FNAS Journal of Mathematical Modeling and Numerical Simulation
Print ISSN: 3027-1282
www.fnasjournals.com
Volume 2; Issue 4; December 2025; Page No. 98-113.
DOI: https://doi.org/10.63561/jmns.v2i4.1127

Predicting Red Wine Sales Using Deep Learning Algorithms for Market Performance Optimisation

[1]Nwali, M.E., [1]Nwele, C., [2]Ugah, J.O., & *[1]Onwudebelu, U.
[1]Department of Computer Science/Informatics, Alex Ekwueme Federal University Ndufu Alike (FUNAI), Abakaliki, Ebonyi State, Nigeria.
[2]Department of Computer Science, Ebonyi State University, Ebonyi State, Nigeria..

***Corresponding author email**: ugochukwu.onwudebelu@funai.edu.ng

Abstract

The demand for accurate sales forecasting within the wine industry has increased as producers, distributors, and retailers seek to improve production planning, inventory management, and marketing strategies. This study presents a deep learning-based approach for forecasting red wine sales using time-series data. Traditional statistical methods such as ARIMA and Exponential Smoothing have demonstrated limited performance when applied to volatile, nonlinear datasets typical of wine markets. To overcome these challenges, the proposed model employs recurrent neural networks (RNNs) and other deep learning architectures to capture long-term dependencies and complex relationships among features. The model was developed using Python and evaluated against traditional statistical models using metrics such as Mean Squared Error (MSE). The results demonstrate that deep learning methods outperform classical models in predictive accuracy and robustness. This research provides a foundation for intelligent forecasting systems that can enhance market optimization and support data-driven decision-making in the wine industry.

Keywords: Red wine forecasting, Deep learning, Time series analysis, Sales prediction, Neural networks

Introduction

Accurate sales forecasting is essential in the wine industry, where production planning, logistics, and marketing activities depend on anticipating consumer demand. The red wine market is particularly volatile due to seasonality, shifting consumer preferences, and broader economic fluctuations. Traditional forecasting approaches such as ARIMA have been widely applied; however, their linear structure limits their ability to model complex, nonlinear sales patterns. Advances in machine learning and deep learning now offer more robust alternatives capable of capturing long-term dependencies and intricate temporal relationships in time-series data. This study investigates the use of deep learning algorithms to predict red wine sales, with the goal of developing a robust forecasting framework that supports market optimization. The importance of accurate sales forecasting extends beyond planning and is fundamental to effective retail operations. Retailers rely on reliable forecasts to optimize inventory purchasing, balance stock levels, and minimize both stockouts and overstocking (Agatz et al., 2006). Precise forecasts also enhance logistics coordination by informing warehouse capacity planning, transportation scheduling, and labor allocation. In marketing, sales projections guide promotional strategies, budget allocation, and channel selection. From a financial perspective, forecasts support budgeting, revenue estimation, and cash-flow planning. Human resource planning similarly benefits from demand visibility, enabling retailers to align staffing levels with expected sales volumes (Bower & Maxham, 2012). Global forecasting models provide an additional advantage by learning patterns across multiple related time series. These models can mitigate the noise and irregularities present in individual series, resulting in more stable and accurate predictions (Babić Rosario et al., 2016).

Cite this article as:
Nwali, M.E., Nwele, C., Ugah, J.O., & Onwudebelu, U. (2025) Predicting red wine sales using deep learning algorithms for market performance optimisation. *FNAS Journal of Mathematical Modeling and Numerical Simulation*, *2*(4), 98-113. https://doi.org/10.63561/jmns.v2i4.1127

Current research on machine learning applications within alcoholic-beverage production has primarily focused on quality assessment, flavor profiling, and sensory analysis of Western wines. Models such as QSOR (Schwindt et al., 2023) predict sensory characteristics by establishing relationships between volatile compounds and perceived flavor attributes. Similarly, the Pi model proposed by Tiwari et al. (2022) applies the Buckingham Pi theorem to quantify mathematical relationships among chemical components in wine, demonstrating the potential of data-driven modeling in enology. The growth of electronic commerce has also transformed purchasing behavior, with online shopping becoming increasingly popular. However, many e-stores still function more like automated vending systems, offering limited engagement because they lack human-like sales interaction. To address this gap, Huang and Lin (2005) developed an autonomous virtual sales agent designed to simulate the role of a clerk in online retail environments. Their intelligent agent, known as Isa, dynamically adapts its persuasion and negotiation strategies based on buyer characteristics and learns effective selling behaviors without explicit instructions from the retailer. Results from their field experiment demonstrated that Isa successfully increased customers' product evaluations, willingness to pay, and overall satisfaction with the e-store, while simultaneously enhancing seller surplus. This highlights the broader role of intelligent systems in improving digital commerce performance and shaping consumer purchasing decisions.

Object-oriented programming languages (OOPLs) remain foundational in modern information technology; however, many widely used languages were not originally designed with security as a primary consideration. During software development, coding errors and overlooked loopholes often result in vulnerabilities, which can pose significant risks to both individuals and organizations when exploited. The severity of these vulnerabilities depends largely on the underlying coding flaws, and while some developers inadvertently produce insecure code, others remain unaware of the associated security implications. Onu et al. (2023) investigated common security vulnerability patterns across four popular OOPLs — Java, C++, Ruby, and Python — with particular emphasis on Python. Using data from the Common Vulnerabilities and Exposures (CVE) database, they applied Moving Average Forecast (MAF) and Weighted Moving Average Forecast (WMAF) techniques to predict vulnerability patches for 2023. Their findings revealed that WMAF outperformed MAF in forecasting accuracy. The study further offered recommendations for minimizing future security risks.

Time-series techniques have also been widely applied to sales forecasting, with evidence showing that combined forecasting models often outperform single-model approaches. Vijayalakshmi et al. (2010) designed an intelligent forecasting engine using genetic algorithms to select optimal model combinations, demonstrating the potential of evolutionary methods in enhancing time-series forecasting accuracy. Global forecasting models such as recurrent neural networks (RNNs) leverage information across multiple related time series, offering improved predictive power compared to univariate approaches like ARIMA (Januschowski et al., 2020). Machine learning continues to advance forecasting applications across domains. Islam et al. (2021) employed an Extreme Gradient Boosting (XGB) framework to model US Dollar–Indonesian Rupiah exchange rates, achieving strong performance with RMSE of 0.2358 and MAPE of 0.1164%. Similarly, online product reviews have been shown to influence pricing behavior, with e-tailers tending to reduce prices when faced with increasing negative reviews (Duan et al., 2022).

Classical time-series forecasting methods such as ARIMA and Exponential Smoothing have been widely applied due to their simplicity and effectiveness under stable conditions. However, these linear models often fail to capture the nonlinear and rapidly changing dynamics of modern markets. Machine learning approaches—including decision trees, random forests, and support vector machines—address some of these limitations by modeling nonlinear relationships, yet they remain constrained in their ability to represent sequential dependencies inherent in time-series data. Deep learning architectures such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs) overcome this challenge by learning temporal patterns directly from data. Empirical studies by Makridakis et al. (2022a) and Livieris et al. (2021) show that LSTM-based models consistently outperform traditional statistical approaches in time-series forecasting, reinforcing their suitability for applications such as red wine sales prediction. In related studies, Ugah and Onwudebelu (2023), Onwudebelu et al. (2024), and Onwudebelu et al. (2025) applied predicting techniques not to wine sales but to meteorological data, specifically the NiMet weather dataset. Using a curated 35-year dataset, they trained machine learning models—including Logistic Regression, Random Forest (RF), and XGBoost—to predict flooding occurrences. Their findings demonstrated strong

Cite this article as:
Nwali, M.E., Nwele, C., Ugah, J.O., & Onwudebelu, U. (2025) Predicting red wine sales using deep learning algorithms for market performance optimisation. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(4), 98-113. https://doi.org/10.63561/jmns.v2i4.1127

predictive capabilities, with RF and XGBoost achieving high classification accuracy and well-calibrated probability estimates, attaining an AUC of approximately 0.98.

