https://www.periodicos.unis.edu.br/mythos/

LOGISTICS AND TRANSPORTATION: FREIGHT DEMAND FORECASTING USING SARIMA AND XGBOOST

LOGÍSTICA E TRANSPORTE: PREVISÃO DA DEMANDA DE FRETE UTILIZANDO SARIMA E XGBOOST

Eduardo Modesto de Melo[1*], Fabrício Pelloso Piurcosky[2]

[1] *Computer Engineer and MBA in Data Science and Analytics (USP). University of São Paulo - USP, Brazil. eduardomodesto@gmail.com*

[2] *Postdoctoral Researcher in Social Sciences, Politics and Territory from the University of Aveiro, Portugal. Ph.D. in Business Administration from the Federal University of Lavras (UFLA). University Center Integrado, Campo Mourão-PR, Brazil. fabricio.pelloso@grupointegrado.br*

Editorial Details
Double-blind review system
Research article

Article History:
Received: March 5, 2026
Revised: June 6, 2026
Accepted: July 15, 2026
Published online: July 17, 2026

Editor-in-Chief:
Rodrigo Franklin Frogeri

Technical Editor:
Eufrásia de Souza Melo

Funding:
This study received no external funding.

How to cite this article:
Melo, E. M. de; Piurcosky, F. P. (2026). Logistics and Transportation: Freight Demand Forecasting Using Sarima and Xgboost. **Mythos**, v. 18, 1, 483-501. https://doi.org/10.36674/mythos.v18i1.1066

***Corresponding Author:**
Eduardo Modesto de Melo
eduardomodesto@gmail.com

Abstract

Accurate freight demand forecasting is essential for improving logistics planning and supporting decision-making in e-commerce supply chains. This study compares the performance of two forecasting approaches—Seasonal Autoregressive Integrated Moving Average (SARIMA) and Extreme Gradient Boosting (XGBoost)—for predicting daily freight demand measured by transported weight. The research followed the CRISP-DM methodology using the Brazilian Olist public e-commerce dataset. After data preprocessing, exploratory analysis, stationarity testing, and feature engineering, multiple SARIMA and XGBoost models were developed and evaluated using chronological train-test splitting, cross-validation, and Mean Absolute Percentage Error (MAPE). The SARIMA models incorporated seasonal differencing and Box-Cox transformations, whereas the XGBoost models included calendar-based variables, moving averages, and moving standard deviations. The results demonstrate that feature engineering substantially improved predictive performance. The best XGBoost model achieved a MAPE of 3%, considerably outperforming the best SARIMA model, whose predictive accuracy remained limited despite data transformations. These findings indicate that machine learning techniques combined with temporal feature engineering provide superior freight demand forecasts for e-commerce logistics. The proposed approach offers a practical decision-support tool for transportation planning, resource allocation, and operational efficiency while providing a reproducible computational workflow through publicly available source code and processed data.

Keywords: E-commerce, Machine Learning, Seasonality, Sales Order Data, Decision Support Systems.

ISSN: 1984-0098 | e-ISSN: 3086-0210

Resumo

A previsão precisa da demanda de frete é essencial para otimizar o planejamento logístico e apoiar a tomada de decisão em cadeias de suprimentos do comércio eletrônico. Este estudo compara o desempenho de duas abordagens de previsão, o modelo Sazonal Autorregressivo Integrado de Médias Móveis (SARIMA) e o Extreme Gradient Boosting (XGBoost), para estimar a demanda diária de frete medida pelo peso transportado. A pesquisa seguiu a metodologia CRISP-DM utilizando o conjunto de dados público *Brazilian E-commerce Dataset by Olist*. Após o pré-processamento dos dados, análise exploratória, testes de estacionariedade e engenharia de atributos, foram desenvolvidos e avaliados múltiplos modelos SARIMA e XGBoost por meio de divisão cronológica entre treinamento e teste, validação cruzada e Erro Percentual Absoluto Médio (MAPE). Os modelos SARIMA empregaram diferenciação sazonal e transformações Box-Cox, enquanto os modelos XGBoost incorporaram variáveis de calendário, médias móveis e desvios-padrão móveis. Os resultados demonstram que a engenharia de atributos melhorou significativamente o desempenho preditivo. O melhor modelo XGBoost obteve MAPE de 3%, superando expressivamente o melhor modelo SARIMA. Os achados evidenciam que técnicas de aprendizado de máquina associadas à engenharia de atributos temporais constituem uma alternativa robusta para previsão da demanda de frete em operações de comércio eletrônico, contribuindo para o planejamento do transporte, a alocação de recursos e a eficiência operacional.

Palavras-chave: Comércio eletrônico, Aprendizado de máquina, Sazonalidade, Dados de pedidos de vendas, Sistemas de apoio à decisão.

Data available

To ensure transparency, reproducibility, and facilitate future research, all source code developed for this study, together with the processed dataset used in the analyses, has been made publicly available. The complete implementation, including data preprocessing procedures, model development, and forecasting routines, can be accessed through the following Google Colab repository:

https://colab.research.google.com/drive/1OJt9zG27du8rFB3WyC1Dfev05OfFExfk?usp=sharing

The repository provides a fully executable computational environment, allowing researchers and practitioners to reproduce the reported results, inspect the implementation details, and adapt the workflow for related freight demand forecasting and logistics applications.

Declaration on the Non-Use of Artificial Intelligence in the Preparation of the Manuscript

The authors declare that no artificial intelligence (AI) tools were used in the generation, editing, or analysis of the content presented in this manuscript. All ideas, text, data interpretation, and manuscript preparation were solely the work of the authors. The authors take full responsibility for the integrity and originality of the submitted work.

1 INTRODUCTION

Planning the transportation of goods and products within the field of Logistics and Supply Chain Management is a critical process for the competitiveness of any organization, particularly in the retail and e-commerce sectors. In these contexts, road transportation is the most commonly used mode, requiring organizations to make data-driven decisions. Currently, technological tools are available to support this need by enabling analytical and predictive decision-making processes (Novack, 2019).

Organizations increasingly allocate a portion of their budgets to technological investments in order to remain competitive. This scenario has created opportunities for companies specialized in the development of Transportation Management Systems (TMS), which invest in research and development to design applications that employ mathematical algorithms capable of performing complex operations related to demand forecasting, transportation cost estimation, and workforce allocation optimization for logistics execution and delivery stages. These activities require an increased flow of information derived from the systems and processes that comprise the supply chain (Novack, 2019).

Customer satisfaction through the delivery of high-quality products enables retail organizations to expand their market share or operate as intermediary partners in marketplace platforms, where other retailers leverage these platforms to sell directly to consumers. In a market characterized by international competition, with companies operating across multiple countries and serving consumers through complex processes and routes, delivery time has become an increasingly significant factor in customer satisfaction metrics. The ability to deliver high-quality products within shorter lead times and at lower freight costs (Prim, 2020) is essential to maintaining competitiveness, especially in the retail sector.

Consequently, there is a growing effort to develop tools that enhance the ability of logistics and supply chain systems to forecast and plan demand fulfillment regardless of customer location. In this context, Machine Learning algorithms combined with the implementation of Neural Networks have gained relevance in both academic research and industrial applications (Saha et al., 2022). In a highly competitive environment where consumer behavior is extensively analyzed and e-commerce platforms seek to retain customers through indepth studies of satisfaction and repurchase intention, Machine Learning (ML) techniques have become valuable decision-support tools in online retail operations (Nagel & dos Santos, 2017).

