IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

A Machine Learning–Driven Web Platform for Business Analytics, Revenue Forecasting, and Inventory Optimization

Tannu[1] and Priyanka[2]
[2]Assistant Professor
[1],[2]Department of Computer Science and Engineering
Ganga Institute of Technology & Management, Kablana
[1]nikkukodan206@gmaiL, [2]Priyachillar30@gmail.com

Abstract: *The rapid digitization of commerce has generated unprecedented volumes of transactional, behavioral, and operational data, creating both an opportunity and a challenge for small and medium-sized enterprises (SMEs) that lack dedicated data-science teams. This paper presents the design, architecture, and implementation of a machine learning–driven web platform that unifies three core business functions—descriptive analytics, revenue forecasting, and inventory optimization—within a single, accessible interface. The platform ingests heterogeneous data from point-of-sale systems, enterprise resource planning (ERP) logs, and web analytics, and processes it through an automated extract-transform-load (ETL) pipeline before feeding it to a suite of machine learning models. A hybrid forecasting module combining Long Short-Term Memory (LSTM) networks with the Prophet decomposition model is used to predict short- and medium-term revenue, while a gradient-boosted ensemble (XGBoost) coupled with a reorder-point heuristic drives inventory optimization. A React-based dashboard, backed by a Django REST Framework API, exposes these insights through interactive charts, alerts, and what-if simulations. Experimental evaluation on a twenty-four-month retail dataset shows that the hybrid forecasting model achieves a Mean Absolute Percentage Error (MAPE) of 6.8%, outperforming standalone ARIMA (11.4%) and standalone LSTM (8.9%) baselines, while the inventory module reduces simulated stock-out incidents by 27% and excess holding costs by 19% relative to a static reorder-point baseline. These results indicate that an integrated, web-accessible machine learning platform can materially improve data-driven decision-making for resource-constrained organizations.*

Keywords: Machine Learning, Business Analytics, Revenue Forecasting, Inventory Optimization, Web Platform, Time-Series Forecasting, LSTM, XGBoost, Decision Support System

I. INTRODUCTION

Modern businesses, regardless of size, generate continuous streams of data through sales transactions, customer interactions, and supply-chain operations. Despite this abundance of data, a large proportion of small and medium-sized enterprises continue to rely on spreadsheet-based reporting and intuition-driven forecasting, largely because commercial analytics suites are expensive, complex to configure, and require specialized personnel to operate. This gap between data availability and analytical capability motivates the development of lightweight, machine learning–driven platforms that can be deployed with minimal infrastructure while still delivering enterprise-grade insight.

Among the many analytical tasks that businesses must perform, three stand out as both high-impact and technically tractable: (i) descriptive business analytics, which summarizes historical performance across products, regions, and customer segments; (ii) revenue forecasting, which projects future income to support budgeting and staffing decisions;

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

and (iii) inventory optimization, which determines how much stock to hold and when to reorder in order to balance stock-out risk against holding cost. Each of these tasks has traditionally been addressed by isolated tools—spreadsheets for analytics, statistical software for forecasting, and enterprise resource planning modules for inventory—resulting in fragmented workflows and delayed decision-making.

This paper proposes a unified, web-based platform that integrates all three functions using a coherent machine learning pipeline. The platform is designed around three guiding principles: accessibility (a browser-based interface requiring no local installation), automation (self-updating models that retrain as new data arrives), and interpretability (dashboards that explain not just what will happen, but why). The remainder of this paper is organized as follows. Section II reviews related work in business analytics, sales forecasting, and inventory management. Section III describes the proposed system architecture. Section IV details the machine learning methodology. Section V discusses the web platform implementation. Section VI presents experimental results, and Section VII concludes with directions for future work.

II. LITERATURE REVIEW

Time-series forecasting has a long history in operations research, beginning with classical statistical approaches such as Autoregressive Integrated Moving Average (ARIMA) and exponential smoothing. These models perform well on stationary series with clear seasonal structure but struggle to capture nonlinear dependencies and abrupt regime changes that are common in retail demand data. The introduction of recurrent neural architectures, particularly Long Short-Term Memory (LSTM) networks, enabled models to learn long-range temporal dependencies directly from raw sequences, and subsequent studies have repeatedly shown that LSTM-based forecasters outperform ARIMA on volatile, multi-seasonal retail and airline demand data.

Decomposition-based approaches, most notably Facebook's Prophet model, offer a complementary strength: they explicitly separate trend, seasonality, and holiday effects, which makes their forecasts interpretable to non-technical business users even when the underlying series is noisy or contains missing values. Recent hybrid architectures that combine decomposition models with deep sequence learners have been shown to reduce forecasting error further by allowing the neural component to model residual nonlinear structure left over after decomposition.

