Groundnut Price Forecasting

Module 4

Agricultural Market Analytics

M.Sc. Agriculture Analytics

Submitted By:

Nidhi Gusai

ID: 202319015

Drashti Nayakpara

ID: 202319019

Divya Prajapati

ID: 202319024

Submitted To:

Dr. Prity Kumari

(Assistant Professor & Head)

Contents:

List of Tables:

List of Figures:

Abstract

This project aims to predict groundnut prices by analyzing and modeling time series data sourced from Agmark net for Gondal Agricultural Produce Market Committee (APMC) in Rajkot District, Gujarat. Various methods were employed, including traditional time series analysis models, advanced machine learning techniques, and statistical models. Specifically, models such as ARIMA, SARIMA, ARIMAX, SARIMAX, ARCH, GARCH, VAR, VARMAX, RANDOM FOREST REGRESSOR, GRU and LSTM were developed to forecast groundnut prices. The model selection was guided by four performance criteria: the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

The results demonstrate the effectiveness of historical price data in accurately predicting future price trends. This study provides valuable insights for stakeholders, offering a robust methodology for informed decision-making based on historical pricing patterns. By addressing risk management and strategic planning in the dynamic groundnut market, this study is a practical tool for farmers, traders, the oilseed processing industry, and policymakers.

Random Forest and GRU are highly stable models, demonstrating low RMSE percentages (6.76% and 10.09%, respectively) and strong performance in capturing data patterns effectively.

Keywords: Machine learning, Price Forecasting, Groundnut, VAR, LSTM, SARIMA, Statical model, Random Forest

Acknowledgements

On completion of the research work as part of the M.Sc. degree program offered as Joint Education Program by DAIICT, IIRS and AAU, I would like to remember with great gratitude all the fascinating people who made it possible with their relentless support and insightful guidance.

This wouldn’t have been possible without the solicitous guidance of my supervisors Dr. Prity Kumari who have been instrumental in shaping the research and guiding me towards the completion of research. They have been helpful guiding me throughout this journey whenever I needed their support.

I would like to thank my batchmates, DAIICT batchmates who have been the support system for me throughout this journey. They helped me to correct my mistakes and guided me through this journey. I am very happy to share a journey with them.

Last but not the least, the support from my family has been instrumental for me to complete the course. Their hard work and supporting words had inspired me to keep going.

1. Introduction

Groundnut is a significant oilseed and cash crop in India. It is an excellent source of protein, iron, calcium, and vitamins in addition to high-quality edible oil. It is also known by various name like peanut, wondernut, and poor man's cashew nut[1]. Peanuts are also known by many other local names such as earthnuts, ground nuts, goober peas, monkey nuts, pygmy nuts and pig nuts. Peanut or groundnut (*Arachis hypogaea*), is a species in the legume or "bean" family. The peanut was probably first domesticated and cultivated in the valleys of Paraguay. It is an annual herbaceous plant. Groundnut is the major oil seed crop in India and it plays a major role in providing raw materials to various industries and creates employment opportunities for millions of people[2].

Price forecasting plays a pivotal role in agricultural commodity trading and price analysis. Agricultural commodity prices, including those of groundnut, are inherently unpredictable due to their dependence on various factors such as weather conditions, pest infestations, and global market dynamics[3]. Natural calamities like droughts and floods further exacerbate the volatility of these prices, making accurate price modeling and forecasting a critical aspect of agricultural economics[4]. Historically, government-controlled food prices minimized the relevance of price forecasting. However, with the advent of liberalization and globalization, food prices are now largely determined by domestic and international market forces. This shift has amplified price variability, making reliable forecasting techniques essential for mitigating risk and enabling informed decision-making[5].

Groundnut is the third largest oilseed produced in the world and stands second largest oilseed crop of India. Asian and African countries have major shares in groundnut production. Asia accounts for about 50% of area and 60% of world groundnut production[6]. China ranks first in production of groundnut, with a share of about 42% to overall world production. India (16-18%) is the second largest producer of groundnut after China and then the United States of America (6-8%). Groundnut production, within the country, is mainly concentrated in five states including Gujarat, Andhra Pradesh, Tamil Nadu, Karnataka, Rajasthan and Maharashtra accounting for nearly 90% of the total production of groundnut in the country (Reddy and Reddy, 2011). The remaining groundnut cultivated area is scattered in the states of Madhya Pradesh, Uttar Pradesh, Punjab, and Odisha[7]. Gujarat is the single largest as well as the best quality producer of groundnuts accounting for over 40% of total groundnut produced in the country. Within Gujarat, the Gondal Agricultural Produce Market Committee (APMC) in Rajkot district serves as a vital hub for groundnut trade. The region’s favourable agro-climatic conditions and well-established market infrastructure make it a significant centre for groundnut cultivation and trading.

Groundnuts have diverse uses, including as a source of edible oil, fodder, and direct consumption as nuts. They also play a significant role in India’s export economy, with substantial demand in international markets[8]. However, price fluctuations, driven by unpredictable factors like rainfall variability and global market trends, pose challenges to all

stakeholders. Therefore, developing robust forecasting models is crucial for efficient market monitoring and planning.

This study leverages historical groundnut price data for Gondal APMC to analyze and model price trends. By employing advanced time series forecasting techniques, the study aims to provide accurate predictions that can guide stakeholders in making informed decisions. These forecasting models are expected to reduce uncertainty and enhance the resilience of groundnut stakeholders in navigating the complexities of agricultural markets.

1.1 Objective:

- Analyze time series price data for groundnut crop
- Price forecast using statistical models
- Price forecast using machine learning and deep learning models
- Compare model performance based on MSE and RMSE

2. Review of Literature

Forecasting of Arrival and Price of Groundnut in Rajasthan by ARIMA Model for Livelihood of Farmers

The study focused on forecasting groundnut arrival and prices in Rajasthan using ARIMA models. They used data from 2005 to 2021 and identified ARIMA (2,1,1) as the best model for price prediction, achieving an R-squared value of 0.852. The study highlighted seasonal trends, with significant increases in price predictions from 2022 to 2026. This model provided insights into market dynamics and aided stakeholders in planning based on expected supply and demand fluctuations[9].

Forecasting Oilseeds Prices in India: Case of Groundnut

They forecast groundnut prices in India using the ARIMA model, aiding farmers in planning acreage and marketing strategies during the kharif season. It demonstrates strong model performance ($R^2 = 0.95$ for India) and highlights the utility of time-series forecasting for reducing price volatility. However, the reliance on historical data and the exclusion of external factors limit its applicability in dynamic market condition[10].

Price Forecasting of groundnut: By an artificial neural network

A study on groundnut price forecasting used ANN models to predict prices in the Hiriyur market, selecting the best-performing network (NN2) based on Minimum Average Absolute Error and R^2 values. The dataset was divided into 155 training and 31 testing points, with forecasts generated using Forecaster XL in Excel. Ex-ante and ex-post results demonstrated the model's effectiveness for accurate market predictions[11].

Modelling and Forecasting Wholesale Groundnut Prices in Bikaner District of Rajasthan for Marketing Intelligence

The ARIMAX model has been effectively used for time series analysis of groundnut yield and production in the Surguja district of Chhattisgarh, utilizing weather variables as explanatory factors. Variables such as maximum and minimum temperatures, which exhibit a strong positive correlation with crop yield and production, were selected as inputs for the model. This approach highlights the importance of integrating climatic factors for accurate agricultural forecasting and decision-making[12].

Statistical Modeling on Groundnut Prices in Adoni Market using Hybrid Timeseries Models

In this study, an attempt was made to forecast the prices of Groundnut in Adoni market by different models namely ARIMA, GARCH and Hybrid (ARIMA-GARCH) model. Among all, the best fitted model was recognized as ARIMA (1,1,1) - GARCH (1,1), due to the better model selection criterion. By using this model Prices of January 2024, was forecasted as Rs. 6984.69 per Quintal[13].

Modelling and Forecasting Wholesale Groundnut Prices in Bikaner District of Rajasthan for Marketing Intelligence

Three forecasting models namely ARIMA, SARIMA and ECM models were used to forecast the wholesale prices of groundnut in Bikaner district of Rajasthan. The results show that ARIMA model is the best fit model for groundnut prices. Among the ARIMA modelling, ARIMA modelling with external predictors predict better than the one with no predictor variables. Multivariate ECM model showed higher forecasting error compared to SARIMA. The results show that, there is no one specific model that can be said to be best fit for forecasting of all crops[14].

3. Study Area

- Area: Gujarat has 101 lakh hectare of Net Sown Area and 128 lakh hectare of total cropped Area.
- Groundnut growing area: 19.09 lakh ha.

