2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) | 978-1-6654-3521-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSES52305.2021.9633975

Quotidian Sales Forecasting using Machine Learning

Spuritha M[1], Cheruku Sai Kashyap[2], Tejas Rakesh Nambiar[2], Dendukuri Ravi Kiran[2], N.Srinivasa Rao[2] and G.Pradeep Reddy[3]

[1]Dept. of IT, G.Narayanamma Institute of Technology and Science, Hyderabad, Telangana, INDIA
[2]Dept. of ECE, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, Telangana, INDIA
[3]School of Electronics Engineering, VIT-AP University, Amaravati, Andhra Pradesh, INDIA
E-mail : m.spurithareddy@gmail.com, chkashyap1999@gmail.com, tejasnambiar32@gmail.com, davidhume3690@gmail.com, nsrao.21@gmail.com, pradeep.19phd7025@vitap.ac.in

Abstract :- Retailers have been experiencing a drop in their sales due to the rise of E-commerce facilities. This poses a problem where the retail stores need to efficiently manage and price their products to increase their sales. Hence the need for efficient sales prediction and dynamic pricing arises. A forecasting model which can effectively predict the sales of a retail store will help retailers compete in the market. With this intent, the paper proposes a model based on XGBoost whose learners are fitted to the store- product subsets with optimum parameters to increase the overall performance of sales prediction. The proposed model predicted sales for 10 stores with 50 products, with average MAPE, RMSE and R^2 values of 11.98 %, 6.63 and 0.76 respectively. In addition, dynamic pricing is applied to the forecasted results which specifies the optimum price of a product based on its demand.

Keywords—Dynamic Pricing, Machine Learning, Retail Stores, Sales Forecasting, XGBoost.

I. INTRODUCTION

Retail stores find it difficult to compete in the market with E-commerce services due to the lack of prediction analysis of crucial factors such as sales and product demand. With inadequate consideration of the future demands in the market, refilling of stocks into their stores is not done efficiently [1]. Due to this, cash flow and maintenance of the products become a hassle. Furthermore, there could be a potential problem of insufficient quantity of products which in turn reduces customer satisfaction and makes it difficult to maintain steady sales. Customers can find most things online, thus retailers with little awareness of demand will eventually fail.

Possible solutions to solve this problem can be inferred from the E-commerce services that use sales forecasting to maintain track of their product sales by forecasting based on their aggregate data using a precise machine learning model [2] [3]. This provides them with information on each product's demand and sales allowing them to compete with E-commerce services.

Sales forecasting for retail stores requires consideration of factors such as actual demand with respect to the sales, seasonality, financial indicators, and market mix being among a few [4] [5]. But for real-time accurate forecasting in retail stores, consideration of weather and holidays is also required which the forecasting algorithms used by E-commerce sites ignore due to them being a virtual marketplace.

With retail stores and customers sharing an interdependent relationship, it becomes imperative to employ demand forecasting mechanisms that provide accurate results [6]. Hence, in order to implement such accurate forecasting for retail stores, robust algorithms are needed that work well under dynamic conditions [7]. Though sales and demand forecasting provides valuable insights about the store's inventory, it still doesn't solve the issues many stores face due to the lack of a proper pricing system.

E-commerce services use dynamic pricing that enables one to change the prices of products as per the fluctuations in demand. It enables strategic pricing of products using pricing algorithms based on an hourly basis, the initial cost of the product, supplementary attributes, and a few others [8] [9]. However, the algorithms don't consider the external environment factors. This concept proves to be inefficient for the retail stores as they require dynamic pricing on a daily basis, and need to consider inputs from the external environment.

To overcome these challenges an extreme gradient boosting (XGBoost) algorithm based solution is proposed, which considers the external environments such as weather and holidays, and builds a regression tree. With the help of gradient boosting, an effective sales forecasting technique is proposed. A new dynamic pricing algorithm is also proposed which uses the output of the sales forecasting algorithm to assist the retailers in pricing their products on a daily basis.

