Journal of Electronic & Information Systems
https://ojs.bilpublishing.com/index.php/jeis

ARTICLE

A Multi-Model Output Fusion Strategy Based on Various Machine Learning Techniques for Product Price Prediction

Diwei Zhu[1,2], Xiaoyang Chen[3], Yunxiang Gan[4*]

1 Hive AI, San Francisco, CA 94105, USA
2 New York University, New York, NY 10003, USA
3 Radiawave Co., Ltd., Shen Zhen 518100, China
4 Moloco, Redwood City, CA 94063, USA

ARTICLE INFO

Article history
Received: 1 March 2022
Accepted: 17 March 2022
Published Online: 30 March 2022

Keywords:
Component
Multi-model fusion
Product price prediction
Machine learning

ABSTRACT

In the digital era, precise product price prediction becomes crucial for enhancing competitiveness in the online marketplace. This paper presents a hybrid model framework that enhances the accuracy of online product price predictions by integrating several machine learning algorithms, including Linear Regression, Decision Trees, and Gradient Boosting. The objective of this approach is to leverage the distinct advantages of each model to address their individual limitations and create a robust unified predictive model. This integration allows for improved handling of complex data relationships and diverse market dynamics that are typical in online sales environments. The results demonstrate that the hybrid model achieves superior prediction accuracy, as reflected in reduced Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics, and an exceptionally high R^2 value compared to single-model approaches. These outcomes underscore the efficacy of combining multiple predictive models to enhance the precision of price forecasts in the highly competitive online marketplace. This model fusion strategy not only provides more accurate pricing predictions but also offers strategic insights into the optimal pricing strategies for businesses looking to enhance their market position.

1. Introduction

Products are the foundation of any business. Whether physical or digital, a product is what businesses offer to their customers to satisfy their needs and desires. The importance of products goes beyond their immediate use. They are key to building a company's brand, driving customer loyalty, and generating revenue. Every successful company, from tech giants to small retailers, depends on the products they create or sell. A product that meets the customer's expectations and offers value can lead to long-term success. On the other hand, a poorly made or overpriced product can harm a company ' s reputation and hurt its bottom line.

*Corresponding Author:
Yunxiang Gan,
Moloco, Redwood City, CA 94063, USA;
Email: yg281@scarletmail.rutgers.edu

DOI: https://doi.org/10.30564/jeis.v4i1.7566

In the digital age, many products are sold online, and this shift has increased the importance of pricing strategies. Online platforms have made it easier for consumers to compare prices, and this has created a highly competitive marketplace where pricing plays a crucial role. For businesses, setting the right price for their products can be the difference between attracting or losing customers. If the price is too high, potential buyers might turn to competitors. If it's too low, businesses may struggle to maintain profitability. Therefore, accurately predicting and adjusting product prices is essential for businesses to succeed in the online marketplace.

Online product price prediction is a complex task. It requires analyzing vast amounts of data and understanding various factors such as demand, supply, production costs, market trends, and consumer behavior. The need for accurate price prediction has led to the use of artificial intelligence (AI) and machine learning (ML) models [1–4]. Over the past decade, AI, ML and computer optimization methods have advanced rapidly, providing businesses with new tools to predict prices more accurately than ever before. These models can process large datasets, identify patterns, and make predictions that humans would struggle to achieve manually in many domains. For example, Dai et al. proposed a markov process and deep-reinforcement learning-based method for safety evaluation of traffic system [5]. Let et al. utilized chemical industrial clusters-based data analysis method for green supply chain management optimization [6]. Xiong et al. developed ensemble model of attention mechanism-based DCGAN and autoencoder for effective noised OCR classification, which demonstrated the effectiveness of AI models [7]. In addition, there are also many other methods that applied in different fields [8–10].

The development of AI models for price prediction has gone through several stages. Initially, simpler models such as linear regression were used to predict prices. These models relied on basic relationships between variables, such as how demand changes with price. However, as the market became more complex and data volumes increased, more advanced models like decision trees, random forests, and neural networks were introduced. These models could capture more intricate relationships and interactions between different factors affecting price. AI has also made it possible to process unstructured data, such as customer reviews and social media posts, to gain deeper insights into consumer sentiment and its impact on prices.

Despite the progress made with AI and ML models, predicting product prices with a single model has limitations. No single model can perfectly capture all the factors that influence pricing in a constantly changing market. For instance, linear regression models are good at finding simple relationships but may miss more complex patterns. More advanced models like random forests or neural networks can handle complexity but are sometimes prone to overfitting or may require a lot of computational power. Each model has strengths and weaknesses, and relying on a single model may result in suboptimal predictions.

