iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85367

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

Time Series Driven E-commerce Sales Forecasting and Intelligent Recommendation System

Bhavani S[1], Deepthi V T[2], Nanditha B[3], Mrs. Kaveri[4]
[1, 2, 3]*Department of CSE, Jyothy Institute of Technology, Bengaluru, India*
[4]*Assistant Professor, Department of CSE, Jyothy Institute of Technology, Bengaluru, India*

Abstract: ***The rapid growth of e-commerce has increased the importance of accurate sales forecasting and personalized product recommendations in data-driven business applications. This paper presents Time Series Based E-commerce Sales Forecasting and Intelligent Recommendation System, a web-based application that integrates machine learning-based sales forecasting with intelligent product recommendation techniques. The forecasting component uses the XGBoost Regressor to predict future daily sales based on historical sales data and engineered temporal features, including lag values, rolling statistics, and calendar-based attributes. The recommendation component implements three independent approaches: K-Nearest Neighbors (KNN)-based collaborative filtering for frequently bought products, content-based filtering using TF-IDF and cosine similarity for identifying similar products, and popularity-based ranking for recommending top-selling products within a selected department. The application is developed using React.js for the frontend and FastAPI for the backend, providing an interactive interface for accessing forecasts and generating product recommendations. MongoDB Atlas is used exclusively for user authentication data, while forecasting and recommendation datasets and model artifacts are maintained separately. The system combines predictive analytics and recommendation techniques within a single web application, providing users with an integrated platform for exploring future sales trends and product suggestions.***

Keywords: ***E-commerce, Time Series Forecasting, XGBoost Regressor, Product Recommendation System, Collaborative Filtering, K-Nearest Neighbors, Content-Based Filtering, TF-IDF, Cosine Similarity, FastAPI, React.js.***

I. INTRODUCTION

The rapid expansion of electronic commerce has transformed the way businesses operate and how customers discover and purchase products. E-commerce platforms generate large volumes of transactional data containing information about orders, products, purchasing patterns, and sales over time. Analyzing this data can help businesses understand historical performance, anticipate future demand, and provide relevant product suggestions to customers. However, extracting meaningful insights from large datasets requires effective forecasting techniques and recommendation methods. Sales forecasting is an important component of data-driven decision-making in e-commerce. By analyzing historical sales patterns, forecasting models can estimate future sales and help businesses understand potential changes in demand. Traditional approaches based primarily on historical averages may not adequately represent complex relationships and variations in sales data. Machine learning techniques offer an alternative by learning relationships between historical observations and relevant features. In this project, the XGBoost Regressor is used to forecast future daily sales using historical sales data and engineered features, including lag values, rolling statistics, and calendar-based attributes. Product recommendation is another important aspect of e-commerce applications. Customers may encounter difficulty identifying relevant products when a platform offers a large product catalog. Recommendation systems address this challenge by suggesting products based on purchasing patterns, product characteristics, or popularity. Different recommendation techniques serve different purposes: collaborative filtering identifies relationships in user–product interactions, content-based filtering identifies products with similar characteristics, and popularity-based methods rank products according to observed purchasing activity. This paper presents Time Series Based E-commerce Sales Forecasting and Intelligent Recommendation System, a web-based application that integrates sales forecasting and product recommendation capabilities within a single platform. The forecasting component uses the XGBoost Regressor, while the recommendation component provides three independent methods: K-Nearest Neighbors (KNN)-based collaborative filtering for frequently bought products, TF-IDF and cosine similarity for similar-product recommendations, and popularity-based ranking for top-selling products within a selected department. The application is developed using React.js for the frontend and FastAPI for the backend. Its interactive interface enables users to access sales forecasts, select recommendation methods, and view the resulting product suggestions. MongoDB Atlas is used exclusively to manage user authentication data, while the forecasting and recommendation datasets and model artifacts are maintained separately.

©IJRASET: All Rights are Reserved

By integrating predictive analytics and multiple recommendation techniques into one web application, the proposed system provides a unified environment for exploring future sales trends and generating product suggestions. The project demonstrates how machine learning models and an interactive web interface can be combined to support analytical and product-discovery tasks in an e-commerce context.