The rest of the paper is structured as follows, Section II presents the proposed method, Section III contains the implementation, while results and discussion are detailed in Section IV, and Section V discusses the overall conclusion. Finally, the future scope is explored in Section VI.

II. Proposed Method

A machine learning-based retail store forecasting model is designed where data was collected from ten retail stores for 3 years (i.e., 2017-2019) in a locality. The weather stack API is used to obtain weather insights of stores for the collected period in order to consider the weather scenario.

Exploratory Data Analysis (EDA) is performed on the dataset and various plots are drawn to get an insight into the sales of the products in different stores over years. Feature engineering is performed on the dataset by which relevant sub- features are determined from the original features.

Table I. Sample Records of the Dataset

S. No	Date	Store ID	Product ID	Sales	Price (in Rs.)	Weather
1	2017-01-01	1	1	110	300	Windy
2	2017-01-02	1	1	70	270	Windy
3	2017-01-03	1	1	82	282	Rainy
4	2017-01-04	1	1	61	297	Rainy
5	2017-01-05	1	1	75	300	Sunny
6	2017-01-06	1	1	101	298	Sunny
7	2017-01-07	1	1	89	299	Windy
8	2017-01-08	1	1	81	302	Sunny
9	2017-01-09	1	1	74	296	Rainy
10	2017-01-10	1	1	66	299	Windy

This approach aids in constructing a stronger machine learning model by increasing the dimensionality of the dataset. Multivariate analysis is done on the features and a correlation matrix is plotted to get the statistical measure of the relationship between the variables. The dataset is divided into subsets based on store item combinations followed by feature selection. The best features from a group of predictor features are chosen via feature selection. The dataset is split into training and testing partitions. Following the splitting process, extreme gradient boosting (XGBoost) is applied to build the prediction model. Choosing the right collection of hyperparameters for the machine learning algorithm is crucial since they influence the model's fit.

After being trained, the model with the tuned parameters is used to forecast the test set's target values. The predicted and actual values for the test set are compared and inferences are drawn in the form of various metrics. The model is evaluated on various metrics which include MAPE, RMSE, and R^2 score.

After obtaining a forecast of sales across the stores, dynamic pricing of the products is done. The predicted sales of the product are used to segregate the product into three tiers i.e. (a) Most in-demand (b) Ideal demand (c) Low demand. The product's demand is determined by its previous sales history at the retail store.

Based on the tier into which the product demand is assigned, dynamic pricing is done based on a predefined formula for the respective tiers.

III. Implementation

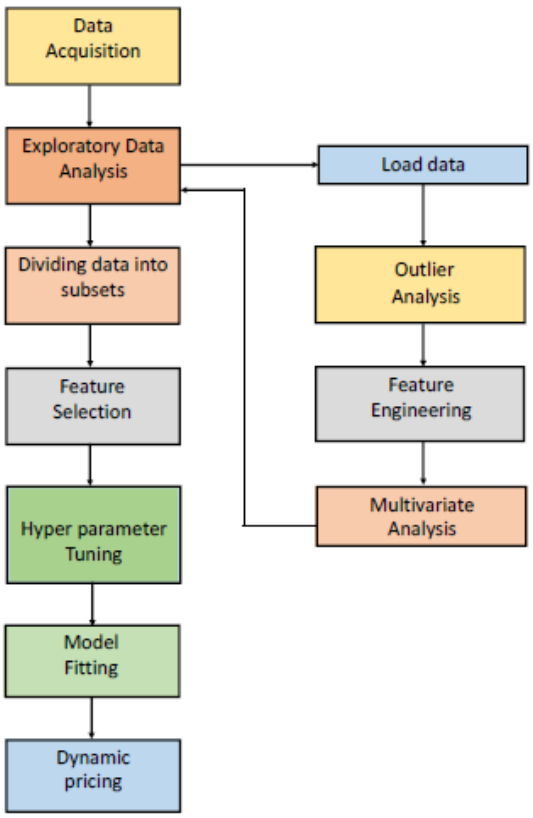

Fig. 1. Implementation of the proposed system

The date, store id, product id, number of products sold, and the price on that specific date are present in the dataset, which was gathered from retail stores. The weather stack API is used to add a new column that describes the weather on that particular day to the dataset.

