

Forecasting of Retail Produce Sales Based on XGBoost Algorithm



Yakup Turgut and Mustafa Erdem

Abstract Sales forecasting of vegetables and fruits imposes a challenging task for the retailers because the demand for them varies depending on several factors, such as temperature, season, holiday. Poor sales forecasting can cause too much cost for retailers since these products are unusable after deterioration. Also, people tend to consume these products freshly. This research aims to compare the forecasting performance of traditional statistical and new machine learning methods. We apply seasonal ARIMA to forecast daily sales of fruits and vegetables as a traditional method. As a machine learning algorithm, we apply LSTM and XGBoost algorithms. The results indicate that the XGBoost algorithm gives more accurate results than the other two methods.

Keywords Machine learning · Perishable foods · Sales forecasting · SARIMA · XGBoost · LSTM

Introduction

Determining the stock levels of perishable products is more complicated than nonperishable products due to their short shelf life and customer behavior toward them [2, 26]. Therefore, it is necessary to develop different stock policies (van Donselaar 2006) and supply chain (SC) strategies [7, 28] for these foods. Fruits and vegetables, which are largely studied in this study, are classified as perishable foods with particular storage characteristics. Managing the fruits and vegetables supply chain is complex and difficult because of their fluctuating demand pattern. Several factors,

Y. Turgut (✉)

Industrial Engineering Department, Management Faculty, Istanbul Technical University, Istanbul, Turkey

e-mail: turgut16@itu.edu.tr

M. Erdem

Mechatronics Engineering Department, ITU Institute of Science and Technology, Istanbul Technical University, Istanbul, Turkey

e-mail: erdemm@itu.edu.tr

such as weather conditions (Agnew and Tornes 1995), price changes [15], seasonality [19], are identified as the causes of variation in the demand of these foods. For Arunraj and Ahrens [5], factors can be classified as controllable factors, partially controllable factors, or uncontrollable factors. The authors state that the first includes price and product characteristics, the second includes substitution and cannibalization, and third includes events, weather, seasonality, and the number of customers. If organizations do not take into account these factors in stock management, they may suffer financial losses due to the waste of food and loss of customers. Optimized replenishment of these foods at the retail level is key to reducing the waste of these foods and increasing the efficiency of the fruits and vegetables supply chain. Optimal replenishment orders depend upon the sales forecasting [22]. Sales forecasting do not only enable retail managers to cope with stochastic demand, but also helps to maintain a competitive advantage in SC management.

Although the replenishment process for these foods usually performed manually by managers based on experience and the point of system (POS system) the necessity of analytical techniques has been understood lately, and more recent research has occurred in the demand forecasting of the perishable foods based on traditional statistical methods (causal models, time series and econometric methods) and machine learning (ML) methods. Sankaran [23] uses the Seasonal ARIMA model to forecast the daily demand for onion at a wholesale market and conclude that forecasting performance is satisfactory with an erratic demand. Raju et al. [21] investigate the factors causing the stochastic demand for perishable foods and examines the forecasting performance of linear and nonlinear forecasting methods. It concludes that temperature is the predominant factor that influences the demand, and nonlinear methods generate more accurate results than linear methods. Yang and Sutrisno (2018) bring a new perspective to forecast the demand for the bakery at franchise stores. The idea is to use sales occurring in the early morning hour to forecast the sales of the rest of the day. They also compare the forecasting performance of Feed Forward Neural Network (FFNN) and Regression analysis. They conclude that this approach is very promising to generate online-forecasting, and FFNN gives better results than regression analysis. Sridama and Siribut [24] propose a decision support system for demand forecasting of perishable foods to improve the inventory management of these foods. They analyze the forecasting performance of the following time series methods: Single exponential smoothing, Adaptive-response-rate single exponential smoothing, and Holt's two parameters linear exponential smoothing. They conclude that the Single exponential smoothing method gives better results than others. Huber and Stuckenschmidt (2017) propose a decision support system (DSS) based on the hierarchical clustering approach to obtain demand forecasts of perishable food at different organizational levels. They implemented the proposed DSS in the bakery chain of a company. The authors use multivariate ARIMA as a forecasting method. They conclude that the proposed approach gives acceptable results to increase the efficiency of the supply chain, and also decreases the computational time. The approach enables us to develop replenishment strategies based on product categories exhibiting similar demand patterns. Yang and Hu (2008) apply an ARIMA model to forecast the demand for cabbage.

Chen and Ou [13] propose an extended neural network model to forecast the daily demand for milk in a convenience store, and they compare the proposed model with an ARIMA model. The results indicate that the proposed model generates better results than the ARIMA model. Du et al. [14] develop an algorithm based on the Support vector machine and fuzzy theory to forecast the demand for perishable farm foods. They conclude that the proposed algorithm gives more promising results than radial basis function neural networks.

Although time series forecasting is generally superior to judgemental and econometric forecasting techniques for forecasting retail sales [16], it still lacks capturing the sudden changes in demand due to characterized nature of perishable foods. The forecasting performance of these methods may be improved by using hybridized versions of them [5]. Chen and Ou [10] propose a model which combines gray relational analysis and multilayer functional network model to forecast the sales of perishable food in a convenience store.