II. LITERATURE REVIEW

Time-series forecasting has been widely studied using statistical, machine learning, deep learning, and hybrid approaches. Traditional methods such as ARIMA, SARIMA, and Prophet are used to model trends and seasonal patterns, while machine learning techniques such as XGBoost and LightGBM learn relationships from historical data and engineered features. Existing reviews discuss the strengths and limitations of these approaches for demand and time-series forecasting [1], [8], [11], [13]. Further studies investigate hybrid architectures, including XGBoost–LSTM, CNN–LSTM, ARIMA–LSTM, and GRU–LightGBM, to improve demand and sales prediction [2], [4], [6], [10]. Other research explores model comparison, trend and seasonality decomposition, and the influence of external variables on forecasting performance [5], [7], [9], [12].

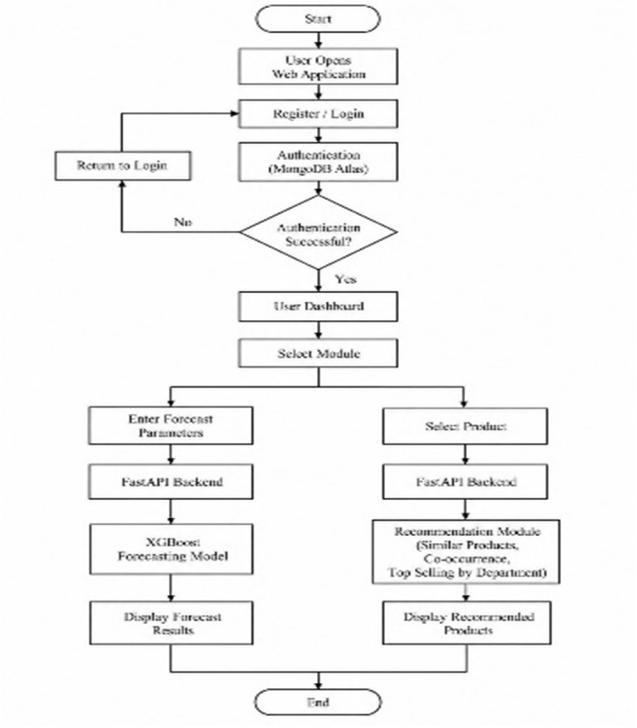
These studies informed the selection of an XGBoost Regressor for the proposed system. The model uses historical sales data and engineered temporal features to forecast daily e-commerce sales. Instead of combining multiple forecasting models, the proposed system uses a single machine learning model integrated into a web-based application.

Product recommendation is addressed as a complementary functionality. The proposed system implements three separate approaches: item-based collaborative filtering using K-nearest neighbours to recommend frequently bought products, content-based filtering using TF-IDF and cosine similarity to identify similar products, and popularity-based ranking to recommend top-selling products within a selected department.

Although the reviewed studies investigate various forecasting techniques and their applications, the literature survey identifies an opportunity to integrate sales forecasting and multiple recommendation approaches within one e-commerce application. The proposed system addresses this opportunity through a React.js frontend, a FastAPI backend, and separate forecasting and recommendation pipelines.

III. PROPOSED METHODOLOGY

The proposed system integrates time-series-based e-commerce sales forecasting with intelligent product recommendations in a web application. The methodology consists of separate data preprocessing pipelines for forecasting and recommendations, model development, recommendation generation, backend integration, and frontend visualization.



A. Data Collection and Preprocessing

Two datasets are used for different purposes. The Brazilian E-Commerce Public Dataset by Olist is used for sales forecasting, while the Instacart dataset is used for product recommendations.

For forecasting, order, order-item, and payment data are processed to generate daily sales records. The data is aggregated by date to derive daily sales and other sales-related attributes. Missing dates are included in the daily time series, and missing numerical values are filled with zeros. Feature engineering generates lag-based variables, rolling statistics, expanding statistics, and calendar-related features to represent historical sales patterns

For product recommendations, the Instacart order, product, aisle, and department data is combined and preprocessed. The resulting dataset provides product details and purchase interactions required by the recommendation techniques.

B. Sales Forecasting

An XGBoost Regressor is used to predict daily e-commerce sales. The forecasting dataset is divided into training and testing sets, containing 534 and 134 records, respectively. The model is trained using 26 input features, with daily sales as the target variable.

The model uses the squared-error regression objective, 500 estimators, a learning rate of 0.05, a maximum tree depth of 6, subsampling of 0.8, and a column-sampling ratio of 0.8. The trained model is saved and used to generate future daily sales predictions. Forecasts are produced recursively, with each predicted value incorporated into the feature-generation process for subsequent days.

Model performance is evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). The resulting predictions and evaluation metrics are made available to the application for analysis and visualization.

C. Intelligent Product Recommendation

The recommendation module provides three independent recommendation techniques.