Despite these advances, traditional machine learning models are not inherently designed to process sequential data. As a result, time-series datasets must be transformed into tabular form through feature engineering (Alihodžić et al., 2022). Common techniques include generating lag features, calculating moving averages, applying differencing operations, and incorporating temporal categorical variables such as day of week, month, and season (Bergmeir & Benítez, 2017). Handling missing values typically involves forward-filling to preserve temporal consistency. Moreover, data splitting and cross-validation must respect chronological order to avoid data leakage. However, converting time-series data into tabular form may result in the loss of temporal dependencies that cannot be fully captured through engineered features alone (Elsayed et al., 2021). Several studies have explored machine learning applications in dynamic retail environments. Ferreira et al. (2016) developed a regression-tree with bagging model for predicting demand in an e-commerce platform offering highly discounted apparel for short durations. Due to the nonparametric nature of regression trees, conventional linear programming techniques were unsuitable for price optimization; thus, the authors introduced a modified algorithm tailored to the retailer's constraints. In another retail-focused study, Ito and Fujimaki (2017) modeled three customer types to determine revenue-maximizing prices. Given the small dataset, deep learning was unsuitable; instead, an AutoML architecture was applied to efficiently evaluate a large range of models, while Bayesian Optimization was used to determine optimal prices through an efficient search process.

Ensemble methods have gained prominence in time-series modeling due to their ability to combine the strengths of multiple algorithms. Tree-based ensembles such as boosting and bagging are particularly effective. Boosting methods iteratively add weak learners—often decision trees—by minimizing a loss function and correcting errors from earlier estimators (Chen et al., 2021). These ensemble techniques have demonstrated improved accuracy, robustness, and resistance to overfitting when compared to individual models (Dietterich, 2017). In regression and classification tasks, k-fold cross-validation (CV) is commonly used to evaluate model performance. The dataset is shuffled and partitioned into k equally sized folds; each fold is used once as a test set while the remaining folds serve as the training set, resulting in k performance estimates (Hastie et al., 2017). This approach ensures all observations are used for both training and validation and reduces bias in model evaluation. However, because k-fold CV assumes independently and identically distributed (i.i.d.) data (Cerqueira et al., 2017), the required shuffling violates the temporal ordering of observations. Consequently, traditional k-fold CV is unsuitable for time-series forecasting.

Deep learning models have demonstrated strong suitability for time-series modeling due to their ability to learn nonlinear and hierarchical data representations. Modern neural architectures—successfully applied in domains such as image classification, speech recognition, and natural language processing—can also learn temporal dependencies directly from sequential data. Their capacity for generalization enables deep models to capture interactions among multiple related series, facilitating multivariate forecasting using a single architecture. Despite these advantages, deep learning models require extensive hyperparameter tuning, which is computationally intensive and may lead to suboptimal configurations (Krauss et al., 2017). Makridakis et al. (2022a) conducted a large-scale evaluation in a forecasting competition involving over 40,000 hierarchical product sales series from a major retailer. Many of the top-performing teams utilized Light Gradient Boosting Machines (LGBMs), which handle heterogeneous feature types efficiently and offer favorable computational performance compared to other boosting methods. LGBMs are particularly suitable for forecasting tasks involving numerous correlated series (Makridakis et al., 2022b). A notable trend among the leading submissions was forecasting at higher aggregation levels rather than solely at the item level to reduce volatility. Ensembles comprising multiple models across different forecast horizons and hierarchical levels were also frequently employed to enhance predictive accuracy.

Beyond retail forecasting, machine learning has also been applied to modeling complex fermentation processes. Koji making—a critical step in liquor production—involves intricate biochemical interactions influenced by environmental conditions, resulting in highly variable quality outcomes (Zhang et al., 2023b). Predicting multiple koji quality scores is challenging due to heterogeneous reproductive behaviors and uneven score distributions across koji levels. To address this, Zhang et al. (2023a) proposed a multi-task learning framework that utilizes a shared representation layer to extract common features while learning multiple rating tasks simultaneously. Experimental evaluations on real-world koji-making data showed that this approach outperformed benchmark models, improving the accuracy and robustness of koji score prediction.

The aim of this research is to develop a deep learning–based model for forecasting red wine sales. The objectives are as follows:

i. To design and implement a red wine sales forecasting model using deep learning techniques.
ii. To evaluate the performance of the developed forecasting model.
iii. To generate actionable insights from the model's predictive outputs

Research Methodology

This study employed an Object-Oriented Design (OOD) methodology to guide system development. Historical red wine sales data were sourced from Kaggle and subjected to preprocessing steps including normalization, handling missing values, and feature engineering to enhance model performance. A Recurrent Neural Network (RNN) was implemented using Python, TensorFlow, and Keras to predict future sales based on sequences of past observations. The model's accuracy was assessed using Mean Squared Error (MSE). The recurrent layers in the architecture enabled the model to capture temporal dependencies and learn underlying sales patterns effectively.

Data Gathering Technique

Data required for this research were obtained primarily through secondary sources. Relevant information was collected from journals, magazines, newspapers, library resources, and online repositories. The dataset used for model training and evaluation was downloaded from Kaggle, where it was already compiled and labeled for analytical purposes. Secondary data collection provided a cost-effective and efficient means of acquiring large volumes of structured information essential for system analysis.

Analysis of the Existing System

Existing price forecasting systems generally utilize two main approaches: quantitative and qualitative methods.

Quantitative Approach: This approach relies on historical numerical data and applies mathematical or statistical techniques—such as time-series analysis, regression, and machine learning—to discover patterns and trends. Quantitative models generate numerical forecasts and are widely used for structured, data-driven prediction tasks.

Qualitative Approach: The qualitative method incorporates non-numerical information such as market trends, expert opinions, consumer behavior, and environmental factors. Techniques such as surveys, interviews, and focus groups are used to capture emerging patterns. While subjective, qualitative insights are valuable in contexts where numerical data may be limited or insufficient.

Advantages of the Existing System

A key advantage of the existing forecasting framework is its integration of price elasticity into demand prediction. After generating a one-day-ahead baseline forecast, the system adjusts expected demand for minimum and maximum retail prices using learned price elasticity functions. This produces three demand estimates per product—corresponding to low, base, and high price points. Linear programming with relaxation is then applied to determine the price that maximizes revenue. Empirical evaluations show that this approach improves company revenue by approximately 1%.

Disadvantages of the Existing System

The main limitation of the existing system lies in its reliance on a non-parametric regression tree–based model. Due to the structural characteristics of regression trees, traditional linear programming could not be directly applied for optimization. As a result, the authors developed a modified price optimization algorithm tailored to the model's constraints. Although effective, this workaround adds complexity and limits scalability.

High-Level Model of the Proposed System

Cite this article as:
Nwali, M.E., Nwele, C., Ugah, J.O., & Onwudebelu, U. (2025) Predicting red wine sales using deep learning algorithms for market performance optimisation. *FNAS Journal of Mathematical Modeling and Numerical Simulation*, *2*(4), 98-113. https://doi.org/10.63561/jmns.v2i4.1127

High-level models provide simplified representations of the system using diagrams, formulas, or descriptive text to support design decisions. The high-level model of the proposed system (Figure 2) illustrates the workflow from user interaction to forecast generation. In the proposed system, a user creates an account and logs into the platform. After selecting a target date for forecasting, the system processes the input and generates a sales prediction. The result is visualized in a plotted graph where the Y-axis represents the predicted sales values and the X-axis represents the corresponding dates or time steps. This visualization supports intuitive interpretation and decision-making.