Given the widespread adoption of Business Intelligence tools and data analytics, there is an increasing interest in applying ML techniques to the large datasets available within organizations. These datasets are generated through the use of Enterprise Resource Planning (ERP) systems and auxiliary applications that support supply chain operations and fleet management for goods transportation (Brunheroto, 2022).

Regarding demand forecasting, various methods can be applied depending on the forecasting horizon, short, medium, or long-term, each offering distinct strategic advantages to organizations within their respective markets (Hyndman, 2021). Some models based on decision trees, such as Gradient Boosting Machines and Random Forests, have already been applied to demand forecasting and evaluated for performance within the retail sector (Pavlyshenko, 2019).

Demand forecasting capability enables organizations to plan inventory levels and logistics capacity more effectively. By forecasting transported volumes by region and product category, decision-making processes related to resource allocation and facility location can be improved, leading to reduced freight costs and more efficient inventory management. Additionally, such planning can contribute to job and income generation in regions identified as strategically viable. This study proposes a freight demand forecasting approach in which the weight of a given good or product is considered the response variable. However, it is acknowledged that additional variables should be incorporated to develop a more comprehensive and accurate predictive process, which falls outside the scope of this work. Furthermore, the complexity associated with different product categories and their distinct characteristics in transportation planning is not addressed in this study.

2 METHODOLOGY

The objective of this study is to develop two supervised Machine Learning (ML) predictive models to forecast freight demand expressed as product weight to be transported, and to compare their respective performance over a given period using the SARIMA method (Hyndman, 2021) and XGBoost (Bruce, 2019). The study follows the CRISP-DM methodology, with emphasis on the Data Preparation, Modeling, and Model Evaluation phases (Chapman et al., 1999).

All analyses were conducted using the Python programming language (version 3.11.8) within a Jupyter Notebook environment, hosted on the Google Colab platform. Data cleaning and transformation procedures, as well as the implementation of forecasting algorithms, were performed using this same environment. Model performance was subsequently compared using Cross-Validation and the Mean Absolute Percentage Error (MAPE) metric across all generated models.

In order to make the process more reproducible, we have described the computational workflow in more detail. The analyses were conducted in Python 3.11.8 in a Jupyter Notebook hosted in Google Colab. The toolset was pandas and NumPy for data development, SciPy and statsmodels for statistical and time-series processing, scikit-learn for data partitioning and evaluation, XGBoost for gradient-boosted regression, and Matplotlib for visualization. In the XGBoost models, we used 100 boosting trees (n_estimators = 100) and the model formulation was as follows. The predictions were made on Business Day, Holiday, Weekend, 7-day moving average and 7-day moving standard deviation. We split the dataset chronologically; 80% of the observations are for training and 20% for testing without randomly shuffling the time series. Any XGBoost parameters that are not explicitly changed in the notebook were kept at the default XGBoost release. For the exact version of all libraries installed, it should be exported from the shared public notebook (i.e., pip freeze) and stored together with the source code.

2.1 Problem Understanding and Variable Definition

This study seeks to address the question of freight demand, measured in terms of transported weight, over a given period through the application of two predictive models.

As this research focuses on logistics planning aimed at forecasting transportation demand, the following attributes were considered:

1. Attributes related to customer orders, such as product, purchase date, and delivery date.
2. Attributes related to order processing, including order status (e.g., delivered, canceled, approved, invoiced).
3. Attributes related to the purchased product, such as weight (in grams) and volume (in cubic centimeters).
4. Attributes related to customer location, including state (UF) and city.

Based on these attributes, it is assumed that a transaction profile can be constructed, providing a plausible foundation for logistics demand planning using the data collected for this study.

2.2 Data Collection

At this stage, the data were obtained through secondary data collection, using the Brazilian E-Commerce Public Dataset (Olist & Sionek, 2018), which is publicly available on the Kaggle platform under the CC BY-NCSA 4.0 license.

Among the datasets publicly available, only those listed below were used in this study:

1. Orders Dataset: the primary dataset used in this research, as it contains customer transaction information, purchase and delivery dates, order status at the time of data collection, order approval

date, carrier processing and delivery dates, estimated and actual delivery dates, and a unique customer identifier enabling integration with other datasets.

2. Order Items Dataset: contains information about products purchased in each customer order, with a oneto-many (1–n) relationship. This dataset also provides the link between orders and products through a unique product identifier.

3. Products Dataset includes, among other attributes, the unique product identifier, product dimensions in centimeters, and weight in grams.

4. Customers Dataset: provides, among other attributes, the unique customer identifier referenced in the orders dataset, as well as city and state (UF), allowing approximate customer localization.

5. Other datasets: although not used in the data analysis phase, the database also includes customer geolocation data, seller listings from the Olist platform, and datasets containing customer reviews and comments related to orders and products.

2.3 Data Preparation

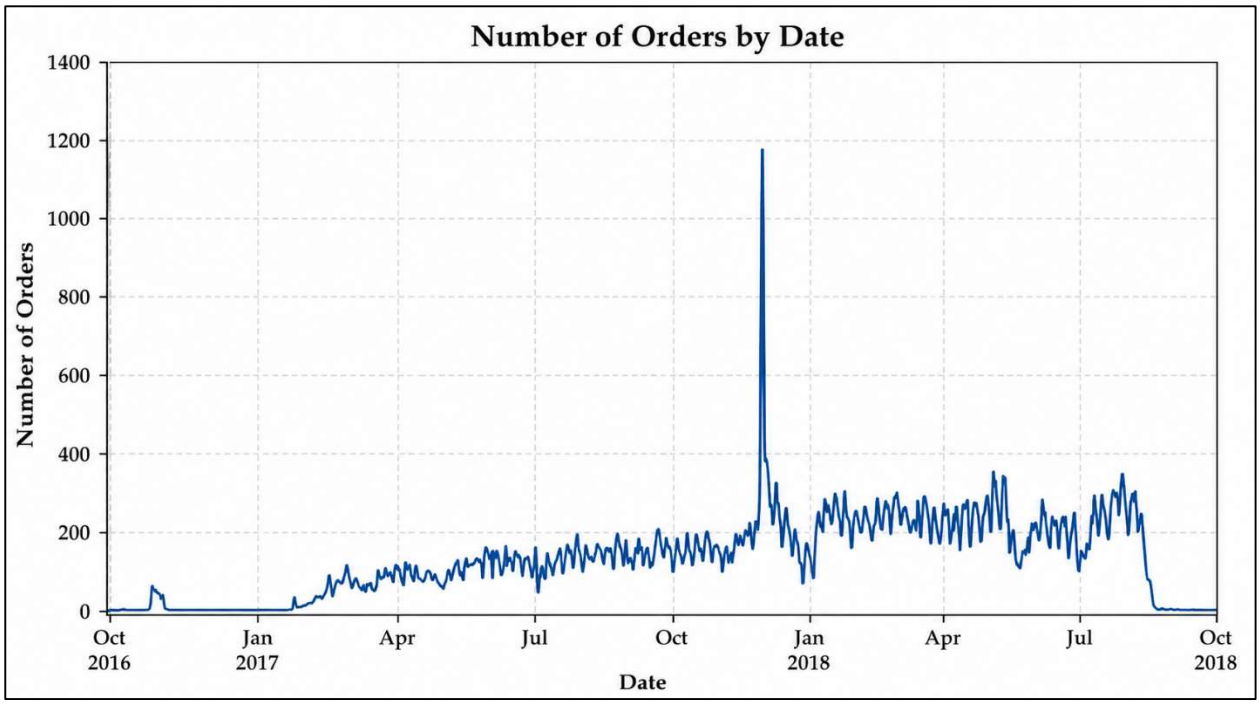
(A) **Exploratory Data Analysis**

Following data collection and understanding, it was observed that the primary dataset consists of 99,440 records, with data spanning from September 4, 2016, to October 10, 2018. These records are distributed across customers located in different regions and cities throughout Brazil.