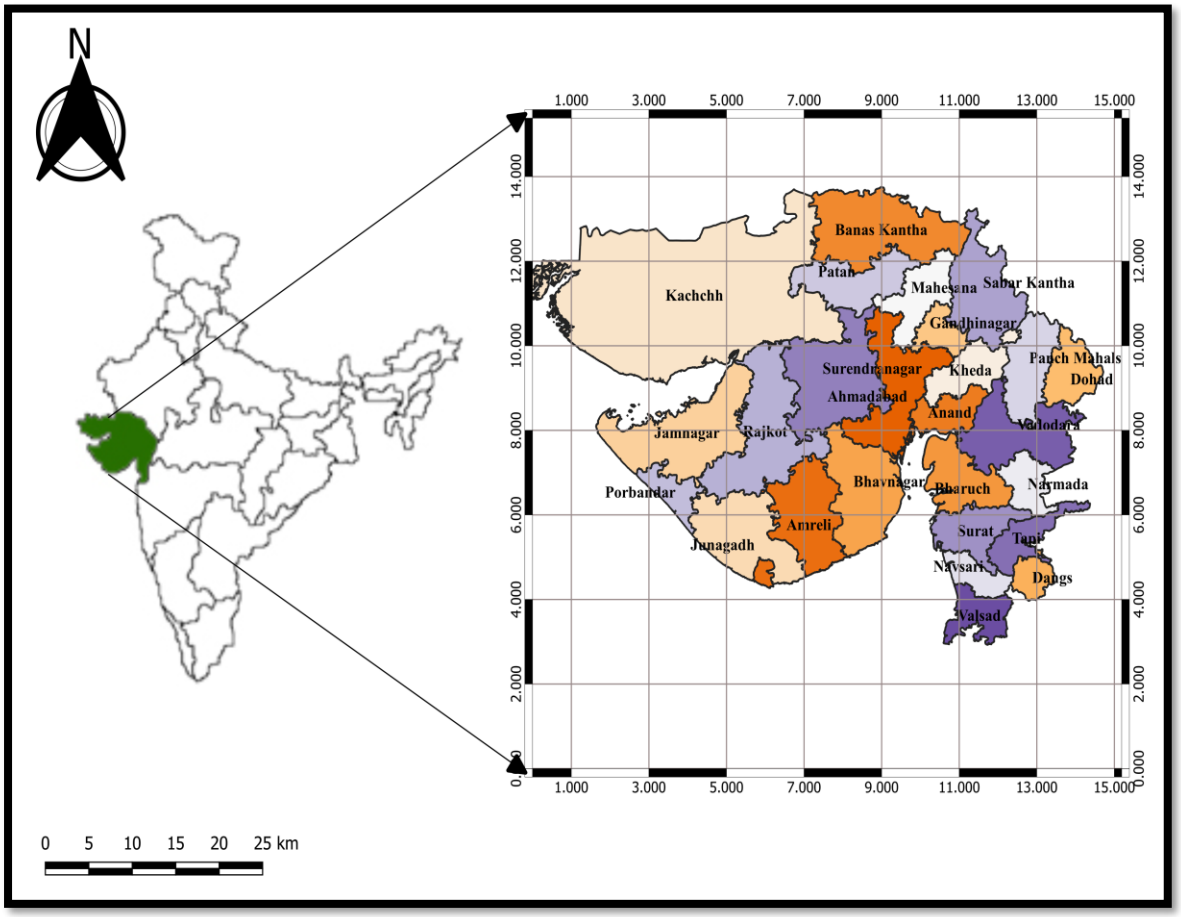

Figure 1: Study Area

4. Dataset

Groundnut production, within the country, is mainly concentrated in five states including Gujarat, Andhra Pradesh, Tamil Nadu, Karnataka, Rajasthan and Maharashtra.

In This study daily price data for groundnut has been collected from the AGMARKNET website for the Gondal market in Gujarat over 14 years, from 1st January 2010 to 20th November 2024, to predict prices.

Details	**Gondal APMC Price data**
Commodity	Groundnut
State	Gujarat
District	Rajkot
Market	Gondal
Time period	1 Jan 2010 to 20 Nov 2024
Price/ Arrival	Both
Data available(days)	5438
Minimum Price	2705
Maximum Price	11705

Table 1: Descriptive Statistic of Data

Statistical parameters	**Before Preprocessing Price**	**After Preprocessing Price**
count	7823	5438
mean	4552.529	4788.887
std	1278.452	1308.155
min	0	2055
25%	3580	3734.25
50%	4355	4680
75%	5305	5852
max	11705	11705

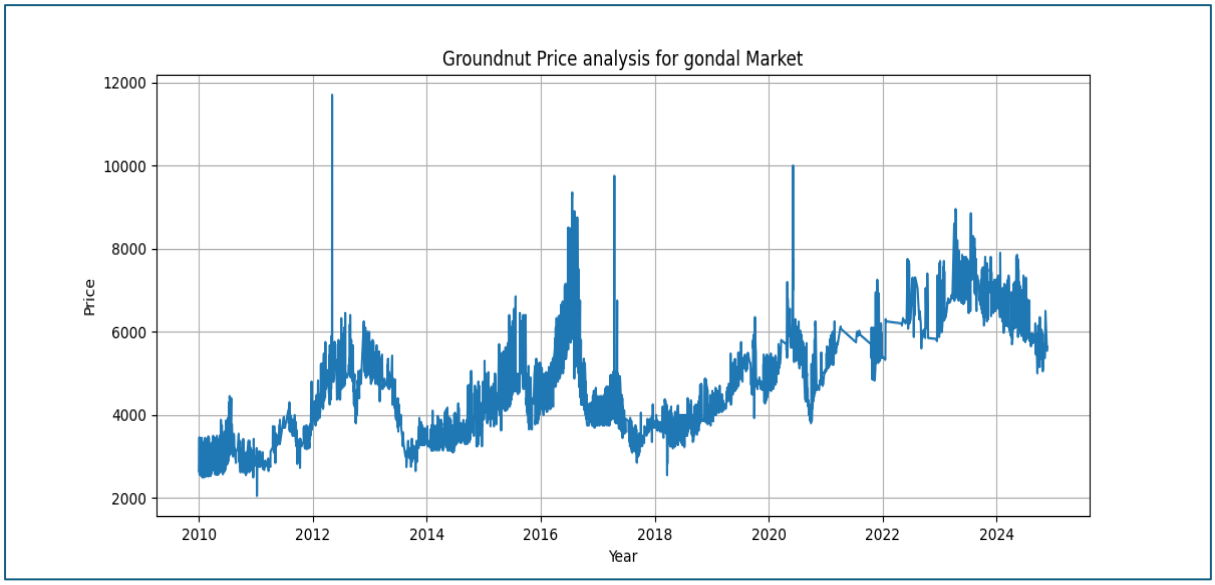

Figure 2: The Price Variation of Groundnut over the 14 years

5. Work Flow

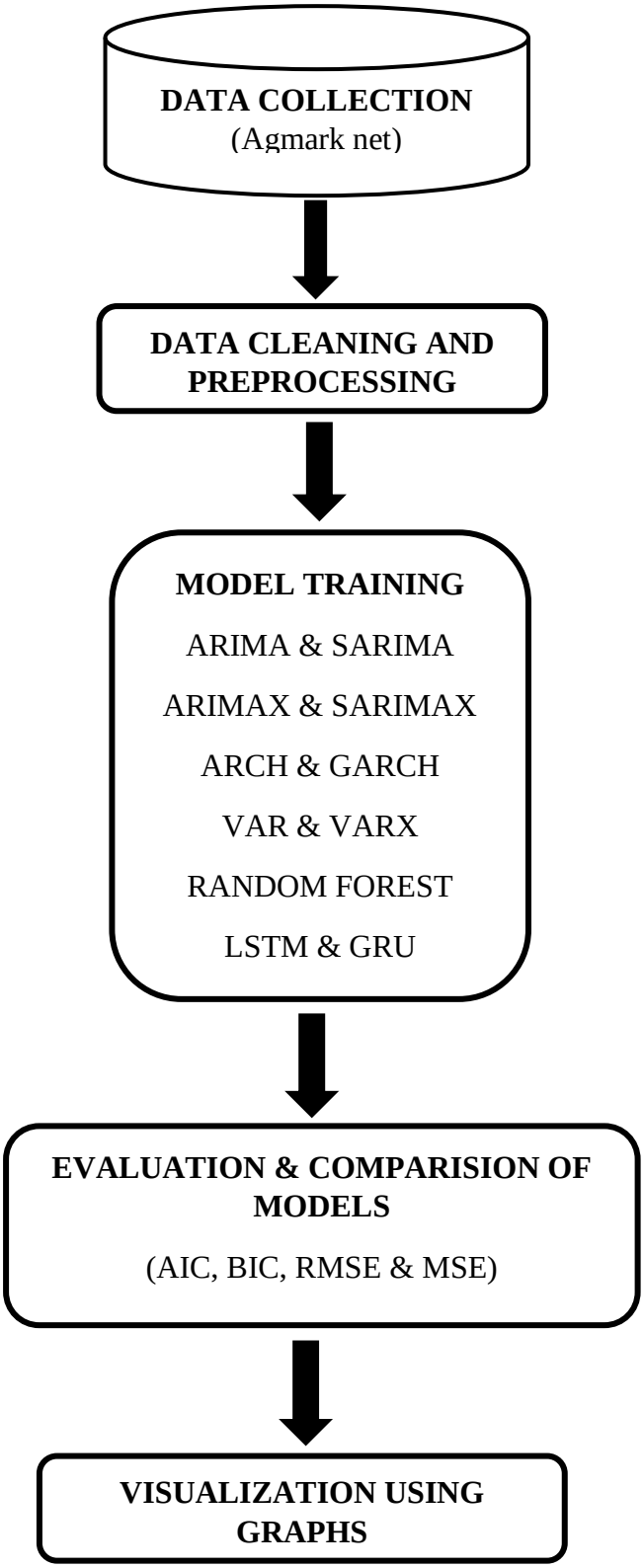

Figure 3: Work Flow

6. Methodology

Different statistical models, Machine learning models, and deep learning models are used in this study.

- **The statistical models are:** ARIMA, SARIMA, ARIMAX, SARIMAX, ARCH, GARCH, VAR and VARMAX
- **The machine learning models are:** Random Forest
- **The deep learning model is:** LSTM and GRU

6.1 Data cleaning:

The **groundnut price and arrival data** from Gondal APMC, spanning approximately 14 years (around 7,000 days), was obtained from Agmarknet. However, the raw dataset contained **duplicates and missing values**, which required thorough cleaning and preprocessing before analysis.

1. **Duplicate Removal**: All duplicate entries were identified and removed to ensure data integrity.
2. **Handling Missing Values**: Missing values in the dataset were filled using a **7-day rolling average** approach, which smooths fluctuations and preserves the overall trend.