The implementation of the proposed system is shown in Fig. 1 and is explained through the following subsections.

A. Exploratory Data Analysis

EDA is the process of grooming and modifying the dataset in a way that enhances the performance of the proposed machine learning model. It is an ensemble of various sub-processes.

1) Load data: The dataset is stored in the form of a CSV file. The dataset consists of 6 columns and 547,500 rows, for the years 2017-2019. The columns in the dataset include store id, product id, sales, price data, and weather description for the respective dates as shown in Table I.

2) Outlier analysis: The outlier distribution of the data is determined by using Box and whisker plot as shown in Fig. 2. By assuming a Z-score threshold of 3.5 and detecting the outliers, it is observed that store 7 had the highest percentage of outliers (0.47%) and store 3 had the lowest percentage of outliers (0.37%) in their respective sales data.

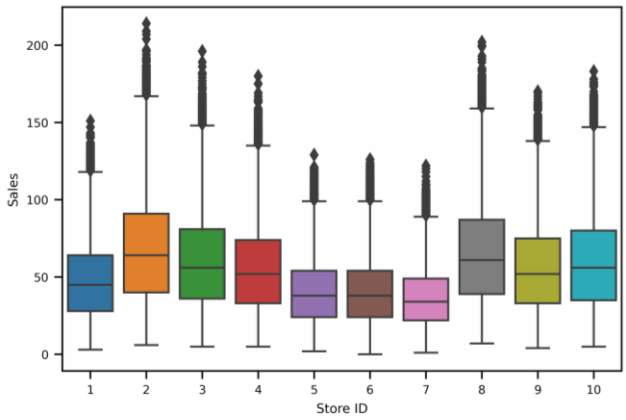

Fig. 2. Box and whisker plot for sales-data distribution in individual stores.

3) Feature engineering: Feature engineering is used to extract additional features from the dataset which helps in attaining a better fit for the model [10]. In the proposed approach, date-related features are created, such as day, month, year, weekend, start_of_month, is_month_end, is_holiday. Additional time-series features are introduced after the dataset is divided based on store and product combinations. Lag features are added through which the given sales value at time t-1, the sales value at t+1 can be predicted. These features are created using the shift() method. Lag values of sales up to seven days are constructed. The mean, minimum, maximum, and standard deviation values of a rolling window with size seven for sales are added to the sub-dataset respectively using the rolling() function.

4) Multivariate analysis: Multivariate analysis is carried out by constructing a correlation matrix [11]. Fig. 3 represents the heat map representation of the matrix. The columns with lesser variance may not serve well towards model building and hence must be omitted. From Fig. 3 it can be deduced that is_month_end and day columns have the least correlation, whereas rolling_mean and rolling_max show the most correlation with the target sales column [12].

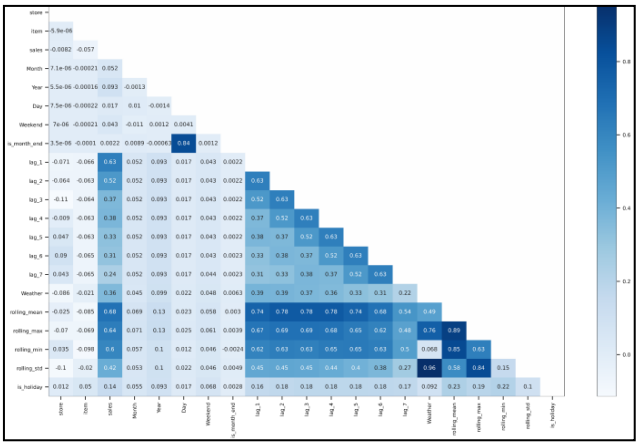

Fig. 3. Correlation matrix.