Consequently, an exploratory analysis was conducted to examine data aggregation patterns, aiming to identify regions with the highest number of orders, monthly demand variations, and potential anomalies in the dataset. Figure 1 illustrates the distribution of the number of orders per month, highlighting the presence of anomalies that require further investigation.

Figure 1

Distribution of Orders per Month from 2016 to 2018



Source: Developed by the authors.

Several outliers were identified in the dataset:

1. **Data gaps** were observed between September and October 2016, with several days presenting no orders or only a single order. This pattern recurs during November and December 2016 and again toward the end of the data collection period starting in August 2018. The underlying causes of these gaps are subject to further investigation and, therefore, were not considered in this study.

2. **A specific date exhibited an exceptionally high order volume** compared to other days, with 1,176 orders recorded on November 24, 2017. This spike was identified as corresponding to the "Black Friday" event held in Brazil during that year.

With the exception of the Black Friday event recorded in the sample, observations meeting the aforementioned conditions were excluded from the dataset in order to avoid potential distortions in the algorithms responsible for training and executing the predictive models.

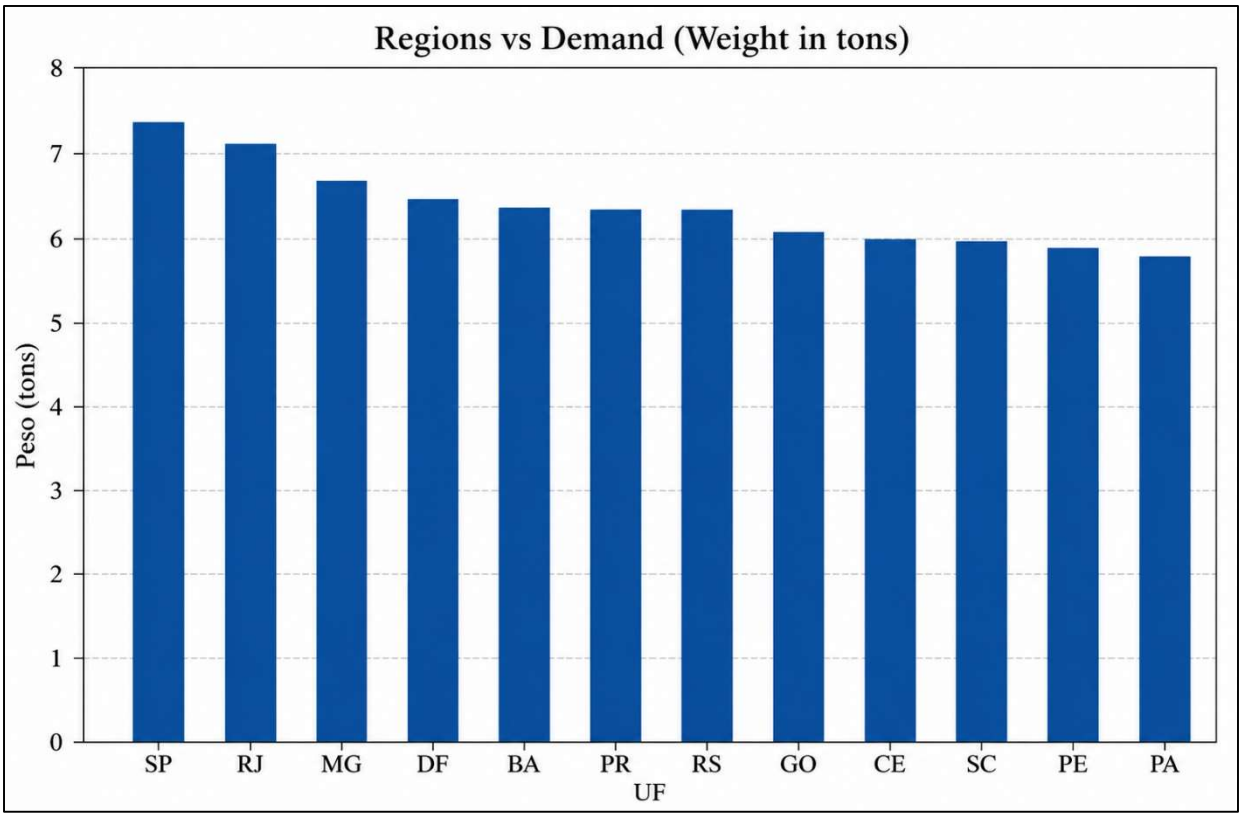
To compute the total weight in grams for each order and enable daily aggregation, the orders dataset was merged with the order items dataset, resulting in a consolidated order–product relational database.

Additionally, customer state (UF) and city information associated with each order were incorporated for a one-year period spanning from July 1, 2017, to June 30, 2018.

The resulting dataset excluded samples in which the order status did not indicate completed delivery, ensuring that only records corresponding to orders effectively delivered to customers—and thus completing the logistics transportation process—were retained. In this final stage, the state of São Paulo was identified as the region with the highest demand, or highest number of delivered orders, as illustrated on a logarithmic scale in Figure 2.

Figure 2

Demand (Weight in Tons, Logarithmic Scale) by Region in Brazil



Source: Developed by the authors.