While these hybrid methods have provided considerable improvement in forecasting accuracy, much of it was not focusing especially on forecasting the fruits and vegetable sales at the retail levels. Furthermore, the forecasting accuracy is still required to improve by applying new algorithms. To improve the forecasting accuracy, this study focuses on the application of a gradient boosting ML algorithm, Extreme Gradient Boosting algorithm (XGBoost) due to its capabilities to handle the sparse data (Chen and Guestrin 2016), and computational efficiency, and it is popularity in ML competitions (Chen and He 2015). The results of the model are encouraging and show that XGBoost outperforms than classical SARIMA and LSTM models.

The rest of the paper is organized as follows: Sect. 2 outlines our method, which is used in this study and presents the description of data. Section 3 discusses the results of the performance of applied methods. Finally, Sect. 4 gives the conclusion and future research of the study.

Methodology

Description of Data

This study uses daily sales data of vegetables and fruits from a supermarket in Istanbul, Turkey, as a case study from January 2014 to December 2017. It would be more appropriate to conduct the forecasting on the product level due to the product-specific nature of demand pattern. Unfortunately, aggregate daily sales of vegetables and fruits are considered due to a lack of sales data on the product level. Figure 1 shows the daily sales data of vegetables and fruits in the time series plot. This time series plot shows that there is no obvious trend in data. There is an increasing trend in the sales of vegetables and fruits on some dates, such as the last week of the year.

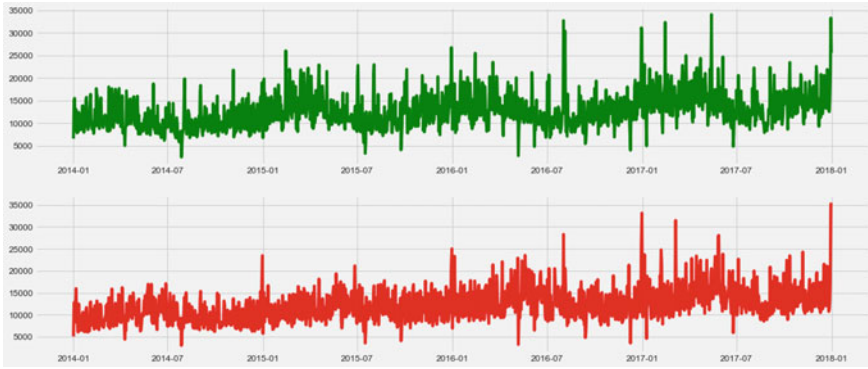


Fig. 1 The annual sales of vegetables and fruits (green indicates vegetables, red indicates fruits)

This cyclic pattern is repeated every year. This weekly or seasonal variation may be attributable to some causes such as the impact of weather or holidays.

Time Series

A time series is a sequential set of observations measured at successive times [8]. There are four key components of time series data, which should be analyzed before applying an algorithm: (1) Trend, (2) Seasonality, (3) Cyclical, and (4) Irregularity. Trend describes the general direction of observations over a long time. Seasonality explains the variation in the observations over a period of one year, usually caused by weather conditions, holidays, vacations, etc. Cyclical refers to the nonperiodic variations caused by circumstances, which occur in a repeating pattern. The duration of these variations lasts several years. Irregularity refers to the random variations in observations caused by unforeseeable reasons, such as earthquakes, floods, epidemic diseases, etc. Expectedly, these variations do not have a particular pattern. Time series plots may reveal these patterns or a combination of these patterns [3].

One of the most critical behaviors of a time series is stationarity. The stationarity of a time series data indicates its statistical behavior in time. When a time series exhibits this property, the statistical behavior of that series does not change in time. This means that it has a constant probability distribution. A time-series data must have this property because nonstationary data cannot be forecasted due to its unstable nature. If a time series data does not have stationarity behavior, it should be converted to a stationary form before performing any forecasting. There are two options to check the stationarity of a time series data: (1) Rolling statistics: plot the rolling average (moving average) and see how it varies with time, (2) ADF (Augmented Dickey-Fuller Test) provides a formal statistical test to detect the stationarity property. The null hypothesis claims that the time series is nonstationary.

Seasonal ARIMA (SARIMA) Model

ARIMA stands for the autoregressive moving average, and it has three parameters: (p, d, q). AR component is referred to the use of past values in the regression equation for the series Y. The parameter p indicates the number of lags used in the model. MA component represents the error of the model as a combination of previous error terms. The parameter q specifies the number of terms to include in the model. The parameter d represents the degree of differencing in the integrated component. Differencing a series involves simply subtracting its current and previous values d times. It is used to stabilize the data to satisfy the stationarity assumption [9]. SARIMA (Seasonal ARIMA) is an extension to ARIMA, which allows the direct modeling of seasonal behavior of data. SARIMA model is represented by the following notation: ARIMA (p, d, q) (P, D, Q)_s. The lower-case (p, d, q) is the same as the nonseasonal ARIMA model. The upper-case (P, D, Q) represents the seasonal parameters of the model. The subscripted letter s indicates the length of the period in each season. For example, in monthly data, s = 12. Let d and D are nonnegative integers. A SARIMA model general form is given in Eq. (1)