6.2 Exploratory Data Analysis (EDA):

- Conducted EDA to uncover insights into the **long-term trends** in both **prices** and **arrivals** over the 14 years.
- Visualized patterns to understand seasonal variations, correlations, and anomalies, aiding in model selection and hypothesis formation.

6.3 Model Training and Evaluation:

- Time series forecasting models were applied to the training data, and their performance was evaluated on the testing set.

7. Models

7.1 Statistical Models:

7.1.1. ARIMA (Auto-Regressive Integrated Moving Average):

ARIMA is a widely used time series analysis and forecasting method effective for predicting future values when a discernible pattern or trend exists. It combines autoregression (AR), differencing (I), and moving averages (MA) to model different aspects of time series data. The components of ARIMA include:

- **AutoRegressive (AR):** Captures the relationship between the current value and its previous values, denoted by "p" in ARIMA (p, d, q).
- **Integrated (I):** Represents differencing to make the series stationary, denoted by "d" in ARIMA (p, d, q).
- **Moving Average (MA):** Models the error term as a combination of past error terms, denoted by "q" in ARIMA (p, d, q)[15].

Formula

$$y_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t$$

$\mathbf{y_t}$: The observed value of the time series at time t.

$\boldsymbol{\mu}$: The mean (constant) term. It represents the long-term mean level of the series

$\boldsymbol{\phi_i}$: The autoregressive coefficients

$\mathbf{y_{t-i}}$: The past p values of the time series

$\boldsymbol{\theta_j}$: The moving average coefficients

$\boldsymbol{\epsilon_{t-j}}$: Past q error terms (or residuals)

Methodology:

1. Conduct ADF test to check stationarity.
2. Apply differencing if necessary to remove trends or seasonality.
3. Use PACF to determine the order of "p" and ACF for "q."
4. Fit the ARIMA model using Maximum Likelihood Estimation (MLE).
5. Compare models using AIC and BIC values to identify the optimal parameters.
6. Validate the model through residual diagnostics and forecasting accuracy.

7.1.2. SARIMA (Seasonal ARIMA):

SARIMA extends ARIMA to include seasonal components, making it ideal for time series with regular seasonal patterns. It adds seasonal parameters (P, D, Q, m):

- **Seasonal AutoRegressive (P):** Captures the influence of past seasonal values.
- **Seasonal Differencing (D):** Removes seasonal trends.
- **Seasonal Moving Average (Q):** Models the relationship between seasonal errors.
- **m:** Specifies the length of the seasonal cycle[16].

Formula

$$\Phi_p(B^s) \cdot (1 - B)^d y_t = \mu + \Theta_q(B^s) \cdot \epsilon_t$$

s: The seasonal period

Φ_i: Coefficients for the seasonal autoregressive terms.

1−B: Represents the first difference operation

y_t: The dependent variable at time t

μ: mean level of the stationary series after differencing

Θ_j: Coefficients for the seasonal moving average terms

Methodology:

1. Identify seasonality in the data using seasonal plots or decomposition.
2. Perform differencing to remove seasonality.
3. Extend ARIMA by including seasonal terms (P, D, Q, m).
4. Fit the SARIMA model and tune parameters using AIC and BIC.
5. Validate results using residual analysis and forecast performance.

7.1.3. ARIMAX (ARIMA with Exogenous Variables):

ARIMAX incorporates exogenous variables (Arrival) that influence the dependent time series. These variables are included to improve model accuracy by accounting for their impact[17].

Formula

$$y_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \sum_{k=1}^{m} \beta_k X_{t-k} + \epsilon_t$$

yt: The observed value of the time series at time t

μ: The constant (intercept) term, representing the mean level of the series

p: The order of the AR part, indicating how many past values of yt are used

ϕ_i: Coefficients of the autoregressive terms

y_{t-i}: Past values of the time series

q: The order of the MA part, indicating how many past error terms are included

θ_j: Coefficients of the moving average terms

ϵ_{t-j}: Past error terms (residuals)

X_{t-k}: External variables (predictors) that are not part of the time series but influence it

β_k: Coefficients that measure the impact of the exogenous variables on yt

m: The number of lags for the exogenous variables

Methodology:

1. Identify and preprocess exogenous variables.
2. Include these variables as predictors in the ARIMA model.
3. Fit the ARIMAX model and evaluate its performance using AIC, BIC, and residual diagnostics.[18]

7.1.4. SARIMAX (Seasonal ARIMAX):

SARIMAX combines SARIMA with exogenous variables (Arrival) to capture both seasonal patterns and external influences on the time series[19].

Formula

$$\Phi_p(B^s) \cdot (1-B)^d y_t = \mu + \Theta_q(B^s) \cdot \epsilon_t + \sum_{k=1}^{m} \beta_k X_{t-k}$$

B: Backshift operator

s: Seasonal period

Φ_i: Coefficients of the seasonal autoregressive terms

d: Order of differencing to remove non-seasonal trends and make the series stationary

Θ_j: Coefficients of the seasonal moving average terms

X_{t-k}: External predictors (variables) that influence the dependent variable yt

β_k: Coefficients representing the impact of the exogenous variables

m: Number of lags for the exogenous variables

Methodology:

1. Identify seasonality and external variables.
2. Combine ARIMA components with seasonal and exogenous terms.
3. Fit SARIMAX using appropriate (p, d, q) and (P, D, Q, m) values.
4. Validate the model with residual diagnostics and performance metrics.

7.1.5. ARCH (Autoregressive Conditional Heteroskedasticity):

Autoregressive Conditional Heteroskedasticity (ARCH) is a statistical model used to analyze and model a time series's volatility or variance, particularly in financial econometrics. These models are designed to capture the changing or conditional variance in a time series, recognizing that volatility can vary over time[20].

Concepts of ARCH Model:

1. **Heteroskedasticity:** ARCH models address the issue of heteroskedasticity, which refers to the phenomenon where the variance of the errors in a time series is not constant over time. In financial markets, for example, volatility tends to exhibit clustering, with periods of high volatility followed by periods of low volatility.
2. **Conditional Variance:** ARCH models assume that the variance of the error term at each time point is a function of past observations. The conditional variance is modeled as an autoregressive process, where past squared residuals contribute to the current conditional variance.

Formula

$$y_t = \mu + \epsilon_t$$

$$\epsilon_t = \sigma_t \cdot z_t$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2$$

$\mathbf{y_t}$: The observed time series value at time t

$\boldsymbol{\mu}$: The constant (mean) term, representing the average value of the time series

$\boldsymbol{\epsilon_t}$: The error term or innovation at time ttt, representing deviations from the mean

$\boldsymbol{\sigma_t}$: The **conditional standard deviation** of ϵt

$\mathbf{z_t}$: A sequence of i.i.d. standard normal random variables (zt~N(0,1))

$\boldsymbol{\sigma_t^2}$: The conditional variance of the error term at time t

$\boldsymbol{\alpha_0}$: A constant term, representing the base level of variance

$\boldsymbol{\alpha_i}$: Coefficients for the lagged squared error terms

q: The order of the ARCH model, indicating how many past squared residuals are included in the model

Methodology:

1. Test for heteroskedasticity in residuals from ARIMA and SARIMA.
2. Fit the ARCH model to capture variance dynamics.
3. Validate the model with diagnostic tests and analyze volatility predictions.

7.1.6. GARCH (Generalized ARCH):

GARCH extends ARCH by modeling both autoregressive and moving average components for variance, making it suitable for capturing persistent volatility[21].

Formula

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

$\boldsymbol{\sigma_t^2}$: Conditional variance at time t

ω: Constant term, providing a base level of variance

$\boldsymbol{\alpha_i}$: Coefficients for the **lagged squared residuals**

$\boldsymbol{\beta_j}$: Coefficients for the **lagged conditional variances**

q: Order of the ARCH part, representing how many past squared errors are included

p: Order of the GARCH part, representing how many past conditional variances are included

Methodology:

1. Fit GARCH to residuals from ARIMA/SARIMA models.
2. Optimize parameters for both ARCH and GARCH terms.
3. Validate the model using diagnostic tests and volatility forecasts.

7.1.7. VAR (Vector Autoregression):

VAR is a multivariate time series model that captures the linear interdependencies between multiple variables. Each variable in the system is modeled as a linear function of its own past values and the past values of all other variables in the system[22].

- **V (Vector):** Refers to the multivariate aspect of the model, where multiple time series variables are analyzed together.
- **AR (Autoregression):** Indicates that each variable is regressed on its own lagged values and those of the other variables.

Hyperparameter

Order: 20

Formula

$$Y_t = \mu + \sum_{i=1}^{p} A_i Y_{t-i} + \epsilon_t$$

μ: A vector of constant terms

$\sum_{i=1}^{p} \mathbf{A_i Y_{t-i}}$**:** Lagged values of Yt

$\boldsymbol{\epsilon_t}$**:** A vector of error terms

Methodology:

1. Identify the variables to include in the model and ensure that all are stationary using tests like ADF.
2. Determine the optimal lag order (p) using information criteria such as AIC or BIC.
3. Fit the VAR model with the selected lag order.
4. Perform Granger causality tests to analyze the causal relationships among variables.
5. Validate the model by examining residuals and forecasting accuracy.

7.1.8. VARMAX (Vector Autoregression with Exogenous Variables):

VARMAX extends VAR by incorporating exogenous variables that are not influenced by the endogenous variables within the system. These external factors help improve the forecasting accuracy of the model[23].

- **V (Vector):** Refers to multiple dependent variables analyzed simultaneously.
- **AR (Autoregression):** Indicates regression on past values of both endogenous and exogenous variables.
- **X (Exogenous):** Represents external predictors that influence the system but are not influenced by it.

