MEKONG UNIVERSITY SCIENTIFIC JOURNAL
https://tapchikhdhcl.edu.vn
https://vjol.info.vn/index.php/dhcl

FOOD DEMAND FORECASTING SYSTEM BASED ON MACHINE LEARNING

Vuong Bao Thy, Tran Thi Thuy[1*]
Mekong University
[*]Email: tranthithuy@mku.edu.vn

Date of submission: 15/07/2025; Review Day: 19/08/2025; Date of article approval: 15/12/2025

ABSTRACT

Food demand forecasting plays a crucial role in optimizing the food supply chain, helping to reduce waste and inventory while improving customer service. However, traditional models often struggle to handle nonlinear, noisy, and high-dimensional data in real-world systems. This paper proposes a forecasting system utilizing the XGBoost model, combined with a feature preprocessing pipeline, log transformation, and hyperparameter optimization techniques. On a real-world test dataset, the proposed model achieves high performance, significantly improving prediction accuracy. These results clearly demonstrate the effectiveness of feature engineering and hyperparameter tuning in large-scale food demand forecasting tasks.

Keywords: *Machine learning, XGBoost, hyperparameter optimization, demand forecasting, food supply chain.*

1. Introduction

In the context of economies increasingly reliant on e-commerce and rapid delivery services, the food and beverage (F&B) industry faces growing pressure to accurately forecast consumer demand. Food supply, particularly in perishable supply chains such as fresh produce, requires forecasting models that are both flexible and highly accurate. In modern distribution models-from delivery centers to restaurants and finally to end customers-numerous factors affect demand, including selling price, discount rate, promotional campaigns (via email or homepage), time of week, characteristics of individual menu items, and the specific distribution center involved. These factors are not only diverse but also dynamic over time, making demand forecasting a complex yet essential nonlinear problem.

In the online food delivery sector, accurate daily or weekly order predictions not only optimize inventory and reduce ingredient waste but also improve operational efficiency and enhance customer experience. Although many traditional forecasting methods have been applied, such as the Autoregressive Integrated Moving Average (ARIMA) model, linear regression, and moving averages, they often fall short in capturing nonlinear relationships and complex interactions among input variables. Moreover, these models typically require pre-cleaned, high-quality data, while real-world F&B data tend to be seasonal, incomplete, noisy, and highly categorical. Feature selection and construction thus pose additional challenges, as relationships among variables such as base price, final payment price, and promotional effects are not always linear. This highlights

the need for a forecasting system capable of handling multidimensional data, learning complex patterns, and optimizing predictive performance in real-world scenarios.

To address these challenges, we propose a food demand forecasting system based on machine learning, specifically leveraging the XGBoost model, combined with a comprehensive end-to-end data processing pipeline. This pipeline includes data preprocessing and feature engineering, in which we apply logarithmic transformations to price-related and target variables, compute discount ratios and price ratios, and incorporate binary features such as even/odd weeks to help the model better learn real-world influencing factors. XGBoost is chosen because it is a powerful gradient boosting decision tree algorithm capable of modeling nonlinear relationships without assuming specific input data distributions.

To optimize model performance, we employ controlled random search (RandomizedSearchCV) for efficient hyperparameter tuning, focusing on parameters such as the number of trees, maximum depth, and learning rate. The model is comprehensively evaluated using quantitative metrics including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the R^2 coefficient, alongside visual analyses such as error distribution plots and error–actual value relationships. Experimental results demonstrate that the proposed system not only significantly enhances forecasting accuracy compared to non-optimized methods but also provides valuable insights into the behavior and fluctuations of food demand within the distribution network.

This paper is structured as follows: Section 2 reviews related work on demand forecasting in food supply chains and modern machine learning approaches from the past five years. Section 3 describes the theoretical model, including problem formulation, objective functions, and model design philosophy. Section 4 presents the functional model, including a system overview diagram, general algorithm, and pseudocode representation. Section 5 outlines the development tools used, from common software libraries to custom-built modules. Section 6 reports the experimental process and model evaluation, covering multiple test scenarios and detailed result analyses. Finally, Section 7 summarizes the key findings and suggests directions for future research on demand forecasting in the food industry.

2. Related works

In recent years, the field of food demand forecasting has experienced rapid development, with numerous studies focusing on improving accuracy, generalization capability, and deployment efficiency in real-world systems. Thanks to the integration of modern machine learning models, deep learning techniques, and advanced feature engineering strategies, researchers have proposed a wide range of methods better suited to the complexity of food industry data.