The is_month_end and day columns are dropped as they have less correlation with the sales variable. The initial dataset is divided into 500 subsets, each with an average of 1095 elements based on store and product.

B. Feature Selection

The best features from a given set of predictor features are chosen via feature selection. It helps to increase the performance of the machine learning model by decreasing over- fitting, increasing accuracy, and reducing the computation time [13] [14]. The proposed approach uses the SelectKBest() function from the sklearn library to perform feature selection. A total of 15 features have been selected from the initial set of features. The features selected are lags (1 to 7), rolling window features (mean, min, max, std), is_holiday, month, weekend and year.

C. Hyperparameter Tuning and Model Fitting

After selecting the best features for model building, the XGBoost model is to be fitted. Ideal values for the model parameters are of utmost priority as they decide the performance of the model to a great extent. One way to approach the problem of assigning ideal values to the hyperparameters is, to heuristically set the hyperparameters. In most cases, there might be another set of hyperparameters that may result in better model

fitting than what is set by intuition [15]. To overcome such setbacks, tuning is introduced. Two of the most common tuning strategies include GridSearchCV and Randomized Search CV. Grid Search CV works by training the model on a subset of the hyperparameter values provided whereas Randomized Search CV picks subsets at random, saving more time than Grid Search CV [16] [17]. The proposed approach uses Randomized Search CV to tune the parameters and the model with tuned parameters is then fitted on the train set and used to predict the sales of the test set.

D. Dynamic Pricing

The predicted sales of a product procured from the forecasting model are used to assign a dynamic price to the product for that specific day. The sales of the product are used to assign the product into one of the 3 tiers: (a) Most in-demand (b) Ideal demand (c) Low demand.

The demand for the product is assigned based on the previous sales record of the product at the retail store. If the individual product sale on the previous day was "N", all the products with predicted sales "P" greater than or equal to "N+1" are assigned in the "Most in-demand" tier. All the products with predicted sales lesser than or equal to "N-1" are assigned in the "Low demand tier". The rest of the products with "N" predicted sales are assigned in the "Ideal demand" tier. The most in- demand tier products have a 20% surge in their price, and Low demand tier products face a 20% fall in their price whereas Ideal demand tier products don't have any change in their price as shown in Fig. 4.

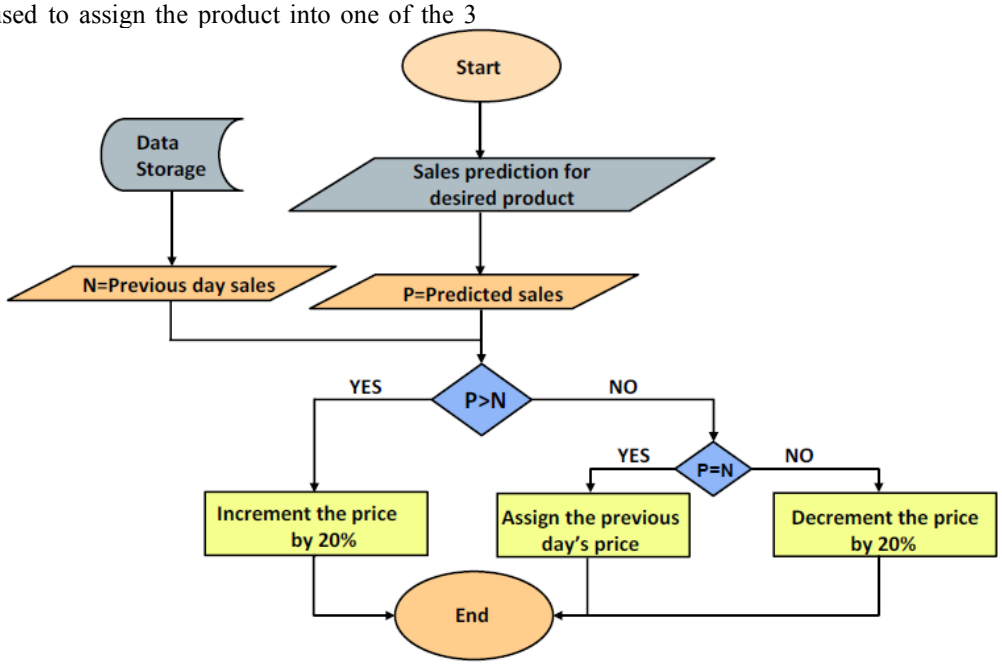

Fig. 4. Implementation of dynamic pricing algorithm.