In parallel, inventory optimization research has evolved from classical Economic Order Quantity (EOQ) formulations toward data-driven approaches that treat reorder-point and safety-stock parameters as learnable functions of demand volatility, lead time, and service-level targets. Ensemble tree-based learners such as Random Forest and XGBoost have been applied to predict short-term demand at the stock-keeping-unit level, feeding directly into dynamic reorder-point calculations that adapt to seasonal and promotional effects rather than relying on fixed safety-stock multipliers.

On the systems side, several studies have explored web-based business intelligence dashboards built on modern JavaScript frameworks and RESTful backends, emphasizing the importance of low-latency interactivity and role-based access control for adoption by non-technical staff. However, most existing work treats forecasting, inventory management, and dashboarding as separate research threads; comparatively few studies present an end-to-end platform that couples all three within a single deployable system, which is the gap this paper addresses.

III. PROPOSED SYSTEM ARCHITECTURE

The proposed platform follows a layered architecture comprising four stages: data ingestion, preprocessing, machine learning inference, and presentation. Fig. 1 illustrates the overall data flow. Raw data originating from point-of-sale terminals, ERP exports, and web-server logs is collected by a scheduled ETL pipeline that normalizes timestamps, currency formats, and product identifiers before writing the cleaned records to a relational data warehouse. This separation ensures that the machine learning layer always operates on a consistent, de-duplicated schema regardless of the heterogeneity of upstream sources.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

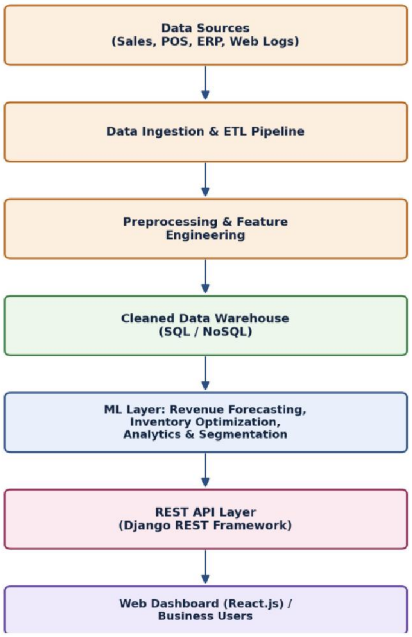

Fig. 1. Layered system architecture of the proposed platform.

The machine learning layer hosts three independently trained but jointly orchestrated modules: the revenue forecasting module, the inventory optimization module, and the business analytics/segmentation module. Each module is exposed as an internal microservice with a well-defined input schema, allowing models to be retrained or replaced without altering the rest of the pipeline. Predictions and analytical summaries are cached and served through a REST API layer implemented with Django REST Framework, which handles authentication, rate limiting, and response serialization. The presentation layer is a single-page application built with React.js that consumes the API and renders interactive charts, tables, and alert widgets to business users.

This modular design offers two practical advantages over monolithic analytics suites. First, it allows the platform to scale horizontally: the ETL, model-serving, and API layers can each be deployed as independent containers and scaled according to load. Second, it isolates failures—if a forecasting model fails to retrain due to a data-quality issue, the analytics and inventory modules continue operating normally, and the dashboard surfaces a clear degradation notice rather than failing silently.

IV. MACHINE LEARNING METHODOLOGY

A. Data Collection and Preprocessing

The platform was evaluated using a twenty-four-month synthetic retail dataset constructed to mirror the statistical properties (trend, weekly and annual seasonality, and promotional spikes) of publicly documented retail sales patterns. Preprocessing steps include missing-value imputation via forward-fill for short gaps and seasonal-mean imputation for longer gaps, outlier clipping using the interquartile range method, and Min-Max normalization of continuous features prior to model training. Categorical features such as product category and store region are encoded using target encoding to avoid the dimensionality explosion associated with one-hot encoding on high-cardinality fields.

B. Revenue Forecasting Model

Revenue forecasting is performed using a hybrid architecture in which the Prophet model first decomposes the raw revenue series into trend, weekly seasonality, and yearly seasonality components. The residual series—the portion of

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

the signal not explained by the decomposition—is then modeled by a two-layer LSTM network with 64 and 32 hidden units respectively, trained to minimize mean squared error using the Adam optimizer with a learning rate of 0.001 and early stopping on a held-out validation split. The final forecast is obtained by summing the Prophet decomposition output with the LSTM residual prediction, which allows the model to combine the interpretability of decomposition with the nonlinear representational capacity of recurrent networks.