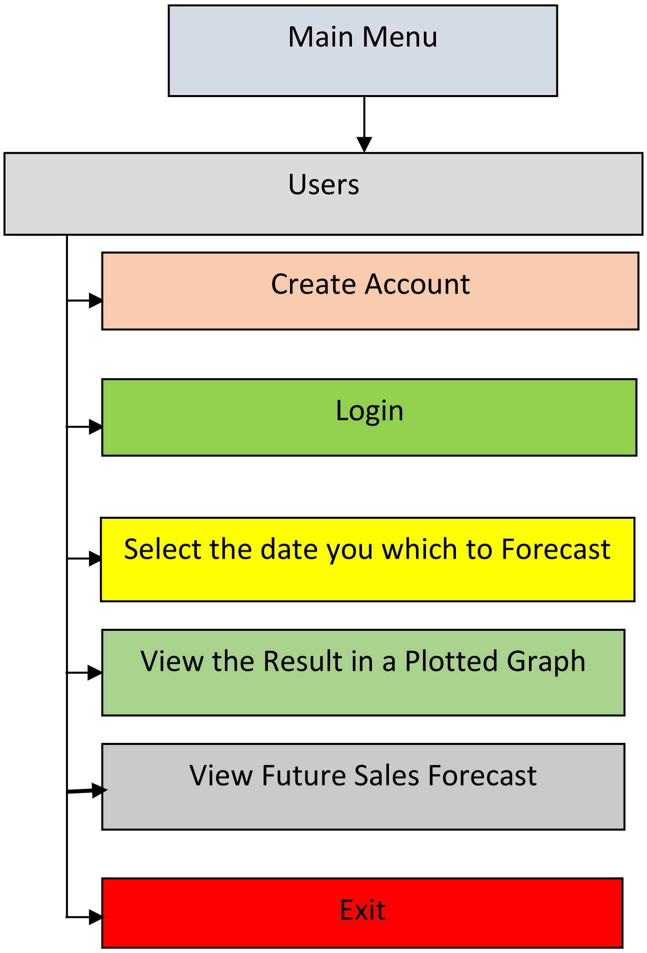

Main Menu
Users
Create Account
Login
Select the date you which to Forecast
View the Result in a Plotted Graph
View Future Sales Forecast
Exit

Figure 2: High Level Model of the Proposed System

Analysis of the Proposed System

The proposed system utilizes a deep learning–based forecasting model to predict red wine sales. Users begin by creating an account and logging into the platform. Upon successful authentication, the system presents the main interface, where users select a target date—comprising day, month, and year—for forecasting. The system validates the selected date to ensure correctness before processing the request. Once validated, the model analyzes historical data and generates a forecast for the specified period. Results are presented visually using a plotted graph, where the X-axis represents the forecasted sales values and the Y-axis denotes the corresponding time intervals. This visualization enhances interpretability and supports data-driven decision-making.

Justification of the Proposed System

The proposed system effectively addresses limitations observed in existing red wine sales forecasting methods. Current systems often fail to deliver accurate predictions, leading to stockouts, lost sales, and inventory mismanagement. In contrast, the developed model leverages deep learning to produce reliable forecasts of future sales trends. By providing precise sales predictions and graphical representations, the model supports better inventory planning and reduces risks related to overstocking or understocking. This improvement demonstrates the added value of integrating advanced forecasting techniques into sales management systems.

System Design

The system design aims to meet user requirements while optimizing the use of available software and hardware resources. A top-down design approach was adopted to decompose system requirements into well-structured high-level components. The system interfaces were developed using front-end technologies such as HTML, CSS, and JavaScript, providing an intuitive and user-friendly experience. The forecasting engine was implemented in Python, which supports efficient handling of numerical computations and deep learning model execution. This design approach ensures modularity, scalability, and maintainability, enabling seamless system enhancement and future integration of additional features.

System Architecture

The proposed system follows a client–server architecture, which offers several advantages:

a. *Scalability*: Client–server systems allow the incremental addition of servers to handle increased data processing or storage demands. When a server becomes overloaded, additional servers can be deployed to distribute the workload efficiently.
b. *Heterogeneous Client and Server Support:* The architecture supports multiple device types and operating systems, enabling flexibility in deployment. Middleware ensures compatibility by translating communication between systems using different vendor software.
c. *Separation of Concerns*: Thin client–server models following Internet standards clearly separate presentation logic, application logic, and data access logic. For example, presentation logic is implemented using HTML or XML, ensuring modularity and ease of maintenance.
d. *Fault Tolerance*: If a server within the system fails, only applications dependent on that server are affected. The failed server can be replaced without disrupting the entire system, ensuring higher availability and system reliability.

This architectural design promotes robustness, scalability, and interoperability, making it well-suited for web-based forecasting applications.

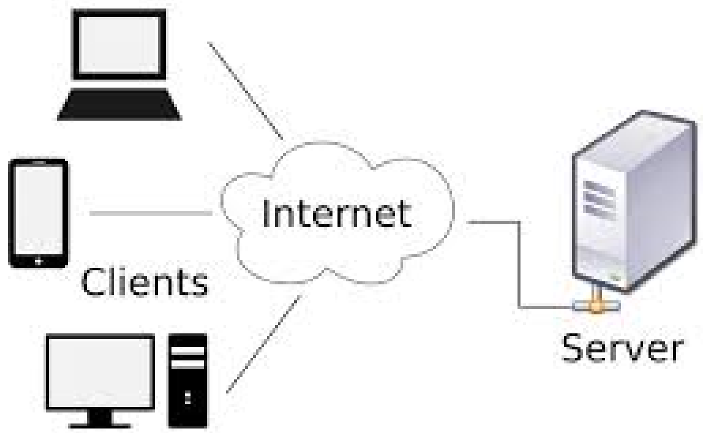


Figure 3: A Two-tier Client-Server System Architecture

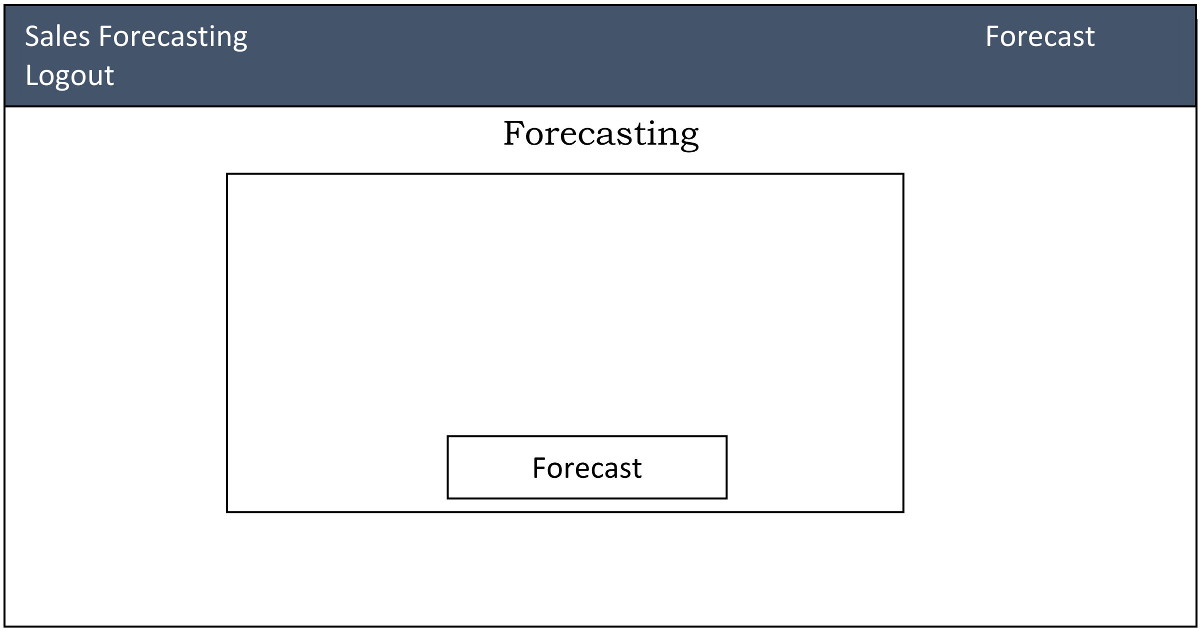
Figure 3 illustrates the client–server architecture adopted in this system. In this structure, the server handles data

Cite this article as:
Nwali, M.E., Nwele, C., Ugah, J.O., & Onwudebelu, U. (2025) Predicting red wine sales using deep learning algorithms for market performance optimisation. *FNAS Journal of Mathematical Modeling and Numerical Simulation*, *2*(4), 98-113. https://doi.org/10.63561/jmns.v2i4.1127

management and processing tasks, while the client focuses on executing the application interface and presenting information to the user. Because the architecture consists of only two interconnected components—the client and the server—it is classified as a two-tier architecture.

