© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715663

Big Mart Product Sales Estimator Using Machine Learning

PATAN SAMEENA[1], MOGAL SIDDIKHA BEGUM[2], G B THARUN[3], B SUPREETH[4], A KALYAN KUMAR[5]
[1, 2, 3, 4, 5] *Madanapalle Institute of Technology and Science*

Abstract- ***The retail industry significantly benefits from sales prediction since it facilitates the management of inventories and the making of business decisions. The use of machine learning techniques to create a Big Mart Product Sales Estimator for the purpose of predicting product sales at various retail outlets is the central theme of this paper. The model's foundation is built on historical sales data, which includes features of key products and outlets such as product category, weight, visibility, maximum retail price, outlet size, and location. To boost the accuracy of predictions, data preprocessing methods are applied such as handling of missing values, categorical encoding, and feature scaling. Three machine learning models—Linear Regression, Decision Tree, and Random Forest—are created and assessed with baseline performance metrics. The experiments suggest that Random Forest Regression is the best model since it provides higher accuracy and lower prediction error than the other models. The system that has been proposed helps retailers to make informed decisions based on data, to optimize stock levels and to enhance their sales performance overall.***

Keywords— Big Mart Sales, Machine Learning, Sales Forecast, Retail Data, Regression Models.

I. INTRODUCTION

Sales forecasting is one of the most important activities in the retail industry because it affects directly the management of stocks, supply chain planning, and overall business profits. By having a precise prediction of product sales, retailers can prevent problems like having too much stock, running out of stock, and therefore losing revenue, thus reflecting operational efficiency positively [1]. Although large retail chains have been growing at a fast pace and historical sales data are more accessible, still the use of traditional forecasting methods is not enough to reveal complex sales patterns.

Machine Learning methods have been widely recognized for sales prediction mainly because of their power to work with huge amounts of data and to discover nonlinear relationships among various product, outlet, and customer-related factors [2]. The retail sales are affected by different attributes like product type, price, store size, place, and visibility which makes predicting them a difficult contention [3]. Modern data-driven techniques are capable of managing such multidimensional data very efficiently and granting more precise forecasts.

The Big Mart dataset is considered the gold standard for evaluating sales prediction models as it reflects real-world retail situations with different product and outlet features [4]. Different algorithms like regression, and ensemble-based methods have been adopted for the purpose of getting more precise predictions [5]. The research presents a Big Mart

Product Sales Estimator based on machine learning that is designed to assist with decision-making and retail performance improvement through accurate predictions and evaluations.

II. LITERATURE REVIEW

The prediction of sales has been one of the main topics in research associated with retail analytics because of its crucial role in demand planning and inventory control. Initially, sales forecasting was done using traditional statistical methods like time series analysis and linear regression. These methods are easy to use in practice but they often lack the ability to represent the complex nonlinear interdependencies between several different factors influencing sales like pricing, store location, and customer behavior.

The machine learning field is continuously evolving, and so are the techniques and fanfared models in retail sales forecasting. One of the prominent ones is the Decision Tree that has the ability to interpret with simplicity, alongside the plus of being able to work on

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715663

both kinds of variables (numerical and categorical). Nonetheless, it still runs the risk of being overfitted when the datasets are large and noisy. In order to cope with this drawback, methods from ensemble learning such as Random Forest have been utilized, which amalgamate several decision trees to not only increase the accuracy of the forecasting but also the robustness of the model.

Various research has shown that Random Forest Regression can be productively used for forecasting retail sales based on the features of the product and the outlet. These benefits come from the fact that models' capabilities like Random Forests include having a capacity to tolerate missing values, variance reduction, and simulating complex feature interactions, thus being a perfect fit for large datasets from the retail industry.

The application of gradient boosting techniques like XGBoost and LightGBM for sales forecasting has become more common recently. On the other hand, they generally need considerably more hyperparameter tuning and higher computational resources than Random Forest models.Not with standing the strides made in the prediction of sales using machine learning, a number of the systems that are already in place are still more concerned with the accuracy of the models than with resolving the day-to-day hurdles imposed by the following: data cleaning, selecting attributes, and scalability. Hence, it is necessary to have a unified and effective sales prediction system that gives the same importance to accuracy, interpretability, and applicability in the real world.