To overcome the limitations of individual models, researchers and data scientists have explored the idea of combining multiple models. By using a hybrid or ensemble approach, it is possible to take advantage of the strengths of different models while minimizing their weaknesses. In this paper, we propose a hybrid model prediction framework for online product price prediction shown in **Figure 1**. Instead of relying on one model, we combine the predictions of several different machine learning models, such as linear regression, decision trees among others. By doing this, we aim to improve the overall accuracy of price predictions. This hybrid model approach has the potential to provide businesses with more reliable price predictions, helping them stay competitive in the fast-paced online marketplace. By using multiple models and combining their outputs, we can capture a wider range of factors that influence price changes and make predictions that are both accurate and flexible.

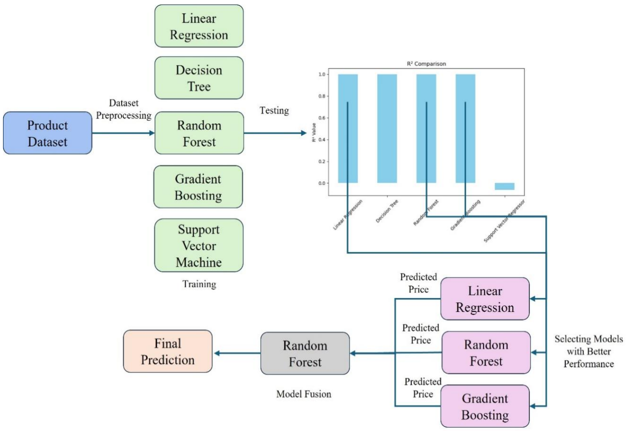

Figure 1. The workflow of multi-model output fusion strategy.

2. Literature Review

2.1 Price Prediction

In this part, we explore a variety of methods that researchers have historically used to predict prices. For many decades, academics have focused on forecasting the prices of commodities sensitive to market changes. The study of oil prices began earnestly in 1931, notably with Hotelling, a prominent economist who introduced mathematical approaches for analyzing markets for natural resources. His work laid the foundational theories that many later economists built upon in the field of energy economics.

Traditional statistical methods have consistently been applied in price prediction across various markets. For example, Lagarto et al. employed the ARIMA model to analyze and predict prices in the Iberian electricity market[11], demonstrating how past data could inform future price trends. As the field of prediction evolved, researchers realized that traditional methods struggled with non-linear data, which is common in volatile markets. This realization led scholars to seek out new methods better suited for dynamic and fluctuating data types. Among these new approaches, Hsu and Chen applied a grey model to predict electricity demand, a method valued for its ability to handle insufficient and uncertain data[12]. Similarly, Gonzalez et al. utilized a hidden Markov model to forecast spot prices in Spain's electricity market, showing that these advanced models could capture the randomness and unpredictability inherent in such data[13].

Modern prediction techniques have increasingly incorporated non-linear models due to their ability to adapt to the complex fluctuations found in data[14–16]. Yousefi et al. used wavelet theory models to forecast crude oil prices[17]. Their approach, which integrates time-scale localization techniques, significantly improved short-term prediction accuracy by comparing its results with futures market data. Neural networks, renowned for their extensive capabilities in large-scale and non-linear processing, have become popular in forecasting non-linear price movements[18]. These networks, composed of layers of interconnected nodes, are adept at identifying patterns in time-series data that traditional models might miss. For instance, Kimoto et al. developed a modular neural network to predict stock prices on the Tokyo Stock Exchange[19], achieving not only accurate predictions but also profitable outcomes in market simulations.

The advancement in predictive methodologies has not been limited to stock or commodity markets. In the realm of consumer products, innovative approaches have been tested to predict prices of second-hand items using mixed data inputs. Fathalla and Salah designed a deep model architecture combining long short-term memory (LSTM) networks and convolutional neural networks to estimate the minimum and maximum prices of used goods[20]. They also developed a method to assess the quality of items, which helped refine their price predictions.

3. Method

3.1 Dataset Preparation

In this study, we utilized a dataset sourced from Kaggle comprising 1000 entries associated with laptop specifications and their corresponding market prices. The dataset encompasses seven features, which include both categorical and numerical data types. The features are as follows: 'Brand' (a categorical string representing the manufacturer), 'Processor_Speed' (a numerical value indicating the CPU speed in GHz), 'RAM_Size' (the size of RAM in gigabytes), 'Storage_Capacity' (hard drive storage in gigabytes), 'Screen_Size' (measured diagonally in inches), 'Weight' (laptop weight in kilograms), and 'Price' (the target variable, representing the market price of the laptop in US dollars).

For preprocessing, the categorical 'Brand' feature underwent one-hot encoding to transform it into a numerical format suitable for machine learning algorithms. All numerical features were normalized to ensure uniformity in scale, which is crucial for optimizing the performance of many predictive models. The dataset was split into training and testing subsets, with 70% of the data allocated for training and the remaining 30% used for testing. This division allows for a comprehensive evaluation of the model's performance on unseen data, ensuring that our predictive insights are both robust and reliable. The distributions of features in this dataset are shown in **Figure 2**. In addition, **Figure 3** and **Figure 4** provide box plots of numerical features and correlation maps.