IV. Results And Discussion

The metrics used to assess the proposed system's accuracy are Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and R^2 score.

Mean Absolute Percentage Error:

MAPE is the average of the absolute percentage errors of forecasts as in (1). The average MAPE of the proposed model is 11.98%.

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| * 100 \quad (1)$$

M - MAPE
n - number of fitted points
A_t - actual value
F_t - forecast value

Root Mean Square Error:

It's the root of the mean of squared differences between actual and predicted observations as in (2). The proposed model has an average RMSE value of 6.63.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \quad (2)$$

y_j - actual value
$\hat{y}_j$ - predicted value
n - number of observations

R^2 score:

For regression models, the R^2 score is a measure of goodness-of-fit. R^2 is calculated by finding the variation of the difference of sum of squares and total sum as in (3). The proposed model's average R^2 score is 0.76, indicating that it is better at fitting the data.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (3)$$

SS_{res} - sum of squares of the residual errors
SS_{tot} - total sum of the errors

The last six months of the dataset are utilized to evaluate the algorithm and forecast sales for the same time period. Fig. 5 indicates the depiction of sales from the train, test sets, and predicted sales for product 50 in store 10.

The actual and predicted sales in the test set and the errors are shown in Fig. 6. The overall sales from the test set for product 50 in store 10 are 24979, whereas the predicted sales are 25106. There is a difference of 127 which is considered as an error.

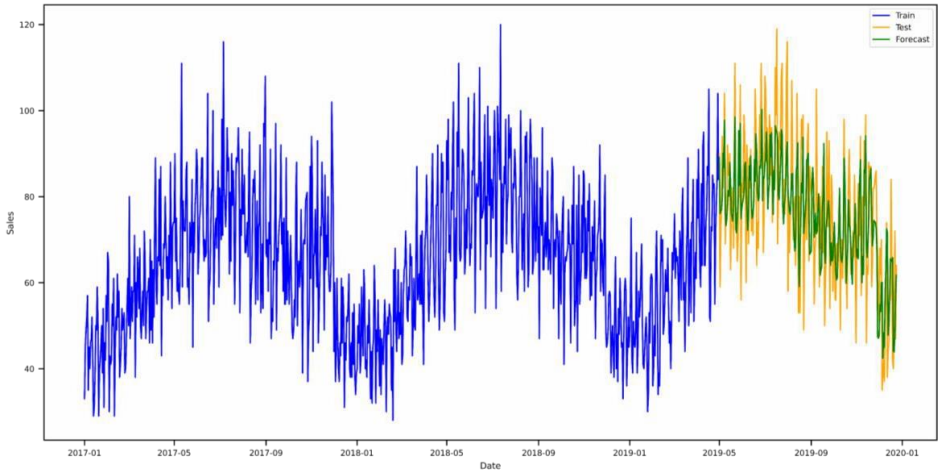

Fig. 5. Sales forecast using the proposed model.