Main Menu Design

The main menu serves as the primary navigation interface, providing users with access to different functional modules of the application through a set of organized links and buttons. During the design process, careful attention was given to the layout, positioning, and structuring of these elements to promote ease of use and enhance the overall user experience. The layout of the main menu is presented in *Figure 4*.



Sub Menu Design

The sub menu design is the interface design for the login input into the system. The interface is displayed immediately the user gains access to the model. *Figure 5* shows the input before login can be successful. Figure 5 is the interface that's display whenever a user wants to login or access the model.

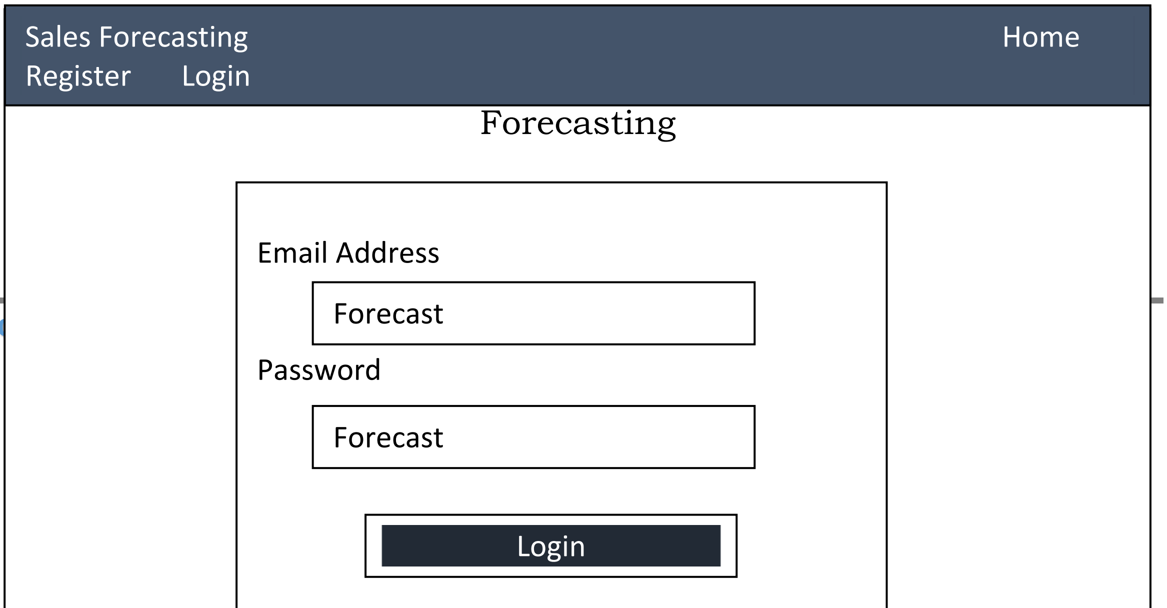


Figure 5: Login Interface Sub Menu Design

Figure 5: Sales Forecasting Register Login

Program Module Design

The developed system is a one user type, which consists of the users that registers with the system. The flowchart in figure 6 shows the flow of information in the design.

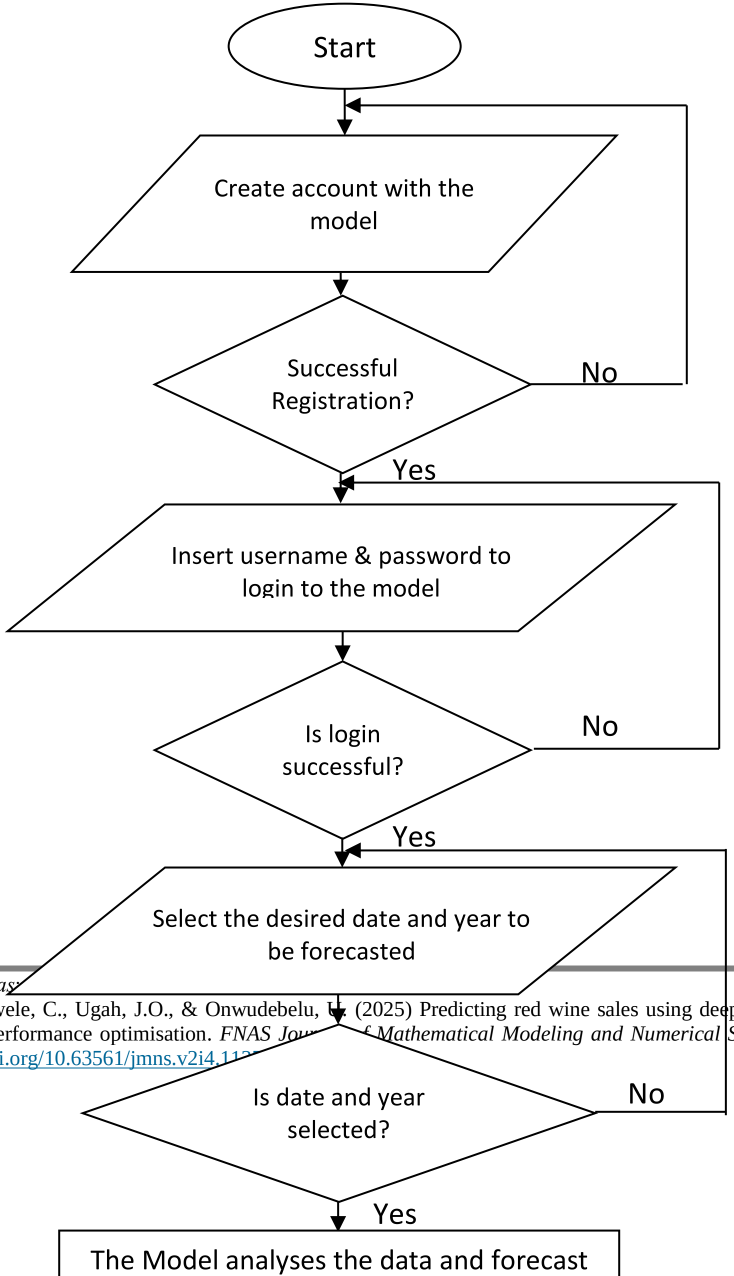

Start
Create account with the model
Successful Registration?
No
Yes
Insert username & password to login to the model
Is login successful?
No
Yes
Select the desired date and year to be forecasted
Is date and year selected?
No
Yes
The Model analyses the data and forecast

Figure 6: Program Module

Figure 6 is the program Module showing full process of the design where a user will first of all register with the model to acquire username and password to enable the user login to the system. If login is success then the user can input some parameters to predict future sales forecasting.

Database Design and Structure

One database was used; the name of the database is wine_sales was created with a sql statement “create database”, it contains relational table that stores information of registered users. The table structure is shown in table 1 and 2.

Table 1: Users

FIELD NAME	DATA TYPE	FIELD SIZE
Id	Int	11
Username	Varchar	15
Password	Varchar	15

Table 1 is the table that contains users’ login details

Table 2: Registration

FIELD NAME	DATA TYPE	FIELD SIZE
Id	Int	11
Fullname	Varchar	30
Email	Varchar	25
Password	Varchar	15
Confirm_password	Varchar	15

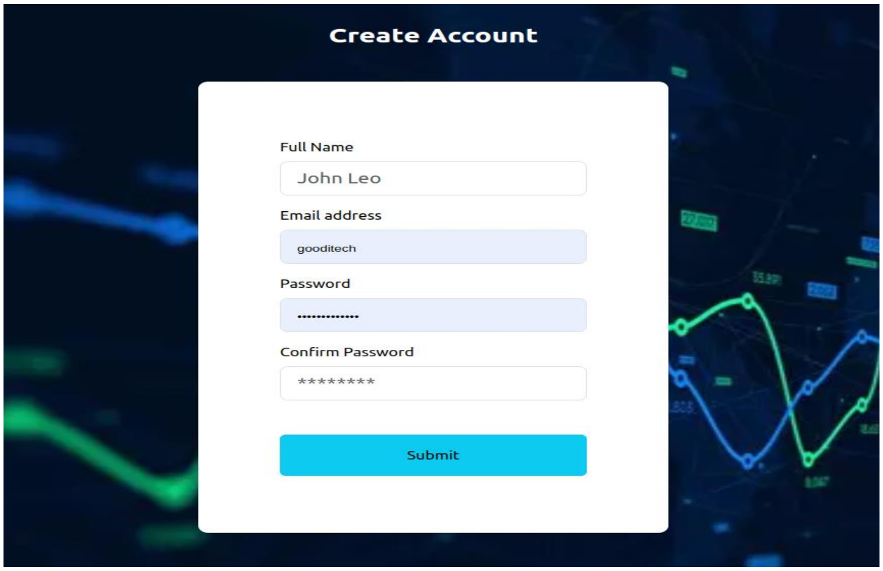
Table 2 presents the complete set of registration information provided by users during account creation.