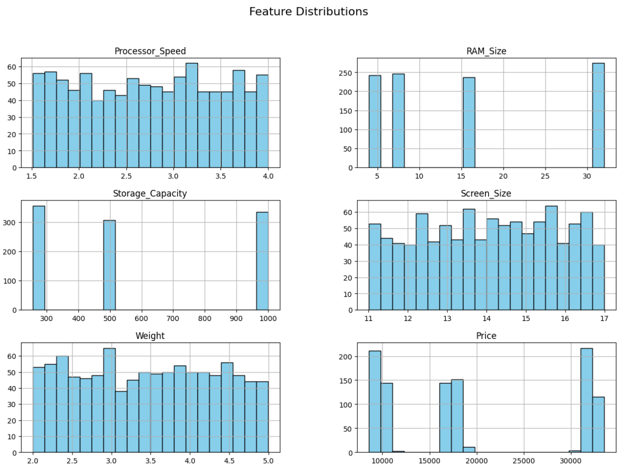

Figure 2. The distributions of features in this dataset.

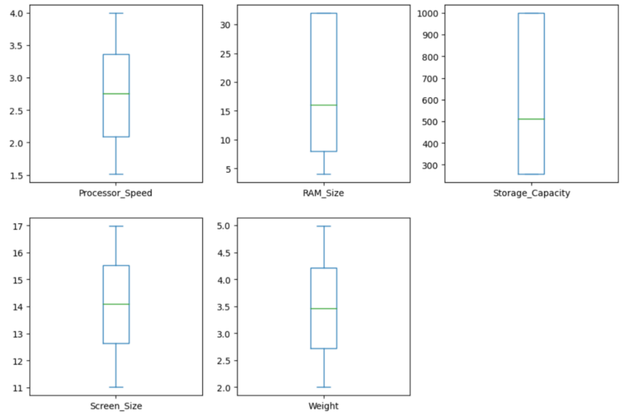

Figure 3. Box plots of numerical features.

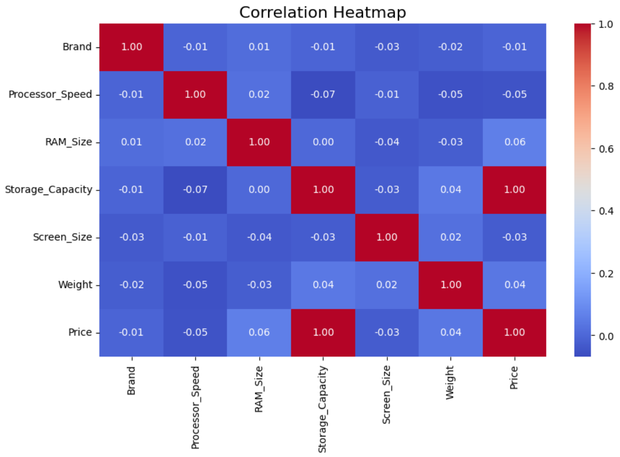

Figure 4. The correlation heatmaps of features.

3.2 The Introduction of Used Machine Learning Models

1) Linear regression

Linear regression[21–24] is a fundamental statistical and machine learning technique used for predicting a dependent variable based on one or more independent variables. It models the relationship between the input variables and the predicted output by fitting a linear equation to observed data. The basic idea of linear regression is to establish a linear relationship between the input and output, effectively drawing a straight line that best fits the data points. This line is the line of best fit and is determined by finding the parameters that minimize the sum of the squared differences between the observed values and the values predicted by the model.

Simple linear regression deals with one independent variable and is used to predict a single output. It is straightforward and often used to determine the strength and direction of the relationship between two variables. For example, predicting house prices based on square footage would typically use simple linear regression. For more complex situations where multiple factors influence the outcome, multiple linear regression is used. This involves several independent variables that together predict the value of the dependent variable. For instance, predicting house prices could be modeled on not just square footage, but also the number of bedrooms, age of the property, and proximity to amenities Linear regression is widely used due to its simplicity, ease of interpretation, and basis for understanding more complex machine learning algorithms. It is a powerful tool for making predictions and decisions in various fields, from economics to engineering.

2) Decision tree

A decision tree[25–27] is a popular machine learning model used for both classification and regression tasks. It operates by breaking down a dataset into smaller subsets based on an attribute value test. This process is repeated recursively on each derived subset in a tree-like fashion. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

The structure of a decision tree consists of nodes, branches, and leaves. Each internal node represents a "test" or "question" on an attribute, each branch represents the outcome of that test (leading to another question or a conclusion), and each leaf node represents a class label or a final decision. The paths from the root to the leaf represent classification rules or regression paths.

One of the main advantages of decision trees is their simplicity and transparency. They are easy to understand and interpret, as they mimic human decision-making more closely than other algorithms—a feature often referred to as the white-box model. Users can see exactly how decisions are made, making them useful for tasks where transparency is important, such as in finance and healthcare. Decision trees are also flexible and can handle both numerical and categorical data. They can be used for a wide variety of problems, ranging from customer segmentation to the prediction of loan defaults.