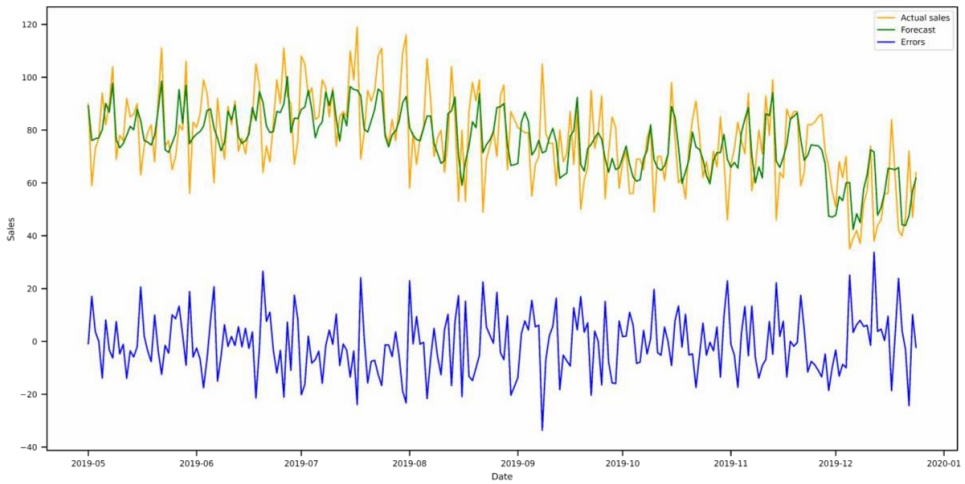

Fig. 6. Comparison of actual and predicted sales.

TABLE II. METRICS USED FOR EVALUATION OF THE PROPOSED MODEL

	Average MAPE	Average RMSE	Average R^2
XGBoost	11.98%	6.63	0.76

It can be inferred from Table II, that the average MAPE, RMSE and R^2 values across all products and store models are 11.98%, 6.63, and 0.76. The forecasted sales of the last 6 months of the dataset are used to assign a dynamic price for the product for the respective day. Considering the previous day's sales of the product 50 in store 10 was 80, and the dynamic pricing of the product is calculated using this threshold. Table III shows the dynamic price for product 50 of store 10 for the dates 01-10- 2019 to 12-10-2019.

TABLE III. DYNAMIC PRICING OF PRODUCT 50 AT STORE 10

S. No	Date	Store ID	Product ID	Old price (in Rs.)	Sales prediction	Dynamic price (in Rs.)
1	2019-10-01	10	50	300	85	320
2	2019-10-02	10	50	300	88	320
3	2019-10-03	10	50	300	90	320
4	2019-10-04	10	50	300	86	320
5	2019-10-05	10	50	300	82	320
6	2019-10-06	10	50	300	80	300
7	2019-10-07	10	50	300	78	240
8	2019-10-08	10	50	300	73	240
9	2019-10-09	10	50	300	72	240
10	2019-10-10	10	50	300	80	300
11	2019-10-11	10	50	300	91	320
12	2019-10-12	10	50	300	90	320

V. CONCLUSION

Sales forecasting and dynamic pricing can thus be used to solve the problem of stock allocation and pricing in retail establishments. Dynamic pricing aids in the efficient use of resources and the growth of a retail store's sales. The forecasting model forecasted sales for 10 stores with 50 products, with average MAPE, RMSE and R^2 values of 11.98%, 6.63 and 0.76 respectively.

VI. FUTURE SCOPE

This proposed system can be further enhanced by introducing a crowd estimation system. This would allow retailers to get an estimate of the crowd and manage their stocks accordingly. This could result in better stockpiling and improved customer service.

REFERENCES

[1] D. Ge, Y. Pan, Z.-J. (Max) Shen, D. Wu, R. Yuan, and C. Zhang, "Retail supply chain management: a review of theories and practices," J. Data, Inf. Manag., vol. 1, no. 1–2, pp. 45–64, May 2019, doi: 10.1007/s42488-019-00004-z.

[2] Z. Camurdan and M. C. Ganiz, "Machine learning based electricity demand forecasting," in 2017 International Conference on Computer Science and Engineering (UBMK), Oct. 2017, pp. 412–417, doi: 10.1109/UBMK.2017.8093428.

[3] D. R. Kiran, A. Rohith, K. D. Reddy, and G. P. Reddy, " Water Sharing Marketplace Using IoT," Springer Advances in Intelligent Systems and Computing, vol. 1245, pp. 543-551, 2021, https://doi.org/10.1007/978-981-15-7234-0_50.