Hyperparameter

Order: 20

Formula

$$Y_t = \mu + \sum_{i=1}^{p} A_i Y_{t-i} + \sum_{j=1}^{q} B_j \epsilon_{t-j} + \sum_{k=1}^{m} C_k X_{t-k} + \epsilon_t$$

σ_t^2: Conditional variance at time t

ω: Constant term

α_i: Coefficients for the lagged squared residuals

β_j: Coefficients for the lagged conditional variances

q: Order of the ARCH (Autoregressive Conditional Heteroskedasticity) component

p: Order of the GARCH (Generalized ARCH) component

Methodology:

1. Preprocess the data and ensure all variables (endogenous and exogenous) are stationary.
2. Determine the optimal lag order for endogenous variables and the inclusion of exogenous variables using AIC or BIC.
3. Fit the VARMAX model by including exogenous variables as additional regressors.
4. Evaluate the model using residual diagnostics and forecasting performance.
5. Use the model to make multivariate forecasts and assess how exogenous variables influence endogenous variables.

7.2. Machine Learning Models:

7.2.1. Random Forest:

Training of Random Forest: Bootstrap Sampling:

Generate multiple subsets of the training dataset by sampling with replacement. Each subset may have duplicate records.

The size of each subset is the same as the original training dataset.

Decision Tree Building:

For each bootstrap sample:

Select a random subset of features at each split (feature subsampling).

Grow a decision tree using this subset without pruning, ensuring the tree goes to its maximum depth.

Split Criterion:

Use splitting criteria to decide the best feature and threshold for each split:

Gini Impurity (for classification).

Entropy (for classification).

Mean Squared Error (for regression)

Random Forest (RF) is a supervised learning algorithm used for classification, regression, and even time series forecasting tasks. It builds an ensemble of decision trees, combining their predictions to improve accuracy and reduce overfitting. For time series forecasting, RF can handle lagged features, trends, and seasonality by transforming the temporal data into supervised learning problems. Splitting criteria like Information Gain (IG) and Gini Index are used to identify the best splits, ensuring the model captures meaningful patterns in both tabular and sequential data. RF is particularly effective in handling high-dimensional datasets and ranking feature importance, making it versatile for diverse applications.

Hyperparameters

n_estimators: 100

random_state: 42

Formula

$$\boldsymbol{Gini = 1 - \sum_{i=1}^{C} p_i^2}$$

$$\hat{y}_t = \frac{1}{T}\sum_{i=1}^{T} f_i(x_t)$$

C: number of classes

p_i: probability (or proportion) of class iii in the dataset

y_t: The predicted or estimated value of yt at time t

T: The total number of predictions or models being aggregated

$f_i(x_t)$: The prediction made by the i-th model (or function) at time t using the input xt

Methodology:

1. Preprocess data by handling missing values and scaling.
2. Split data into training and testing sets.
3. Train the Random Forest regression model.
4. Evaluate model performance using R^2, MSE, and MAE metrics.

7.3. Deep Learning Model:

7.3.1. LSTM (Long Short-Term Memory):

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to overcome the limitations of traditional RNNs in capturing and learning from long-term dependencies in sequential data. LSTMs are widely used in various applications, including natural language processing, time series analysis, and financial market predictions.

Components of LSTM:

1. **Cell State:** LSTMs maintain a cell state, which serves as a memory unit that can capture information over long sequences. This enables LSTMs to remember important information and selectively update or forget it as needed.

2. **Three Gates:** LSTMs have three gates—input gate, forget gate, and output gate—that control the flow of information through the cell state.
   I. 1.Input Gate: Regulates the flow of new information into the cell state.
   II. 2.Forget Gate: Manages the removal or "forgetting" of information from the cell state.
   III. 3.Output Gate: Determines the information to be output based on the cell state.

3. **Hidden State:** The hidden state in LSTMs is responsible for carrying information throughout the sequence. It acts as a filtered version of the cell state and is used for making predictions.

4. **Training and Backpropagation Through Time (BPTT):** LSTMs are trained using backpropagation through time, similar to traditional RNNs. However, LSTMs mitigate the vanishing gradient problem associated with RNNs, allowing them to learn long-range dependencies more effectively.

Hyperparameters

Layer: 1
neurons: 50
Optimizer: Adam
Learning rate: 0.001
Batch size: 64
Split: 80:20

Formula

Forget Gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Input Gate

$$it = \sigma W i \cdot \mathrm{h}t-1, xt + bi$$

Cell Candidate

$$\mathrm{C}t = \tanh \mathrm{W}C \cdot \mathrm{h}t-1, xt + bC$$

Cell State Update

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t}$$

Output Gate

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Hidden State Update

$$h_t = o_t \cdot \tan \mathrm{h}(C_t)$$

σ: sigmoid activation function

W_f: weight matrix associated with the forget gate

h_{t-1}: previous hidden state (output from the previous time step)

x_t: input at time t

b_f: bias term for the forget gate

W_i: weight matrix associated with the input gate

b_i: bias term for the input gate

i_t: output of the input gate

tanh: hyperbolic tangent activation function

WC: weight matrix for the cell candidate

bC: bias term for the cell candidate

C_{t-1}: previous cell state

W_o: weight matrix for the output gate

b_o: bias term for the output gate

Methodology:

1. Normalize data to accelerate model convergence.
2. Prepare data into sequences (X, y) for supervised learning.
3. Build the LSTM model with input, hidden, and output layers.
4. Compile the model with appropriate loss function and optimizer.
5. Train the model on the training dataset and validate it using test data.
6. Evaluate performance using metrics like MSE and MAE and visualize predictions.

7.3.2. GRU (Gated Recurrent Unit):

Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture designed to efficiently capture dependencies in sequential data, similar to LSTMs, but with a simpler structure. GRUs address the vanishing gradient problem, enabling effective learning of long-term dependencies. They are widely used in natural language processing, time series forecasting, and speech recognition.

Components of GRU:

1. **Update Gate**:
   The update gate determines how much of the previous memory should be retained and how much of the new information should be added.
2. **Reset Gate**:
   The reset gate controls how much of the past information to forget or ignore. It helps the GRU focus on the relevant portions of the past data.
3. **Candidate Hidden State**:
   A temporary state calculated based on the reset gate's influence, which is used to generate the final hidden state.
4. **Hidden State**:
   The GRU's output for the current time step, calculated as a combination of the candidate hidden state and the previous hidden state, controlled by the update gate.

Hyperparameters

Layer: 1

neurons: 50

Optimizer: Adam

Learning rate: 0.001

Batch size: 64

Split: 80:20

Formula

Update Gate

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

Reset Gate

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

Candidate Hidden State

$$\widetilde{h_t} = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h)$$

Final Hidden State

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t}$$

σ: sigmoid activation function, ensuring that zt is between 0 and 1.

W_z: weight matrix for the update gate.

$[h_{t-1}, x_t]$: concatenation of the previous hidden state and the current input.

b_z: bias term for the update gate.

Z_t: controls the amount of previous hidden state to keep and the amount of candidate hidden state to incorporate

W_r: weight matrix for the reset gate

b_r: bias term for the reset gate

W_h: weight matrix for the candidate hidden state

b_h: bias term for the candidate hidden state

$r_t \odot h_{t-1}$: element-wise multiplication

tanh: hyperbolic tangent activation function, which outputs values between -1 and 1

h_t: potential new hidden state based on the current input and the previous hidden state, modulated by the reset gate

Methodology:

1. Scale the data to speed up training.
2. Format data into input-output pairs for supervised learning.
3. Construct a GRU model with input, hidden, and output layers.
4. Choose an appropriate loss function and optimizer (e.g., Adam).
5. Train the model on training data and validate on test data.
6. Use performance metrics (MSE, MAE) and visualize predictions

8. Results

8.1. ARIMA:

Best model: ARIMA(5,1,3)(0,0,0)[0]