3) Random forest

Random Forest[28–30] is an advanced machine learning algorithm that builds on the simplicity of decision trees with enhanced accuracy and robustness. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. This ensemble approach helps in handling overfitting, which is a common problem with single decision trees.

The process of creating a Random Forest model involves generating multiple trees using a technique called "bootstrap aggregating" or "bagging." In this technique, different subsets of the dataset are randomly selected with replacement to train multiple decision trees. Each tree in the forest is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. Moreover, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is the best among a random subset of the features is chosen. This strategy of combining random subsets of data with random subsets of features leads to a greater tree diversity, which trades a slight increase in bias for a larger decrease in variance, resulting in an overall better model.

Random Forest has several strengths that make it very useful for practical applications. Firstly, it can be used for both classification and regression tasks, and it's easy to measure the relative importance of each feature on the prediction. This is helpful in feature selection where we aim to identify which factors are most influential in predicting the outcome. Random Forests are also very versatile and can handle large datasets with higher dimensionality. They can automatically handle missing values and maintain accuracy for a large proportion of data missing.

4) Gradient boosting

Gradient Boosting[31–33] is a powerful machine learning technique that builds on the concept of boosting, where weak learners (typically decision trees) are combined sequentially to create a strong overall model. Unlike Random Forests, which build trees independently and combine their outcomes at the end, Gradient Boosting builds one tree at a time. This iterative approach focuses on continuously reducing the errors made by the previous trees.

The process starts with a base model that makes simple predictions, which could be as basic as predicting the average value of the target variable. After the initial predictions, the algorithm calculates the residuals or errors between the predicted and actual values. Subsequent trees are then built, each one focusing on correcting the errors made by the previous ones. Each tree in the sequence is called a weak learner, and it specifically aims to improve the shortcomings of the previous tree.

A key element of Gradient Boosting is the learning rate, which controls how fast the model learns. A smaller learning rate means that each tree has a smaller impact on the final outcome, requiring more trees in the model but often leading to better generalization (reducing the risk of overfitting). Conversely, a higher learning rate makes the model learn faster, which might lead to fitting the training data too closely.

One of the significant advantages of Gradient Boosting is its flexibility to be used for both regression and classification problems. It is also highly effective in handling various types of data, including missing values and heterogeneous features (numeric and categorical).

5) Support vector machine

Support Vector Machine (SVM)[34–36] is a powerful, versatile machine learning algorithm primarily used for classification tasks, but it can also be adapted for regression. At its core, SVM is designed to identify the optimal boundary, or decision surface, that best separates different classes in a dataset. This boundary is chosen to maximize the margin, which is the distance between the nearest data points of different classes (known as support vectors) and the decision surface itself.

Linear SVM is its simplest form, SVM is used to create a linear boundary between classes. This linear classifier works best when the data is linearly separable, meaning the classes can be separated with a straight line (or hyperplane in higher dimensions). The algorithm finds the hyperplane that has the largest minimum distance to the training samples, maximizing the margin between the closest samples of separate classes (support vectors). SVMs are highly appreciated for their strong generalization capabilities. They don't just look for any line that can separate the classes; they look for the line that can do so with the maximum margin. This inherent capacity to generalize makes them very robust, especially against overfitting when compared to other algorithms. The problem of finding the optimal hyperplane is an optimization problem, which means SVMs are based on solid mathematical foundations. The solution to the SVM optimization problem ensures that the model makes the most balanced decision possible, minimizing error.

3.3 Multi-Model Fusion Strategy for Product Price Prediction

In this research, we commence training five distinct models and then proceed to introduce an innovative fusion strategy for models aimed at boosting the accuracy and reliability of predictive analytics. This methodology initiates with the input of the original dataset into three varied models for the purpose of training. Utilizing the most effective models—specifically Linear Regression, Random Forest, and Gradient Boosting—predictions are generated for both training and test datasets. The predictions from these top-performing models are subsequently consolidated to create a new feature set. The core of this strategy involves using these combined features to train and refine the Random Forest model further. We execute this experimental framework using the sklearn library in a Python 3 environment. The process for the model fusion strategy includes the following phases: 1) Load the dataset and divide it into training and test segments, maintaining a 70:30 split. Ensure consistency in data splits across all models by setting the random state parameter to a fixed value. 2) Train the three regression models: Linear Regression, Random Forest, and Gradient Boosting. Parameters are iteratively adjusted to discover the most effective settings for each. 3) Following the determination of the best individual model, amalgamate the predictions from these models to forge a new set of input features. 4) Employ these merged features to train and enhance a sophisticated Random Forest model. 5) Measure the efficacy of the various models using statistical metrics such as RMSE[37], MAE[38] and R^2[39].