ISSN: 1984-0098 | e-ISSN: 3086-0210

(B) **Data Treatment**

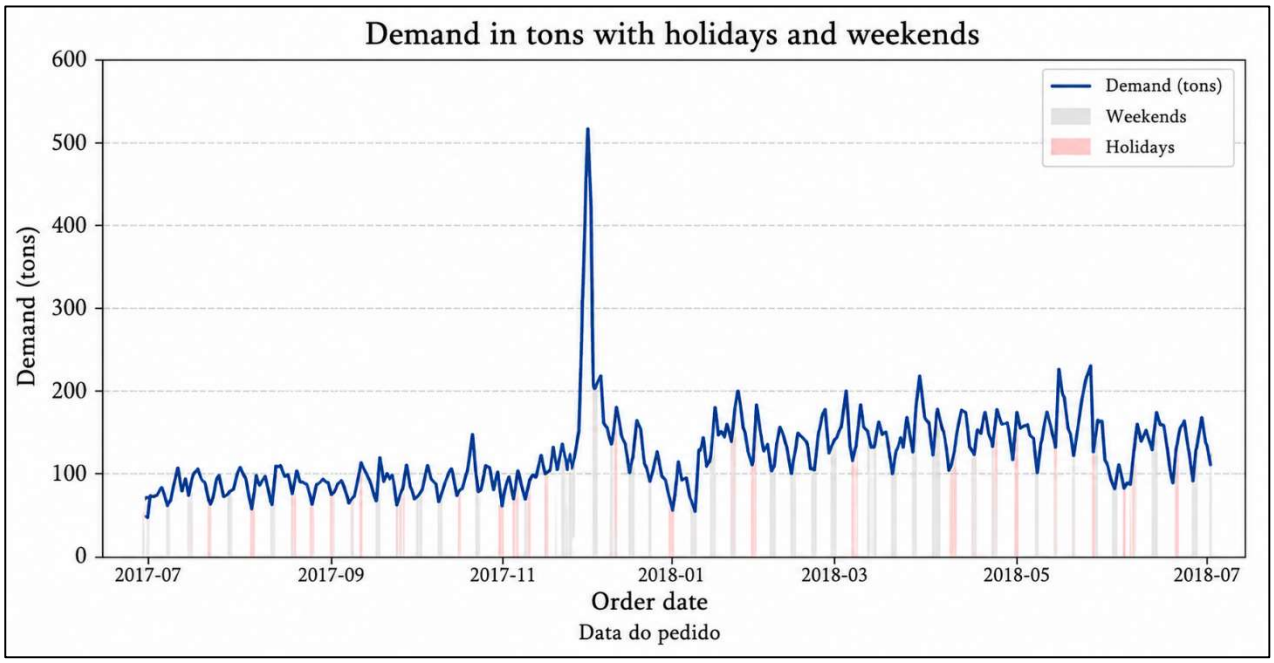
Once the dataset to be used as the basis for modeling was obtained, it became necessary to identify its characteristics within the context of a time series. The ordinal qualitative variable *purchase order date* was used as the temporal reference for the time series, as it represents the moment when an order was placed and provides the logistics planning department with visibility into freight demand, measured by weight, for a given day. In this study, this variable is referred to as *Order Date*.

To better understand the behavior of the data, a transformation was applied to the dataset to classify each date as either a business day or a non-business day (holiday or weekend) for the calendar years 2017 and 2018, as illustrated in Figure 3. This transformation enabled the identification of a clear weekly seasonality pattern, including a significant decline in demand during weekends and holidays.

At the end of the data treatment process, a dataset comprising 364 days was obtained, containing daily freight demand measured in tons.

Figure 3

Demand in Kilograms for Weekdays, Weekends, and Holidays



Source: Developed by the authors.

Conceptually, the data treatment process resulted in the construction of a time series of the form

$$\{Xt\}nt{=}1 = \{X1, X2, \dots, Xn\},$$

where Xrepresents the quantitative variable *Weight* and $t$denotes time, expressed by the discrete quantitative variable *Order Date*.

Upon examining the variability in the data, Seasonal-Trend Decomposition using Loess (STL) was applied due to the presence of outliers, with the objective of assessing whether the time series residuals approximate white noise, as recommended in the literature (Hyndman, 2021), following the initial analysis.

Conceptually, the time series was decomposed using both additive and multiplicative models, defined respectively as

$$y_t = T_t + S_t + E_t$$

and

$$y_t = T_t \cdot S_t \cdot E_t,$$

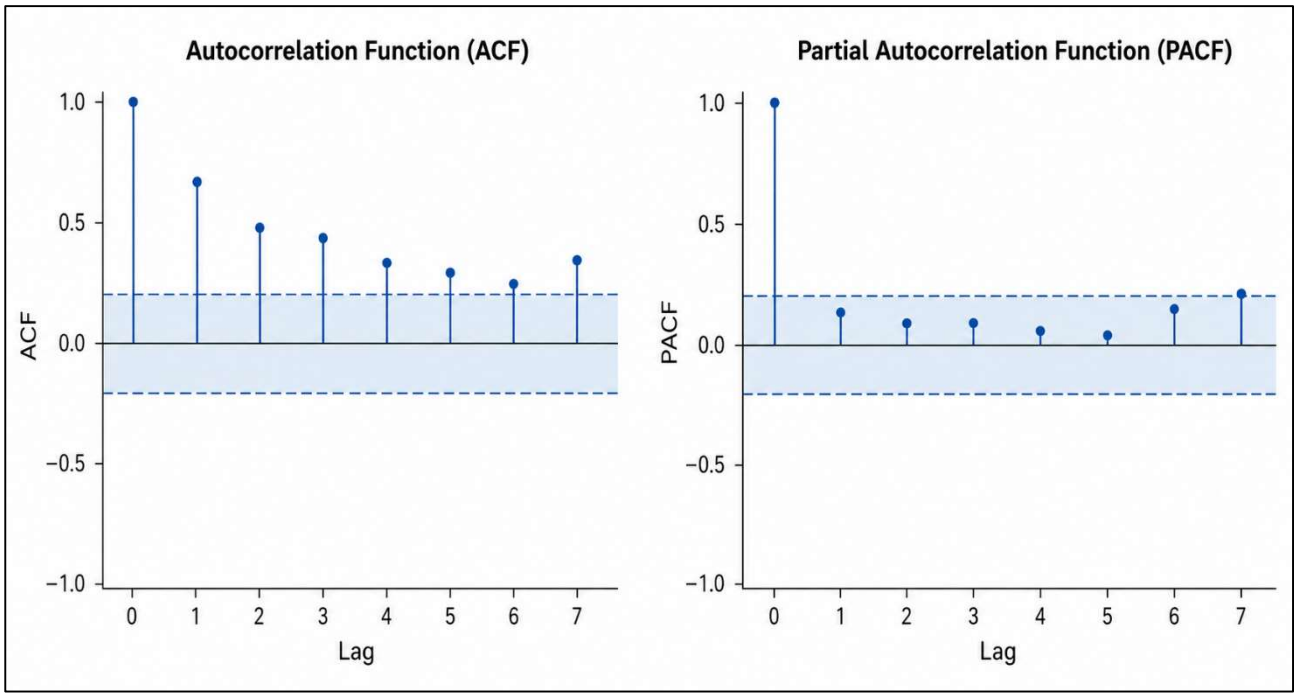
where $y$represents the value of the *Weight* variable, and T, S, and Ecorrespond to the trend, seasonal, and error components at time t.

From the seasonal decomposition, the following components were obtained:

- **Trend**: The trend component indicates an increase in demand over the final six months of the sample period, as illustrated in Figure 4. This behavior is consistently observed in both the additive and multiplicative decompositions.
- **Seasonality**: A clear weekly seasonality pattern was identified, characterized by higher demand on business days and lower demand on non-business days. This behavior was confirmed through the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) analyses, in which business days exhibited low correlation, while non-business days showed higher autocorrelation, as presented in Figure 5.
- **Residuals**: The residual component displayed behavior consistent with the identified seasonality when the additive decomposition method was applied.