System Implementation

The implementation phase represents a critical component of the Software Development Life Cycle (SDLC), as it transforms the system design into an operational software product. In this stage, design specifications are translated into executable code, enabling the system to function as intended. Once the coding process is completed, several implementation activities follow, including testing, documentation, deployment, and user training. The system was deployed using a XAMPP server environment to support web execution and database connectivity. During testing, user interactions with the newly developed application were monitored to verify system behavior and identify potential issues. Detected errors were corrected to ensure alignment with current web development requirements and to enhance system reliability. After comprehensive testing and refinement, the application was prepared for organizational deployment and end-user adoption. The hardware requirement includes: (i) NVIDIA GeForce RTX 2060/2070/2080 or RTX 3060/3070/3080: These offer a good balance of performance and cost. (ii) Intel Core i7 or i9 / AMD Ryzen 7 or 9: A powerful multi-core CPU to handle data preprocessing, augmentation, and other tasks. (iii) 16 GB RAM minimum: Adequate for smaller datasets. (iv) 500 GB to 1 TB: For faster data access and to store the dataset, model checkpoints, and logs. (v) Ubuntu 18.04 or 20.04 LTS or Windows 10/11 is require. For the software requirement, the following are needful: (i) Python 3.6 or higher (ii) CUDA Toolkit: Compatible version for your GPU. (iii) cuDNN: Corresponding version with CUDA.

Main Menu Implementation

After a user successfully logs in and their credentials are validated, the system displays the main menu interface. This interface serves as the central navigation hub, allowing users to access different system modules and carry out essential functions. The implemented main menu layout is presented in *Figure 7*, reflecting the final structure developed during the implementation phase.



Create Account

Full Name: John Leo

Email address: gooditech

Password: ....

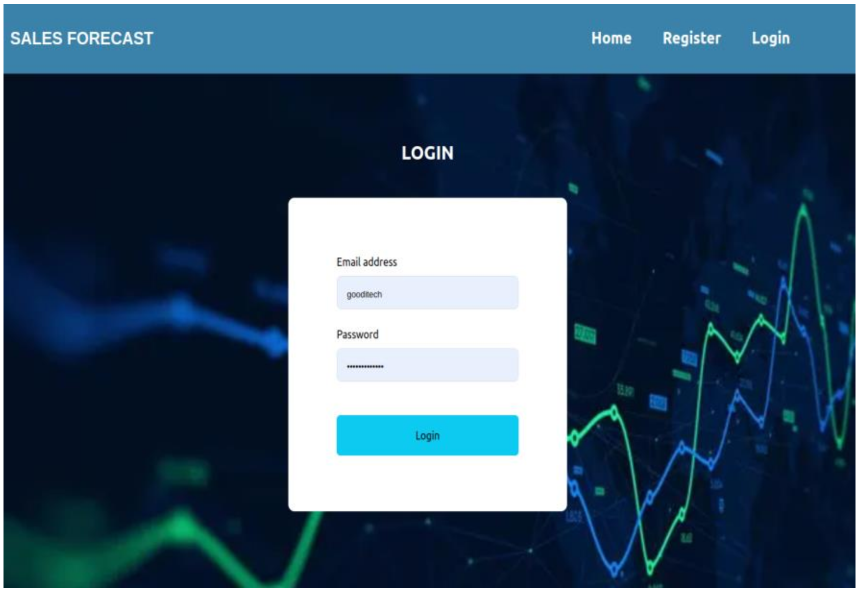
Confirm Password: ********

Figure 7: Implementation of the Main Menu

Figure 7 display after successful login from users.

Sub Menu Implementation

There are different sub menu in the system, one of them is the input sub menu implementation. The sub menu implementation is shown in the Figure 8. Figure 8 is the login input where users can supply their correct username and password to gain access to the main menu of the system.



Email address

goodtech

Password

••••••••••••

Figure 8: Login Input Sub Menu Implementation

Results

The experimental results indicate that the deep learning model achieved superior performance compared to traditional statistical methods. The RNN-based model effectively captured the nonlinear and seasonal trends in red wine sales data. The Mean Squared Error (MSE) demonstrated a significant reduction, confirming the model's improved predictive capacity. The visualization of forecast results revealed stable and consistent predictions over extended periods. These findings suggest that deep learning techniques can enhance decision-making in inventory management, marketing, and production planning. After successfully inputting the parameters, the system will automatically use the historical data to forecast future red wine sales price. The red wine sales forecast result is showed in Figure 9.

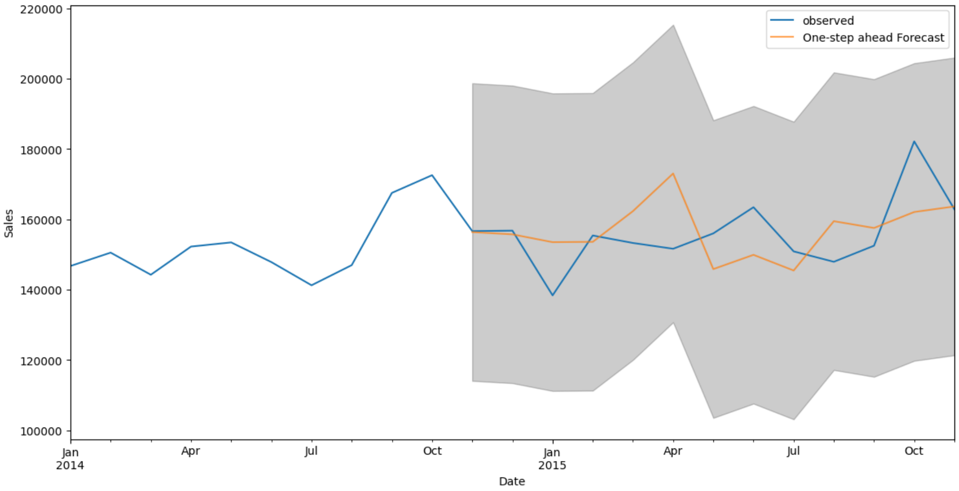


Figure 9: The Red Wine Sales Forecast

Cite this article as:
Nwali, M.E., Nwele, C., Ugah, J.O., & Onwudebelu, U. (2025) Predicting red wine sales using deep learning algorithms for market performance optimisation. *FNAS Journal of Mathematical Modeling and Numerical Simulation*, *2*(4), 98-113. https://doi.org/10.63561/jmns.v2i4.1127