$$\emptyset_P(B)\Phi_P(B^S)Y_t = \theta_q(B)\Theta_Q(B^S)\varepsilon_t \quad (1)$$

$$\emptyset_P(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_P B^P \quad (2)$$

$$\Phi_P(B^S) = 1 - \Phi_S B^S - \Phi_{2S} B^{2S} - \dots - \Phi_{PS} B^{PS} \quad (3)$$

$$\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q \quad (4)$$

$$\Theta_Q(B^S) = 1 - \Theta_S(B^S) - \Theta_{2S}(B^{2S}) - \dots - \Theta_{QS}(B^{QS}) \quad (5)$$

where $\{X_t\}$ is the original series; $Y_t = (1 - B)^d(1 - B^s)^D X_t$ is differenced series to eliminate seasonality component; B is the lag operator; $\emptyset(B)$ and $\theta(B)$ are polynomials of order p and q, respectively; $\Phi(B^S)$ and $\Theta(B^S)$ are polynomial in B of degrees P and Q respectively; p is the order of nonseasonal autoregression; d is the number of regular differences; q is the order of nonseasonal moving average; P is the order of seasonal autoregression; D is the number of seasonal differences; Q is the order of seasonal moving average; and S is the length of the season [9]. The steps of the model are explained in the section of the application of the SARIMA model.

Long Short-Term Memory (LSTM)

Feedforward neural networks are not suitable for sequential data due to their fixed-size input/output. Therefore, they cannot be used to model memory. On the other hand, recurrent neural networks (RNN) are designed for capturing information from time-series data. In an RNN, thanks to the recurrence relation, each state is dependent on all previous computations. In theory, RNNs are capable of remembering information for long sequential data. However, in practice, this is not feasible due to the vanishing/exploding gradient problem. A similar problem is observed in deep feedforward networks. The source of this problem is the nature of RNN, which is using the same weight matrix to compute all the state updates. Even though the theory states that RNN can be used to learn long-term dependencies, due to vanishing/exploding gradient problems, they only seem to limit themselves to learn short-term dependencies.

LSTM solves the vanishing gradient problem and gives more accurate results compared to regular RNN. LSTM consists of three gates (forget, input, and output) and one cell state [17]. These are defined as follows:

$$f_t = \sigma(w_f h_{t-1} + w_f x_t) \quad (6)$$

$$i_t = \sigma(w_i h_{t-1} + w_i x_t) \quad (7)$$

$$o_t = \sigma(w_o h_{t-1} + w_o x_t) \quad (8)$$

$$\tilde{c}_t = \tanh(w_c h_{t-1} + w_c x_t) \quad (9)$$

$$c_t = (i_t * \tilde{c}_t) + (f_t * c_{t-1}) \quad (10)$$

$$h_t = o_t * \tanh(c_t) \quad (11)$$

Here f , i , o are forget, input and output gates respectively, c is cell state, h is a hidden state and x is the input. The complete structure of the LSTM is illustrated in Fig. 2.

LSTM can hold a combination of different information blocks at each time step. The main advantage of LSTM comes from the cell state. Cell state provides the possibility of explicitly information writing or removing. This cell state can only be altered by the gates which are responsible for letting the information pass through.

From previous studies, we know that the convolution operation works well to extract features as local input patches. This allows modular and efficient data representations. In our forecasting problem, we accept the time as a spatial dimension and process the data by applying 1D convolution operations to extract subsequences

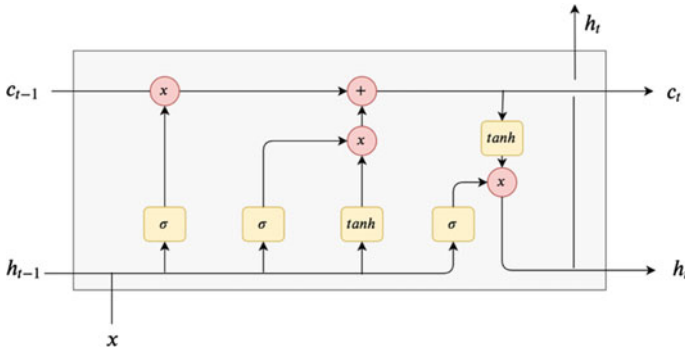


Fig. 2 LSTM structure

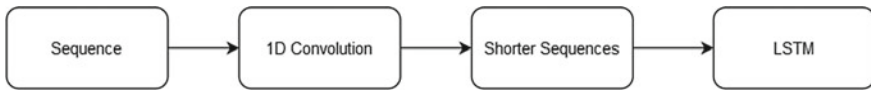


Fig. 3 Data feed processing

from the sequence. This allows us to recognize local patterns since the same transformation is applied to every patch. However, it is not possible to get reasonable results in forecasting problems just by using convolution operation. Since the convolution operation processes the input patches independently, it is not sensitive to the order of the timestep, unlike LSTM. One way to use the advantageous feature of convolution is to use it as a preprocessing step before LSTM. Figure 3 illustrates our data processing step combined with LSTM.