Frequently Bought Products: Item-based collaborative filtering uses a user–product interaction matrix and a K-nearest neighbours model with cosine distance to identify products associated with similar purchase patterns.

Similar Products**:** Content-based filtering represents product information using Term Frequency–Inverse Document Frequency (TF-IDF). Cosine similarity is then used to identify products with similar textual attributes, including product names, aisles, and departments.

Top-Selling Products by Department**:** A popularity-based method ranks products using a weighted score derived from purchase counts and the number of unique customers. Users can select a department to retrieve its top-ranked products.

These approaches allow recommendations to be generated according to purchase interactions, product similarity, or product popularity.

D. Web Application Integration

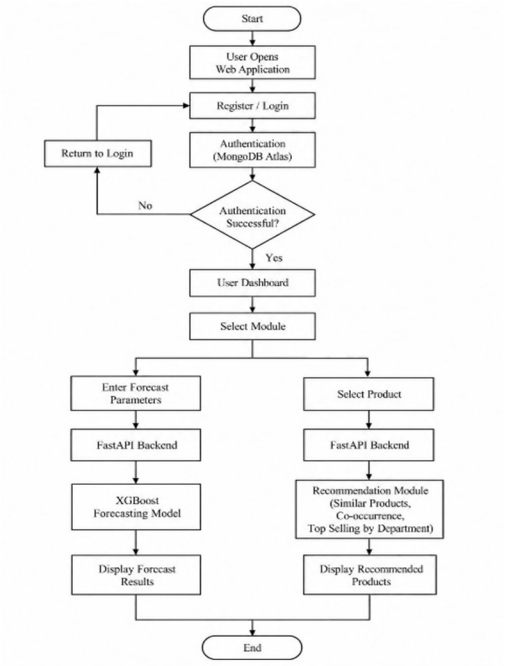
The forecasting and recommendation components are integrated through a FastAPI backend and a React.js frontend. The backend loads the trained models and supporting datasets to process requests and return results to the interface. Users can request sales forecasts, explore forecasting analytics, search for products, select a recommendation type, and view recommended products.

MongoDB Atlas is used exclusively to store user authentication information. Forecasting datasets, recommendation datasets, trained model artifacts, and generated outputs are maintained separately rather than stored in MongoDB Atlas. Plotly and Matplotlib support data visualization within the project.

IV. SYSTEM ARCHITECTURE

The proposed system follows a client–server architecture that integrates sales forecasting, product recommendation, user authentication, and data visualization. The frontend is developed using React.js, HTML, CSS, and JavaScript, while the backend is implemented using FastAPI and Python. Forecasting and recommendation operations are handled by separate processing modules, and MongoDB Atlas is used exclusively for user authentication data.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com



A. Frontend Layer

The frontend provides the user interface for interacting with the system. Users can access the forecasting dashboard, request future sales predictions, view analytics, search for products, and select recommendation methods. Forecasting results and product recommendations are presented through the web interface, with visualizations supported by Plotly and Matplotlib.

B. Backend Layer

The FastAPI backend receives requests from the frontend and routes them to the appropriate service. The forecasting service generates predictions using the trained XGBoost model and the processed sales dataset. The recommendation service loads the required model artifacts and generates results using the selected recommendation technique. The backend returns the processed results to the frontend for display.

C. Sales Forecasting Module

The forecasting module uses the preprocessed Olist sales dataset and the trained XGBoost Regressor. Historical sales features are supplied to the model to generate daily predictions. For multi-day forecasting, predictions are generated recursively and saved to a separate output file. The backend reads the generated forecast and returns the requested results and summary statistics to the frontend.

D. Product Recommendation Module

The recommendation module uses the processed Instacart dataset and separately trained model artifacts. Frequently bought products are generated using item-based collaborative filtering, similar products are identified using TF-IDF and cosine similarity, and top-selling products are ranked using purchase popularity within a selected department. The service returns the recommendations according to the user's selected method and input.

E. Database and Data Storage Layer

MongoDB Atlas stores user authentication information, including user name, email, and password-related account data. The forecasting dataset, recommendation dataset, trained models, prediction outputs, and recommendation artifacts are maintained separately as files. Forecast results and recommendation results are not stored in MongoDB Atlas.

The interaction between these components enables users to access forecasting and recommendation features through a single web application while keeping authentication storage separate from machine learning data and artifacts.