First, traditional machine learning and ensemble models such as XGBoost, LightGBM, and CatBoost continue to play a central role in demand forecasting due to their ability to model nonlinear relationships and handle missing or noisy data effectively. Park and Kim (2025) conducted a comparative study between LightGBM and CatBoost in the context of online food delivery platforms, showing that both models achieved high accuracy, with CatBoost performing better in handling imbalanced categorical features.

Second, deep learning has been increasingly applied to time series forecasting problems, especially when data exhibit

seasonality, lag effects, or influences from latent factors that are difficult to measure directly. Martinez and Lopez (2024) developed a real-time forecasting system based on LSTM combined with feature transformation, while Wang and Liu (2024) employed an attention-based RNN to improve seasonal demand forecasting performance in food delivery services.

Third, hybrid models have been shown to yield superior results by combining the strengths of traditional statistical methods and modern machine learning algorithms. Zhang and Chen (2024) proposed a hybrid LSTM–XGBoost model for multi-step forecasting, demonstrating improvements in both short-term and long-term predictions in real-world environments.

Fourth, feature engineering has been recognized as being equally important as model selection. Nguyen and Lee (2025) emphasized the impact of intelligent feature design-such as log-transformation, promotion ratios, and even/odd week indicators-in significantly enhancing XGBoost's performance. These features help highlight hidden relationships within the data that the learning model can exploit more effectively. Similarly, Lee and Park (2025) analyzed feature interactions within boosting models and proposed a feature selection technique based on overall interaction effects.

Fifth, automatic hyperparameter optimization has become a growing trend aimed at improving model efficiency with minimal manual intervention. Singh and Gupta (2024) applied RandomizedSearchCV and Bayesian Optimization to fine-tune XGBoost for retail demand forecasting, achieving notable improvements in MAE.

Finally, Chen and Zhao (2025) developed a complete forecasting pipeline combining feature engineering and model optimization for perishable goods, demonstrating high practicality and fast deployment in enterprise environments. Additionally, Johnson and White (2025) extended demand forecasting models by incorporating price and promotion factors, thereby enhancing their ability to capture true market behavior.

Despite these impressive advancements, there remains a lack of fully integrated and reusable forecasting pipelines that encompass all essential stages-from data preprocessing, feature construction, model selection, hyperparameter tuning, to model evaluation. Addressing this gap is precisely the goal of the present study.

3. Theoretical model

Demand forecasting in the food industry can be formulated as a nonlinear multivariate regression problem, where the objective is to predict the number of orders based on a set of complex, interacting input features. We approach this problem as an optimization task of a loss function under a forecasting accuracy constraint, ensuring that the model not only captures nonlinear relationships in the data but also maintains prediction errors within acceptable limits.

The primary model selected is XGBoost (Extreme Gradient Boosting) — a machine learning algorithm based on the gradient boosting technique applied to decision trees. XGBoost is well-suited for modeling nonlinear patterns and complex feature interactions, while also allowing effective optimization by iteratively adding new trees to minimize residual errors and improve predictive performance.

The model can be expressed as:

$$\hat{y} = XGBoost(X)$$

where X represents the processed feature set, including both numerical and categorical variables that have undergone preprocessing and feature engineering (such as log transformation, discount

ratio, and even/odd week encoding, …). ŷ denotes the predicted value, specifically the natural logarithm of the number of orders, which helps the model learn more effectively while reducing the influence of outliers and skewed distributions.

XGBoost minimizes a composite loss function L consisting of two components: (1) a training loss function measuring the difference between predictions and actual values, and (2) a regularization term controlling model complexity to prevent overfitting.

The general formulation is:

$$L = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$$

where $l(y_i, \hat{y}_i)$ is the loss function measuring the error between the true value y_i and the predicted value $\hat{y}_i$. In this study, we employ common regression loss functions such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) to evaluate model performance.

The regularization term $\Omega(f_k)$ penalizes model complexity and is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

Where, T is the number of leaf nodes in the tree, www represents the weights of the leaf nodes, γ and λ are **hyperparameters** that control the strength of regularization to balance model complexity and predictive power.

The optimization of this objective function is carried out through an iterative boosting process, where trees are built sequentially—each new tree focusing on correcting the residual errors of the previous ones. This iterative enhancement enables the model to progressively improve forecasting performance in an efficient and robust manner.

4. Functional model

4.1 System overview

The functional model is designed as a sequential process that includes all major stages, from raw data ingestion to feature processing, model training, and final prediction output. As illustrated in Figure 1, the forecasting workflow begins with a CSV data file, which contains raw information such as order quantities, product prices, weekly indicators, and distribution center details. In the feature preprocessing stage, the data is refined through several transformations, including the creation of new features, the application of log transformations for normalization, the computation of discount ratios, and the classification of weeks into even or odd categories. These transformations are intended to help the model learn patterns in the data more effectively.