XGBoost (Extreme Gradient Boosting)

XGBoost is a scalable end-to-end novel machine learning algorithm based on tree learning gradient boosting that can handle sparse data in a highly efficient way. The mathematical formulation of the model is described by Chen and Guestrin (2016) as follows: Eq. (12) describes the objective function (also called loss function) which should be minimized at iteration t where $\hat{y}_i^{(t)}$ indicates the prediction of i th instance at i th iteration; $\Omega(f_i)$ describes the regularization term, which helps to model to avoid over-fitting results.

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_i(x_i)) + \Omega(f_i) \tag{12}$$

At each iteration, a new function f_t is added, which provides the best improvement for the model. It is not possible to solve Eq. (12) by using traditional optimization methods. So second-order Taylor approximation is used to get a solvable form by traditional methods and Eq. (13) is obtained. Where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial_{\hat{y}^{(t-1)}}^2 l(y_i, \hat{y}^{(t-1)})$ are first and second-order terms respectively at iteration t . When we remove the constant term in Eq. (13), we obtain Eq. (14).

$$\mathcal{L}^{(t)} = \sum_{i=1}^n \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (13)$$

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (14)$$

If regularization term $\Omega(f_t)$ is replaced by $\gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$, then objective function takes its final form as follows:

$$\begin{aligned} \tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\ &= \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \end{aligned} \quad (15)$$

The optimal value of weights w_j at leaf j is obtained by using Eq. (16) and the optimal value is calculated by Eq. (17). This equation can be understood as a scoring function for a tree structure q . This score is similar to the impurity score for evaluating decision trees.

$$w_j^* = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (16)$$

$$\tilde{\mathcal{L}}^{(t)}(q) = - \frac{1}{2} \sum_{j=1}^T \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \quad (17)$$

It is a real challenge to enumerate all possible trees. The authors propose Eq. (18) to calculate the score of each tree structure for evaluating each one to select the best split where I_L and I_R denotes the instances set of left and right nodes after the split; $I = I_L \cup I_R$.

$$\mathcal{L}_{split} = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\sum_{i \in I} g_i}{\sum_{i \in I} h_i + \lambda} \right] \quad (18)$$

Application of SARIMA Model

The steps involved in building a SARIMA model are as follows:

1. **Identification of Model Parameters:** the initial step of the SARIMA is to determine the values of parameters. The autocorrelation, partial autocorrelation, and Augmented Dickey-Fuller (ADF) test results for vegetables and fruits are shown in Fig. 4. The plots indicate that there is no regular or seasonal trend. ADF results also prove that both data satisfy stationarity property at 0.05. The