©IJRASET: All Rights are Reserved

V. IMPLEMENTATION

The proposed system is implemented using React.js for the frontend and FastAPI with Python for the backend. It integrates sales forecasting, product recommendation, data visualization, and user authentication through separate modules. The forecasting and recommendation components use independently processed datasets and trained model artifacts.

A. Sales Data Processing and Forecasting Model

The Olist Brazilian e-commerce dataset is processed using order, order-item, and payment information to generate daily sales records. Feature engineering generates lag features for 1, 7, 14, and 28 days, rolling statistics over 7-, 14-, and 28-day windows, expanding statistics, and cyclical calendar features. The resulting dataset is divided into training and testing sets containing 534 and 134 records, respectively.

An XGBoost Regressor is trained using 26 input features, with daily sales as the target variable. The model uses 500 estimators, a learning rate of 0.05, a maximum depth of 6, subsampling of 0.8, and a column-sampling ratio of 0.8. The trained model is saved as a model artifact and used to generate future sales predictions.

B. Forecast Generation and Evaluation

The forecasting module generates daily predictions recursively for a user-specified forecast horizon. Each predicted value is incorporated into the feature-generation process for the following day. The forecasts are saved separately and processed by the backend to calculate summary statistics, including total, average, maximum, and minimum predicted sales.

Model performance is evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination ((R^2)). The evaluation module saves the metrics and test-set predictions for use in the analytics dashboard.

C. Product Recommendation Implementation

The recommendation module uses the preprocessed Instacart dataset and implements three independent recommendation techniques. The frequently bought products method applies item-based collaborative filtering using a user–product interaction matrix and a K-nearest neighbours model with cosine distance. The similar products method uses Term Frequency–Inverse Document Frequency (TF-IDF) to represent product names, aisle information, and department information, followed by cosine similarity. The top-selling products method ranks products using a weighted popularity score based on purchase counts and the number of unique customers, with weights of 0.7 and 0.3, respectively.

The trained recommendation models, product catalogs, interaction matrices, and similarity artifacts are stored separately and loaded by the recommendation service. Users can search for products, select a recommendation method, and specify the number of recommendations to retrieve results through the application.

D. Backend and Frontend Integration

FastAPI provides endpoints for forecast generation, product search, and recommendation requests. The forecasting service executes the prediction process and returns the generated forecasts and summary statistics. The recommendation service loads the required artifacts and invokes the selected recommendation technique before returning the results.

The React.js frontend provides interfaces for sales forecasting, analytics, and product recommendations. Forecasting results and sales analytics are displayed through the dashboard, while the recommendation interface presents products according to the selected method and user input. Plotly and Matplotlib are used for data visualization.

E. Authentication and Data Storage

MongoDB Atlas is used exclusively to store user authentication information, including user name, email, and password-related account data. Forecasting datasets, recommendation datasets, trained models, prediction files, and recommendation artifacts are maintained separately. Forecast results and product recommendations are not stored in MongoDB Atlas.

VI. RESULTS AND DISCUSSION

The proposed system integrates daily e-commerce sales forecasting with three product recommendation techniques through a web-based interface. The forecasting model is evaluated using a held-out test dataset, while the recommendation module provides frequently bought products, similar products, and top-selling products by department.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

A. Sales Forecasting Results

The XGBoost Regressor was trained using 534 records and evaluated using 134 test records with 26 input features. Its performance was assessed using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination ((R^2)).

The evaluation results are presented below :

Evaluation Metric	Obtained Value
Training records	534
Testing records	134
Input features	26
MAE	604.7324
RMSE	1006.7889
MAPE	1.3844
(R^2)	0.9965

The MAE represents the average absolute difference between actual and predicted sales values, while RMSE gives greater weight to larger prediction errors.

MAPE expresses prediction error relative to actual values, and (R^2) indicates how well the model explains the variation in the target values. The obtained (R^2) value of 0.9965 indicates a strong fit on the evaluated test data. These results describe the evaluation performed on the available test dataset and do not guarantee identical performance on future data.

B. Forecast Visualization and Analytics

The application presents forecast results through the forecasting interface and analytics dashboard. Users can request forecasts for a specified period and view summary statistics, including total, average, maximum, and minimum predicted sales. The dashboard also provides sales-related analytics and visual comparisons between actual and predicted values. Plotly and Matplotlib are used for data visualization.