III. PROPOSED METHODOLOGY

The suggested methodology is centered around the development and deployment of a powerful machine learning system capable of forecasting product sales in different retail stores. The system adheres to a well-organized pipeline consisting of data gathering, cleaning, feature extraction, model training, assessing performance, and making the final prediction. This holistic technique guarantees precise forecasting and practical applicability in actual retail situations.

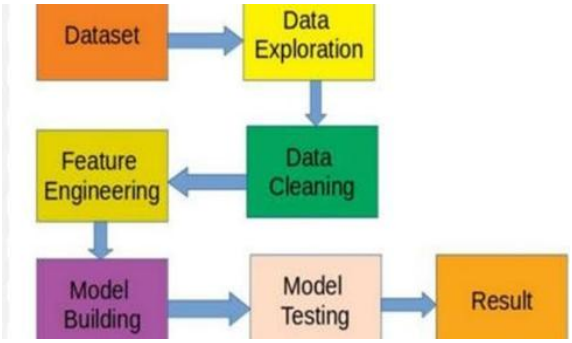

Fig 3.1: Steps for Processing

1. Dataset

The Big Mart sales dataset is the starting point of the process; it contains historical sales percentage information of products sold in different outlets. The dataset is made up of product- related characteristics like type of item, weight of item, item visibility, and maximum retail price, as well as outlet-related characteristics like outlet size, outlet type, and location. The target variable is sales at the item outlet.

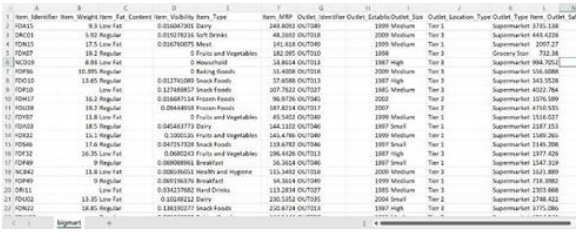

Fig 3.2 : Big Mart dataset

2. Data Exploration

Data exploration is a method used to gain insight into the dataset's structure, distribution, and properties. By means of statistical analysis and visualizations, the data exploration process reveals patterns, trends and associations among features and sales. This stage subsequently assists in recognizing missing data, outliers, and data skewness, thus directing the next steps in data preprocessing.

3. Data Cleaning

The cleaning of data stage deals with the inconsistencies and errors occurring in the data set. In the case of numerical attributes, missing values will be filled with the most suitable imputation techniques whereas in categorical data missing values will be replaced with the most appropriate category. Duplicate records are eliminated and outliers are removed in order to make the data quality and constantly reliable.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715663

4. Feature Engineering

Feature engineering is the process of extracting useful information from raw data and turning it into features for better model performance. Categorical variables are converted into numbers, and numerical features have their ranges adjusted if needed. Besides, new features like outlet age are derived, while irrelevant or duplicate attributes are eliminated in order to cut down the feature space.

5. Model Building

Following feature engineering, the next step is to create machine learning models that will predict sales. The models that are trained on the processed dataset include Linear Regression, Decision Tree Regression, and Random Forest Regression. During the training stage, the models become acquainted with the correlations between the input features and the sales output.

6. Model Testing

The performance of the trained models is assessed with the help of a distinct test data set. The models are compared and the best one is selected based on the evaluation metrics of Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared score.

7. Result

At last, the model yielding the most accurate results and smallest prediction error is chosen. The sales forecasts are examined and disclosed, thus enabling retailers to base their decisions on data regarding stock control, demand predictions, and general planning.