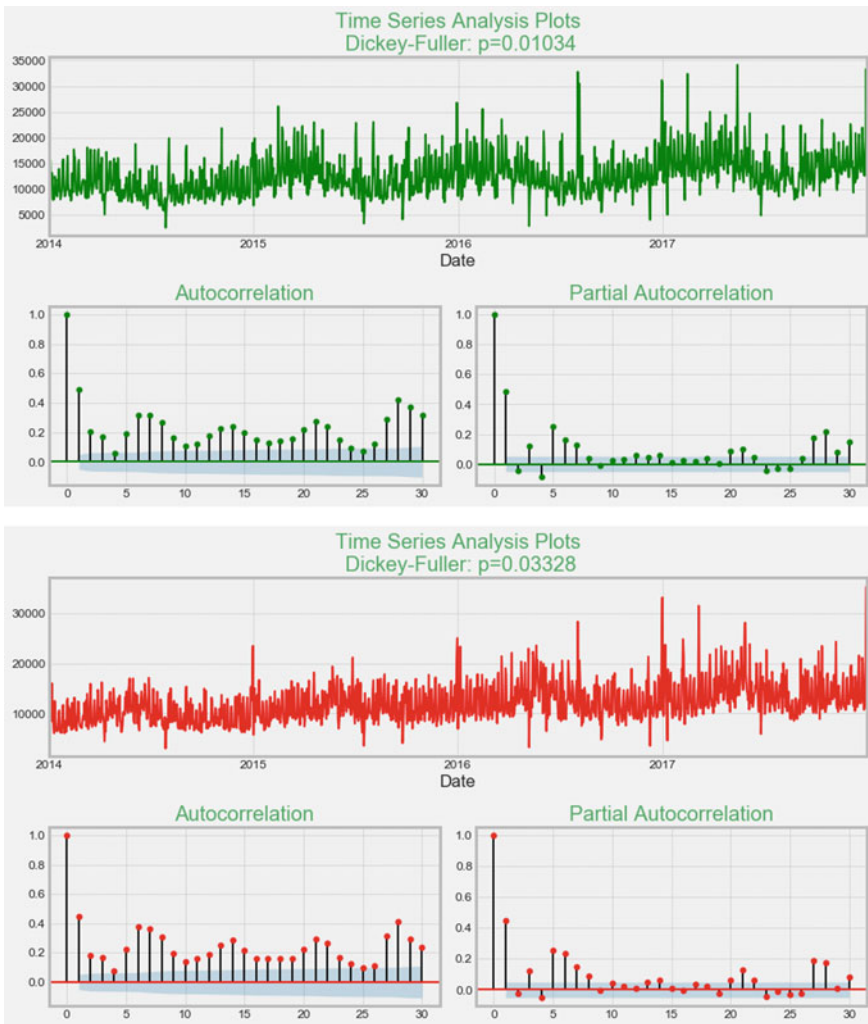


Fig. 4 Seasonal and stationary analysis of fruits and vegetables respectively

best values of parameters should be identified. To determine the parameters of the model, Akaike’s Information Criterion (AIC) is commonly utilized. It is calculated as follows:

$$AIC(p) = n \ln\left(\frac{RSS}{n}\right) + 2K \tag{19}$$

where n is the number of observations, and RSS is the residual sums of squares. The parameters providing the minimum AIC value will be set as model parameters. Another approach for determining appropriate parameters of the model is to analyze (ACF) and (PACF) plots.

2. **Estimation of Model Parameters:** A grid search approach is employed for determining the best forecasting model in this study. ARIMA (p, d, q) (P, D, Q) m model requires six parameters: p, d, q, P, D, and Q. The value of m is set as 12 because used data are monthly with a period of 12. The AIC values of evaluated models are shown in Table 1. According to Table 1, SARIMA (1, 1, 1) × (1, 0, 1)₁₂ shows the lowest AIC value. Thus, this model should be considered as the best forecasting model. According to Table 1, the AIC value of SARIMA (1, 1, 2) × (0, 0, 1)₁₂ is the lowest.
3. **Diagnostic of Model:** In this step, the statistical importance of the selected SARIMA model is determined. The statistical test results of the SARIMA (1, 1, 1) × (0, 0, 1, 1) model are shown in Table 2. The second column indicates the weight of the coefficients. Since all values of P > |z| are less than

Table 1 AIC Values of SARIMA models

(p, d, q)	(P, D, Q, m)	AIC values
(0, 1, 0)	(0, 0, 0, 1)	21,115.035
(1, 1, 0)	(0, 0, 0, 1)	21,029.448
(0, 1, 1)	(0, 0, 0, 1)	20,763.500
(1, 1, 1)	(0, 0, 0, 1)	20,683.540
(1, 1, 2)	(0, 0, 0, 1)	20,655.100
(2, 1, 3)	(0, 0, 0, 1)	20,659.035
(0, 1, 2)	(0, 0, 0, 1)	20,663.134
(1, 1, 3)	(0, 0, 0, 1)	20,656.342

Table 2 Results of the diagnostics test of the SARIMA (1, 1, 2) × (0, 0, 1, 1) model

	coef	std err	z	P > z	[0.025	0.975]
ar.L1	-0.2861	0.071	-4.027	0.000	-0.425	-0.147
ma.L1	-0.3550	0.064	-5.550	0.000	-0.480	-0.230
ma.L2	-0.5916	0.060	-9.943	0.000	-0.708	-0.475
sigma2	9.581e+06	3.5e-05	2.74e+11	0.000	9.58e+06	9.58e+06

0.05 (significance level), we can conclude that all coefficients are statistically significant.

- 4. **Forecasting:** The model with estimated parameters is used to make forecasts. Data from January 2014 to December 2016 are used as a training set, and the remaining data is used as test data. We forecast the periods between January and December 2017 daily. Figure 5 shows the model forecast values and the actual value curve. Mean absolute percentage error (MAPE) on the test data set is 24.57 and 24.43% for fruits and vegetables respectively.