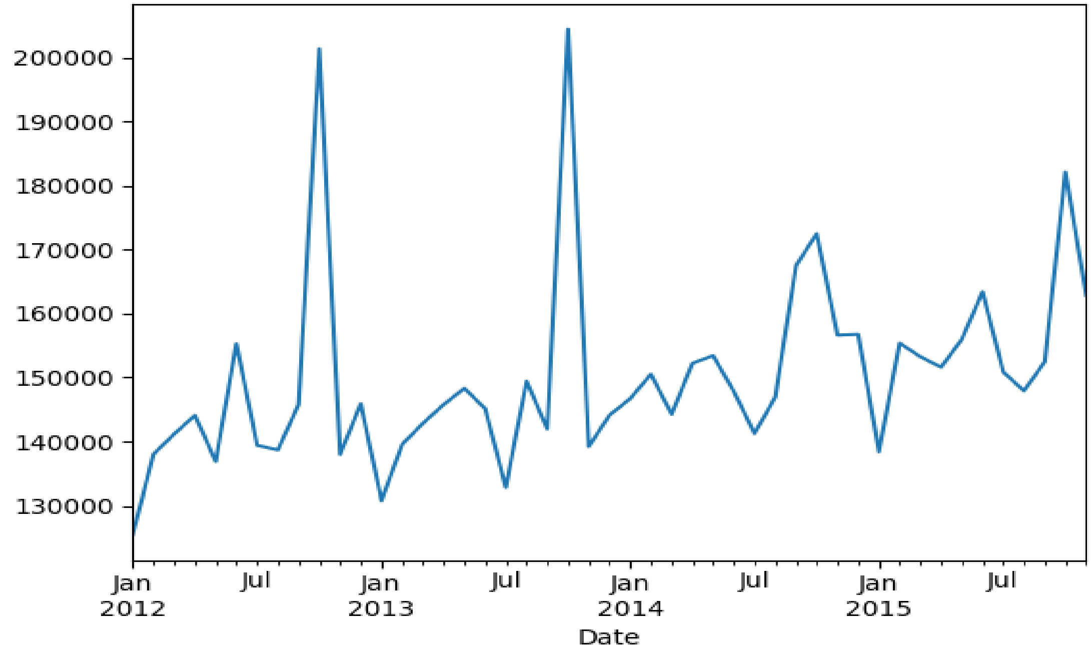


Figure 10: Price Trend within 4 Years of Red Wine

Figure 10 shows the chart of the price trend within 4 years of red wine. While Figure 11 is the chart of price fluctuation in the dataset overtime.

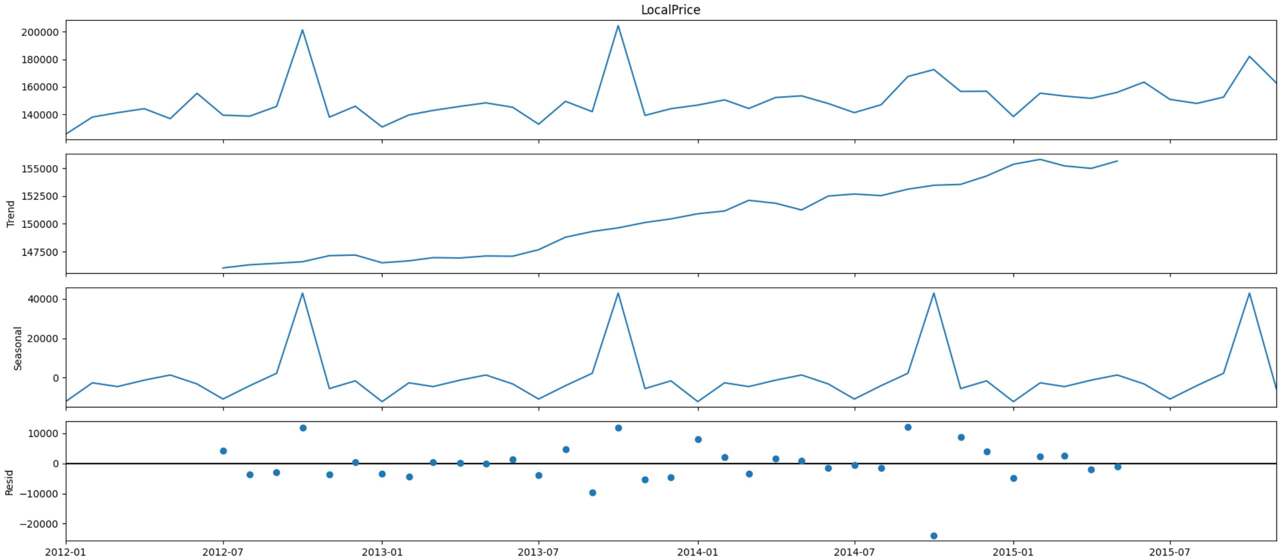


Figure 11: The chart shows Price Fluctuation in the Dataset Overtime

Discussion

The plot illustrates the standardized residuals across time and includes three diagnostic components. The *Normal Q–Q plot* compares the residual quantiles with those of a standard normal distribution to assess normality. The *Histogram* with estimated density displays the distribution of the residuals alongside a kernel density estimate (KDE) and an overlaid theoretical normal curve , indicating that the residuals closely follow a normal distribution. Finally, the *Correlogram* presents the autocorrelation function (ACF) of the residuals, with the shaded region denoting the confidence intervals used to evaluate temporal dependence.