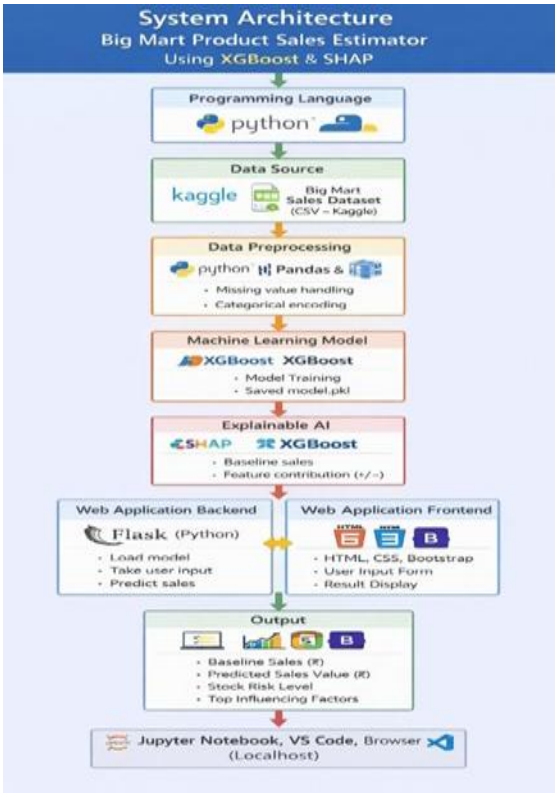

Fig 3.2: Architecture Diagram XGBoost Regressor Model

The XGBoost Regressor is an excellent performing algorithm for regression based on gradient boosting, which is why it is referred to as an advanced ensemble learning algorithm. This project is utilizing the XGBoost Regressor to forecast the sales of Big Mart products through the identification of intricate nonlinear interactions between product and outlet characteristics.

The process of the model is to develop numerous decision trees successively, and every new tree is trained to rectify the prediction mistakes of the earlier trees. This step-by-step

learning process reduces the loss function using gradient descent, resulting in higher prediction exactness. XGBoost applies regularization methods to control overfitting and hence increase the performance of the model in generalization.

During model training, the XGBoost Regressor takes in the preprocessed dataset as its input, and at the same time, crucial hyperparameters such as learning rate,

maximum tree depth, number of estimators, and subsampling rate are tuned. These parameters influence the speed of learning of the model, its complexity and its robustness.

After being trained, the XGBoost Regressor produces reliable sales forecasts for data that has not been seen before. The evaluation of the model's performance is done with the help of regression metrics including Root Mean Squared Error (RMSE) and R-squared score. XGBoost, with its efficiency, scalability, and high predictive power, is a perfect match for large retail datasets like Big Mart sales data.

IV. IMPLEMENTATION

The application of the Big Mart Product Sales Prediction system is done through XGBoost Regressor, which is a potent gradient boosting algorithm for regression problems. The whole system is built with Python and renowned data science libraries.

A. Tools and Technologies

The primary programming language for the implementation is Python. Data is manipulated and preprocessed with Pandas and NumPy. Exploratory data analysis and result visualization take place using Matplotlib and Seaborn. Model development and evaluation are done using the XGBoost library plus Scikit-learn utilities.

B. Data Preprocessing

The Big Mart sales dataset is loaded in CSV format. Mean imputation is used to handle the missing values in the numerical attributes like item weight whereas the filling of the missing categorical values is done with the most frequent category. The encoding techniques are used to convert the categorical features into numerical form. The feature scaling is applied wherever it is needed to have a consistent input for the model.

C. Train–Test Split

The processed dataset gets split into training and testing sets through the train-test split method. The training set is used for the XGBoost model creation while the testing set is kept for the evaluation of the model performance on data, not seen before.

D. XGBoost Model Training

The XGBoost Regressor is set with proper hyperparameters like learning rate, number of estimators, maximum depth, and subsampling ratio. The model is trained on the training dataset where it gradually creates decision trees and reduces prediction error with the help of gradient boosting. XGBoost's regularization methods are effective in controlling overfitting and thus make the model better at generalization.

E. Model Evaluation

The model undergoes a testing process with the test dataset after the training phase. The evaluation of model performance is done through the use of regression metrics, including Root Mean Squared Error (RMSE) and R-squared score. These metrics are helpful for understanding the accuracy of predictions and the trustworthiness of the model.

F. Sales Prediction

After the validation, the loaded XGBoost Regressor is put in charge of predicting the sales of products across various outlets. The retailers are helped to forecast the demand patterns and consequently, to take better decisions on managing their stocks and planning sales through the use of the predicted values.

The BigMart Product Sales Prediction system is executed utilizing Python, XGBoost Regressor, and a Flask web application. The entire system is split into the two core components: training the model and providing predictions through the web.