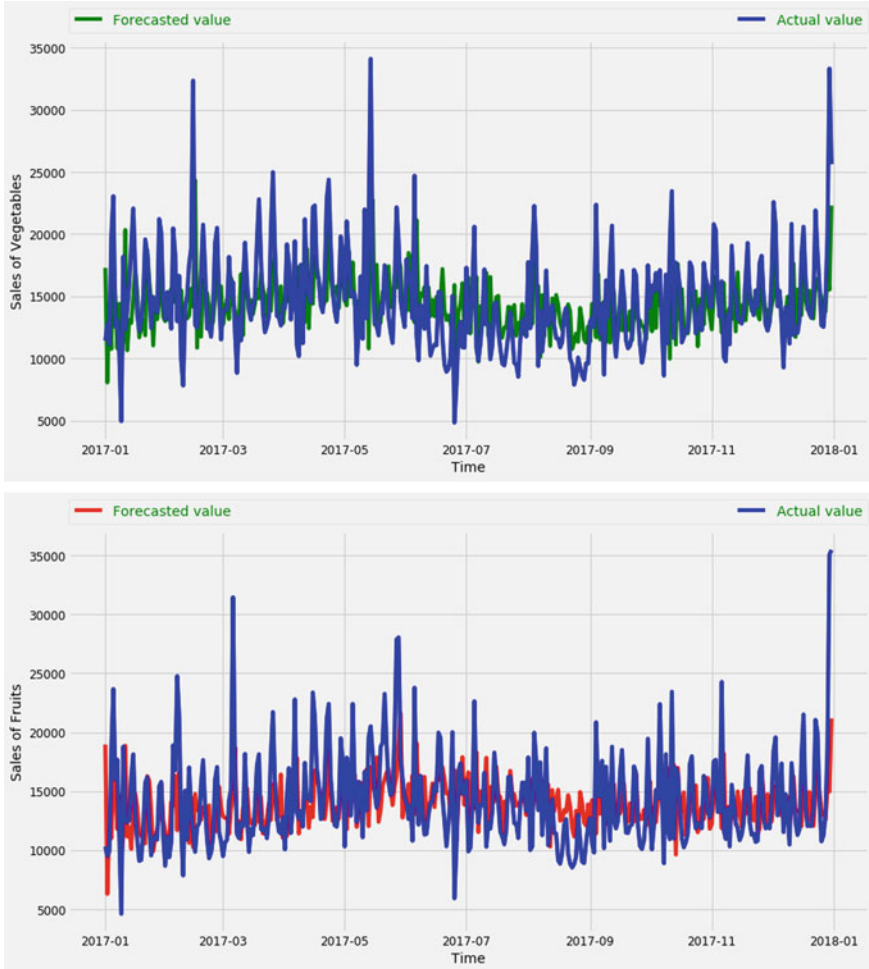


Fig. 5 Forecast value and actual value fitting curve

Application of LSTM Model

The algorithm is implemented in Python with the Pytorch library.

1. **Model Parameters:** Our network consists of 1 hidden layer with 9 neurons. The output activation function is linear. The input to the network is our 17 handcrafted features. Mean squared error loss is selected for loss calculation. A popular ADAM optimization algorithm was selected to optimize network weight values. Hyperparameters of the model and optimization algorithm are selected based on trial-and-error.
2. **Data Processing:** Total data split into two parts as follows: first three-year data as training and the last year as testing. The windows size of the 1D convolution operation was selected as 12. Before feeding the inputs into the network, additional scaling/normalization processes applied as in regular feedforward neural networks to make the learning step more stable.
3. **Forecasting:** Neural network trained for 2000 epochs. No regularization method was applied to the implementation.

Training loss for vegetable sales data is illustrated in Fig. 6. MAPE on test data is 24.6, which is inside an acceptable boundary compared to previous studies. Figure 8 shows the prediction of vegetable data. To forecast fruit data, the same LSTM network is retrained by using the same hyperparameters. Figures 7 and 8 show the results for fruit data. In this case, MAPE is 22.5.

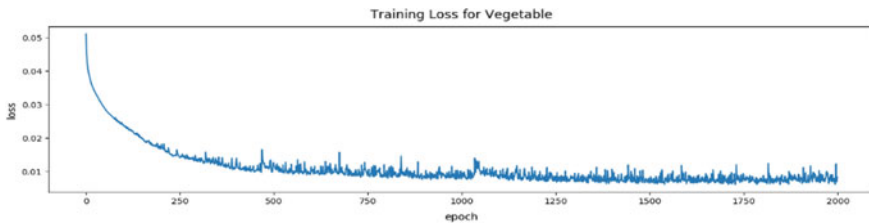


Fig. 6 LSTM training loss for vegetable

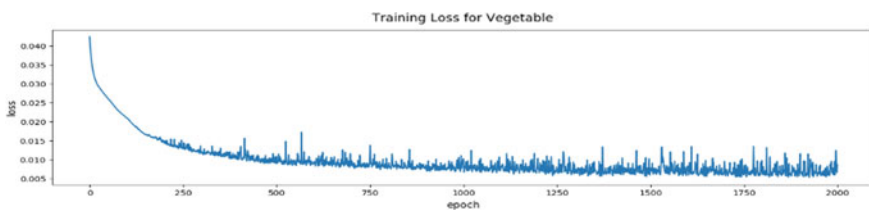


Fig. 7 LSTM training loss for fruit



Fig. 8 Forecast value and actual value fitting curve for vegetable (top) and fruit (bottom)

Application of XGBoost (Extreme Gradient Boosting)

The algorithm is implemented in Python using open-source XGBoost libraries. An open-source framework provides a fast and easy implementation of the algorithm. The following steps are involved in the implementation of the algorithm.