4. Results and Discussion

4.1 The Performance of Different Machine Learning Models

Table 1 and **Figures 5** through 11 meticulously present the comparative performance of Linear Regression, Decision Tree, Random Forest, Gradient Boosting, and Support Vector Regression models on a test dataset. The assessment is based on standard metrics: MAE, RMSE, and the R^2: 1) Linear Regression shows excellent performance with an MAE of 153.64, an RMSE of 191.74, and an R^2 of 0.99958. The prediction curve in Figure 5 closely matches the true values, indicating a high accuracy and consistency of this model. 2) Decision Tree has a higher MAE and RMSE of 232.26 and 291.45, respectively, with an R^2 of 0.99903. Despite its simplicity, the decision tree model exhibits robust predictive power, although the variance in predictions is higher compared to linear regression, as seen in **Figure 6**. 3) Random Forest improves on the Decision Tree with an MAE of 173.23, an RMSE of 215.31, and an R^2 of 0.99947. **Figure 7** demonstrates that Random Forest predictions are more consistent with the true values than those of the Decision Tree, showcasing its capability to handle overfitting better. 4) Gradient Boosting presents an MAE of 165.28, an RMSE of 204.51, and an R^2 of 0.99952. As depicted in **Figure 8**, this model provides a tightly fitted prediction curve that aligns well with the actual data points, reflecting its effectiveness in sequential improvement from the errors of previous trees. 5) Support Vector Regression (SVR) dramatically differs in performance with an extremely high MAE of 7777.55 and RMSE of 9686.28, with a negative R^2 of -0.061. **Figure 9** illustrates a significant deviation of SVR predictions from the true values, indicating poor model performance and potential issues in kernel or parameter settings.

The bar charts in **Figure 9** and **Figure 10** provide a stark visualization of the performance discrepancies among the models. SVR's error metrics are disproportionately higher than those of other models, suggesting its inapplicability for this particular dataset or need for substantial parameter tuning. In addition, **Figure 11** sharply contrasts the effectiveness of each model in explaining the variability of the response data. All models except SVR have an R^2 value close to 1, indicating excellent predictability. The negative R^2 for SVR confirms its unsuitability for the data, as it fails to account for the variance around the mean.

Table 1. The performance of different models in testing dataset.

Model Name	MAE	RMSE	R^2
Linear Regression	153.643727	191.737282	0.999584
Decision Tree	232.261282	291.451099	0.999039
Random Forest	173.225308	215.310736	0.999476
Gradient Boosting	165.275236	204.513808	0.999527
Support Vector Regressor	7777.551840	9686.279784	−0.061316

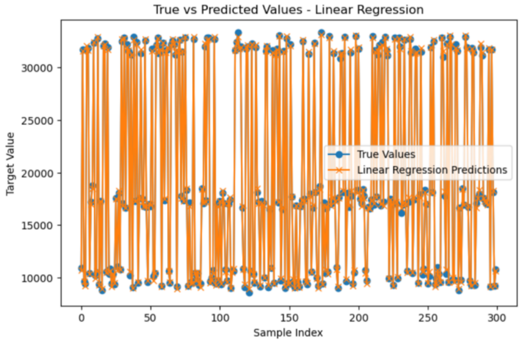

Figure 5. The prediction curve based on linear regression.

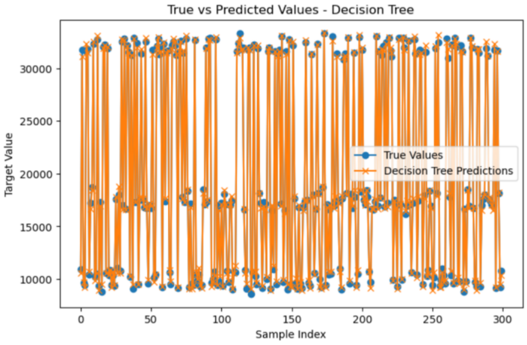

Figure 6. The prediction curve based on decision tree.

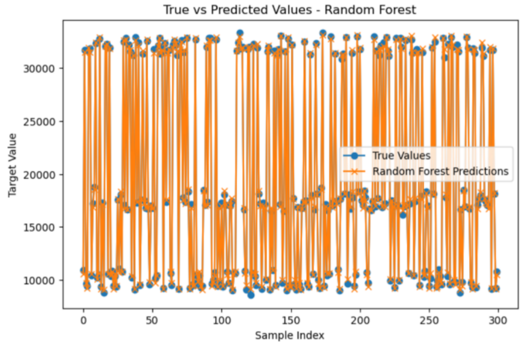

Figure 7. The prediction curve based on random forest.

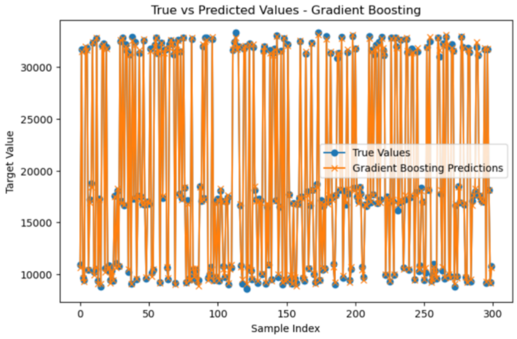

Figure 8. The prediction curve based on gradient boosting.