Model Training

Initially, the BigMart sales dataset is first imported from a CSV file using Python data processing libraries. The dataset is subjected to rigorous preprocessing prior to model training in order to make sure that the data quality and consistency are at their best. Mean imputation is used to handle the missing values in the numerical features such as item weight and outlet size, while the mode is used to fill the categorical features with missing values. This step is beneficial since it not only helps maintain data integrity but also prevents information loss during training. The categorical attributes of outlet type and outlet location are numerically represented through

label encoding and are thus rendered suitable for the machine learning algorithms.

Predictive capability of the model is increased by features engineering based on domain knowledge like season type, price sensitivity and demand trend. The features so derived help to uncover hidden patterns related to consumer behavior and market dynamics that were not directly available in the original dataset. The next step after feature engineering is to split the dataset into training and testing sets so that the model's performance on unseen data can be evaluated without bias.

A model based on XGBoost Regressor is then built with the training dataset that has been prepared earlier. The model's complexity and prediction accuracy are balanced by tuning key hyperparameters like learning rate, maximum tree depth, and number of estimators. The regularization measures put in place by XGBoost help in controlling overfitting and subsequently improving the model's generalization. The saving of the trained model in a .pkl file format that allows efficient loading and reuse for real-time sales prediction in the deployed application is done after the training process has been completed.

Web Application Development Using Flask

A web app built on Flask gives sales forecasting a plain and interactive user interface. Flask is the one chosen since it has a light structure and can easily be combined with the machine learning models. The trained XGBoost regression model saved in .pkl format is uploaded in the Flask application when it runs which allows real-time prediction without the need for retraining the model.

Through the web interface, users are allowed to input main product and outlet details like Item MRP, Item Weight, Item Visibility, Outlet Type, Outlet Location, Season Type, and Demand Trend using input fields and dropdown menus. These inputs are rigorously validated for correctness and consistency before being processed by the backend. The app transforms categorical inputs into numerical values applying the same encoding scheme of model training which guarantees compatibility between the phases of training and prediction.

After the inputs are processed they are fed into the trained XGBoost model for sales prediction. By examining the supplied features the model produces an estimated sales figure. The forecasted outcome is then returned to the front end and prominently displayed on the webpage. This user interface, backend processing, and machine learning model interaction shows the successful deployment of the sales prediction system and offering of an efficient decision- support tool for the purpose of retail planning right at users' hands.

Risk Analysis and Explanation

The sales prediction value generation results is followed by the system performing a risk analysis aimed at helping the retailers to take the right decisions regarding the management of their stocks. The forecasted sales are classified according to High Risk, Medium Risk, or Low Risk levels of stock based on setting threshold values beforehand. A High Risk class means that the sales are expected to be low, probably leading to an overstocking risk if such inventory levels are maintained. Medium Risk stands for balanced sales demand, needing balanced stock planning, whereas Low Risk points to high sales expectation, i.e., the caution to prevent stock shortage through free access to inventory.

To give the prediction results more clarity and make them easier to interpret, SHAP (SHapley Additive exPlanations) is the method used to explain the output of the XGBoost model. SHAP values are calculated for the input features in terms of the respective contribution to the final sales prediction value and are based on the respective increase or decrease of the predicted sales value against a baseline. The features considered for the prediction were the Item MRP, Item Visibility, Outlet Type, Outlet Location, Season Type, and Demand Trend.

The synergy of risk classification and SHAP-based explanation not only allows the users to see the predicted sales but also gives them the insight of what has been the main reasons for the prediction to be made. Improved trust in the model, data-driven decision-making support, and retailers' capability of effectively adjusting pricing, promotions, and inventory strategies are some advantages of the mentioned scenario.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715663

User Interface Implementation (index.html)

The Big Mart Product Sales Estimator has an interactive user interface designed with HTML and CSS, which gives its users a clean, simple, and adaptable front-end to communicate with the system. The communication path between a user and the sales prediction system is therefore this interface. The design focuses on usability and clarity, meaning the users would be able to input the necessary data without having any technical background.