1. **Feature extraction and selection:** We extract 17 features after reading the papers in the area and obtain the relative importance of these features, as shown in Fig. 9. The temperature and dollar rate are the most important features. The categorical variables are converted into the numerical form by using one-hot encoding.
2. **Parameter Tuning:** The XGBoost algorithm has too many parameters, and the values of these parameters highly affect the prediction performance of the model. So it is required to perform hyper-parameter tuning operation to obtain the more appropriate model. It is the most time-consuming part of the implementation of the algorithm. The custom grid search approach is used to find the parameter values. The booster parameters related to the learning task and model complexity

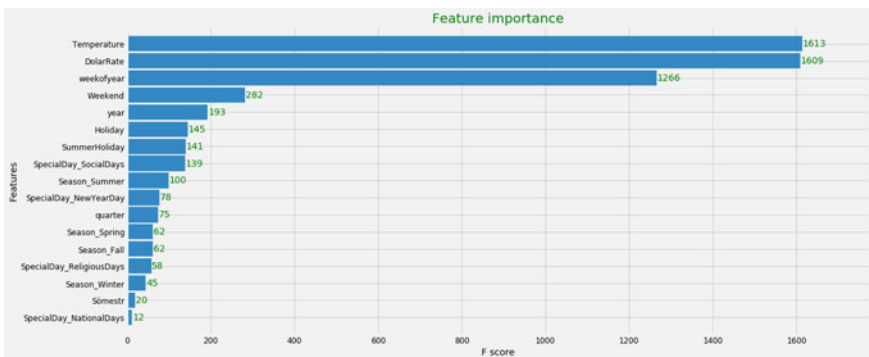


Fig. 9 Feature importance plot

Table 3 Hyper-parameter tuning values

Parameter	Parameter Values	Best Value
max_depth	[0, 10]	1
n_estimators	{100, 200}	100
subsample	[0, 1]	0.16
Colsample_bytree	[0, 1]	1
gamma	{0, 1}	0
reg_alpha	[0, 1]	0
learning_rate	[0.01, 0.1]	0.1
min_child_weight	{1, 3, 5}	1

are tuned by using the given values in Table 3. These values are determined based on the expert's suggestions. Column 2 indicates the used values in the grid search process, and column 3 represents the best value of tuned parameters. Default values are used for the remaining parameters.

3. **Forecasting:** After selecting the best model parameters, we can use the model to make forecasting. Figure 10 shows the model forecast values and the actual value curve. The forecasting results indicate that XGBoost can make better predictions than SARIMA and LSTM (see Table 4).

Summary and Outlook

In this study, we focused on the applicability of the XGBoost algorithm to forecast the daily sales of perishable foods. A specific focus of the study was directed toward two special perishable food categories: vegetables and fruits. The test cases were performed in the retail market. The results show that XGBoost yields better predictions compared to SARIMA and LSTM. The outcomes of this study can give several useful insights for managers, such as the development of stock policy, investments in SC.

Although we obtained meaningful results, some limitations and future research avenues may be emphasized. First, the advantage of using ML techniques highly depends on data availability. Since all factors are not included in the model due to a lack of data availability, we could not fully exploit the advantages of ML methods. Second, the controllable factors, such as product characteristic, promotions are not used in this study. With the availability of all these features, the most relevant features can be identified.

These limitations imply many possible extensions of this study in the future. For example, product-based or location-based estimation can be employed to understand the impact of these factors on sales. Besides, it is possible to investigate different ML algorithms, as well as new methods of combining these algorithms while considering the respective accuracy of each.

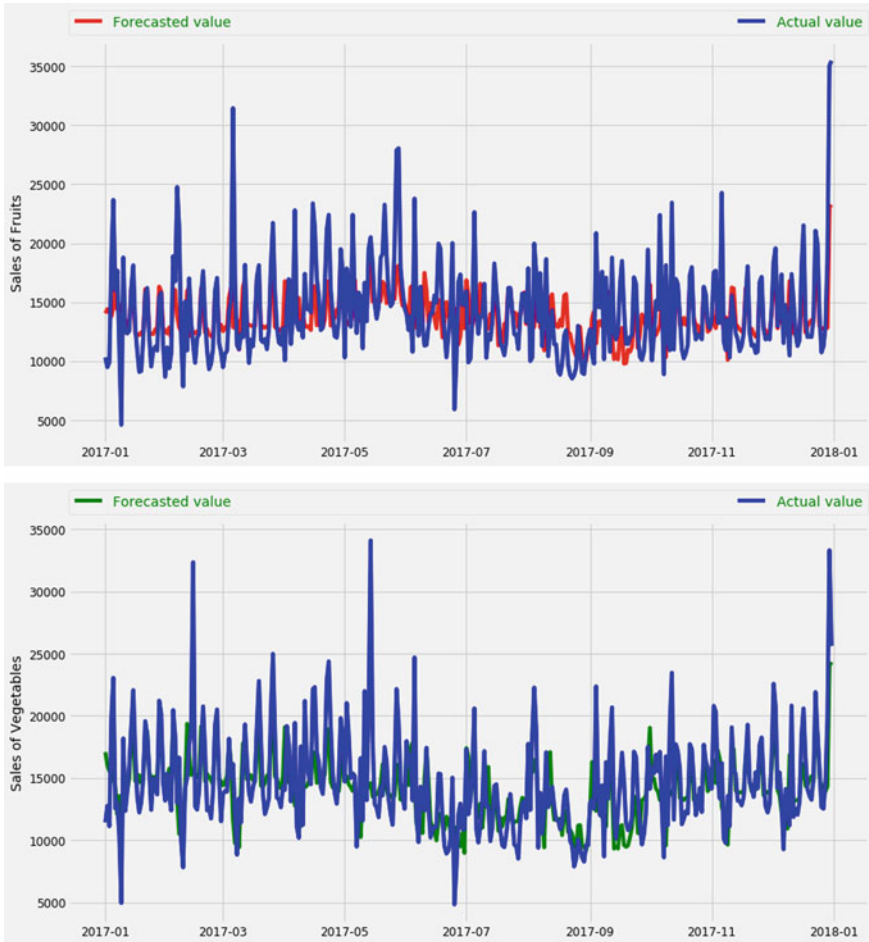


Fig. 10 Forecast value and actual value fitting curve

Table 4 Performance comparison of forecasting methods based on MAPE

Model	MAPE for fruits (%)	MAPE for vegetables (%)
SARIMA	24.43	24.57
LSTM	22.5	24.6
XGBoost	16.57	16.43