Figure 4

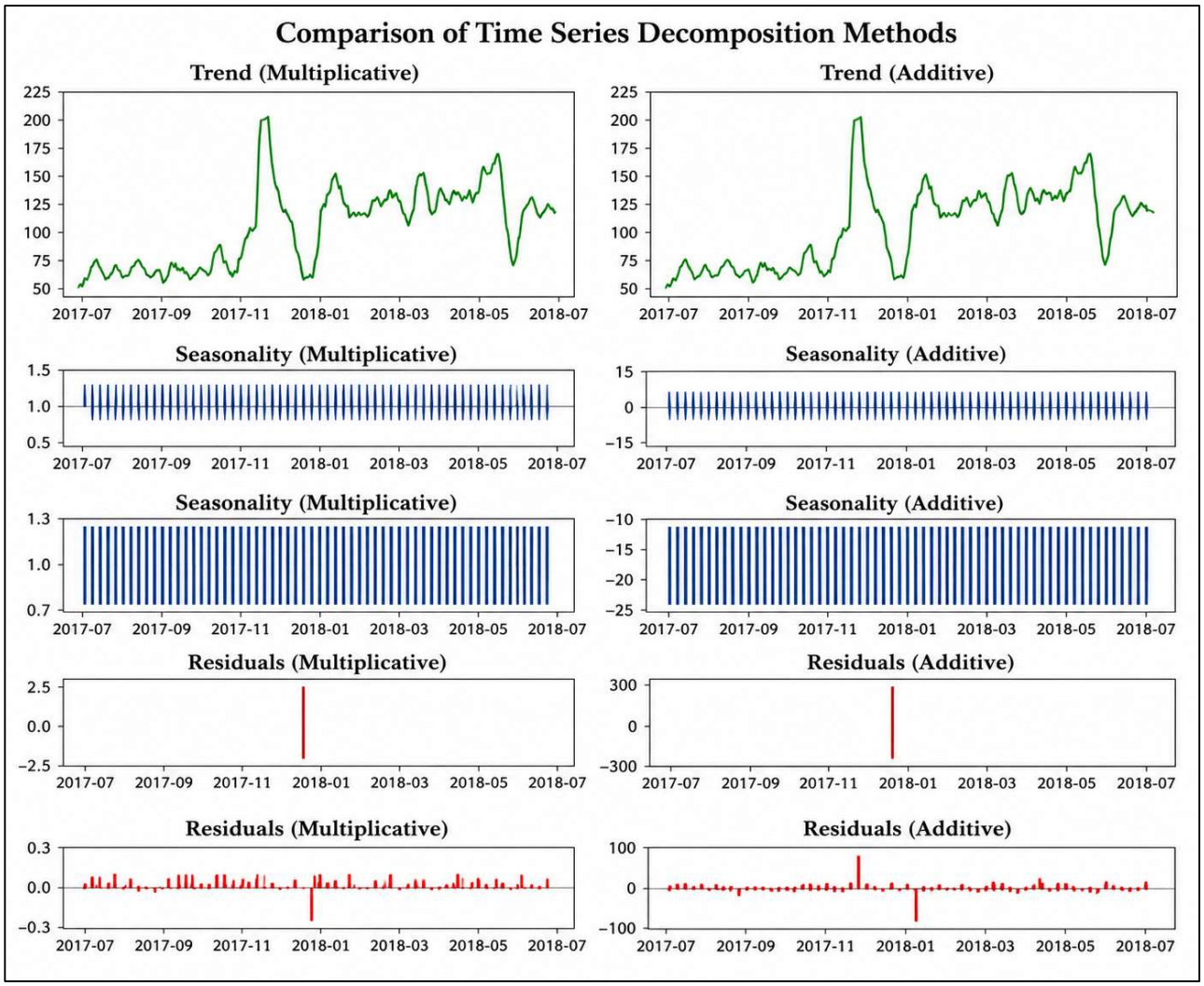
ACF and PACF Plots



Source: Developed by the authors.

Figure 5

Comparison of Time Series Decomposition Using Multiplicative and Additive Methods



Source: Developed by the authors.

In order to determine whether the time series is stationary, two statistical tests were applied: the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test and the Augmented Dickey–Fuller (ADF) test. The respective results are presented in Table 1.

Table 1

Comparison of p-values for Time Series Stationarity

Time Series	KPSS	ADF
Original data from the *Weight* variable	0.01 – Null hypothesis rejected	0.01003 – Null hypothesis accepted
First-order differenced *Weight* data	0.05578 – Null hypothesis accepted	Significantly less than 0.01 – Null hypothesis rejected

Source: Developed by the authors.

Based on these results, the first-order differenced series was considered stationary and suitable for modeling. The resulting dataset from this process consists of the fields listed below and their respective classifications (Morettin & Bussab, 2016), as shown in Table 2.

Table 2

Final Dataset Used for Predictive Model Generation

Field	Data Type	Values
Order Date	Discrete quantitative variable	Date in DD/MM/YYYY format
Weight	Continuous quantitative variable	Real value
Business Day	Nominal qualitative variable	0 = true, 1 = false
Weekend	Nominal qualitative variable	0 = true, 1 = false
Holiday	Nominal qualitative variable	0 = true, 1 = false

Source: Developed by the authors.

(C) **Modeling**

Given the stationarity of the time series after first-order differencing, the modeling phase focused on developing a classical Box–Jenkins model (SARIMA) and a tree-based model using Gradient Boosting (XGBoost). For both approaches, the dataset generated during the data preparation phase was maintained to ensure comparability of model performance based on the obtained results.

Initially, the dataset was divided into two subsets: a training set containing approximately 80% of the observations and a test set comprising the remaining data.

Seasonal ARIMA (SARIMA) Models

1) ARIMA model

For all seasonal ARIMA models, non-seasonal and seasonal orders of (1,1,1) were adopted, with a weekly cycle of seven days. The conceptual formulation of the seasonal ARIMA model is expressed as:

$$(1-\phi 1B)(1-\Phi 1Bm)(1-B)(1-Bm)yt=(1+\vartheta 1B)(1+\Theta 1Bm)\varepsilon t$$

2) Seasonal ARIMA Model Without Transformation

An initial simulation was performed using a SARIMA model applied to the dataset without transforming the *Weight* variable, aiming to assess model behavior and performance using MAPE and CrossValidation metrics. The estimated model is given by:

$$(1 - 0.2697B)(1 - 0.0395B^7)(1 - B)(1 - B^7)y_t = (1 - 0.8304B)(1 - 0.9982B^7)\varepsilon_t$$

3) Seasonal ARIMA Model with Box–Cox Transformation

The Weight data were transformed using the Box–Cox transformation to stabilize variance in the time series (Hyndman, 2021). The Box–Cox transformation is defined as:

$$y_t 0.0522 = \frac{y 0.0522}{0.0522} t - 1$$

The resulting SARIMA model is expressed as:

$$(1 - 0.3353B)(1 - 0.0534B^7)(1 - B)(1 - B^7)y_t = (1 - 0.7871B)(1 - 0.9980B^7)\varepsilon_t$$

4) Seasonal ARIMA Model with Logarithmic and Box–Cox Transformations

In this model, the dataset underwent a base-10 logarithmic transformation followed by a Box–Cox transformation (Fávero & Belfiore, 2017). This approach aimed to address the heteroscedasticity observed in previous models due to the presence of outliers, as identified during the data preparation stage. The optimized lambda value obtained was 1.27097. The Box–Cox transformation and the resulting predictive model are defined as:

$$1.27097 = \frac{y^t 1.27097^{-1}}{1.27097} y_t$$

$$(1 - 0.2937B)(1 - 0.0615B^7)(1 - B)(1 - B^7)y_t = (1 - 0.9991B)(1 - 0.9995B^7)\varepsilon_t$$

5) MXGBoost Ensemble Models

The models generated using the Boosting–Bagging technique with the XGBoost algorithm differ from the ARIMA-based models in that they incorporate hyperparameters tuned according to observed data behavior.