The web page features a structured form that allows the users to provide the necessary product and outlet attributes. Different input fields for numerical data are available for such features as Item MRP, Item Weight, and Item Visibility, thus allowing for very accurate data input. Along with that, drop- down menus are used for features like Outlet Type, Outlet Location, Season Type, and Demand Trend which belong to the categorical type. These user-selected inputs help minimize data entry mistakes and maintain the same standard as that of the model training.

The user then submits the form with a click on the Predict Sales button after entering all required information. The entered data is then securely sent to the Flask backend by means of the POST method. At the server side, the backend processes and checks the values of the input, converts them to the required numerical format and sends them to the trained XGBoost Regressor model for forecasting.

The predicted sales figure is sent to the front-end afterward and then shown on the same web page very clearly. The interface's simplicity and responsiveness detail a very smooth interaction and fast feedback, which would make the application suitable for actual retail use. To sum it up, the user interface is a perfect example of how advanced machine learning models can be brought closer to the end users through an intuitive and user-friendly platform for sales prediction.

V. APPLICATION OVERVIEW

Big Mart Product Sales Estimator is a web-based application based on machine learning that predicts sales of retail products by historical and contextual data. It provides retailers and business analysts with the estimations of product sales considering the key factors like item price (MRP), visibility, weight, outlet type, city tier, seasonal demand, and market demand trends. The system has integrated a trained machine learning model with a user-friendly web interface and is thus able to provide accurate and data-driven sales forecasts.

The application's main purpose is to assist inventory management, demand forecasting, and risk assessment in the retail industry. Users can simply enter details of the product and outlet through an easy-to-use interface, and the backend model goes through these inputs to produce sales forecasts. The app not only predicts sales figures but also tells what factors influence those figures and how much the stock is at risk, which makes it more explaining and helpful for decision-making.

In summary, the application eliminates the disconnect between highly sophisticated machine learning models and the actual business implementations. It allows non-technical people to easily access and use the predictive analytics of the complex models through the simple web platform, thus enhancing operational efficiency, minimizing both overstock and understock situations, and contributing positively to strategic planning in retail settings.

Outputs

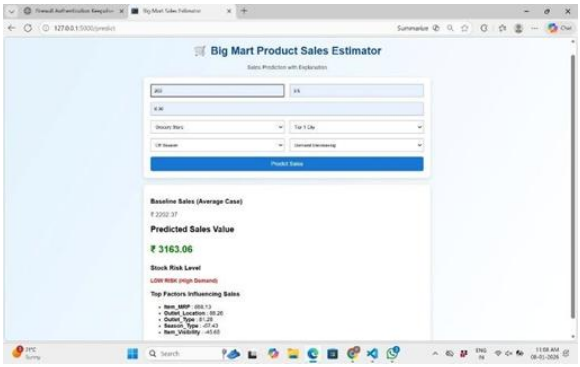

Fig 5.1: Low Risk Prediction

The prediction of Low Stock Risk created by the sales estimation system based on XGBoost is depicted in the figure. The trained model forecasts the sales value after the user provides the product and outlet attributes such as Item MRP, Item Weight, Item Visibility,

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715663

Outlet Type, Location Tier, Season Type, and Demand Trend.

In this scenario, the model predicts a sales value of ₹3163.06, which is a great deal more than the baseline average sales value of ₹2202.37. The system then identifies the stock status as Low Risk (High Demand) because the predicted sales have surpassed the baseline threshold, indicating that the product has a good chance of selling and therefore, it will be safe to have a higher inventory.

To further illustrate the process, the system along with SHAP gives a much clearer view and explainable insights. The most influencing factors contributing to the prediction are Item MRP, Outlet Location, Outlet Type, and Season Type, along with their positive or negative impact values. These clarifications not only help retailers to be aware of the reason behind the high demand of the product but also support the subdivision of their inventories through data-driven decisions.

The Low Risk output is an evidence of the proposed model's capability of accurately predicting sales performance and at the same time giving explanations that help retailers in reducing stock-related risks.