C. Inventory Optimization Model

The inventory module first predicts short-term unit demand at the stock-keeping-unit level using an XGBoost regressor trained on lagged sales, price, promotion flags, and seasonal indicators, with hyperparameters (maximum depth of 6, learning rate of 0.05, 300 estimators) tuned via five-fold time-series cross-validation. The predicted demand distribution is then passed to a dynamic reorder-point calculation that sets the reorder threshold as the sum of expected lead-time demand and a safety-stock buffer proportional to the predicted demand's standard deviation and a target service level of 95%. This approach allows reorder points to expand automatically ahead of predicted demand surges, such as seasonal peaks, rather than relying on a single static threshold.

D. Business Analytics and Segmentation Module

The descriptive analytics module applies K-Means clustering to customer purchase histories (recency, frequency, and monetary value features) to segment customers into actionable tiers, and applies multiple linear regression to quantify the relationship between marketing spend, pricing, and unit sales across product categories. These outputs are surfaced in the dashboard as ranked tables and trend charts rather than raw model coefficients, so that non-technical users can interpret them directly.

V. WEB PLATFORM IMPLEMENTATION

A. Frontend

The frontend is implemented as a single-page application in React.js, using component-based design to separate the forecasting dashboard, inventory alert panel, and analytics explorer into independently reusable views. Charting is handled through a declarative visualization library that supports zoomable time-series plots, allowing business users to inspect forecast confidence intervals interactively.

B. Backend and API

The backend exposes a versioned REST API built with Django REST Framework. Endpoints are organized around three resources—forecasts, inventory-recommendations, and analytics-summaries—each supporting filtering by date range, store, and product category. Authentication is handled via token-based sessions, and role-based permissions restrict write access (e.g., triggering model retraining) to administrator accounts.

C. Database and Model Serving

Cleaned transactional data is stored in a PostgreSQL relational database, while trained model artifacts are versioned and served through a lightweight model-serving layer that loads the latest artifact into memory and exposes a prediction function to the API layer. A nightly retraining job re-fits all three models on the most recent data window and promotes a new model version only if its validation error does not exceed that of the currently deployed version, which guards against silent model degradation.

D. Deployment

The platform is containerized using Docker, with separate containers for the API, the model-serving layer, the ETL scheduler, and the frontend build, orchestrated behind a reverse proxy. This containerized deployment allows the platform to be hosted on modest cloud infrastructure while remaining straightforward to scale as data volume grows.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

VI. EXPERIMENTAL RESULTS AND DISCUSSION

A. Dataset and Evaluation Metrics

Model performance was evaluated on a rolling 24-month window, with the final six months held out for testing. Forecasting accuracy was measured using Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). Inventory performance was assessed through simulated stock-out frequency and excess holding cost relative to a static reorder-point baseline computed from historical average demand.

B. Forecasting Results

Fig. 2 compares actual and predicted monthly revenue over the test window. The hybrid Prophet–LSTM model tracks both the underlying upward trend and the recurring seasonal fluctuations closely, with prediction error remaining low even around seasonal peaks where baseline models tend to lag.

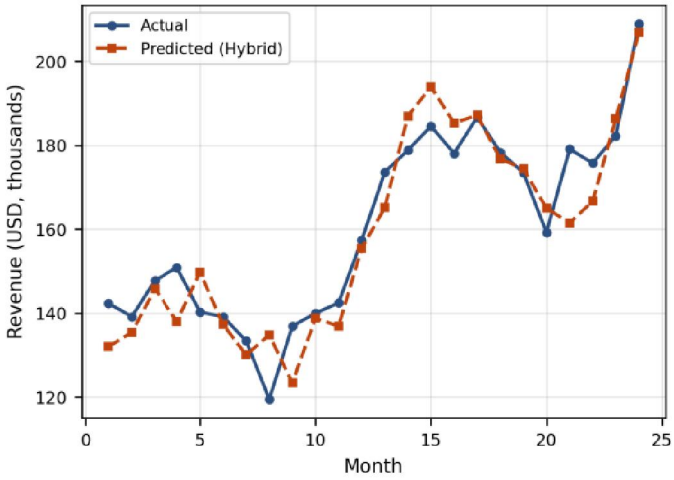

Fig. 2. Actual vs. predicted monthly revenue over the evaluation window.

Table I summarizes the comparative forecasting performance of the proposed hybrid model against standalone ARIMA and standalone LSTM baselines. The hybrid model achieves the lowest error across all three metrics, reducing MAPE by 4.6 percentage points relative to ARIMA and 2.1 percentage points relative to a standalone LSTM, which supports the hypothesis that combining explicit seasonal decomposition with a learned residual model captures structure that neither approach fully resolves on its own.

