multiple years of history.

Key advantages of Prophet:

- **Handles Seasonality:** Automatically accounts for daily, weekly, and yearly patterns
- **Flexible:** Supports holiday effects and irregular sales events
- **Robust:** Handles missing values and outliers gracefully

Prophet uses a decomposable model to isolate trend, seasonality, and holiday components for interpretable predictions.

Spoiler Alert: LightGBM ultimately outperformed Prophet (and last year-based forecasts) but required more effort, including extensive feature engineering, optimizing model settings to improve prediction accuracy(hyperparameter tuning), and trial-and-error experimentation. Prophet required minimal setup and offered decent results with less complexity.

Features

Features explain the "why" behind the data. They are the variables that describe and provide context for sales behavior. For example, a binary "holiday" flag was added to identify weeks affected by holidays.

Only raw sales unit data was generated—no external metadata (e.g., promotions, pricing, weather) was included. Without features, LightGBM performed poorly, while Prophet did comparably to last year's data.

Engineered features added to the LightGBM dataset:

1. **Lagged Variables**
    - Past sales data points used as predictors to capture patterns over different time frames (e.g., last week, last month)
    - Lagged variables calculated for 1, 4, 13, and 52 weeks ago
    - Captures short-term, monthly, quarterly, and annual patterns
2. **Rolling Aggregates**
    - Calculated over 4, 13, and 52-week windows
    - Metrics: Mean, Median, and Standard Deviation
    - Captures trends and volatility across time scales
3. **Clustering Techniques**
    - Applied five clustering methods:
        - K-means
        - Gaussian Mixture
        - Hierarchical
        - Time Series K-means
        - Rolling Sales Trend
    - Groups similar items to uncover shared behaviors

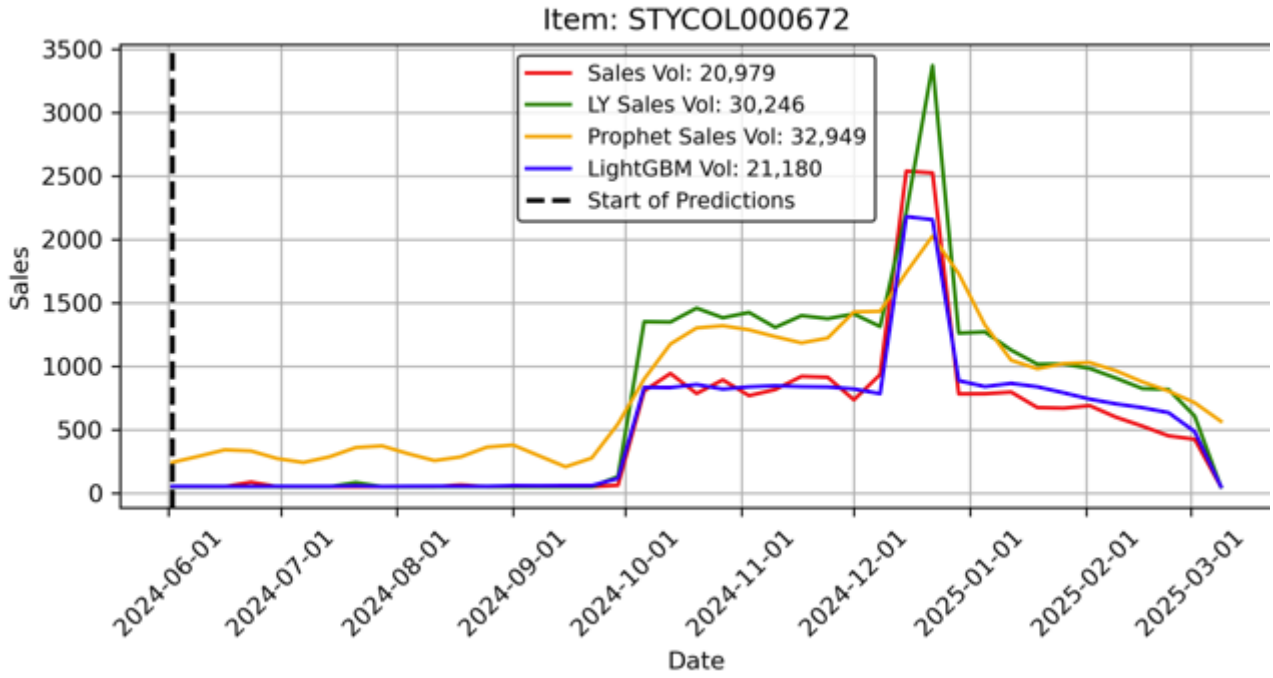
4. **Holiday Indicator**
    - Binary variable to identify holiday-affected weeks
    - Accounts for seasonal sales spikes or dips