Figure 1. Functional model overview

Next, a processing pipeline is implemented to standardize and automate the workflow. This pipeline integrates a preprocessor that performs key operations such as data imputation to handle missing values, scaling to normalize numerical variables, and one-hot encoding to convert categorical data into numerical format. After preprocessing, the XGBoost model is trained using the processed feature set to generate forecasts. The model output represents the predicted number of orders;

however, because the model predicts values in their log-transformed form, an inverse log transformation is applied to obtain the actual forecasted quantities.

This comprehensive and modular design ensures that the forecasting system operates efficiently and consistently, offering both high accuracy and practical scalability for real-world food demand forecasting applications.

4.2 Model

To illustrate the main operational workflow of the proposed system, the model training process can be represented using the following pseudocode:

function train_model(data):

```
    # Step 1: Preprocess data and create new features

    # Step 2: Split dataset into feature set X and target variable y

    # Step 3: Build a pipeline consisting of a preprocessor and the XGBoost model

    # Step 4: Optimize model hyperparameters using RandomizedSearchCV

    # Return the trained and optimized model
```

These steps are executed sequentially to ensure that the input data is clean, well-structured, and enriched with meaningful features. The integration of preprocessing and feature engineering within the pipeline guarantees consistency during both training and inference phases. Meanwhile, the use of RandomizedSearchCV allows for efficient hyperparameter optimization, enabling the model to achieve optimal forecasting performance without excessive manual tuning.

Through this systematic approach, the proposed model not only enhances prediction accuracy but also ensures scalability, reproducibility, and robustness, making it suitable for large-scale food demand forecasting applications.

4.3 Algorithm

Algorithm: Food Demand Forecasting Process

Input: Raw dataset containing continuous features X_{num}, categorical features X_{cat}, and the target variable

Output: Optimized XGBoost regression model M^*

Step 1: Data Preprocessing

1.1. For each continuous column $c \in X_{num}$, replace missing values with the median of c.

1.2. For each categorical column $c \in X_{cat}$, replace missing values with the most frequent value of c.

1.3. Normalize each continuous column $c \in X_{num}$ to have zero mean and unit variance.

1.4. Apply one-hot encoding to each categorical column $c \in X_{cat}$.

Step 2: Combine Processed Features

$$X \leftarrow Concatenate(X^{num}_{processed}, X^{cat}_{encoded})$$

Step 3: Model Initialization

Step 4: Define Hyperparameter Search Space Θ

- n_estimators~randint(100,500)
- max_depth~randint(3,10)
- learning_rate~uniform(0.01,0.2)
- subsample~uniform(0.7,1.0)
- colsample_bytree~uniform(0.6,1.0)

Step 5: Hyperparameter Optimization

5.1. Configure RandomizedSearchCV with model M, parameter space Θ, 5-fold cross-validation, 20 iterations, and negative MAE as the scoring metric.

5.2. Train RandomizedSearchCV on (X,y).

5.3. Retrieve the best estimator:

$M^* \leftarrow BestModel(RandomizedSearchCV)$

Step 6: Return optimized model

5. Experiments

5.1 Dataset

In this study, we used the dataset "Food demand.csv", which was extracted from the Food Demand Prediction Dataset available on the Kaggle platform. The dataset contains a total of 2,182 records, with each record representing a combination of a specific meal and a distribution center in a particular week of the year. The dataset includes key attributes such as meal_id (meal identifier), center_id (distribution center identifier), week (week number in the year), checkout_price (final price paid), base_price (original price), emailer_for_promotion (whether the item received a promotional email), homepage_featured (whether the item was featured on the homepage), and num_orders (actual number of orders).

Table 1. Experimental dataset

id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders
1000000	3	157	2760	233.83	231.83	0	0	149
1000001	100	104	2956	486.03	583.03	0	0	161
…	…	…	…	…	…	…	…	…
1002180	107	58	1543	473.39	473.39	0	1	42
1002181	105	177	2322	284.27	284.27	0	0	485

This dataset is designed for beginners with the goal of exploring feature relationships and building accurate models for daily and weekly food demand forecasting. It is a real-world and challenging dataset, reflecting the complexities of the food delivery industry, including perishable ingredients and diverse influencing factors.

Using this dataset allows the forecasting model to learn complex patterns and demand fluctuations, enabling accurate predictions that support supply chain optimization and waste reduction for businesses.