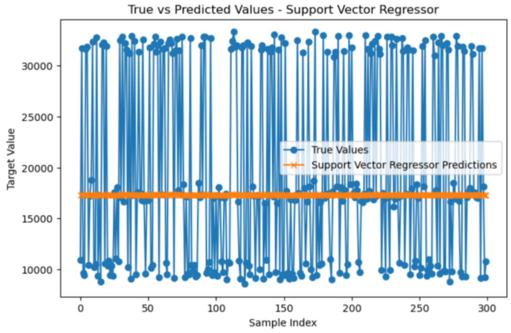

Figure 9. The prediction curve based on support vector regressor.

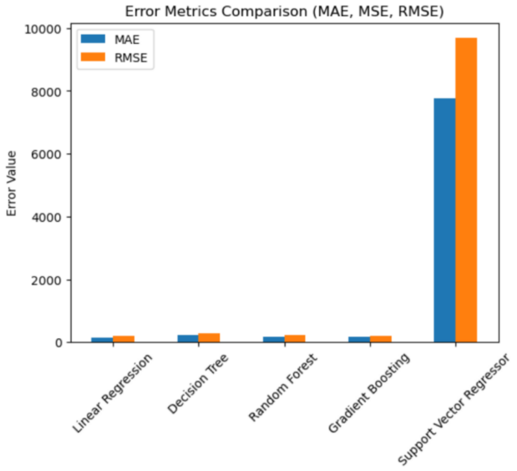

Figure 10. The prediction performance comparison in terms of MAE and RMSE.

The comparative analysis underscores the strengths and limitations of each regression model in handling specific types of data. Linear Regression, Random Forest, and Gradient Boosting demonstrate superior performance and reliability for this dataset, making them preferable choices for similar predictive tasks. The Decision Tree, while less accurate than its ensemble counterparts, offers a simpler, albeit effective alternative. In contrast, SVR's underperformance highlights the critical importance of model and parameter selection based on the nature of the data and the specific requirements of the task. This study thus provides valuable insights into selecting appropriate models for predictive analytics, emphasizing the need for careful consideration of model capabilities and limitations.

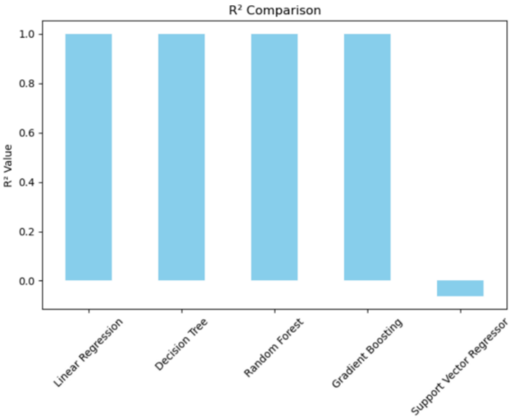

Figure 11. The prediction performance comparison in terms of R².

4.2 The Performance of the Proposed Model Fusion-Based Approach

In this study, we developed and evaluated a novel model designed to enhance prediction accuracy by leveraging the strengths of individual models: Linear Regression, Random Forest, and Gradient Boosting. This proposed model is a fusion that strategically combines the predictions from these models, aiming to capitalize on their individual predictive capabilities to produce a superior unified output.

As detailed in **Table 2**, the proposed model achieves a Mean Absolute Error (MAE) of 149.664, which is lower than that of any individual model used in its construction. Specifically, it outperforms Linear Regression (MAE = 153.643), Random Forest (MAE = 173.225), and Gradient Boosting (MAE = 165.275). Additionally, it attains a Root Mean Square Error (RMSE) of 189.098, also the lowest among the models, further underscoring its enhanced predictive precision.

The superiority of the proposed model is visually represented in **Figure 12**, where the prediction curve demonstrates a closer adherence to the true values compared to the individual models. This visual alignment indicates that the hybrid model not only predicts with lower error margins but also maintains high consistency across the dataset, showcasing a robust balance between bias and variance.

Table 2. The performance of the proposed model compared with the single model.

Model Name	MAE	RMSE	R²
Linear Regression	153.643727	191.737282	0.999584
Random Forest	173.225308	215.310736	0.999476
Gradient Boosting	165.275236	204.513808	0.999527
Proposed model	149. 664622	189.098844	0.999881

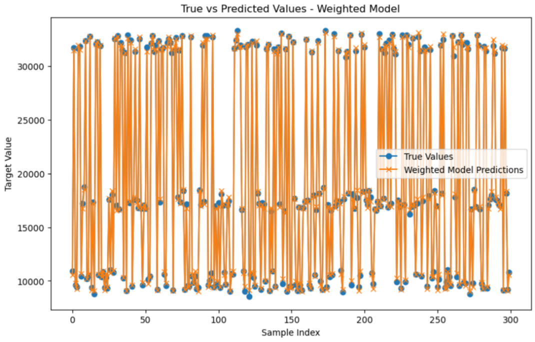

Figure 12. The prediction curve based on the proposed model.