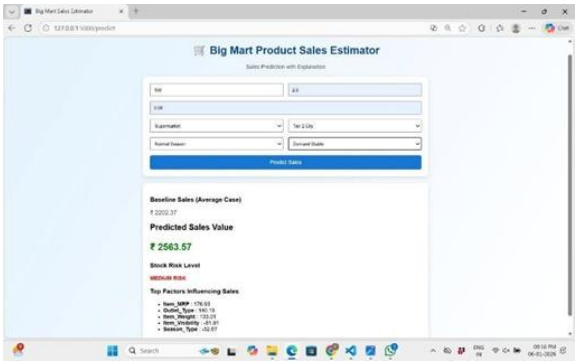

Fig 5.2: Medium Risk Prediction

The depicted scenario in the figure represents the Medium Stock Risk situation that the Big Mart Product Sales Estimator predicted through XGBoost model. The trained model evaluates the expected sales value after the user has input the product and outlet data comprising Item MRP, Item Weight, Item Visibility, Outlet Type, Outlet Location, Season Type, and Demand Trend.

The model in this particular situation arrives at a sales value of ₹2563.57, which is somewhat higher than the baseline average sales value of ₹2202.37. The system classifies the condition of the stock as Medium Risk, thereby indicating balanced demand since the predicted sales are only slightly higher than the baseline. This scenario suggests that the retailers should keep their stock levels at the optimum level to prevent both overstock and stocks out situations.

The system also describes the prediction with the help of SHAP-based feature importance analysis. Among the Item MRP, Outlet Type, and Item Weight are the features that positively affect sales, while Visibility and Season Type cause a negative influence. These explanations are effective in showing the demand patterns to retailers and in providing them with the tools to make better inventory and supply chain decisions.

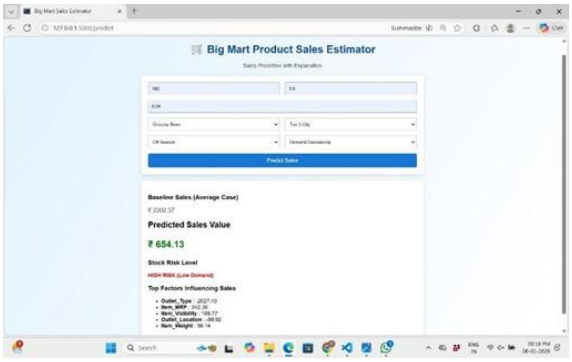

Fig 5.3: High Risk Prediction

The illustration depicts the High Stock Risk scenario produced by the Big Mart Product Sales Estimator through the XGBoost regression model. The system predicts a sales value of ₹654.13 based on the user inputs like Item MRP, Item Weight, Item Visibility, Outlet Type, Outlet Location,

Season Type, and Demand Trend, which is much lower than the baseline average sales value of ₹2202.37.

Because of the large difference under the baseline, the system marks this situation High Risk (Low Demand). This reflects a very high likelihood of overstocking if the inventory is not adjusted accordingly. Such a prediction is especially beneficial for retailers who can

then take preventive actions like cutting stock levels, offering discounts, or changing prices.

In the effort to increase the unit's understanding, the system utilizes SHAP - based explainability to highlight up the most powerful characteristics that are responsible for the prediction. The aspects such as Outlet Type, Item MRP, and Item Visibility are said to have a negative effect on sales, whereas Item Weight has a slight positive impact. The different explanations given ensure that the model's decision- making is exposed and it is easier to support data-driven inventory and sales management through this.

Future Work:
In the upcoming time, the precision of the Big Mart Product Sales Estimator can be enhanced by executing extensive hyperparameter tuning of the XGBoost regression model. The utilization of various techniques like Grid Search, Random Search, and Bayesian Optimization can be done to tweak the parameters such as learning rate, maximum tree depth, and number of estimators which in turn will result in better model performance and lesser prediction errors.

Moreover, the system can be upgraded by introducing time- dependent features like past sales trends, seasonal changes, and demand cycles. The creation of lag-based and rolling statistical features can enable the XGBoost model to detect temporal patterns more accurately and thus, provide trustworthy sales predictions not only during the different times but also under different market conditions.

On the other hand, there is also a possibility of improvement by blending XGBoost with ensemble or hybrid learning strategies. The synergistic effect of XGBoost and other regression models can make the predictions more stable and have wider applicability, particularly in the case of various outlet types, product categories, and different consumer behaviors.