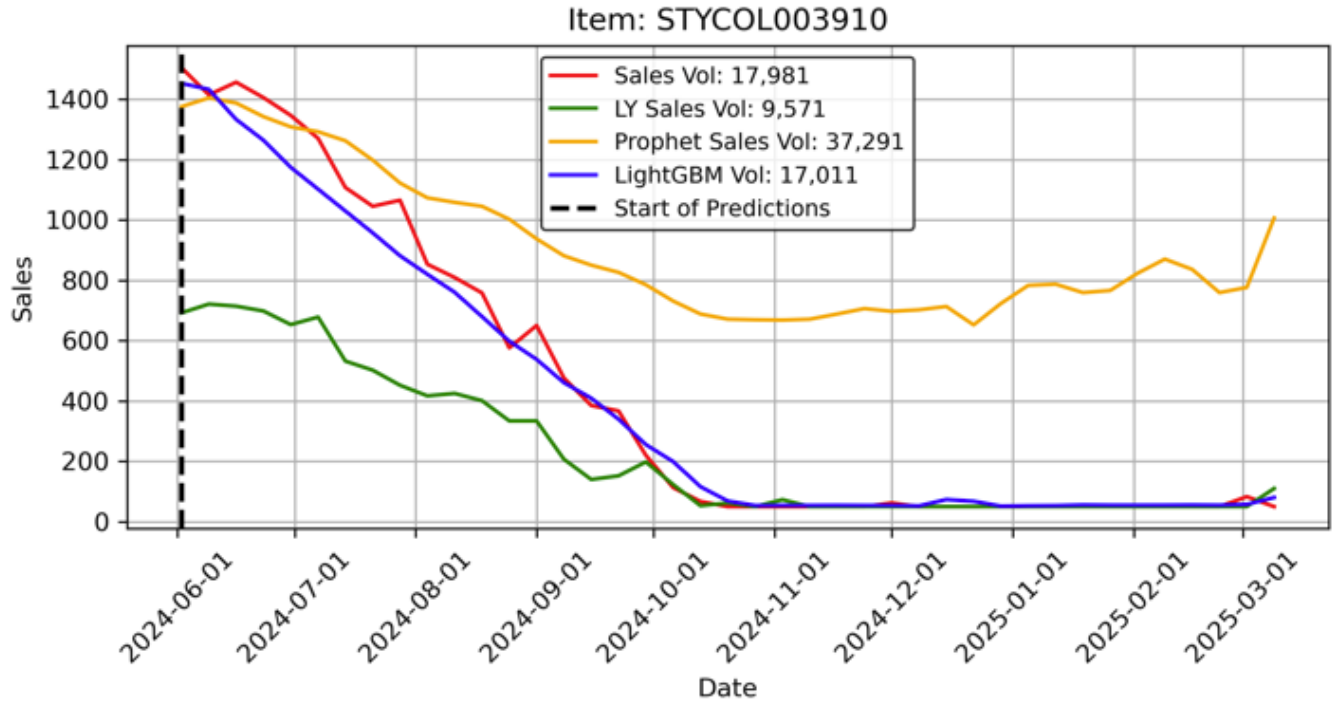
Note: Predicting week 1 is straightforward since lag and rolling features can be calculated from actuals. However, predicting week 2 (and beyond) requires recursive predictions, as prior values are themselves predictions. This restricts batch forecasting. Prophet does not support this iterative approach, which limited its accuracy.