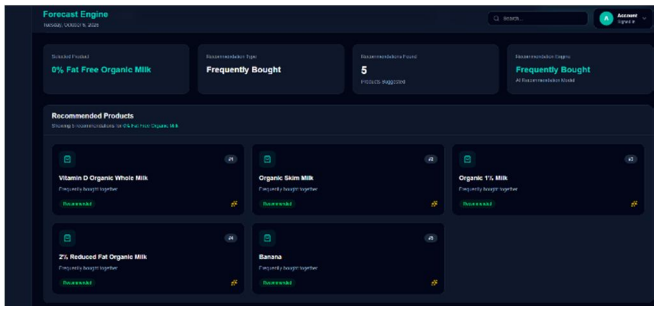


C. Product Recommendation Results

The recommendation module provides three selectable approaches. Frequently bought products are generated through item-based collaborative filtering using K-nearest neighbours. Similar products are identified through TF-IDF representations and cosine similarity. Top-selling products are ranked using a weighted popularity score based on purchase counts and unique customers, with recommendations filtered by department.

These approaches provide complementary product discovery options based on purchase interactions, product attributes, and popularity. The application allows users to search for products or select a department, choose a recommendation method, specify the number of recommendations, and view the resulting product list.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



D. Discussion

The results demonstrate the integration of a machine learning forecasting model and multiple recommendation techniques into a single web application. The forecasting module provides numerical sales predictions and evaluation metrics, while the recommendation module supports different product discovery strategies. The React.js frontend and FastAPI backend connect these functions, and MongoDB Atlas is reserved for authentication data. Forecasting and recommendation datasets, model artifacts, and generated outputs are maintained separately.

VII. CONCLUSION AND FUTURE SCOPE

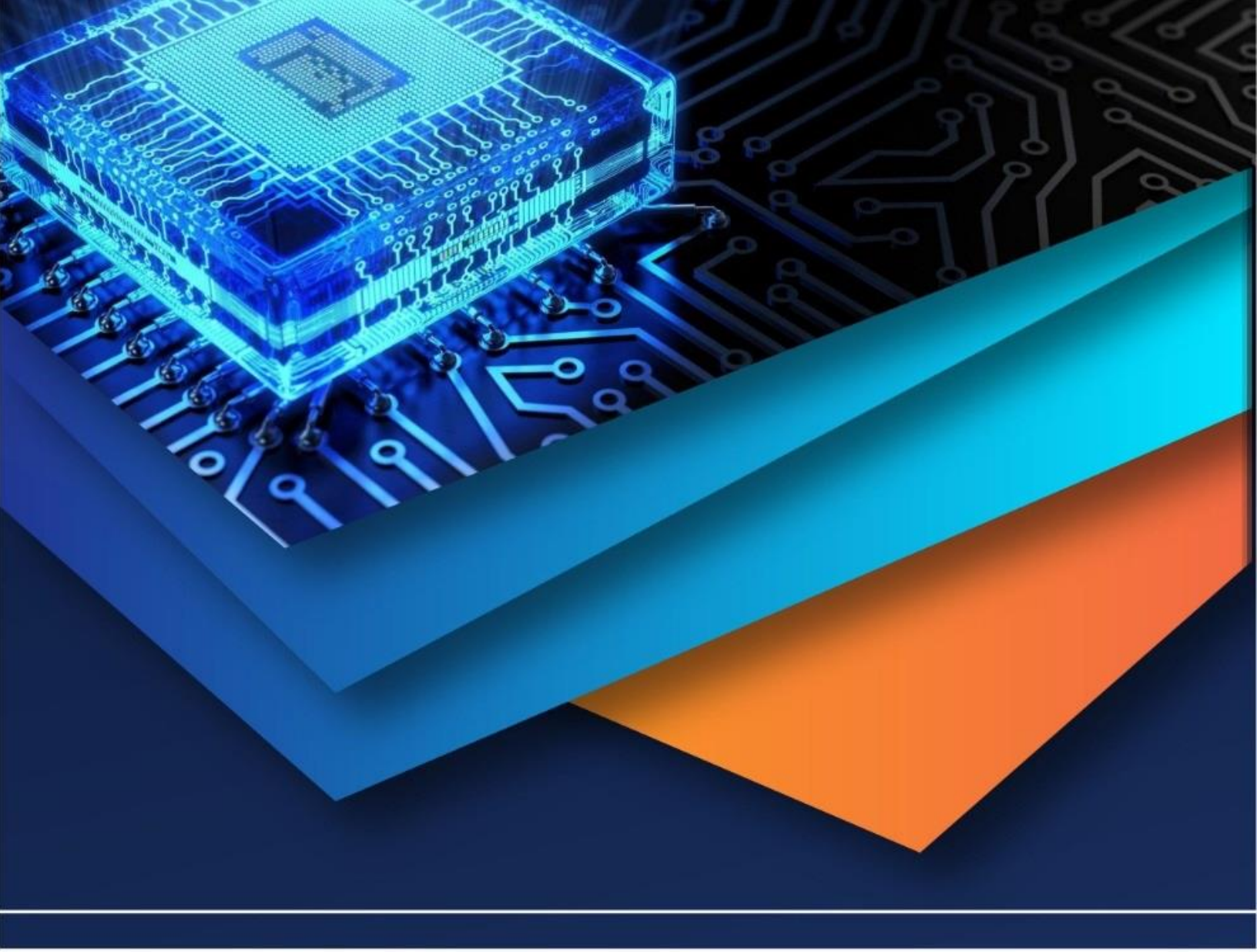
The project, Time Series Based E-commerce Sales Forecasting and Intelligent Recommendation System, presents a web-based application that integrates machine learning-based sales forecasting with intelligent product recommendations. An XGBoost Regressor is used to forecast daily e-commerce sales using historical sales data and engineered temporal features. The system also implements three independent recommendation techniques: item-based collaborative filtering for frequently bought products, content-based filtering for similar products, and popularity-based ranking for top-selling products within a department.