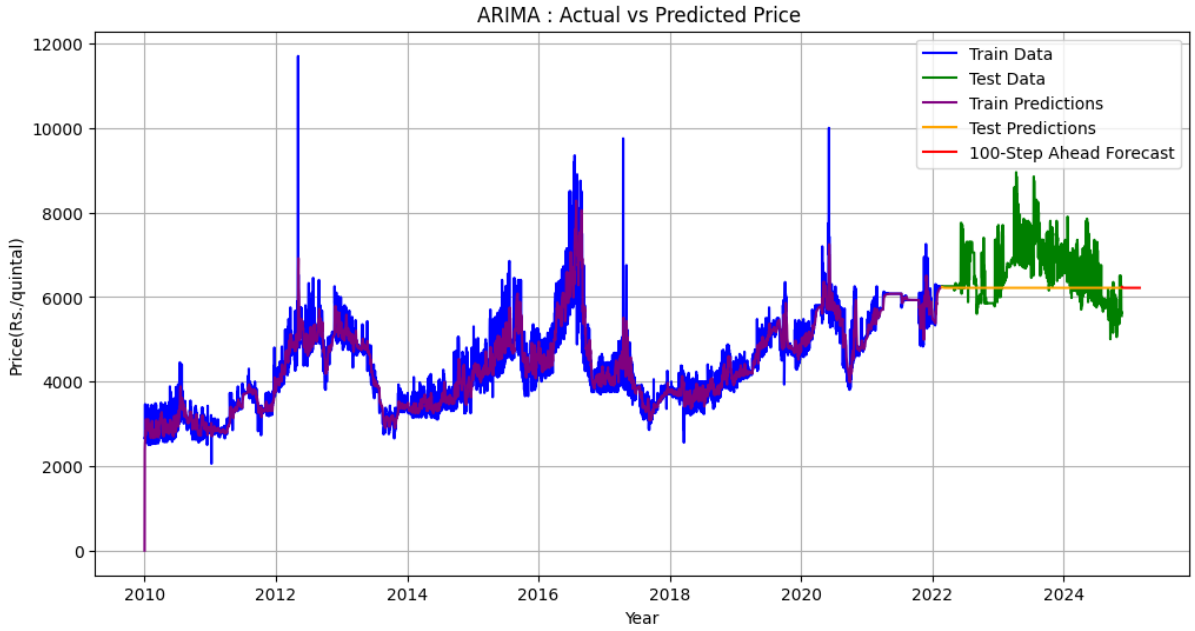

Figure 4: ARIMA Time Series Prediction

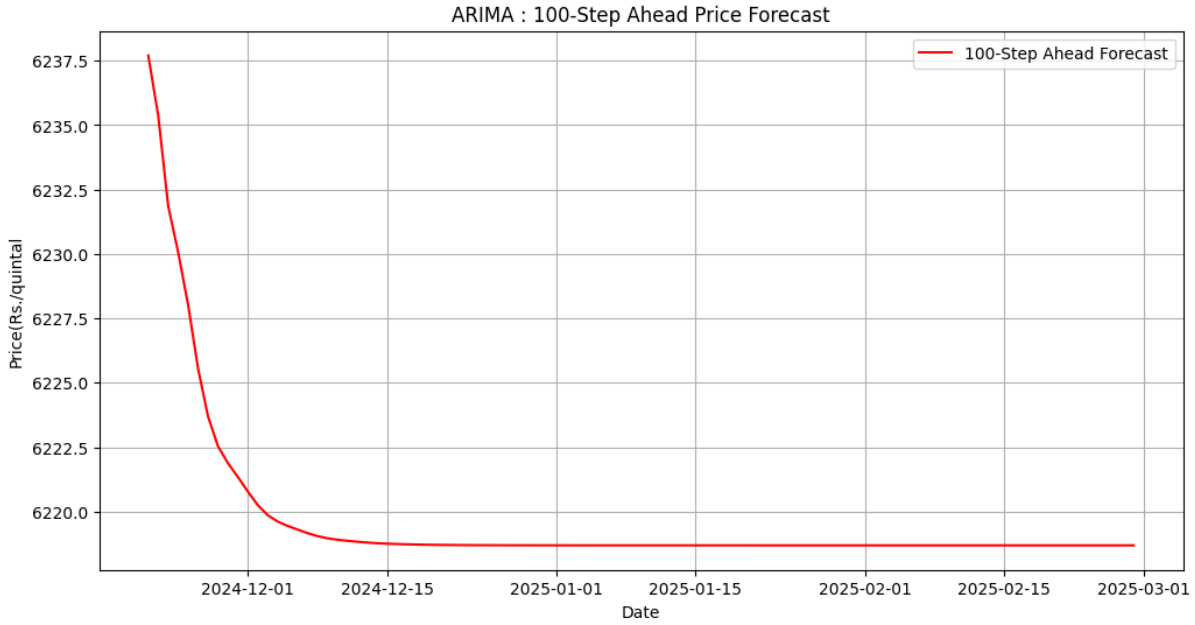

Figure 5: ARIMA Time Series Forecasting 100 Step Ahead

8.2. SARIMA:

Best model: ARIMA(2,0,1)(0,0,1)[7]

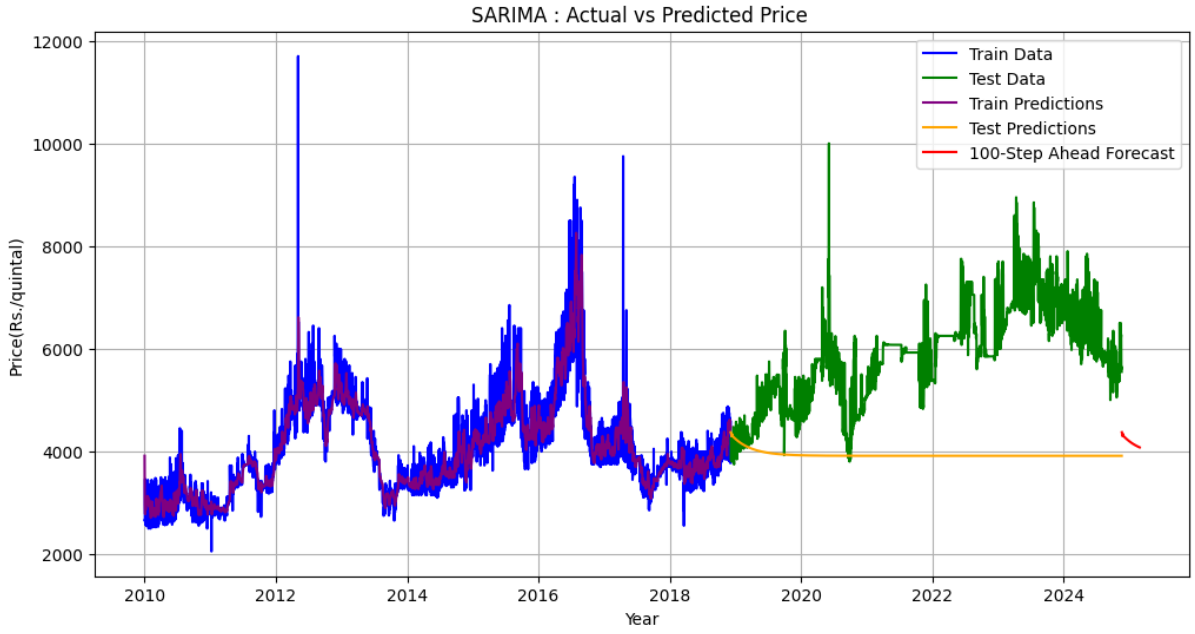

Figure 6: SARIMA Time Series prediction

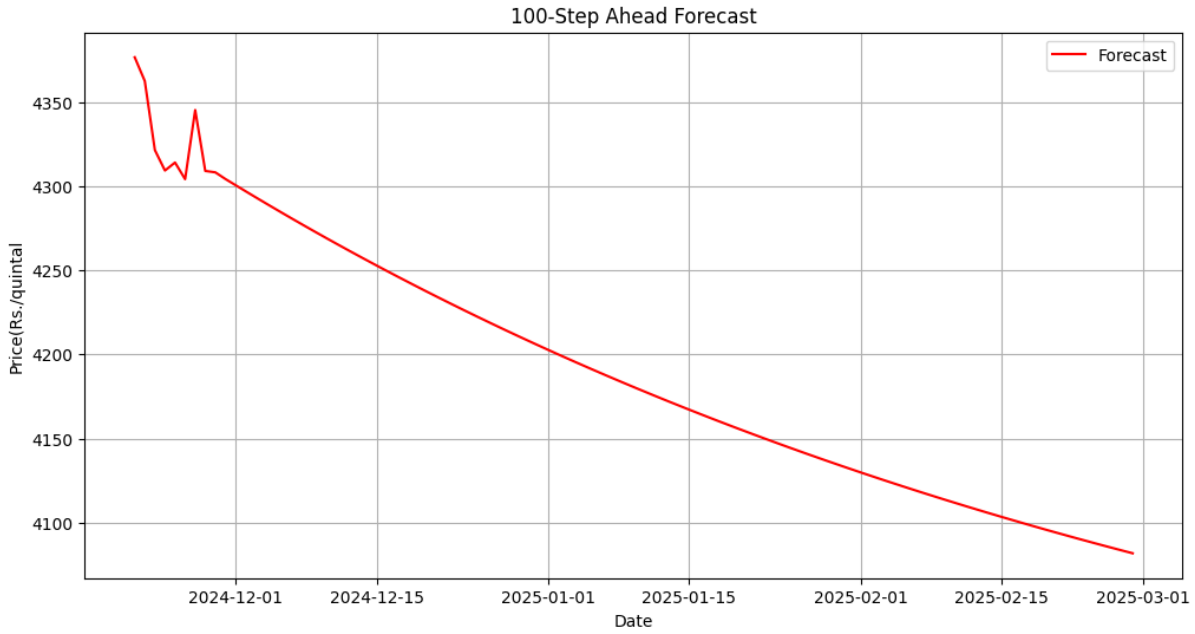

Figure 7: SARIMA Time Series Forecasting 100 Step Ahead

8.3 ARIMAX:

Best model: ARIMA(2,1,5)(0,0,0)[0]