Prediction Visualizations

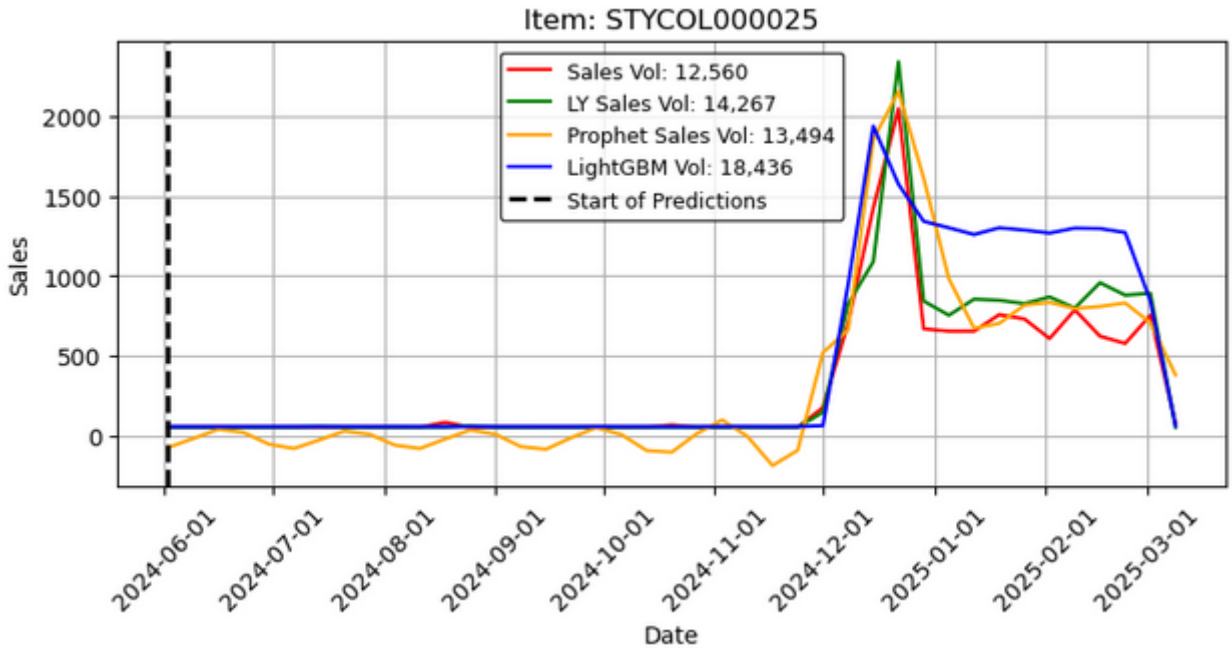
The following plots compare predictions to actual sales. Closer alignment with the red line indicates better predictions.

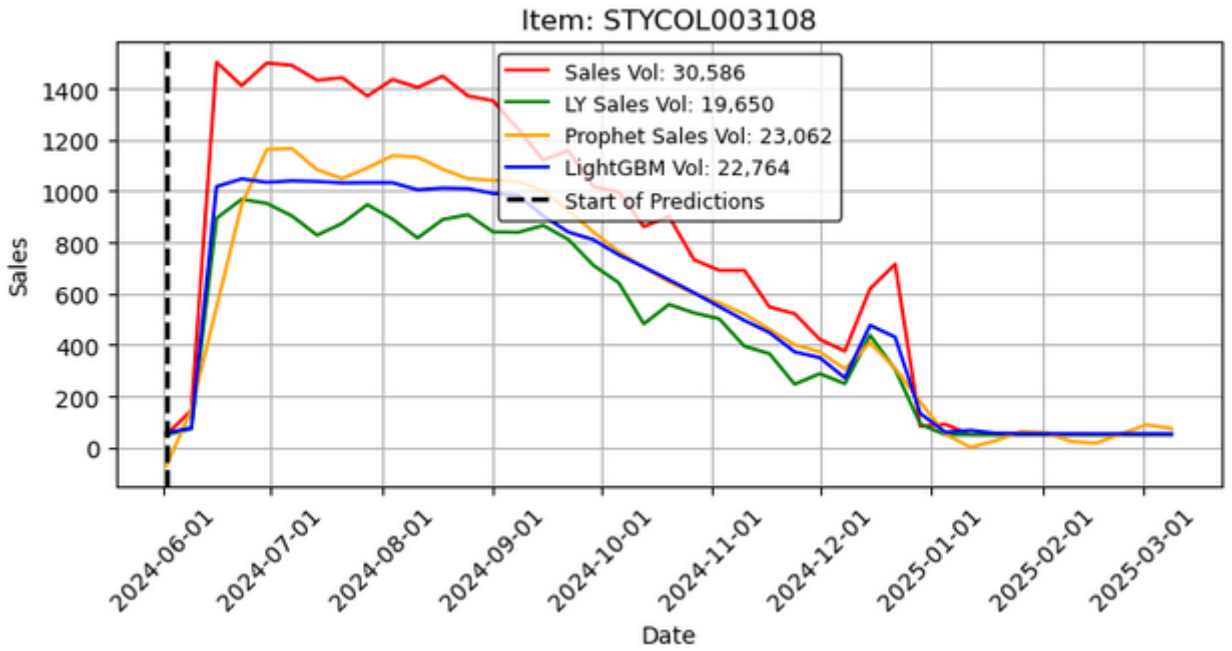
LightGBM Example (Outperforms Prophet and LY)