Three XGBoost models were developed:

1. A model using the original dataset without transformation.
2. A model using Box–Cox–transformed data.
3. A final model incorporating moving average and moving standard deviation features to enhance predictive capacity, using the same number of days as the test dataset.

Conceptually, the learning model for a dataset with $n$observations and predictor variables $x$is defined as:

$$n$$

$$\hat{y} = \sum_{k=1} f_k(w \cdot x)$$

The selected nominal qualitative predictor variables were *Business Day*, *Holiday*, and *Weekend*, as described in Table 2.

(i) Model Using Untransformed Data

A predictive model was developed using the Box–Cox transformation applied to the *Weight* variable, defined as:

$$0.0522 = y_t 0.0522 - 1$$

$$y_t \qquad \frac{\qquad\qquad}{0.0522}$$

The resulting model is expressed as:

$$\hat{y}^{(0.0522)} = \sum_{k=1}^{100} f_k(\text{Business Day,Holiday,Weekend})$$

(ii) Model Using Box–Cox Transformation with Moving Average and Moving Standard Deviation

As an alternative to the previously generated models, a predictive model incorporating a moving average was developed. The moving average was defined as:

$$\text{Moving Average}_t = \text{Window Size} \frac{1}{\qquad\qquad} \sum_{i=t-\text{Window Size}+1}^{t} \text{Weight}_i$$

A similar approach was applied to compute the moving standard deviation:

$$\text{Moving Standard Deviation} \quad (x, w) = \sqrt{\frac{1}{7}\sum_{i=1}^{7}(x_i - \bar{x})^2}$$

The window size was defined as seven days, following the same seasonal assumption adopted in the SARIMA models. The resulting predictive model is expressed as:

$$\hat{y} = \sum_{k=1}^{100} f_k(\text{Business Day,Holiday,Weekend,Moving Average,Moving Standard Deviation})$$

3 RESULTS AND DISCUSSION

Overall, the models generated using the seasonal ARIMA method exhibited heteroscedasticity, and this behavior was found to negatively affect their predictive performance (Hyndman, 2021). It was also observed that the Mean Absolute Percentage Error (MAPE) improved as data transformations were applied during the modeling process. This improvement is further supported by comparisons using the Ljung–Box test to assess residual autocorrelation, as well as the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC) (Shumway & Stoffer, 2011), and the maximum log-likelihood function (Log-Likelihood or Log-lik), which were employed to evaluate overall model fit (Fávero & Belfiore, 2017).

The evaluation criteria for the generated SARIMA models are summarized in Table 4.

Table 4

Results of Predictive Models Generated Using the Seasonal ARIMA Method

Model	Ljung–Box	MAPE	Log-lik	AIC	BIC	Heteroscedasticity
(i) Seasonal ARIMA model without transformation	28%	0.02	-1,384.387	2,778.775	2,797.00	23.19
(ii) Seasonal ARIMA model with Box–Cox transformation	6%	0.01	-52.506	115.012	133.23	90.74
(iii) Seasonal ARIMA model with logarithmic and Box–Cox transformations	79%	0.95	-129.794	269.589	287.81	60.77

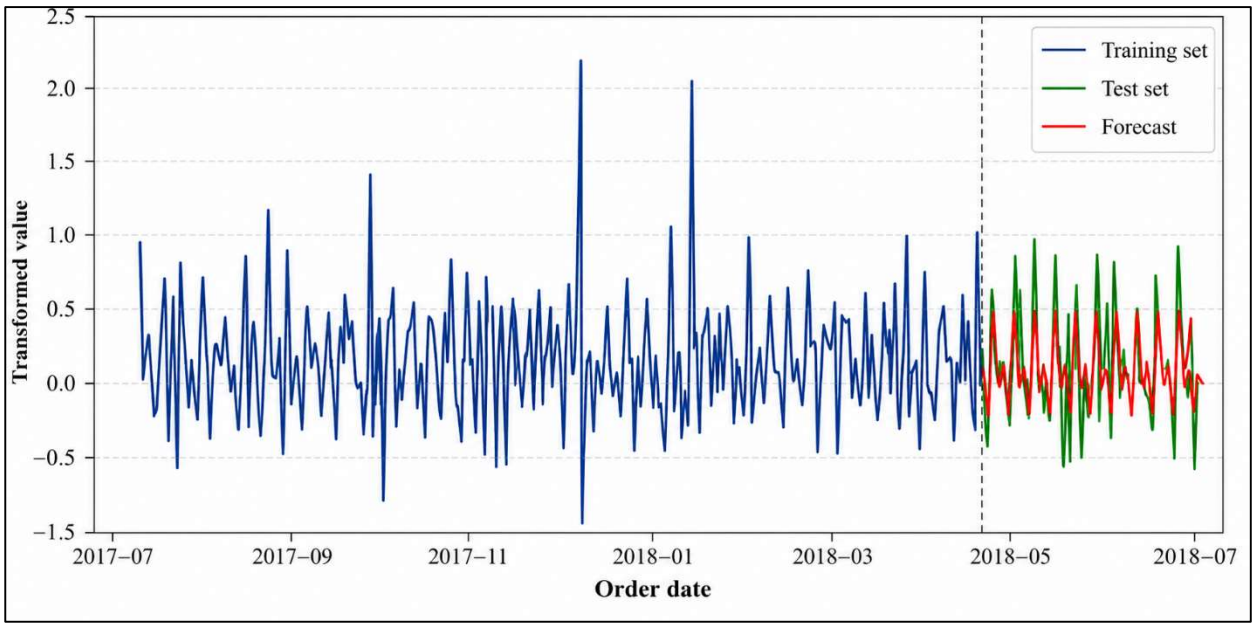
Source: Developed by the authors.

In general, all models exhibited some degree of heteroscedasticity, with model (i) showing the highest level of this behavior. Model (ii) presented a level of heteroscedasticity similar to that of model (iii), although model (iii) demonstrated visually superior adherence to the observed data, despite presenting higher AIC and BIC values than models (i) and (ii).

Figure 6 illustrates the results obtained with model (iii), considering the training, test, and forecast datasets. Adopting a MAPE threshold of approximately 14% as a benchmark for acceptable predictive performance (Pavlyshenko, 2019), model (ii) can be considered more accurate in quantitative terms. However, model (iii) exhibits superior graphical performance, suggesting a better visual fit to the underlying demand dynamics.