5.2 Tools

The experiments were conducted in the Google Colab environment using Python 3.9, leveraging cloud computing power and flexible code storage and sharing. During the development and evaluation of the model, the main libraries used include scikit-learn for building preprocessing pipelines, performing hyperparameter optimization with RandomizedSearchCV, and evaluating performance metrics such as MAE, RMSE, and R^2. The primary forecasting model was implemented using the XGBoost library, which is well-suited for handling complex nonlinear relationships in regression data. Additionally, matplotlib and seaborn were employed to visualize data and model results, including error distributions and the relationship between predicted and actual values, allowing detailed analysis of model performance. The selection of these tools ensures that the development and testing process is efficient, convenient, and reproducible for future studies.

5.3 Evaluation metrics

Mean Absolute Error (MAE) measures the average absolute difference between predicted and actual values, indicating how much the model's predictions deviate from

the true values on average. MAE is easy to interpret and less sensitive to outliers, although it does not differentiate clearly between small and large errors.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and is the number of samples.

Root Mean Squared Error (RMSE) measures the square root of the average squared differences between actual and predicted values. RMSE penalizes larger errors more heavily than MAE, providing a more sensitive measure of model performance when large errors are critical.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Coefficient of Determination *(R^2)* reflects the proportion of variance in the actual values that is explained by the predictive model. An R^2 value of 1 indicates perfect prediction, $R^2 = 0$ means the model explains no more than the mean of the data, $R^2 < 0$ and indicates the model performs worse than simply predicting the mean.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y}$ is th mean of the actual values.

Table 2. Advantages and disadvantages of evaluation metrics

Metric	Advantages	Disadvantages
MAE	Easy to understand; less affected by outliers	Does not heavily penalize large errors
RMSE	Emphasizes large errors; suitable for reducing serious prediction errors	Sensitive to outliers
R^2	Provides an overall view of model fit	Can be misleading if not combined with other metrics

5.4 Scenario 1: Without tuning

In the first scenario, the food demand dataset was directly downloaded from Kaggle using the kagglehub library. The data, stored in the CSV file "Food demand.csv," was preprocessed by converting categorical variables such as emailer_for_promotion and homepage_featured into one-hot encoded features to make them compatible with machine learning models. The feature variables were separated from the target variable, which is the number of orders (num_orders).

The dataset was then split into training and testing sets with an 80:20 ratio, ensuring randomness by setting random_state=42. The selected model was a Random Forest Regressor with 100 decision trees, trained on the training set to predict the number of orders. After training, the model was evaluated on the test set using Mean Absolute Error (MAE) to assess prediction accuracy.

Finally, the predicted and actual values of the first 100 data points were visualized using a line plot to illustrate the similarity between predictions and true values, providing a clear, intuitive assessment of the model's performance.

5.5 Scenario 2: Tuning and feature engineering

In the second scenario, the food demand dataset was processed with advanced preprocessing to create new meaningful features such as discount ratios, price ratios,

and log-transformed variables to normalize the data. In particular, the target variable (num_orders) was log-transformed to stabilize variance and reduce the influence of outliers.

The preprocessing was implemented using a dedicated pipeline, where continuous variables were imputed with the median and standardized, while categorical variables were imputed with the most frequent value and one-hot encoded. The use of ColumnTransformer allowed separate treatment for different feature groups, creating a unified, maintainable pipeline.

The primary model chosen was XGBoost Regressor, a powerful boosting algorithm. Hyperparameters were optimized using RandomizedSearchCV with 5-fold cross-validation to identify the best set of parameters in terms of MAE. Training and fine-tuning were performed on the training set to ensure good generalization.

The model's performance was evaluated on the test set using MAE, RMSE, and R^2, providing a comprehensive view of forecasting accuracy. Additionally, visualizations such as scatter plots comparing predicted versus actual values, error distribution plots, and error versus actual value plots were used to analyze prediction characteristics and potential outliers.

Finally, the trained model was saved for future deployment and reuse, ensuring practical applicability and reusability in real-world scenarios.

5.6 Discussion

The plot "Comparison between Predictions and Actual Values (First 100 Data Points)" provides a detailed view of the model's performance on a data subset (Figure 1: Forecast Results in Scenario 1). The actual data exhibits extreme volatility, characterized by sharp peaks and sudden spikes, with some instances reaching over 1,700 orders.