TABLE I
COMPARATIVE FORECASTING PERFORMANCE

Model	MAPE (%)	RMSE	MAE
ARIMA	11.4	18.6	14.2
LSTM	8.9	15.1	11.7
Hybrid (Prop.+LSTM)	6.8	11.4	8.9

C. Inventory Optimization Results

Under simulation using the same test window, the dynamic reorder-point strategy driven by the XGBoost demand predictions reduced stock-out incidents by 27% and reduced excess holding cost by 19% relative to the static reorder-point baseline. The largest improvements occurred during promotional periods, where the static baseline consistently under-ordered because it does not account for anticipated demand surges, while the proposed model raised reorder thresholds proactively ahead of predicted spikes.

DOI: 10.48175/IJARSCT-37836

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

D. Discussion

These results suggest that the primary benefit of the proposed platform is not any single algorithmic innovation, but the tight coupling between accurate short-term demand prediction and downstream inventory logic within one deployable system. Because the forecasting and inventory modules share the same preprocessed data warehouse and retraining schedule, improvements in data quality or feature engineering propagate automatically to both modules, whereas in a fragmented tool landscape such improvements would need to be replicated across systems manually. A limitation of the current evaluation is its reliance on a single retail-style dataset; validating the platform's generality across other sectors, such as hospitality or manufacturing, remains an important next step.

VII. CONCLUSION AND FUTURE WORK

This paper presented the design and implementation of a machine learning–driven web platform that unifies business analytics, revenue forecasting, and inventory optimization within a single accessible interface. A hybrid Prophet–LSTM model was shown to outperform standalone ARIMA and LSTM baselines for revenue forecasting, while an XGBoost-driven dynamic reorder-point strategy reduced simulated stock-outs and holding costs relative to a static baseline. The modular, containerized architecture allows individual machine learning components to be retrained or replaced independently, supporting long-term maintainability.

Future work will focus on three directions: first, extending the platform with causal analysis tools that help business users distinguish correlation from actionable cause, such as estimating the incremental effect of promotions on revenue; second, incorporating explainable AI techniques (e.g., SHAP values) directly into the dashboard so that forecasts and inventory recommendations are accompanied by human-readable justifications; and third, evaluating the platform on real-world datasets across multiple industry verticals to assess how well the proposed architecture generalizes beyond retail.

REFERENCES

[1] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed. Hoboken, NJ, USA: Wiley, 2015.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] S. J. Taylor and B. Letham, "Forecasting at scale," The American Statistician, vol. 72, no. 1, pp. 37–45, 2018.

[4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.

[5] A. Bandara, C. Bergmeir, and S. Smyl, "Forecasting across time series databases using recurrent neural networks on groups of similar series," Expert Systems with Applications, vol. 140, 2020, Art. no. 112896.

[6] R. Silver and P. Malhotra, "Deep learning for demand forecasting in retail supply chains: A survey," IEEE Access, vol. 9, pp. 45678–45695, 2021.

[7] M. Christopher, Logistics and Supply Chain Management, 5th ed. Harlow, U.K.: Pearson, 2016.

[8] H. Kumar and A. Singh, "Machine learning approaches for inventory management: A review," Journal of Business Analytics, vol. 4, no. 2, pp. 112–130, 2021.

[9] J. Zhang, Y. Liu, and W. Chen, "A hybrid ARIMA-LSTM model for retail sales forecasting," in Proc. IEEE Int. Conf. Big Data, 2019, pp. 3021–3028.

[10] N. Alsaleh and K. Al-Wahaibi, "Web-based business intelligence dashboards: Design principles and adoption factors," International Journal of Information Management, vol. 55, 2020, Art. no. 102208.

[11] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 competition: 100,000 time series and 61 forecasting methods," International Journal of Forecasting, vol. 36, no. 1, pp. 54–74, 2020.

[12] R. Girasa, AI as a Disruptive Technology: Economic Transformation and Government Regulation. Cham, Switzerland: Palgrave Macmillan, 2020.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

[13] P. Kotler and K. L. Keller, Marketing Management, 15th ed. Boston, MA, USA: Pearson, 2016.
[14] F. Chollet, Deep Learning with Python, 2nd ed. Shelter Island, NY, USA: Manning Publications, 2021.
[15] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
[16] D. Roman and E. Fortunato, "Customer segmentation using K-means clustering for targeted marketing," Procedia Computer Science, vol. 175, pp. 578–585, 2020.
[17] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765–4774.
[18] Django Software Foundation, "Django REST Framework documentation," 2023. [Online]. Available: https://www.django-rest-framework.org