The forecasting model is evaluated using MAE, RMSE, MAPE, and (R^2), while the web application presents forecasting results, sales analytics, and product recommendations through a React.js frontend connected to a FastAPI backend. MongoDB Atlas is used exclusively for user authentication data, while datasets, model artifacts, and generated outputs are maintained separately. The integration of these components provides a unified interface for accessing sales forecasts and different product recommendation options. Future work may explore additional forecasting techniques, comparisons across different model configurations, and the incorporation of further temporal or external factors where suitable data is available. Recommendation functionality could be extended through additional personalization techniques and more comprehensive evaluation using recommendation-specific metrics. The application could also be enhanced with larger datasets and further performance and usability evaluation.

REFERENCES

[1] P. G. Giannopoulos, T. K. Dasaklis, I. Tsantilas, and C. Patsakis, “Machine Learning Algorithms in Intermittent Demand Forecasting: A Review,” 2025.
[2] B. Wang and A. B. M. Zain, “A Hybrid XGBoost-LSTM Framework for Supply Chain Demand Forecasting: Empirical Evidence from Retail Multi-Store Data,” 2025.
[3] Z. Mohammed, C. Anas, and M. El Hammoumi, “A Hybrid Learning Framework for Forecasting Uncertainty and Adaptive Inventory Planning in Retail Supply Chains,” 2026.
[4] S. Mansur, K. Sattar, S. E. Hosseini, S. Pervez, I. Ahmad, K. Saleem, and A. Z. Elhendi, “Sales Forecasting for Retail Stores Using Hybrid Neural Networks and Sales-Affecting Variables,” 2025.
[5] P. Yadav and C. R. Barde, “Retail Real-Time Sales Prediction System Using LSTM and XGBoost,” 2025.
[6] C. Wang and J. Wang, “Research on E-Commerce Inventory Sales Forecasting Model Based on ARIMA and LSTM Algorithm,” 2025.
[7] H. Jatte, S. Belattar, and E. K. Haimoudi, “Optimization of Forecasting Performance in the Retail Sector Using Artificial Intelligence,” 2025.
[8] K. Douaioui, R. Oucheikh, O. Benmoussa, and C. Mabrouki, “Machine Learning and Deep Learning Models for Demand Forecasting in Supply Chain Management: A Critical Review,” 2024.
[9] T. Zhou, “Improved Sales Forecasting using Trend and Seasonality Decomposition with LightGBM,” 2023.
[10] Y. Chen, X. Xie, Z. Pei, W. Yi, C. Wang, W. Zhang, and Z. Ji, “Development of a Time Series E-Commerce Sales Prediction Method for Short-Shelf-Life Products Using GRU-LightGBM,” 2024.
[11] S. B. Kwarteng and P. A. Andreevich, “Comparative Analysis of ARIMA, SARIMA and Prophet Model in Forecasting,” 2024.
[12] I. Boukrouh, S. Idriri, F. Tayalati, A. Azmani, and L. Bouhassien, “A Hybrid Approach for Sales Forecasting: Combining Deep Learning and Time Series Analysis,” 2025.
[13] V. I. Kontopoulou, A. D. Panagopoulos, I. Kakkos, and G. K. Matsopoulos, “A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data-Driven Networks,” 2023.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)