Figure 6

Seasonal ARIMA Model (iii) Generated After Box–Cox Transformation

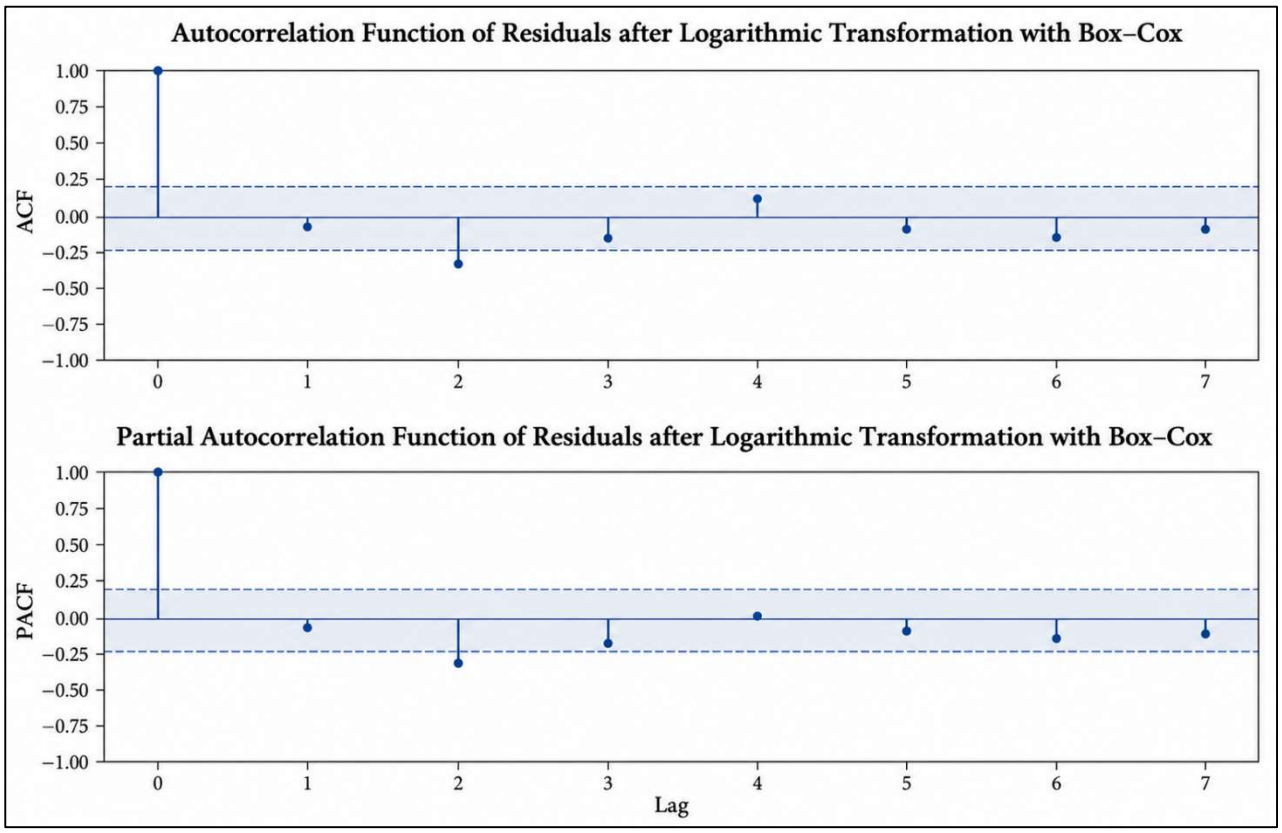


Source: Developed by the authors.

It was also observed that the results of the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) were close to zero across all three models, as shown in Figure 7. This behavior indicates low residual autocorrelation for model (iii), supporting the conclusion that the model exhibits acceptable predictive performance and that its residuals are stationary and randomly distributed.

Figure 7

ACF and PACF Plots of Residuals for the Seasonal ARIMA Model (iii)



Source: Developed by the authors.

The ensemble-based models exhibited very low goodness of fit when generated using either untransformed data or data subjected only to the Box–Cox transformation, as shown in Table 5. Empirically, the XGBoost model demonstrated substantially improved behavior when moving average and moving standard deviation features were incorporated. This enhancement enabled the algorithm to generate predictions from the training dataset that were visually much closer to the test dataset.

Furthermore, the use of tuned hyperparameters, as indicated in Table 3, also contributed to improved algorithm performance, resulting in a model with a low Mean Absolute Percentage Error (MAPE).

Table 5

Results of Predictive Models Generated Using the XGBoost Method

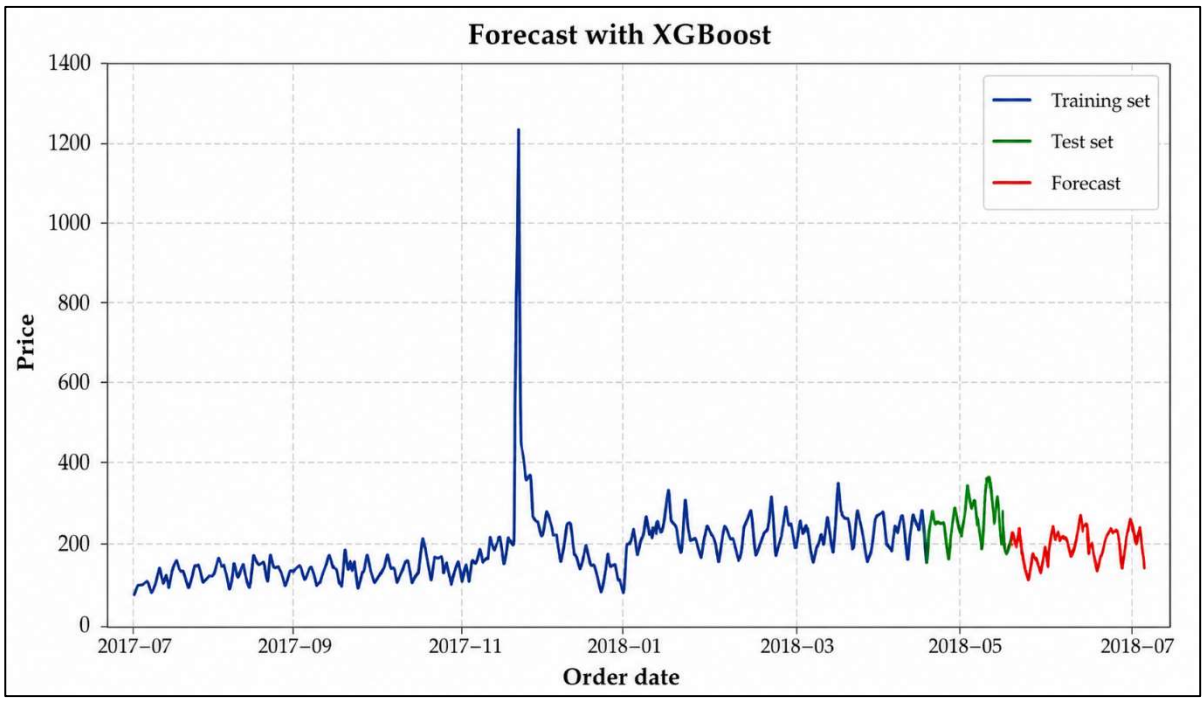
Model	MAPE	MSE
(i) Model using untransformed data	26%	1,472.955
(ii) Model using Box–Cox–transformed data	8%	0.235
(iii) Model using Box–Cox transformation with moving average and moving standard deviation	3%	93.721

Source: Developed by the authors.