References

1. Aburto L, Weber R (2007) Improved supply chain management based on hybrid demand forecasts. *Appl Soft Comput* 7(1):136–144

2. Adebajo D, Mann R (2000) Identifying problems in forecasting consumer demand in the fast-moving consumer goods sector. *Benchmarking Int J*
3. Adhikari R, Agrawal RK (2013) An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613*
4. Agnew MD, Thornes JE (1995) The weather sensitivity of the UK food retail and distribution industry. *Meteorol Appl* 2(2):137–147
5. Arunraj NS, Ahrens D (2015) A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *Int J Prod Econ* 170:321–335
6. Ashagidigbi WM, Adebayo AS, Salau SA (2019) Analysis of the demand for fruits and vegetables among households in Nigeria. *Sci Lett* 7(2):45–51
7. Blackburn J, Scudder G (2009) Supply chain strategies for perishable products: the case of fresh produce. *Prod Oper Manag* 18(2):129–137
8. Brockwell PJ, Davis RA, Fienberg SE (1991) *Time series: theory and methods: theory and methods*. Springer Science & Business Media
9. Brockwell PJ, Davis RA (2016) *Introduction to time series and forecasting*. Springer
10. Chen FL, Ou TY (2009) Gray relation analysis and multilayer functional link network sales forecasting model for perishable food in convenience store. *Exp Syst Appl* 36(3):7054–7063
11. Chen T, He T (2015) Higgs boson discovery with boosted trees. In: *NIPS 2014 workshop on high-energy physics and machine learning*, pp 69–80
12. 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
13. Chen FL, Ou TY (2008) A neural-network-based forecasting method for ordering perishable food in convenience stores. In: *2008 fourth international conference on natural computation*, vol 2. *IEEE*, pp 250–254
14. Du XF, Leung SC, Zhang JL, Lai KK (2013) Demand forecasting of perishable farm products using support vector machine. *Int J Syst Sci* 44(3):556–567
15. Durham C, Eales J (2010) Demand elasticities for fresh fruit at the retail level. *Appl Econ* 42(11):1345–1354
16. Geurts MD, Kelly JP (1986) Forecasting retail sales using alternative models. *Int J Forecast* 2(3):261–272
17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
18. Huber J, Gossman A, Stuckenschmidt H (2017) Cluster-based hierarchical demand forecasting for perishable goods. *Expert Syst Appl* 76:140–151
19. Locke E, Coronado GD, Thompson B, Kuniyuki A (2009) Seasonal variation in fruit and vegetable consumption in a rural agricultural community. *J Am Diet Assoc* 109(1):45–51
20. Paam P, Berretta R, Heydar M, Middleton RH, García-Flores R, Juliano P (2016) Planning models to optimize the agri-fresh food supply chain for loss minimization: a review. *Ref Mod Food Sci* 19–54
21. Raju Y, Kang PS, Moroz A, Clement R, Hopwell A, Duffy AP (2016) Investigating the demand for short-shelf-life food products for SME wholesalers
22. Reddy AS, Chakradhar P, Santosh T (2018) Demand forecasting and demand supply management of vegetables in India: a review and prospect. *Int J Comput Technol* 17(1):7170–7178
23. Sankaran S (2014) Demand forecasting of fresh vegetable product by seasonal ARIMA model. *Int J Oper Res* 20(3):315–330
24. Sridama P, Siribut C (2018) Decision support system for customer demand forecasting and inventory management of perishable goods. *J Adv Manag Sci* 6(1)
25. van Donselaar K, Van Woensel T, Broekmeulen RACM, Fransoo J (2006) Inventory control of perishables in supermarkets. *Int J Prod Econ* 104(2):462–472
26. Van Woensel T, Van Donselaar K, Broekmeulen R, Fransoo J (2007). Consumer responses to shelf out-of-stocks of perishable products. *Int J Phys Distrib Logistics Manag*
27. Yang CL, Sutrisno H (2018) Short-term sales forecast of perishable goods for franchise business. In *2018 10th international conference on knowledge and smart technology (KST)*. *IEEE*, pp 101–105

28. Yang S, Xiao Y, Kuo YH (2017) The supply chain design for perishable food with stochastic demand. *Sustainability* 9(7):1195
29. Yang HX, Hu J (2013) Forecast of fresh agricultural products demand based on the ARIMA model. *Guangdong Agric Sci* 11:52