4.3 Discussion

In the discussion of our research findings, it's crucial to acknowledge that while the fusion of multiple models can yield lower error rates and enhanced predictive performance, there are inherent limitations and challenges associated with this approach that must be considered. But it still has great potential in applications in many fields [40–42]. One of the primary drawbacks of model fusion is the increased complexity of the resulting predictive model. Combining multiple models often leads to a system that is more difficult to interpret than its individual components. For instance, while a single model like Linear Regression provides clear insights into the relationship between variables through its coefficients, a hybrid model obscures these relationships under layers of aggregation and weighting. This loss of interpretability can be a significant disadvantage in applications where understanding the model's decision-making process is as important as the accuracy of its predictions, such as in medical or financial contexts where stakeholders must assess risk and causality. Moreover, hybrid models typically require more computational resources. The process of training multiple models and then combining their outputs can be computationally expensive, especially with large datasets or in real-time applications. This increase in computational demand can limit the practicality of deploying such models in environments with constrained resources. While model fusion can mitigate some of the overfitting risks by blending diverse predictive patterns, there's still a risk that the combined model might adapt too closely to the training data, particularly if the individual models are already complex. Ensuring that each component model is well-regularized and that the fusion mechanism does not exacerbate overfitting is crucial. The success of a hybrid model also heavily depends on the performance of its component models. If one or more of the individual models are poorly tuned or fundamentally misaligned with the data's underlying patterns, the overall performance of the hybrid model can be adversely affected. This dependency necessitates careful selection, tuning, and validation of each component model.

5. Conclusion

The implementation of a hybrid model for online product price prediction has demonstrated significant improvements in prediction accuracy by effectively amalgamating the capabilities of Linear Regression, Random Forest, and Gradient Boosting models. The proposed model not only reduces the Mean Absolute Error and Root Mean Square Error but also closely aligns predicted values with actual market prices, as evidenced by a high R^2 value. While this model fusion approach enhances predictive performance, it also introduces challenges such as increased computational demands and potential complexities in model interpretation. Future work should focus on optimizing these hybrid models to balance computational efficiency and ease of interpretation, ensuring they are practical for real-world applications. Moreover, continuous refinement of component models and their integration strategies will be crucial to adapting to evolving market dynamics and maintaining the relevance of the predictive framework in the competitive landscape of online retail.

Author Contribution

Diwei Zhu is responsible for the conceptualization and design of the hybrid model framework as well as the drafting of the manuscript. Xiaoyang Chen is responsible for data preprocessing and feature engineering. Yunxiang Gan lead on experimental analysis and result interpretation.

References

[1] Dai, W., 2022. Evaluation and improvement of carrying capacity of a traffic system. Innovations in Applied Engineering and Technology. 1–9.

[2] Zhao, Z., Ren, P., Tang, M., 2022. Analyzing the impact of anti-globalization on the evolution of higher education internationalization in China. Journal of Linguistics and Education Research. 5(2), 15–31.

[3] Lei, J., 2022. Efficient strategies on supply chain network optimization for industrial carbon emission reduction. Journal of Computational Methods in Engineering Applications. 1–11.

[4] Xiong, S., Zhang, H., Wang, M., Zhou, N., 2022. Distributed data parallel acceleration-based generative adversarial network for fingerprint generation. Innovations in Applied Engineering and Technology. 1–12.

[5] Dai, W., 2021. Safety evaluation of traffic system with historical data based on Markov process and deep-reinforcement learning. Journal of Computational Methods in Engineering Applications. 1–14.

[6] Lei, J., 2022. Green supply chain management optimization based on chemical industrial clusters. Innovations in Applied Engineering and Technology. 1–17.

[7] Xiong, S., Zhang, H., Wang, M., 2022. Ensemble model of attention mechanism-based DCGAN and autoencoder for noised OCR classification. Journal of Electronic & Information Systems. 4(1), 33–41.

[8] Yu, L., Li, J., Cheng, S., Xiong, S., Shen, H., 2013. Secure continuous aggregation in wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems. 25(3), 762–774.

[9] Xiong, S., Yu, L., Shen, H., Wang, C., Lu, W., 2012. Efficient algorithms for sensor deployment and routing in sensor networks for network-structured environment monitoring. In 2012 Proceedings IEEE INFOCOM. 1008–1016.

[10] Feng, Z., Xiong, S., Cao, D., Deng, X., Wang, X., Yang, Y., et al., 2015. Hrs: A hybrid framework for malware detection. In Proceedings of the 2015 ACM International Workshop on Security and Privacy Analytics. 19–26.