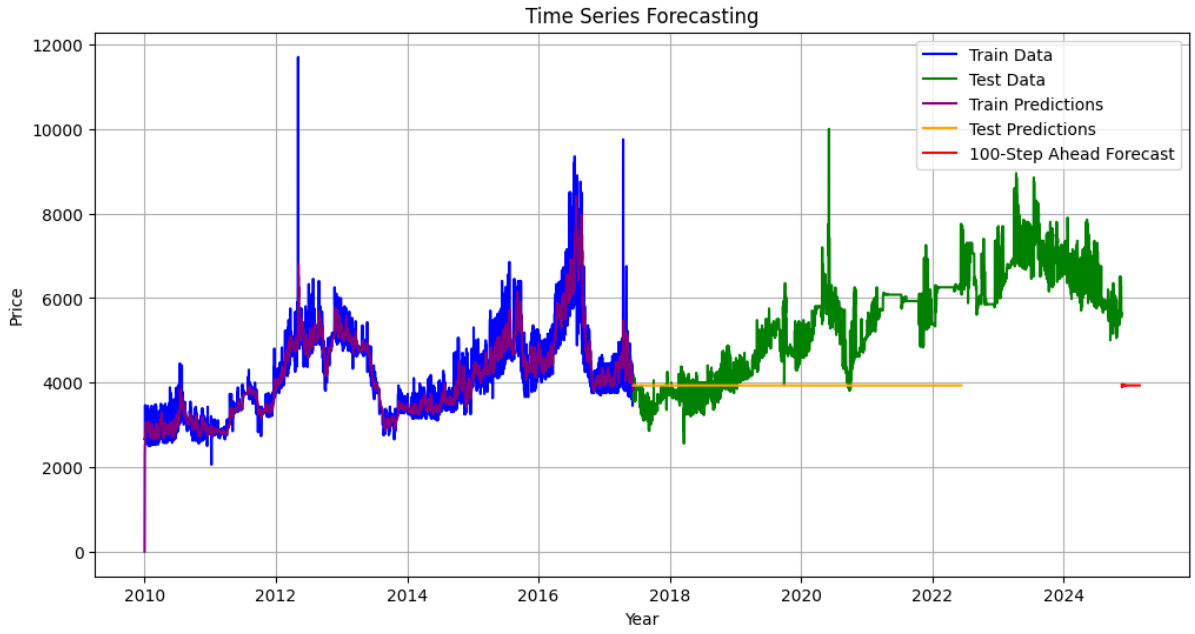

Figure 8: ARIMAX Time Series Prediction

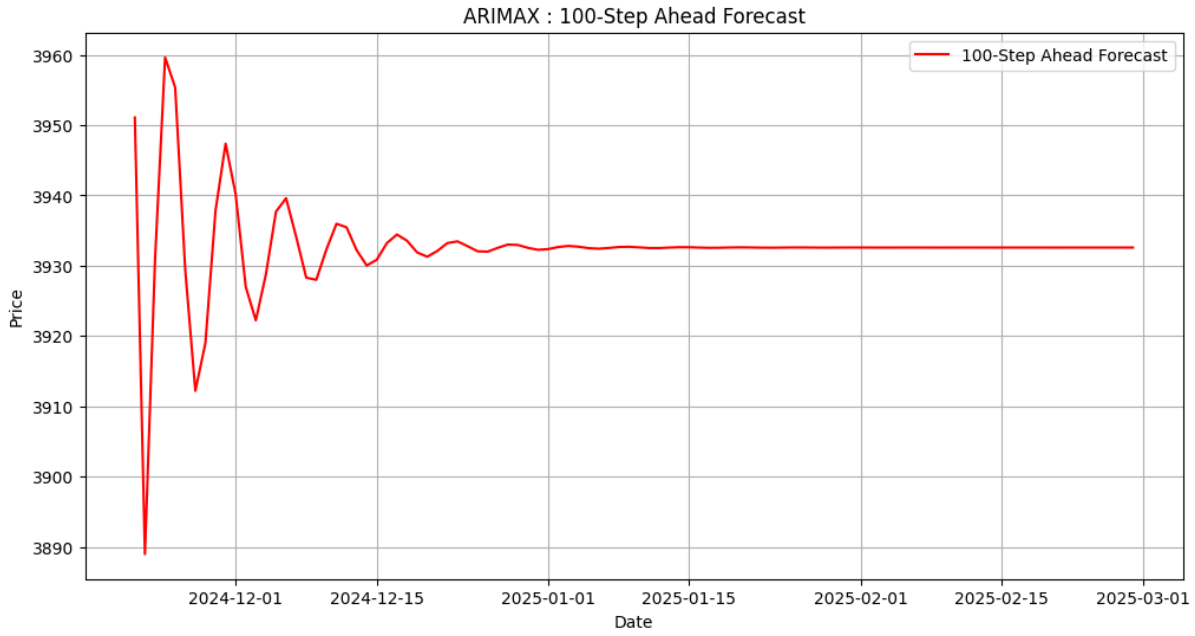

Figure 9: ARIMAX Time Series Forecasting 100 Step Ahead

8.4. SARIMAX:

Best model: ARIMA(5,1,0)(5,1,0)[7]

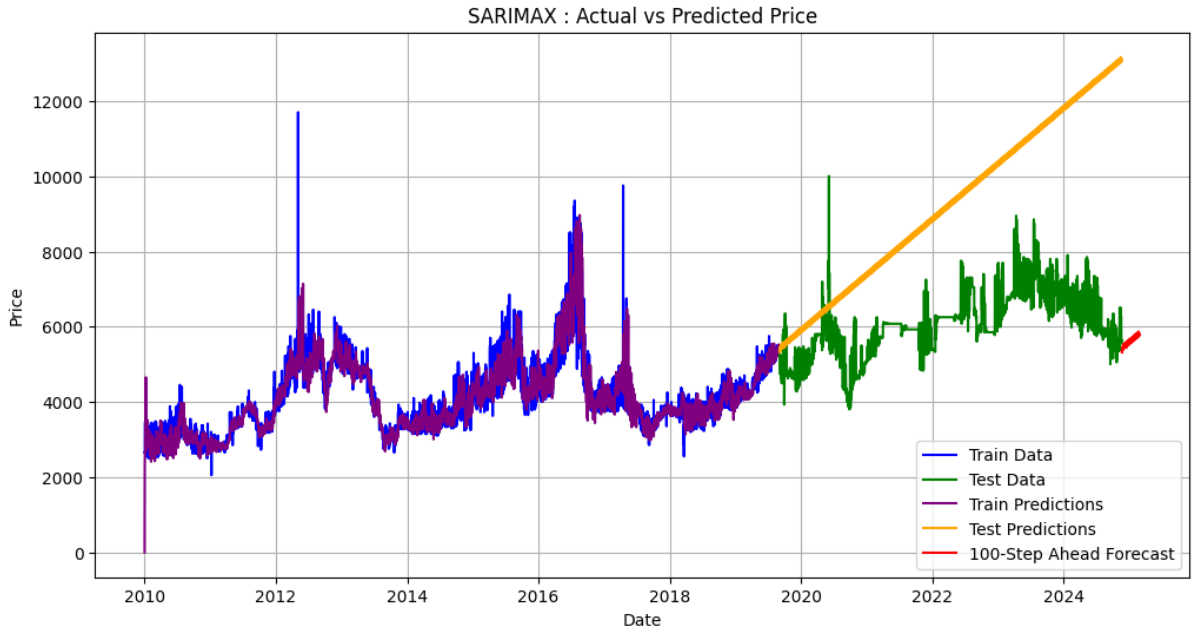

Figure 10: SARIMAX Time Series Prediction

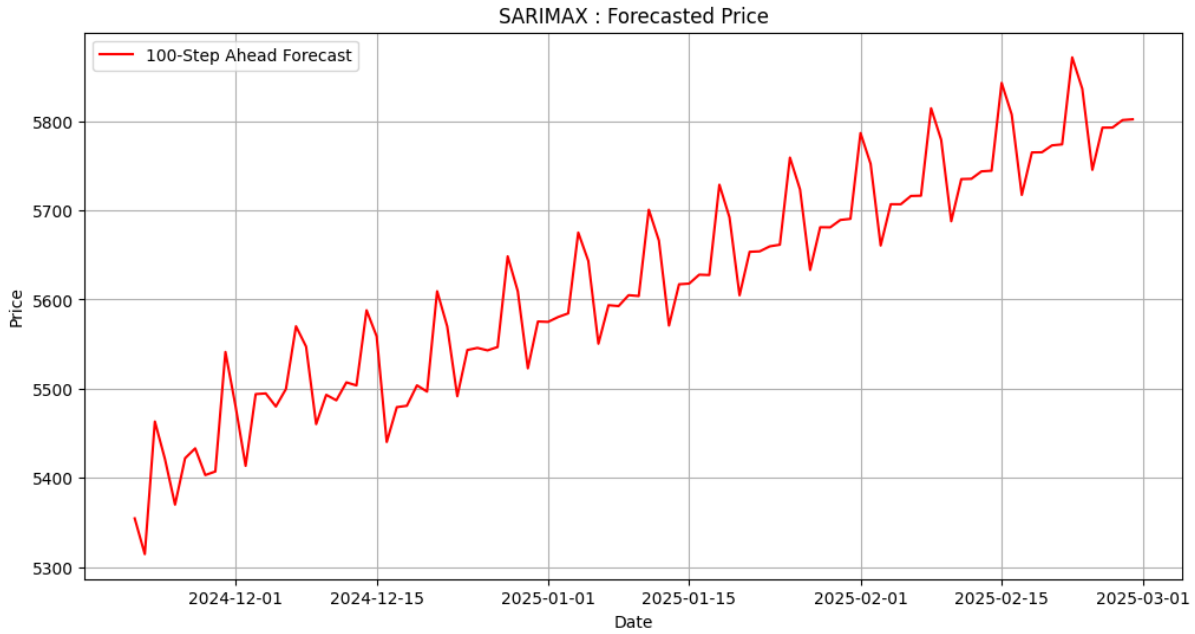

Figure 11: SARIMAX Time Series Forecasting

8.5. ARCH:

Best model: ARIMA(5,1,0)(5,1,0)[7]

