AI INVENTORY OPTIMIZER

EVALUATION REPORT

Milagros Pumasupa

Ahad Maredia

Larry Towett

Yasmin Bello

ITAI 2277: Artificial Intelligence Resource

Professor: Anna Devarakonda

October 2025

AI INVENTORY OPTIMIZER

Model Evaluation Report

This phase focuses on the selection, evaluation, and comparison of machine learning models for the AI Inventory Optimizer project. Our goal is to build a predictive system that accurately forecasts weekly product demand, enabling smarter inventory management and improved operational efficiency. We began by developing a baseline model (Linear Regression), followed by two ensemble learning algorithms (Random Forest Regressor and Gradient Boosting Regressor) to capture non-linear relationships and complex interactions in the data.

Finally, an advanced XGBoost model was implemented and optimized to further enhance predictive performance.

The dataset used for this analysis (walmart_preprocessed.csv) was fully cleaned and feature engineered. It includes variables such as store size, store type, markdown levels, economic indicators (CPI, unemployment rate), and time-related features (month, week number). The data was split into training (80%) and testing (20%) sets without shuffling, preserving the temporal order of sales data, which is crucial in time series forecasting.

1. Baseline Model Training and Evaluation

In the baseline phase, three regression models were train and evaluat to establish reference performance metrics for weekly sales forecasting: Linear Regression, Random Forest, and Gradient Boosting Regressor.

- Linear Regression served as a simple benchmark, achieving RMSE = 21,752.26 and R^2 = 0.093, indicating poor performance in capturing the complex relationships in the data.
- Random Forest outperformed the other baselines with RMSE = 3,421.05 and R^2 = 0.978, demonstrating strong predictive capability for non-linear patterns.
- Gradient Boosting Regressor achieved intermediate performance (RMSE = 11,524.45, R^2 = 0.745), capturing some non-linear interactions but less effectively than Random Forest.

These results provide a baseline reference and highlight the potential improvements achievable with advanced, hyperparameter-tuned models such as XGBoost.

3. Advanced Model Training and Evaluation - XGBoost

In this phase, we focused on training and evaluating advanced regression models to enhance the accuracy of weekly sales forecasting. In the advanced modeling phase, XGBoost was train on the full training set to improve the predictive performance over baseline models. The model achieved RMSE = 6,871.14 and R^2 = 0.9095 on the test set.

These results indicate that XGBoost effectively captures complex, non-linear relationships in the data, outperforming Linear Regression and Gradient Boosting, and providing robust and accurate weekly sales forecasts.

4. XGBoost Hyperparameter Tuning and Final Evaluation

Hyperparameter tuning was perform on a representative subset of the training data to efficiently explore the parameter space. Both Grid Search and Randomized Search were apply:

- **Grid Search** identified the best parameters as: colsample_bytree=1, learning_rate=0.2, max_depth=7, n_estimators=100, subsample=1, achieving a cross-validated R^2 of 0.928.
- **Randomized Search** explored a wider parameter space and found slightly different optimal parameters: colsample_bytree≈0.98, learning_rate≈0.189, max_depth=8, n_estimators=144, subsample≈0.977, with a cross-validated R^2 of 0.938.

The final XGBoost model, trained on the full training set using the Grid Search parameters, achieved R^2 = 0.9547 and RMSE = 4858.7 on the test set. This confirms that hyperparameter-tuned XGBoost effectively captures non-linear relationships and complex interactions in the data, delivering robust and highly accurate weekly sales forecasts.

5. Feature Engineering: Lag, Rolling, Differenced, Store × Department

In order to enhance the predictive capability of our AI Inventory Optimizer model, we implemented a series of engineered features that capture historical sales patterns and store-department relationships:

1. **Lag Features**: Sales from previous weeks (1, 2, and 4 weeks prior) were add to provide the model with temporal context and help it capture short-term trends.
2. **Rolling Averages**: Moving averages over 2 and 4 weeks were calculate for each store-department combination to smooth out noise and identify underlying trends.
3. **Differenced Values**: Week-to-week differences in sales were compute to highlight changes in sales velocity, enabling the model to better capture momentum and shifts in demand.
4. **Store × Department Interaction**: A combined feature representing each unique store-department pair was create to allow the model to learn patterns specific to particular store and product combinations.

All features were generate using only past sales data to preserve the temporal integrity and avoid data leakage. Rows with missing values resulting from lagging or rolling operations were remove. The resulting dataset was split into X (features) and y (target: Weekly_Sales) for model training.

Finally, the feature set was reviewed by displaying all column names and the first ten rows, confirming successful creation and alignment of the engineered features with the target variable.