As a last step, the application will be boosted by allowing integration of real-time data and continuous model retraining. The cloud-based optimized XGBoost model would provide the facility of scalable, real-time sales forecasting, and the support of adaptive learning whenever new sales data comes in, thus making the system more suitable for the large-scale retail sector.

VI. CONCLUSION

The paper introduced a system for estimating the sales of Big Mart products, which relied on the XGBoost regression model to forecast the sales of retail products by taking into consideration the key features of both product and outlet. The model was able to recognize the nonlinear connections within the data and reach high accuracy in predictions by utilizing preprocessing and feature engineering techniques appropriately.

Moreover, the combining of the trained XGBoost model with a web-based app proved its usefulness in forecasting sales in real-time, thus confirming the results of XGBoost being not only a good but also a scalable method for predicting retail sales, therefore aiding decision-making regarding inventory and demand management.

REFERENCES

[1] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.
[2] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.
[3] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Waltham, MA, USA: Elsevier, 2012.
[4] A. Géron, Hands-On Machine Learning with Scikit- Learn, Keras, and TensorFlow, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2019.
[5] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
[6] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” Ann. Stat., vol. 29, no. 5, pp. 1189–1232, 2001.
[7] S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning: From Theory to Algorithms. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[8] M. Kuhn and K. Johnson, Applied Predictive Modeling. New York, NY, USA: Springer, 2013.
[9] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning, 2nd ed. New York, NY, USA: Springer, 2017.
[10] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.
[11] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods," PLOS ONE, vol. 13, no. 3, pp. 1–26, 2018.
[12] K. G. Ramanathan, "Retail sales forecasting using machine learning techniques," Int. J. Comput. Appl., vol. 179, no. 7, pp. 22–27, 2018.
[13] A. Singh and N. Sharma, "Sales prediction using regression techniques," Int. J. Adv. Res. Comput. Sci., vol. 10, no. 3, pp. 45–49, 2019.
[14] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 2nd ed. Melbourne, Australia: OTexts, 2018.
[15] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," Informatica, vol. 31, no. 3, pp. 249–268, 2007.
[16] P. Domingos, "A few useful things to know about machine learning," Commun. ACM, vol. 55, no. 10, pp. 78–87, 2012.
[17] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825– 2830, 2011.
[18] S. Raschka and V. Mirjalili, Python Machine Learning, 3rd ed. Birmingham, U.K.: Packt Publishing, 2019.
[19] A. B. Nassif, M. A. Azzeh, M. Capretz, and D. Ho, "Machine learning techniques for sales forecasting," Procedia Comput. Sci., vol. 170, pp. 123–130, 2020.
[20] R. Carbonneau, K. Laframboise, and R. Vahidov, "Application of machine learning techniques for supply chain demand forecasting," Eur. J. Oper. Res., vol. 184, no. 3, pp. 1140–1154, 2008.
[21] M. Zunic, "Retail demand forecasting using XGBoost," in Proc. IEEE Int. Conf. Big Data, Los Angeles, CA, USA, 2019, pp. 3456–3461.
[22] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," J. Comput. Syst. Sci., vol. 55, no. 1, pp. 119–139, 1997.
[23] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2016.
[24] N. Japkowicz and M. Shah, Evaluating Learning Algorithms. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[25] M. Stone, "Cross-validatory choice and assessment of statistical predictions," J. Roy. Stat. Soc., vol. 36, no. 2, pp. 111–147, 1974.
[26] J. Brownlee, Machine Learning Mastery with Python. Melbourne, Australia: Machine Learning Mastery, 2018.
[27] H. Kagermann, "Data-driven retail analytics," Bus. Inf. Syst. Eng., vol. 7, no. 1, pp. 23–28, 2015.
[28] A. Ng, "Machine learning and AI via brain simulations," Commun. ACM, vol. 55, no. 4, pp. 44–53, 2012.
[29] S. Chakraborty and S. Chatterjee, "Retail sales prediction using ensemble learning methods," Int. J. Data Sci. Anal., vol. 9, no. 2, pp. 67–78, 2020.
[30] P. B. D. Silva and R. Perera, "A comparative study of regression models for sales forecasting," Int. J. Comput. Sci. Inf. Secur., vol. 17, no. 6, pp. 45–52, 2019.