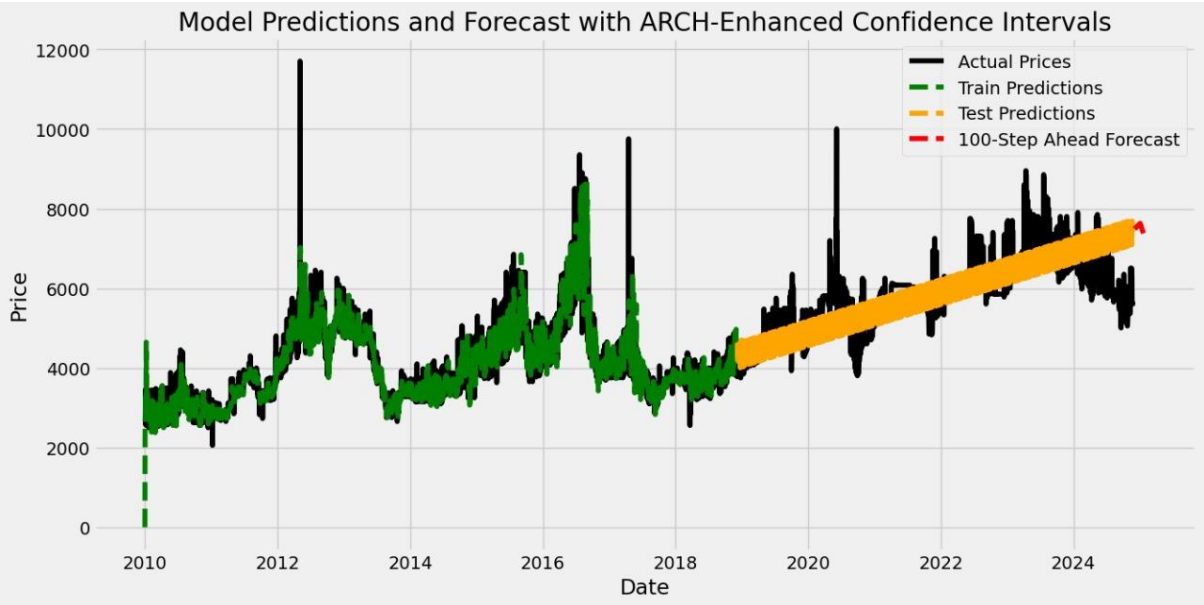

Figure 12: ARCH Time Series Prediction

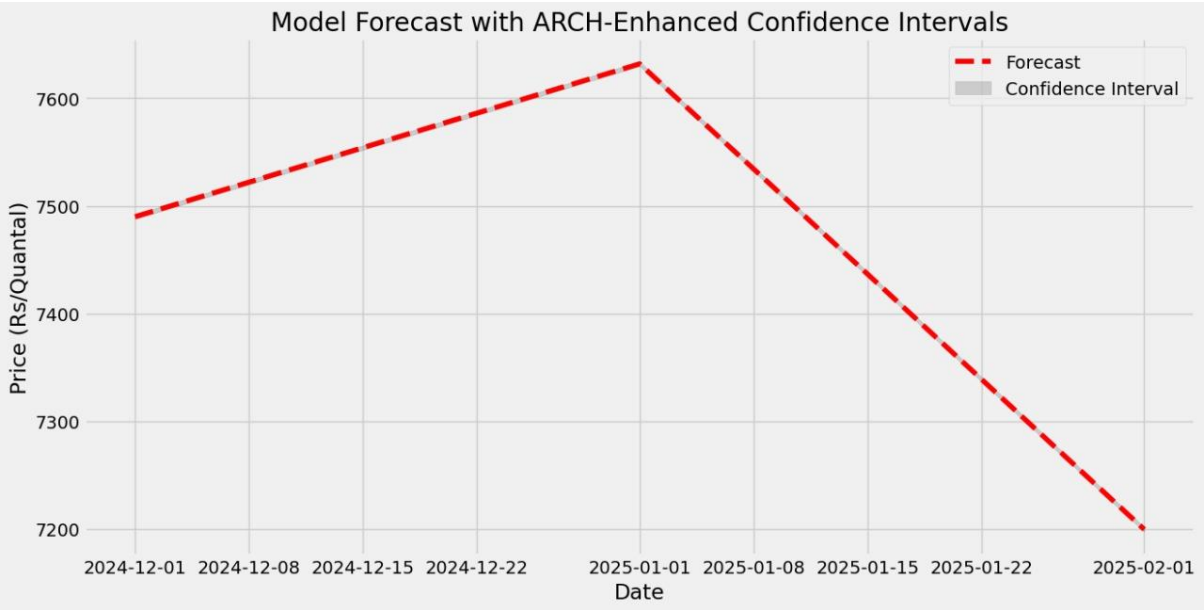

Figure 13: ARCH Time Series Forecasting

8.6. GARCH:

Best model: ARIMA(5,1,0)(5,1,0)[7]

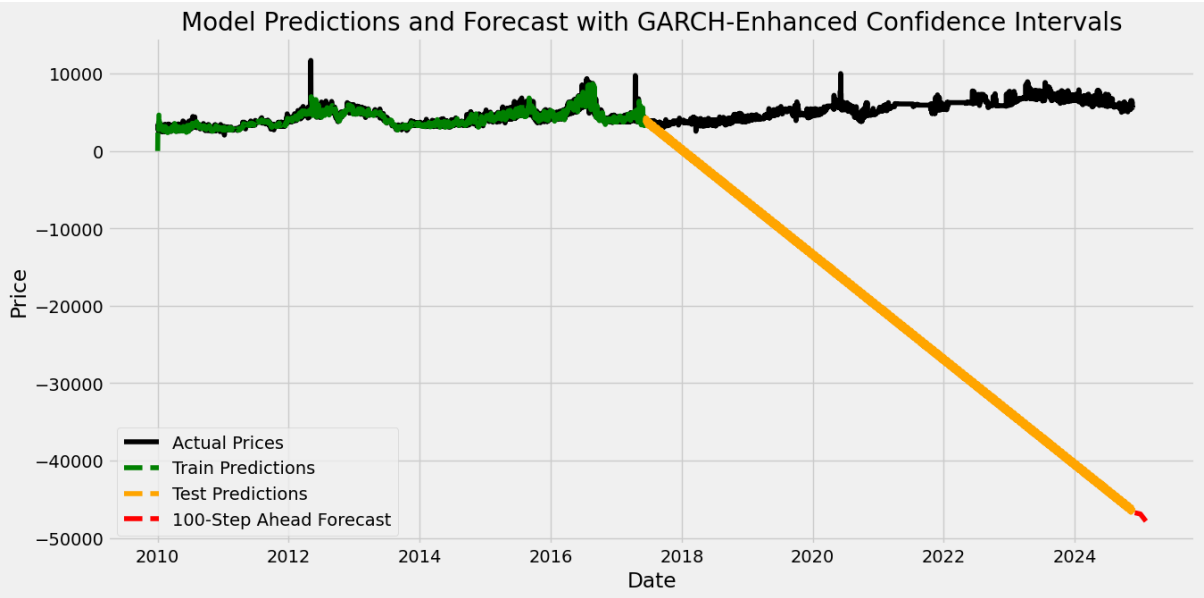

Figure 14: GARCH Time Series Prediction

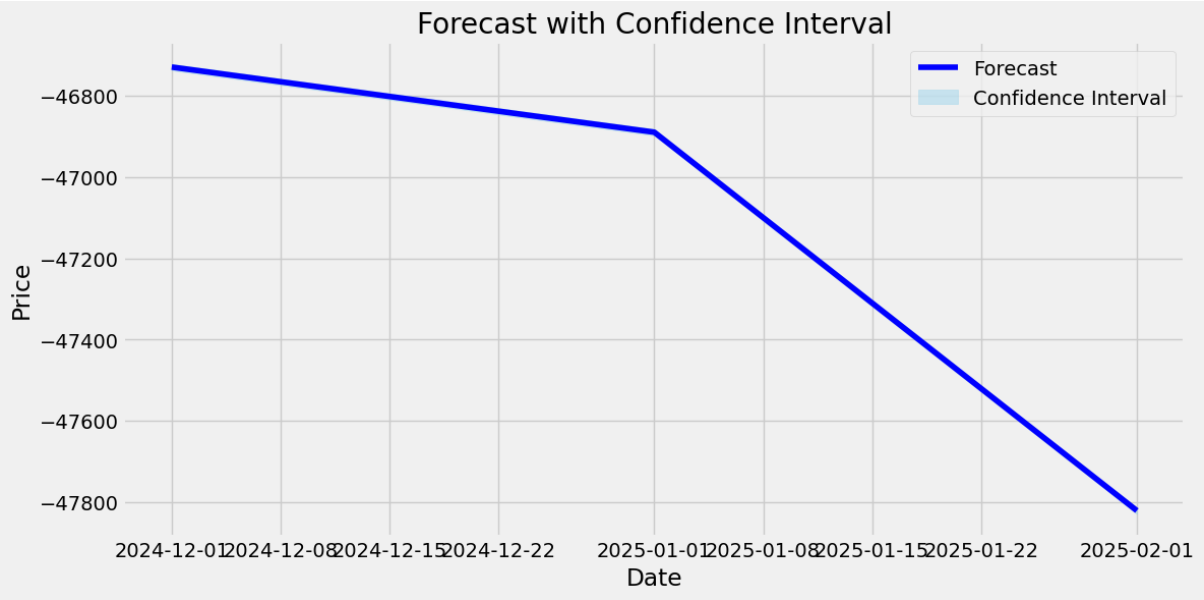

Figure 15: GARCH Time Series Forecasting

8.7. VAR:

The model best fitted at Order (20)