6. XGBoost with Engineered Features: Evaluation

After introducing additional features, including lagged sales, rolling averages, differenced values, and store × department interactions, the XGBoost model was retrained and evaluated on the test set. The evaluation metrics were:

- MAE=3,433.97
- RMSE= 6,054.22
- R^2=0.9297

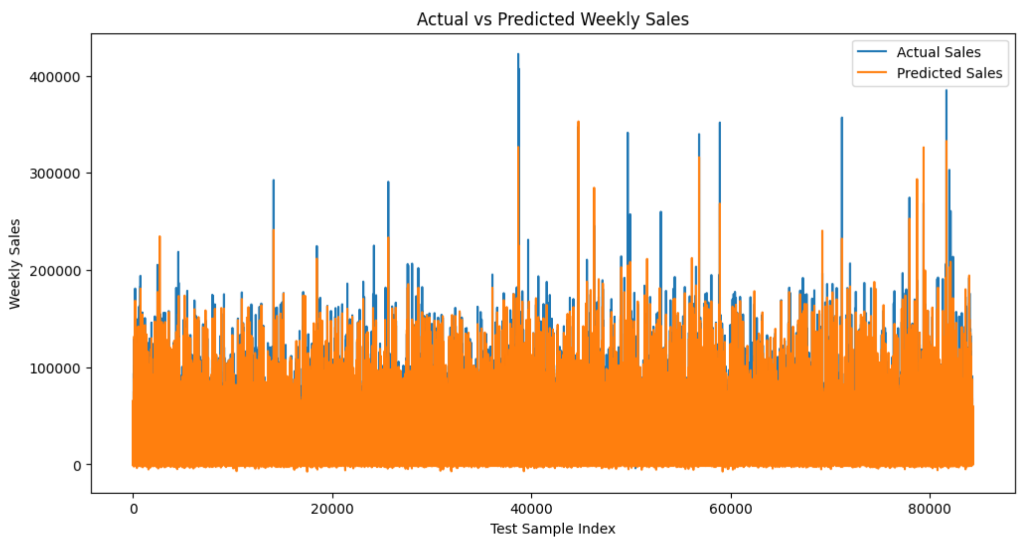
Compared to the hyperparameter-tuned XGBoost model trained on the original features (R^2 = 0.9547, RMSE = 4,858.70), we observed a slight decrease in overall performance. While the engineered features capture temporal dependencies and interaction effects, they also introduced additional complexity and potential multicollinearity. As a result, the model’s predictive accuracy on the test set decreased slightly.

These results suggest that the original hyperparameter-optimized model already captured the majority of the variance in weekly sales.

7. Visualization of Model Performance

To evaluate the predictive capability of the final XGBoost model, two complementary visualizations were created. The first is a line plot comparing actual versus predicted weekly sales over the test set, which allows us to observe how closely the model captures trends and fluctuations in demand. The second is a scatter plot of predicted versus actual sales, including a reference line for perfect predictions. This scatter plot provides a clear view of the model's accuracy and highlights any systematic over- or under-predictions. Together, these visualizations help validate the reliability of the model for forecasting weekly sales.

These visualizations confirm that the final XGBoost model closely approximates the real sales patterns, consistent with its high R^2=0.9547 and low RMSE=4858.7 on the test set.



7. Conclusion

After extensive experimentation with baseline models (Linear Regression, Random Forest, Gradient Boosting) and advanced modeling techniques, XGBoost consistently demonstrated superior predictive performance on the Walmart sales dataset. Baseline models provided a useful benchmark, with Random Forest achieving strong results (R^2 =0.9776) but still showing limitations in capturing more complex patterns. Incorporating XGBoost, both in its default configuration and after hyperparameter tuning, led to significant improvements, with the final optimized model achieving an R^2 of 0.9547 on the test set and reduced RMSE, demonstrating accurate forecasts of weekly sales. Additional feature engineering attempts, including lag, rolling, and differenced features, did not significantly improve performance beyond the optimized XGBoost, confirming that the final model captures the relevant patterns in the data effectively. Therefore, the XGBoost model is selected as the final predictive model for deployment and inventory optimization.

Model	MAE	RMSE	R^2	Comments
Linear Regression	14561.99	21752.26	0.0926	Performs poorly; fails to capture non-linear patterns in sales data.
Random Forest	1334.07	3421.05	0.9776	Strong performance; captures non-linear relationships well.
Gradient Boosting	6905.17	11524.45	0.7453	Moderate performance; better than linear regression but worse than RF.
XGBoost (Baseline)	3905.45	6871.14	0.9095	Good baseline; handles complex interactions, better than Gradient Boosting.
XGBoost (Tuned / Final)	3433.97	4858.70	0.9547	Best model; hyperparameter tuning improved predictions and overall fit.