Among the evaluated XGBoost models, the one that achieved performance closest to acceptable thresholds when jointly considering MAPE and MSE was model (ii). The behavior of this model is illustrated in Figure 8.

Figure 8

XGBoost Model (iii) Generated After Box–Cox Transformation with Moving Average and Moving Standard Deviation



Source: Developed by the authors.

Accordingly, the best-performing models were compared as shown in Table 6, where the superior forecasting quality of the XGBoost model is evident. The algorithm was able to capture the underlying data behavior and reproduce the demand forecast with an error rate of only 3%.

Table 6

Comparison Between Seasonal ARIMA Model (iii) and XGBoost Model (iii)

Modelo	**MAPE**
Seasonal ARIMA (iii) com transformações Logarítmica e Box–Cox	79%
XGBoost (iii) com transformação Box–Cox, média móvel e desvio-padrão móvel	**3%**

Source: Developed by the authors.

These results indicate a substantial performance advantage of the XGBoost-based approach over the Seasonal ARIMA model for the analyzed dataset.

The present findings are consistent with the recent demand-forecasting research that has found that machine learning can perform better when temporal and contextual information is incorporated in the model and the model is explicitly used. Haque et al. (2023) compared multivariate retail demand forecasting based on historical and temporal data and Jahin et al. (2024) showed that deep-learning and machine learning for forecasting supply chain demand with structured preprocessing and engineered temporal features are beneficial. In this study, the reduction in forecasting error after incorporating the 7-day moving average and moving standard deviation also suggests that the advantage of XGBoost is not only the ability of the learning algorithm to learn not only linear relationships and other temporal information but also the nonlinear aspects. Due to the different datasets, forecasting goals, model architectures, and evaluation process used in the different studies, direct numerical comparisons should be considered in a cautiously optimistic manner, but the overall trend is consistent with the growing evidence that machine learning-based models with high-level feature sets and a more robust approach in demand data models can outperform univariate time series models in high demand instances.

From a practical perspective, the forecasting improvement in this paper can be used to drive logistics decision making in the field by providing reliable daily freight predictions to management. This can help managers better predict the size of vehicle and carrier loads, plan loading and warehouse operations, schedule labor when needed, and reduce underutilization and last-minute transportation capacity limitations. In an e-commerce environment, the better freight demand forecast can also be used to make more decisions for regional dispatch and service decision making and reduce avoidable logistics costs and guarantee reliable delivery. The forecasting model should therefore be viewed as an operational tool and retrained as demand patterns, product mix, and operational setup change.

4 CONCLUSION

This study aimed to present the reader with the possibility of developing two predictive models based on different methodological approaches, which must be carefully compared in order to provide organizations with robust arguments to support decision-making. Based on the comparative analysis, it can be concluded that the characteristics of the data, despite being treated and applied to algorithms with distinct underlying assumptions, result in models with substantially different performance levels. This finding highlights the need for tailored techniques and modeling strategies to achieve satisfactory predictive results.

In addition to the traditional Box–Jenkins forecasting approach employed in this study, future research could extend the analysis by incorporating Exponential Smoothing models, thereby offering an alternative perspective within the class of classical forecasting methods. Given the data behavior and the heteroscedasticity observed in the ARIMA models developed herein, alternative approaches based on GARCH-type models may also be explored.

With respect to ensemble-based models, further investigation may involve exploring different combinations of hyperparameters to enhance model performance. Since these models are conceptually based on regression trees, the dataset prepared in this study could be expanded by incorporating additional datasets, increasing the number of nominal qualitative variables and, consequently, the predictive capacity of the models. Moreover, the ensemble approach adopted in this work could be extended by integrating techniques from natural language processing and sentiment analysis, thereby enriching the set of qualitative predictors used in demand forecasting.

Along the same line of investigation, the comparison matrix could be expanded by including other ensemble models, such as LightGBM, using the same datasets to ensure consistency and comparability of results.

Finally, the approach proposed in this study is applicable to any organization that employs e-commerce platforms as a sales channel and is not limited to the specific dataset analyzed. Through the results obtained from the generated models, this work seeks to contribute to organizational decision-making by highlighting alternative demand forecasting strategies.

Since identical datasets may exhibit different behaviors depending on the modeling technique employed, it is essential for decision-making processes to systematically compare multiple approaches in order to achieve optimal performance for a given business area or organizational process.

REFERENCES

Bruce, P., & Bruce, A. (2019). *Estatística prática para cientistas de dados: 50 conceitos essenciais*. Alta Books.

Brunheroto, P. H., Pepino, A. L. G., et al. (2022). Data analytics in fleet operations: A systematic literature review and workflow proposal. *Procedia CIRP, 107*, 1192–1197. https://doi.org/10.1016/j.procir.2022.05.130

Chapman, P., Clinton, J., Kerber, R., et al. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. SPSS.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

Fávero, L. P., & Belfiore, P. (2017). *Manual de análise de dados*. Elsevier.

Haque, M. S., Amin, M. S., & Miah, J. (2023). Retail demand forecasting: A comparative study for multivariate time series. arXiv preprint arXiv:2308.11939. https://arxiv.org/abs/2308.11939

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Prin ciples and practice* (3rd ed.). OTexts. https://otexts.com/fpp3/

Jahin, M. A., Shahriar, A., & Al Amin, M. (2024). MCDFN: Supply chain demand forecasting via an explainable multi-channel data fusion network model integrating CNN, LSTM, and GRU. arXiv preprint arXiv:2405.15598. https://arxiv.org/abs/2405.15598

McKinney, W. (2022). *Python para análise de dados*. Novatec.

Morettin, P. A., & Bussab, W. O. (2016). *Estatística básica* (8ª ed.). Saraiva.

Nagel, M., & dos Santos, C. P. (2017). A relação entre a satisfação com o gerenciamento de reclamações e as intenções de recompra: Detectando influências moderadoras em retail. *Brazilian Business Review, 14*(5). https://doi.org/10.15728/bbr.2017.14.5.4

Novack, R. A., Gibson, B., Coyle, J. J., & Suzuki, Y. (2020). *Transportation: A global supply chain perspective* (9th ed.). Cengage Learning.

Olist, & Sionek, A. (2018). *Brazilian e-commerce public dataset by Olist* [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/195341

Pavlyshenko, B. M. (2019). Machine-learning models for sales time series forecasting. *Data, 4*(1), 15. https://doi.org/10.3390/data4010015

Prim, A. L., & de Freitas, K. A. (2020). Frete barato e entrega atrasada: O dilema do nível de serviço versus custos. *Revista de Administração Contemporânea, 24*(6), 618–631. https://doi.org/10.1590/19827849rac2020190226

Pankratz, A. (1983). *Forecasting with univariate Box-Jenkins models: Concepts and cases*. John Wiley & Sons.

Saha, P., Gudhenia, N., Mitra, R., et al. (2022). Demand forecasting of a multinational retail company using deep learning frameworks. *IFAC-PapersOnLine*. https://doi.org/10.1016/j.ifacol.2022.09.425
Shumway, R. H., & Stoffer, D. S. (2011). *Time series analysis and its applications* (3rd ed.). Springer.

VanderPlas, J. (2016). *Python data science handbook*. O'Reilly Media. https://jakevdp.github.io/PythonDataScienceHandbook/