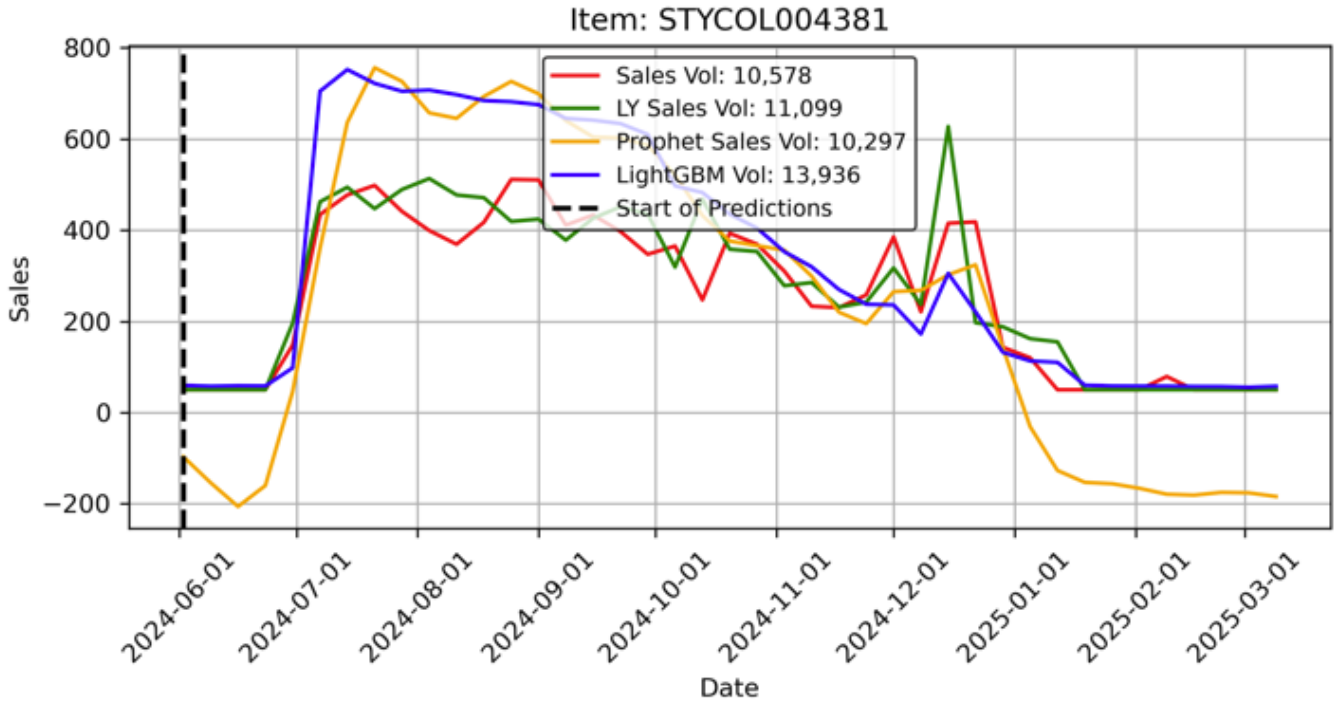


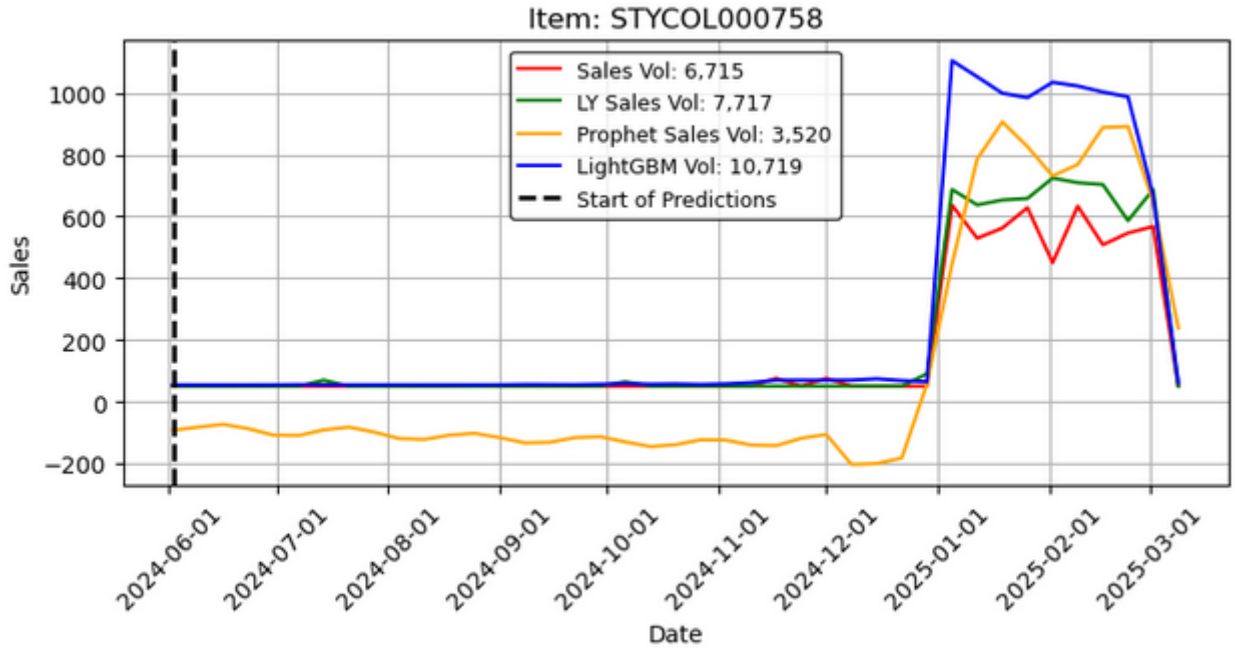
Prophet Example (Outperforms LightGBM and LY)





LY Example (Outperforms LightGBM and Prophet)





Analysis

The framework's ability to adapt to shifting sales trends can lead to more accurate stock levels, minimizing overstock and stockouts. For example, LightGBM's 12% variance from actual sales suggests potential for significant cost savings compared to the 20% variance of last-year methods. Incorporating external data sources like promotions and social media can further enhance ROI.

LightGBM benefited from the use of lag and rolling features. For example, Lag 1 (last week's sales) is a strong predictor of this week's sales. This allowed LightGBM to follow the general curve of last year's data while adapting to short-term volume shifts. Prophet applied a trend on last year's curve but lacked dynamic responsiveness.

Among 5,080 items:

- **LightGBM** was most accurate for **3,208 items**

- **Last Year (LY)** method performed best for **1,026 items**
- **Prophet** performed best for **846 items**

LightGBM correctly forecasted 63% of the time.

Net Absolute Differences in Total Units Predicted

This table shows the net variance of units predicted vs actual sales

Model	Total Sales	Abs Diff to Actual*	% Variance
Actual Sales	96,967,751	-	-
Last Year	96,303,959	19,798,824	20%
LightGBM Prediction	95,283,155	11,828,286	12%
Prophet Prediction	96,064,301	21,752,598	22%