[4] R. Fildes, S. Ma, and S. Kolassa, "Retail forecasting: Research and practice," Int. J. Forecast., Dec. 2019, doi: 10.1016/j.ijforecast.2019.06.004.

[5] G. Tsoumakas, "A survey of machine learning techniques for food sales prediction," Artif. Intell. Rev., vol. 52, no. 1, pp. 441–447, Jun. 2019, doi: 10.1007/s10462-018-9637-z.

[6] Z. Tang, "Mechanism for a New Demand Forecasting Paradigm 'Individual Demand Forecasting,'" in 2010 Third International Conference on Business Intelligence and Financial Engineering, Aug. 2010, pp. 99–103, doi: 10.1109/BIFE.2010.33.

[7] C. Hu, J. Yan, and C. Wang, "Advanced Cyber-Physical Attack Classification with Extreme Gradient Boosting for Smart Transmission Grids," in 2019 IEEE Power & Energy Society General Meeting (PESGM), Aug. 2019, pp. 1–5, doi: 10.1109/PESGM40551.2019.8973679.

[8] T. K. Ghose and T. T. Tran, "A Dynamic Pricing Approach in E-Commerce Based on Multiple Purchase Attributes," Springer Advances in Artificial Intelligence, vol. 6085, pp. 111-122, 2010, https://doi.org/10.1007/978-3-642-13059-5_13.

[9] W. Elmaghraby and P. Keskinocak, "Dynamic Pricing in the Presence of Inventory Considerations: Research Overview, Current Practices, and Future Directions," Manage. Sci., vol. 49, no. 10, pp. 1287–1309, Oct. 2003, doi: 10.1287/mnsc.49.10.1287.17315.

[10] A. González-Vidal, F. Jiménez, and A. F. Gómez-Skarmeta, "A methodology for energy multivariate time series forecasting in smart buildings based on feature selection," Energy Build., vol. 196, pp. 71–82, Aug. 2019, doi: 10.1016/j.enbuild.2019.05.021.

[11] P. Bruno and F. Calimeri, "Using Heatmaps for Deep Learning based Disease Classification," in 2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Jul. 2019, pp. 1–7, doi: 10.1109/CIBCB.2019.8791493.

[12] L. W. Turner and S. F. Witt, "Forecasting Tourism Using Univariate and Multivariate Structural Time Series Models," Tour. Econ., vol. 7, no. 2, pp. 135–147, Jun. 2001, doi: 10.5367/000000001101297775.

[13] R. Spencer, F. Thabtah, N. Abdelhamid, and M. Thompson, "Exploring feature selection and classification methods for predicting heart disease," Digit. Heal., vol. 6, p. 205520762091477, Jan. 2020, doi: 10.1177/2055207620914777.

[14] M. A. M. Hasan, M. Nasser, S. Ahmad, and K. I. Molla, "Feature Selection for Intrusion Detection Using Random Forest," J. Inf. Secur., vol. 07, no. 03, pp. 129–140, 2016, doi: 10.4236/jis.2016.73009.

[15] C. Aguilar-Palacios, S. Munoz-Romero, and J. L. Rojo-Alvarez, "Forecasting Promotional Sales Within the Neighbourhood," IEEE Access, vol. 7, pp. 74759–74775, 2019, doi: 10.1109/ACCESS.2019.2920380.

[16] K. Vijayakumar, C. Arun, "Automated risk identification using NLP in cloud based development environments" , Journal Ambient Intelligence and Humanized Computing", ISSN:1865-5137(Print),1868-5145 (Electronic), (Springer Publisher Sci Indexed, IF:1.588), May 2017 , https://link.springer.com/article/10.1007/s12652-017-0503-7

[17] X. Liao, N. Cao, M. Li, and X. Kang, "Research on Short-Term Load Forecasting Using XGBoost Based on Similar Days," in 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Jan. 2019, pp. 675–678, doi: 10.1109/ICITBS.2019.00167.