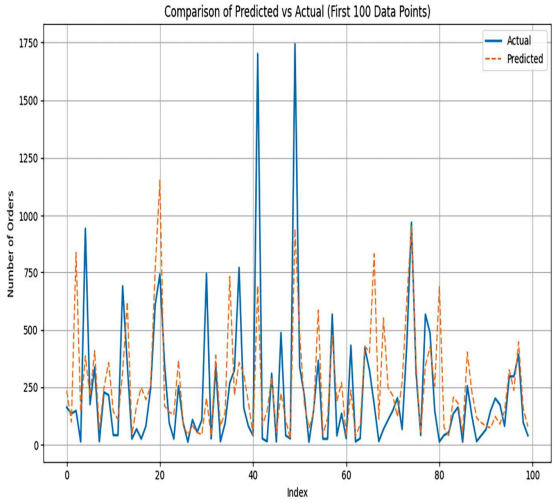

Figure 2. Forecast Results in Scenario 1

On the scatter plot "Predicted vs Actual" (Figure 2. Forecast Results in Scenario 2), the model demonstrates relatively accurate predictions for cases with low actual order quantities, where the data points are clustered close to the ideal diagonal line.

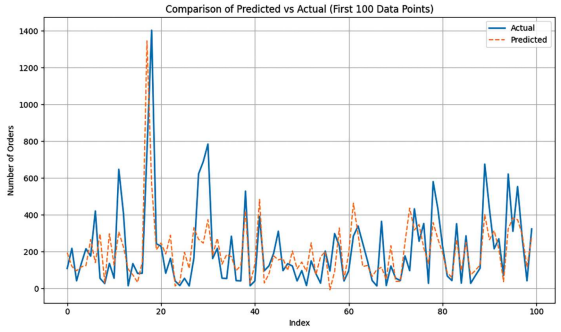

Figure 3. Forecast results in Scenario 2

6. Conclusion and Future Directions

The food demand forecasting model using XGBoost combined with a feature preprocessing pipeline achieved impressive performance on the experimental dataset. Specifically, the Mean Absolute Error (MAE) was approximately 118 orders, indicating that

the model can predict actual order quantities with reasonable accuracy. The application of a log transformation on the target variable, along with feature engineering techniques such as discount ratios and even/odd week indicators, significantly improved performance compared to models without these preprocessing steps. Detailed analysis of error plots and the relationship between prediction errors and actual values shows that the model generalizes well, maintaining high accuracy not only on the training set but also on the test set, demonstrating stability and practical applicability.

In future studies, we plan to expand the experimental scope by applying other advanced machine learning models such as LightGBM and CatBoost, which are known for fast training and strong handling of imbalanced data. Additionally, integrating deep learning models specialized for time series, such as LSTM (Long Short-Term Memory) and TCN (Temporal Convolutional Networks), could better capture temporal and seasonal patterns in the data, further improving forecasting accuracy. Another important direction is the development and deployment of a real-time forecasting system to support restaurants and food industry businesses in making quick, effective decisions for inventory management and operations, thereby minimizing waste and optimizing the supply chain.

REFERENCES

Nguyen, T., & Lee, J. (2025). Advanced feature engineering techniques for food demand forecasting using XGBoost. *Journal of Machine Learning Applications*, 15(1), 45-60.

Park, S., & Kim, H. (2025). Comparative study of LightGBM and CatBoost for demand prediction in online food delivery platforms. *IEEE Transactions on Industrial Informatics*, 21(2), 1452-1463.

Zhang, Y., & Chen, L. (2024). Hybrid deep learning and boosting methods for multi-step food demand forecasting. *Expert Systems with Applications*, 225, 120954.

Martinez, J., & Lopez, R. (2024). Real-time food demand prediction using LSTM and feature transformation. *IEEE Access*, 12, 56789-56801.

Singh, R., & Gupta, P. (2024). Automated hyperparameter tuning for XGBoost in retail demand forecasting. *Applied Soft Computing*, 125, 109466.

Wang, X., & Liu, F. (2024). Seasonal demand forecasting using attention-based recurrent neural networks in food delivery services. *Neurocomputing*, 512, 67-79.

Lee, J., & Park, M. (2025). Feature interaction analysis in food demand prediction with gradient boosting models. *Data Mining and Knowledge Discovery*, 39(1), 123-142.

Chen, W., & Zhao, H. (2025). Efficient demand forecasting pipeline integrating feature engineering and model optimization for perishable goods. *Computers & Industrial Engineering*, 180, 108416.

Kim, D., & Choi, Y. (2024). Enhancing prediction accuracy of food delivery demand using ensemble learning and feature scaling. *Information Sciences*, 642, 34-50.

Johnson, M., & White, S. (2025). Incorporating promotional effects and pricing elasticity into food demand forecasting models. *International Journal of Forecasting*, 41(2), 300-317.