* Absolute difference from actual sales totals

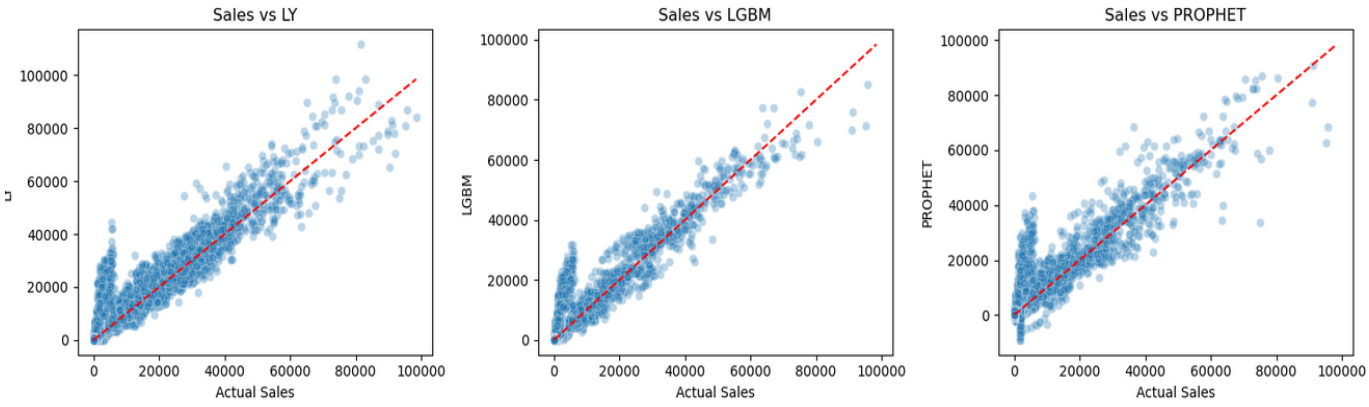
Evaluation Metrics

This table compares the forecasting accuracy of three models—Last Year (LY), Prophet, and LightGBM against actual sales using RMSE, MAE, and MAPE. While LY had the lowest RMSE and MAPE, LightGBM achieved the best MAE, indicating more consistent day-to-day accuracy despite some higher variance in larger errors.

Model	RMSE	MAE	MAPE
LY vs Sales	661.51	211.76	150.61
Prophet vs Sales	862.66	295.17	323.71
LightGBM vs Sales	705.42	199.17	191.49

Prediction Scatter Plots

Below is are the plots of predictions vs actual sales. The closer to the red line (actual sales) the better:



☑ Why LightGBM was the Best Overall Choice

Factor	Verdict	Reason
Visual fit	☑ Best	Closest match to actual sales with minimal systemic bias
Consistency	☑ Strong	Performs well across high and low volume items
Error robustness	⚠ Not the best RMSE/MAPE	Slightly worse RMSE/MAPE likely due to outliers
Scalability & adaptability	☑ High	Easily expandable with more features or data sources
Long-term potential	☑ Best	Learns and improves over time unlike static benchmarks

🧠 Important Considerations

- **LY** performs well on RMSE and MAPE due to its stability, not intelligence. It reflects the **past**, not the **future**.
- **LightGBM** may incur complexity penalties, but benefits from **hyperparameter tuning** and **feature engineering**.
- In retail, where behavior and trends shift frequently, LightGBM's ability to learn and adapt gives it a strategic advantage.

Operationalization & Next Steps

To deploy this framework effectively:

- **Automate** data refresh and model retraining weekly
- **Expand** feature set with external metadata (promotions, weather, social trends)
- **Integrate forecasts** into existing planning dashboards
- **Train** planning teams on interpreting AI-driven insights
- **Continuously monitor** model performance and update as needed
- **Scale** framework to additional product lines and business units

By implementing these steps, retailers can shift from rigid, last-year-based planning to an agile, data-driven in-season and pre-season approach—without requiring an expensive system overhaul.

Results Conclusion

LightGBM outperformed LY and Prophet largely due to its ability to leverage engineered features, especially lag variables like "Lag 1". Prophet lacked access to those features and thus could not adjust as easily.

With more feature-rich inputs, performance could improve further. Examples include:

- Social Media Metrics
- Weather Data
- News and Events

- Customer Feedback
- Search Trends
- Demographics & Psychographics
- Location-Based Data
- Product Popularity Indicators
- Influencer Campaign Data
- Emerging Trends
- Behavioral Analytics

Retailers not capturing these data signals are missing a critical opportunity to improve forecasting and stay competitive. Incorporating external signals into merchandise planning enables smarter, faster, and more responsive decision-making.

This pilot proves that with smart features and appropriate models, both **pre-season** and **in-season** forecasting can be integrated into the planning process—giving teams a valuable new tool. The opportunity grows even more powerful when enriched with external data like weather, social sentiment, and domain-specific insights.