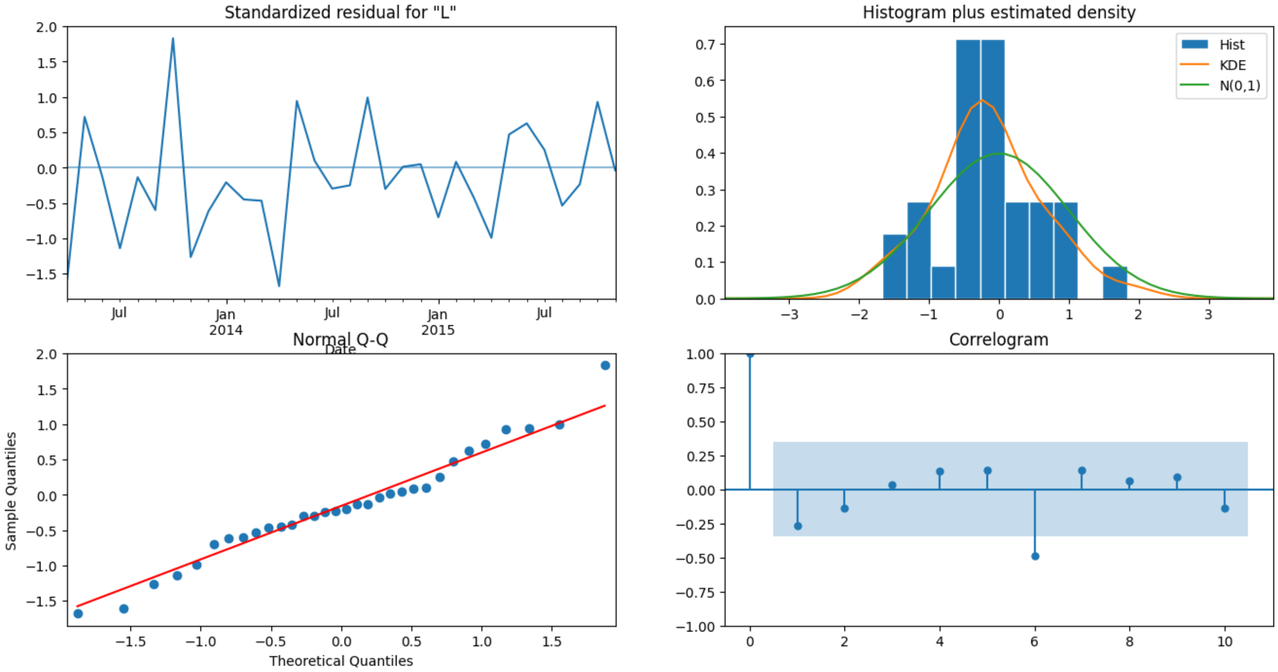


Figure 12: Standardized Residual plot

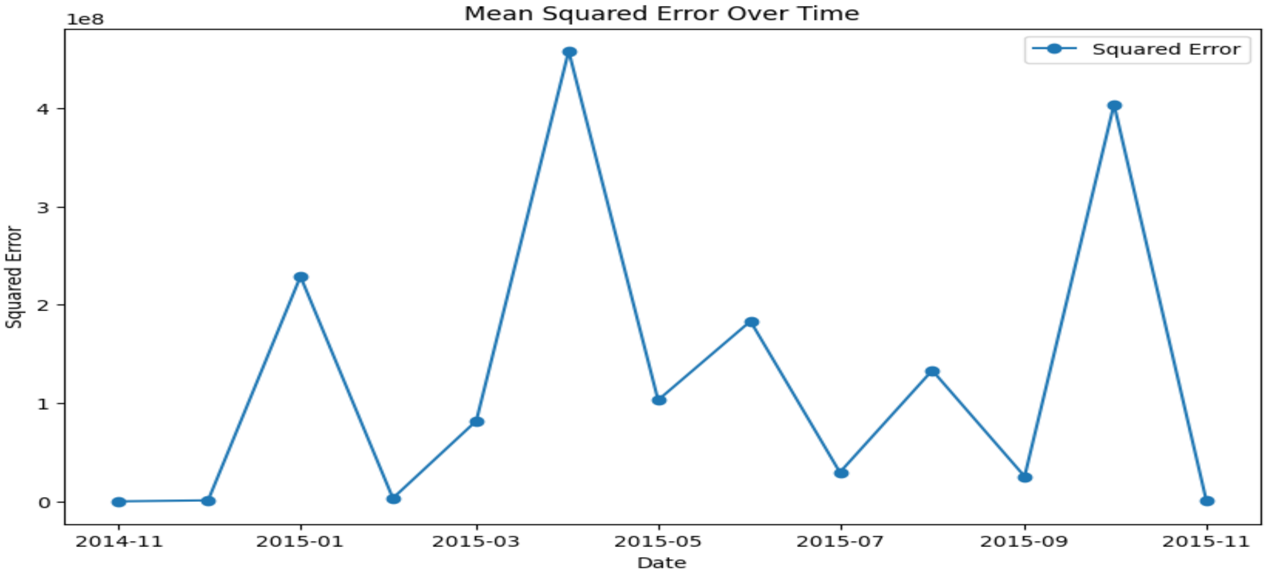


Figure 13: Mean Squared Error (MSE)

Cite this article as:
Nwali, M.E., Nwele, C., Ugah, J.O., & Onwudebelu, U. (2025) Predicting red wine sales using deep learning algorithms for market performance optimisation. *FNAS Journal of Mathematical Modeling and Numerical Simulation*, *2*(4), 98-113. https://doi.org/10.63561/jmns.v2i4.1127

Figure 13 displays the Mean Squared Error (MSE) of the SARIMAX model over time, which is a measure of the average squared differences between the actual values and the model's predictions.

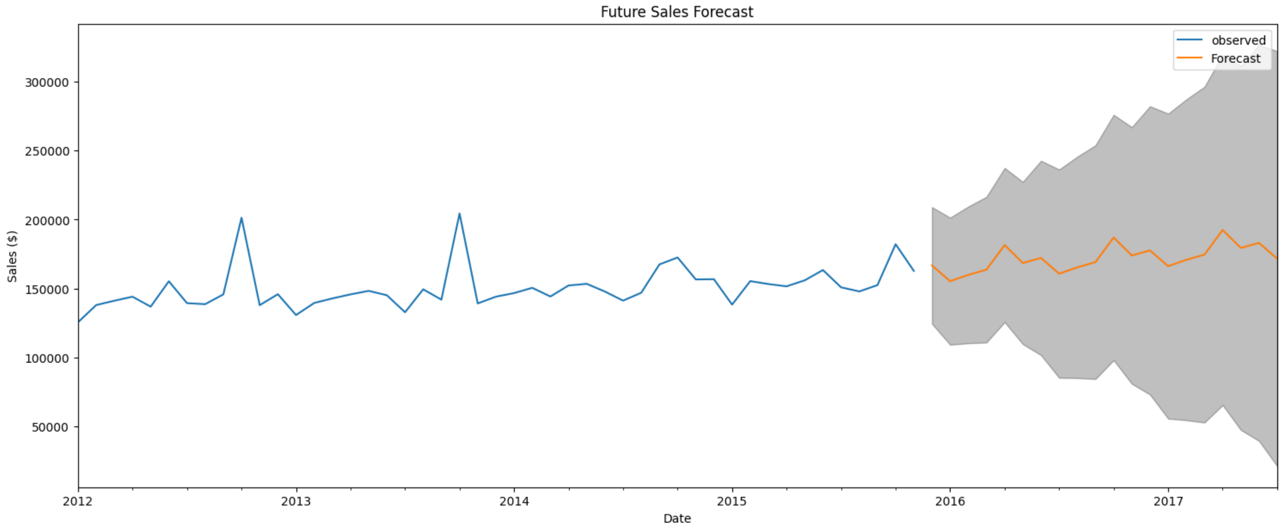


Figure 14: Future Sales Forecast Based on Historical data using SARIMAS model

Figure 14 presents a future sales forecast based on historical data using SARIMAX model. The forecasted sales (orange line) seem relatively stable compared to historical data, suggesting that the model predicts steady sales without large spikes or drops.

A. Performance Evaluation

Performance evaluation involves modeling, measuring, and assessing the operational efficiency of computing and analytical systems. In this study, performance evaluation was conducted to compare different cross-validation techniques for time-series forecasting. The training dataset consisted of 301 days of sales data per product. To ensure that the validation procedure respected the temporal structure and weekly seasonality patterns, the dataset was divided into ten folds. Each fold contained approximately 30 days of observations. For k-fold–based cross-validation variations, each fold served as the test set once while the remaining folds were used for training. For window-based cross-validation methods, the initial training window also contained 30 days of data. This configuration provided a consistent framework for evaluating model performance while maintaining the chronological order required for time-series forecasting.

B. Limitation of the Results

The primary limitation of this research is the restricted size of the dataset. Only one year of data was collected, which is insufficient for capturing long-term patterns, multi-year seasonality, or evolving sales trends—common requirements in time-series forecasting. After preprocessing and feature engineering, only 301 days of usable data per product remained. As a result, the model lacks exposure to seasonal fluctuations, particularly during summer months, which may adversely affect generalization and reduce predictive accuracy for those periods. The limited dataset size may also restrict the model’s ability to learn complex temporal dependencies that typically emerge across multiple years.

Conclusion

This research demonstrates that deep learning models provide an effective and adaptive approach for forecasting red wine sales. The proposed RNN-based forecasting model outperforms traditional statistical techniques by capturing nonlinear and temporal relationships within the data. These capabilities improve the reliability of sales predictions and support better inventory planning and market-oriented decision-making. Future research should incorporate external variables—such as promotional activities, weather conditions, and consumer sentiment—to further improve model

Cite this article as:
Nwali, M.E., Nwele, C., Ugah, J.O., & Onwudebelu, U. (2025) Predicting red wine sales using deep learning algorithms for market performance optimisation. *FNAS Journal of Mathematical Modeling and Numerical Simulation*, *2*(4), 98-113. https://doi.org/10.63561/jmns.v2i4.1127

accuracy and robustness. Additionally, integrating Explainable AI (XAI) techniques would enhance transparency and allow stakeholders to understand the underlying decision logic of the forecasting model. While this study used machine learning and deep learning methods, the broader literature shows that ensemble methods, particularly Gradient Boosted Trees (GBT) and Light Gradient Boosting Machine (LGBM) models, often outperform other regression approaches due to their computational efficiency, adaptability, and interpretability. LGBM, in particular, remains a strong candidate for future extensions of this work, especially in settings requiring rapid computation and scalability.

Recommendations

Based on the model evaluation, the following recommendations are proposed for organizations intending to deploy the forecasting model:

i. Ensure that the input data accurately reflects real-world sales patterns. Data inconsistencies, missing values, or external disruptions can significantly degrade model performance.
ii. This forecasting model is particularly suitable for retailers or brand owners with stable daily sales patterns and minimal competition. In these contexts, sales behavior is more autoregressive and easier to model.
iii. Because sales patterns evolve, organizations should periodically retrain and validate the model to maintain forecasting accuracy.
iv. Accurate forecasts can help optimize stock levels, reduce overstocking risks, and support well-informed pricing strategies.
v. Including multiple years of historical data will enhance the model's ability to detect seasonal trends and improve long-term predictive accuracy.