[11] Lagarto, J., de Sousa, J., Martins, A., Ferrao, P., 2012. Price forecasting in the day-ahead Iberian electricity market using a conjectural variations ARIMA model. In 2012 9th International Conference on the European Energy Market. 1–7.

[12] Hsu, C.C., Chen, C.Y., 2003. Applications of improved grey prediction model for power demand forecasting. Energy Conversion and Management. 44(14), 2241–2249.

[13] González, A.M., Roque, A.S., García-González, J., 2005. Modeling and forecasting electricity prices with input/output hidden Markov models. IEEE Transactions on Power Systems. 20(1), 13–24.

[14] Xiong, S., Li, J., Li, M., Wang, J., Liu, Y., 2011. Multiple task scheduling for low-duty-cycled wireless sensor networks. In 2011 Proceedings IEEE INFOCOM. pp. 1323–1331.

[15] Yu, L., Li, J., Cheng, S., Xiong, S., 2011. Secure continuous aggregation via sampling-based verification in wireless sensor networks. In 2011 Proceedings IEEE INFOCOM. 1763–1771.

[16] Li, J., Xiong, S., 2010. Efficient Pr-skyline query processing and optimization in wireless sensor networks. Wireless Sensor Network. 2(11), 838.

[17] Yousefi, S., Weinreich, I., Reinarz, D., 2005. Wavelet-based prediction of oil prices. Chaos, Solitons & Fractals. 25(2), 265–275.

[18] Lippmann, R.P., 1988. An introduction to computing with neural nets. ACM SIGARCH Computer Architecture News. 16(1), 7–25.

[19] Kimoto, T., Asakawa, K., Yoda, M., Takeoka, M., 1990. Stock market prediction system with modular neural networks. In 1990 IJCNN International Joint Conference on Neural Networks. 1–6.

[20] Fathalla, A., Salah, A., Li, K., et al., 2020. Deep end-to-end learning for price prediction of second-hand items. Knowledge and Information Systems. 62(12), 4541–4568.

[21] Su, X., Yan, X., Tsai, C.L., 2012. Linear regression. Wiley Interdisciplinary Reviews: Computational Statistics. 4(3), 275–294.

[22] Montgomery, D.C., Peck, E.A., Vining, G.G., 2021. Introduction to linear regression analysis. John Wiley & Sons.

[23] Groß, J., 2012. Linear regression. Springer Science & Business Media.

[24] Tranmer, M., Elliot, M., 2008. Multiple linear regression. The Cathie Marsh Centre for Census and Survey Research (CCSR). 5(5), 1–5.

[25] Song, Y.Y., Ying, L.U., 2015. Decision tree methods: Applications for classification and prediction. Shanghai Archives of Psychiatry. 27(2), 130.

[26] Suthaharan, S., Suthaharan, S., 2016. Decision tree learning. Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning. 237–269.

[27] De Ville, B., 2013. Decision trees. Wiley Interdisciplinary Reviews: Computational Statistics. 5(6), 448–455.

[28] Breiman, L., 2001. Random forests. Machine Learning. 45, 5–32.

[29] Ali, J., Khan, R., Ahmad, N., Maqsood, I., 2012. Random forests and decision trees. International Journal of Computer Science Issues (IJCSI). 9(5), 272.

[30] Louppe, G., 2014. Understanding random forests. Cornell University Library. 10.

[31] Natekin, A., Knoll, A., 2013. Gradient boosting machines, a tutorial. Frontiers in Neurorobotics. 7, 21.

[32] Friedman, J.H., 2002. Stochastic gradient boosting. Computational Statistics & Data Analysis. 38(4), 367–378.

[33] Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. Annals of Statistics. 1189–1232.

[34] Hearst, M.A., Dumais, S.T., Osuna, E., et al., 1998. Support vector machines. IEEE Intelligent Systems and Their Applications. 13(4), 18–28.

[35] Kecman, V., 2005. Support vector machines—An introduction. In Support Vector Machines: Theory and Applications. Springer Berlin Heidelberg. 1–47.

[36] Meyer, D., Wien, F.T., 2001. Support vector machines. R News. 1(3), 23–26.

[37] Hodson, T.O., 2022. Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. Geoscientific Model Development Discussions. 2022, 1–10.

[38] Wang, W., Lu, Y., 2018. Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. In IOP Conference Series: Materials Science and Engineering. 324, 012049.

[39] Miles, J., 2005. R-squared, adjusted R-squared. Encyclopedia of Statistics in Behavioral Science.

[40] Xiong, S., Li, J., 2010. An efficient algorithm for cut vertex detection in wireless sensor networks. In 2010 IEEE 30th International Conference on Distributed Computing Systems. 368–377.

[41] Xiong, S., Li, J., 2009. Optimizing many-to-many data aggregation in wireless sensor networks. In Asia-Pacific Web Conference. Springer Berlin Heidelberg. 550–555.

[42] Wang, H., Li, J., Xiong, S., 2008. Efficient join algorithms for distributed information integration based on XML. International Journal of Business Process Integration and Management. 3(4), 271–281.