References

Agatz, N.A., Fleischmann, M., & Nunen, J. V. (2006). E-fulfillment and multi-channel distribution - A review. *Eur. J. Oper. Res.*, 187, 339-356. http://doi.org/10.1016/j.ejor.2007.04.024

Alihodžić, A., Zvorničanin, E., & Čunjalo, F. (2022). A Comparison of Machine Learning Methods for Forecasting Dow Jones Stock Index. In: Lirkov, I., Margenov, S. (eds) Large-Scale Scientific Computing, *Lecture Notes in Computer Science*, 13127. Springer, Cham. https://doi.org/10.1007/978-3-030-97549-4_24

Babić Rosario, A., Sotgiu, F.D., de Valck, K., & Bijmolt, T.H. (2016). The Effect of Electronic Word of Mouth on Sales: A Meta-Analytic Review of Platform, Product, and Metric Factors. *Journal of Marketing Research*, 53, 297 - 318. http://doi.org/10.1509/jmr.14.0380

Bergmeir, C., & Benítez, J. M. (2017). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191, 192–213. https://doi.org/10..1016/j.ins.2011.12.028

Bower, A.B., & Maxham, J.G. (2012). Return Shipping Policies of Online Retailers: Normative Assumptions and the Long-Term Consequences of Fee and Free Returns. *Journal of Marketing*, 76, 110 - 124. http://doi.org/10.1509/jm.10.0419

Cerqueira, V., Torgo, L., Smailovic, J., & Mozetic, I. (2017). A Comparative Study of Performance Estimation Methods for Time Series Forecasting. *In 2017 IEEE international conference on data science and advanced analytics (DSAA)*, 529–538. IEEE. https://doi.org/10.1109/DSAA.2017.7

Chen, L., Wu, W., Zhang, B., Wen, Y. & Yang, C. (2021). Learning from Multiple Time Series: A Deep Disentangled Approach to Diversified Time Series Forecasting. http://arxiv.org/abs/2111.04942

Dietterich, M., Jerath, K., & Srinivasan, K. (2017). Firm Strategies in the "Mid Tail" of Platform-Based Retailing. *Marketing Science*, 30 (5), 757–775. https://doi.org/10.1287/mksc.1110.0656

Duan, Q., Wang, Q., & Song, P. (2022). The Effects of Agency Selling on Reselling on Hybrid Retail Platforms. *International Journal of Electronic Commerce*, 23 (4), 524–556. https://doi.org/10.1080/10864415.2019.1655209

Elsayed, S., Thyssens, D., Rashed, A., Jomaa, H. S., & Schmidt-Thieme, L. (2021). Do We Really Need Deep Learning Models for Time Series Forecasting? Retrieved from http://arxiv.org/abs/2101.02118

Ferreira, K. J., Lee, B. H. A., & Simchi-Levi, D. (2016). Analytics for an Online Retailer: Demand Forecasting and Price Optimization. *Manufacturing & Service Operations Management,* 18 (1), 69–88. https://doi.org/10.1287/msom.2015.0561

Hastie, T., Tibshirani, R., & Friedman, J. (2017). Model Assessment and Selection. *In The elements of statistical learning*, 219–259. https://doi.org/10.1007/978-0-387-84858-7{_}7

Cite this article as:
Nwali, M.E., Nwele, C., Ugah, J.O., & Onwudebelu, U. (2025) Predicting red wine sales using deep learning algorithms for market performance optimisation. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(4), 98-113. https://doi.org/10.63561/jmns.v2i4.1127

Huang, S., & Lin, F. (2005). Designing intelligent sales-agent for online selling. *International Conference on Evolutionary Computation*, 279 – 286. https://doi.org/10.1145/1089551.1089605

Islam, S.F.N., Sholahuddin, A., & Abdullah, A.S. (2021). Extreme gradient boosting (XGBoost) method in making forecasting application and analysis of USD exchange rates against rupiah .*Journal of Physics: Conference Series*, 1722 (1), 012016. https://doi.org/10.1088/1742-6596/1722/1/012016

Ito, S., & Fujimaki, R. (2017). Optimization Beyond Prediction. *In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (Vol. Part F129685, pp. 1833–1841). New York, NY, USA: ACM. Retrieved from https://doi.org/10.1145/3097983.3098188

Januschowski, R., Chocarro, R., Cortiñas, M., Elorz, M., & Mugica, J. (2020). Channel Choice in the 21st Century: The Hidden Role of Distribution Services. *Journal of Interactive Marketing*, 33, 1–12. https://doi.org/10.1016/j.intmar.2020.09.002

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259 (2), 689–702. https://doi.org/10.1016/j.ejor.2016.10.031

Livieris, I.E., Pintelas, E.G., & Pintelas, P.E. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 32, 17351 – 17360. https://doi.org/10.1007/s00521-020-04867-x

Makridakis, D., Hosanagar, K., & Nair, H. S. (2022a). Advertising Content and Consumer Engagement on Social Media: *Evidence from Facebook. Management Science*, 64 (11), 5105–5131. http://pubsonline.informs.org/doi/10.1287/mnsc.2017.2902

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022b). The M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*.

Onu, F. U., Igwe J. S., Ikporo, S. C., Ariom, L. E., Nwamini B. T., & Onwudebelu, U. (2023), Security Vulnerabilities in Some Popular Object-Oriented Programming Languages – Forecast of the Danger Ahead, *Afr. J. Comp. & ICT*, 16, (1/ 2). https://afrjcict.net

Onwudebelu, U., Ugah, J.O., & Fasola, O.O. (2025). Harnessing Disruptive Technologies for Flood Prediction and Advisory Systems through Persuasive Modeling. *Digital Technologies Research and Applications.* 4(3), 113-135. https://doi.org/10.54963/dtra.v4i3.1648

Onwudebelu, U.; Ugah J. O., Uket N. E., E.B., Nnabu-Richard N. E., Oketa, K.C, Achi, I.I. & Afolabi I. Y. (2024). FPAS Model: A Case Study of Flood Prediction and Advisory System. *Sumerianz Journal of Scientific Research.* 2024, 7, 7–19. https://doi.org/10.47752/sjsr.72.7.19

Schwindt, C., Coletto, V., Díaz, M.M. & Ponzoni, I. (2023). Could QSOR Modelling and Machine Learning Techniques Be Useful to Predict Wine Aroma? *Food Bioprocess Technol* 16(1), 24–42. 2023. https://doi.org/10.1007/s11947-022-02836-x

Tiwari, P., Bhardwaj, P., Somin, S., Parr, W. V., Harrison, R. & Kulasiri, D. (2022). Understanding Quality of Pinot Noir Wine: Can Modelling and Machine Learning Pave the Way?, 19, *Foods* 11(19), 3072. https://doi.org/10.3390/foods11193072

Ugah, J.O. & Onwudebelu, U. FPAS System: The New Frontier in Flood Prediction and the Lifesaver for Flood-Prone Communities. *Afr. J. Manag. Inf. Syst. 2023*, 5, 15–35. https://afrjmis.net/wp-content/uploads/2024/08/vol5iss123pap2journalformatpn.pdf

Vijayalakshmi, M.N., murli, V., Menezes, D.B., Menon, R., Divecha, A., Ravindran, R., & Mehta, K. (2010). Intelligent sales forecasting engine using genetic algorithms. *Proceedings of the 19th ACM international conference on Information and knowledge management*.

Zhang, Q., Zeng, Z., Si, K., Luo, Q., Chen, D., & Chen, D. (2023a). Quality Prediction of Brewing Koji based on Multitask Learning, *In 2023 5th International Conference on Internet of Things, Automation and Artificial Intelligence* (IoTAAI 2023), 1-7. https://doi.org/10.1145/3653081.3653090

Zhang, Z., Li, Z., Shu, Z., Liang, J., Bai, W., Fei, Y., Zhao, W. & Zhao, S. (2023b). Effects of different koji making methods on volatile flavor components in the production of fermented soybean flavor Baijiu koji, *Journal of Food Safety and Quality Testing* 14(14), 105–114. http://doi.org/10.19812/j.cnki.jfsq11-5956/ts.2023.14.008