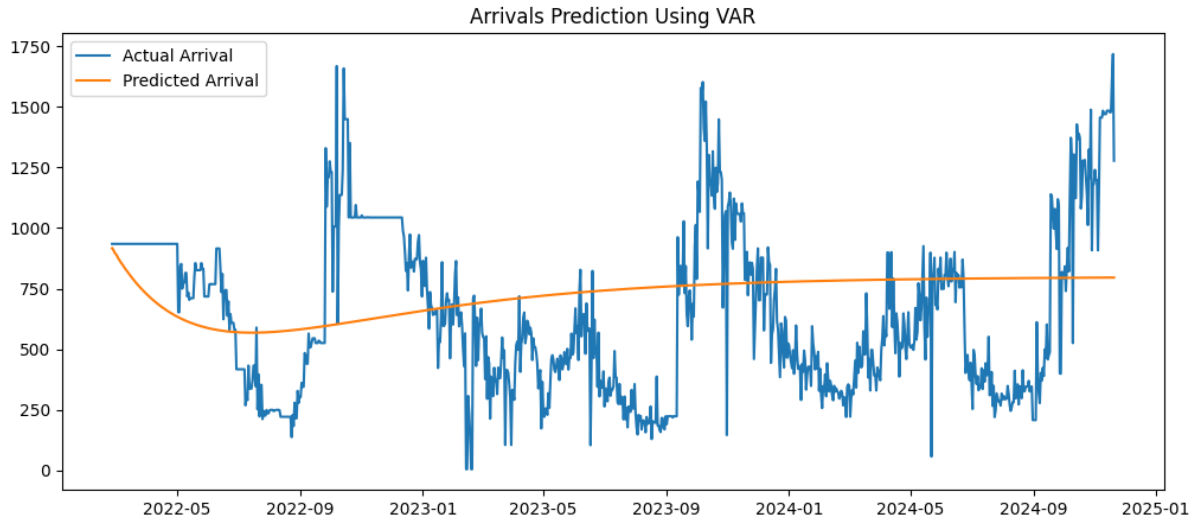

Figure 16: Arrival prediction using VAR

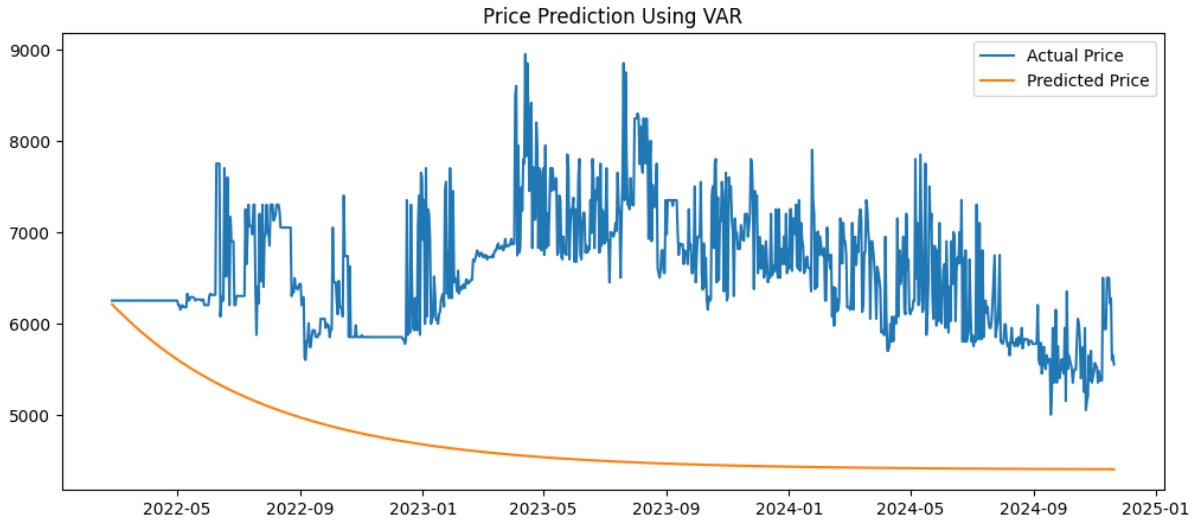

Figure 17: Price prediction using VAR

8.8. VARMAX:

The model best fitted at Order (20)

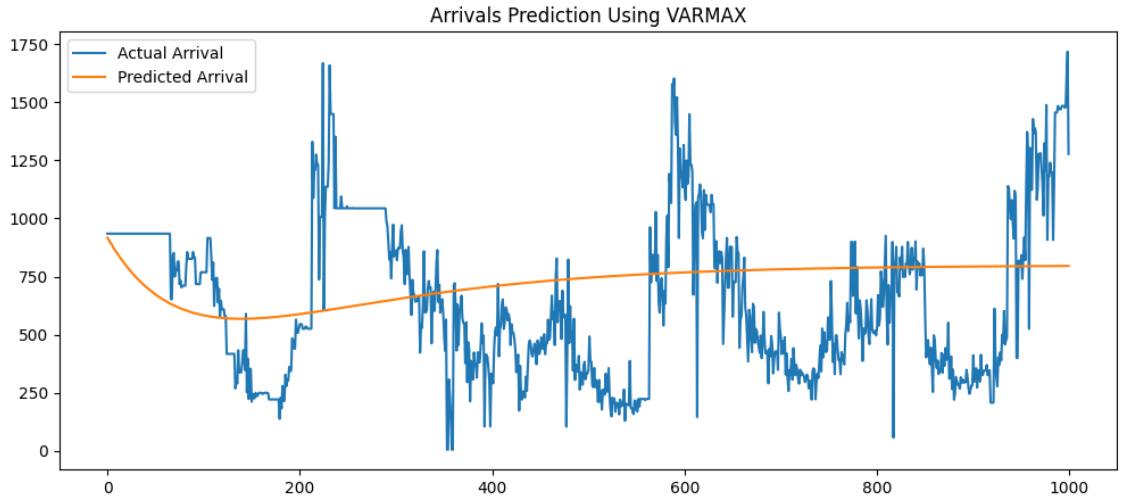

Figure 18: Arrival Prediction using VARMAX

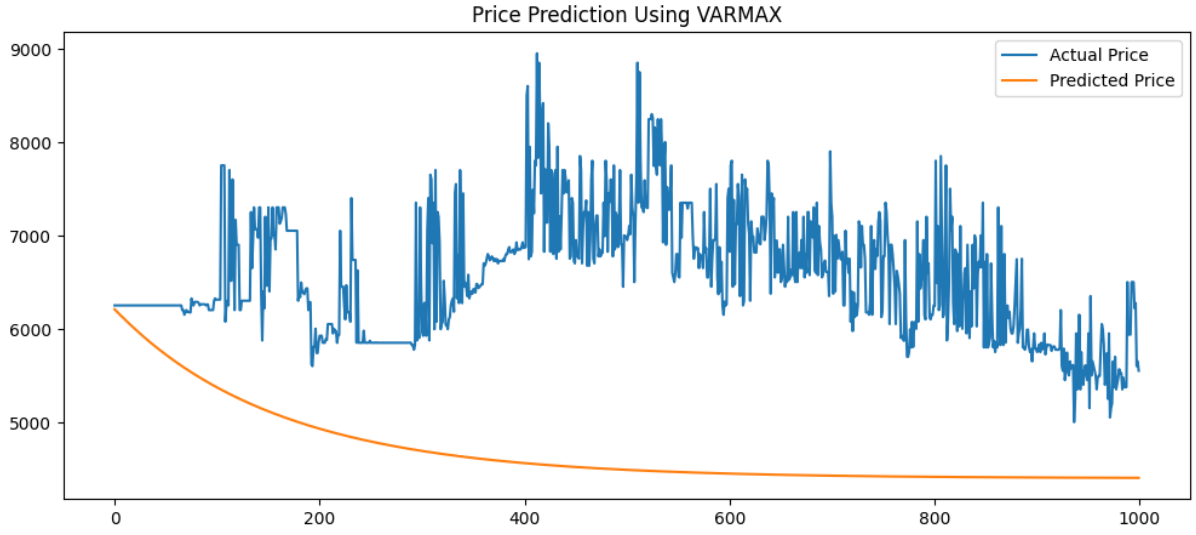

Figure 19: Price Prediction using VARMAX

8.8. RANDOM FOREST REGRESSOR:

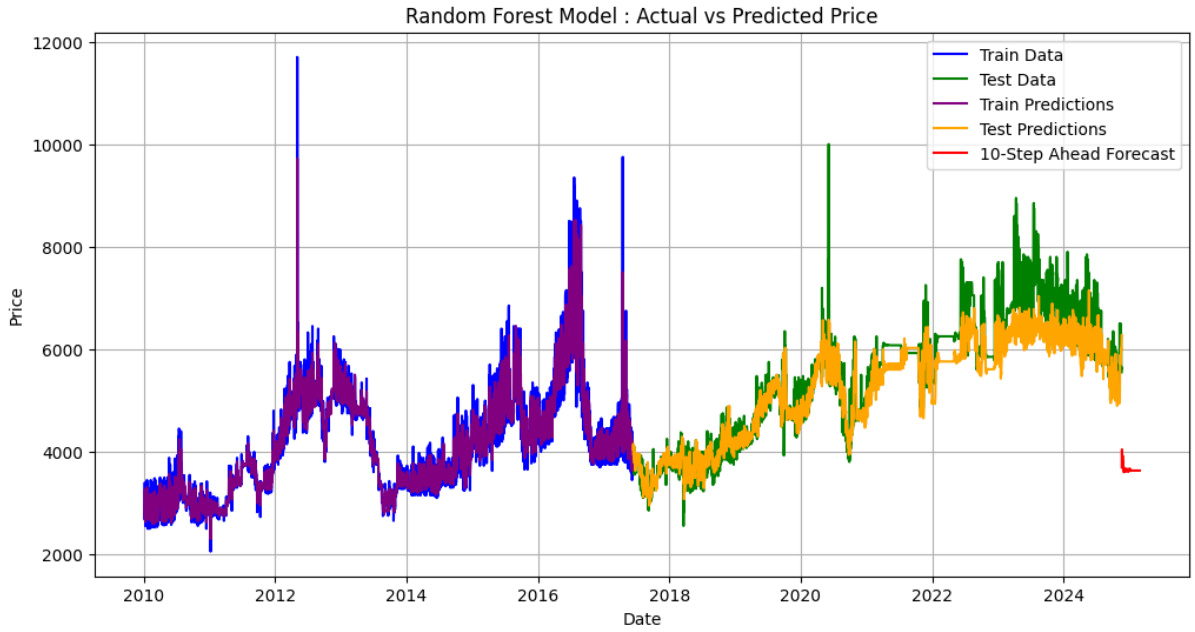

Figure 20: Random Forest Time Series Prediction

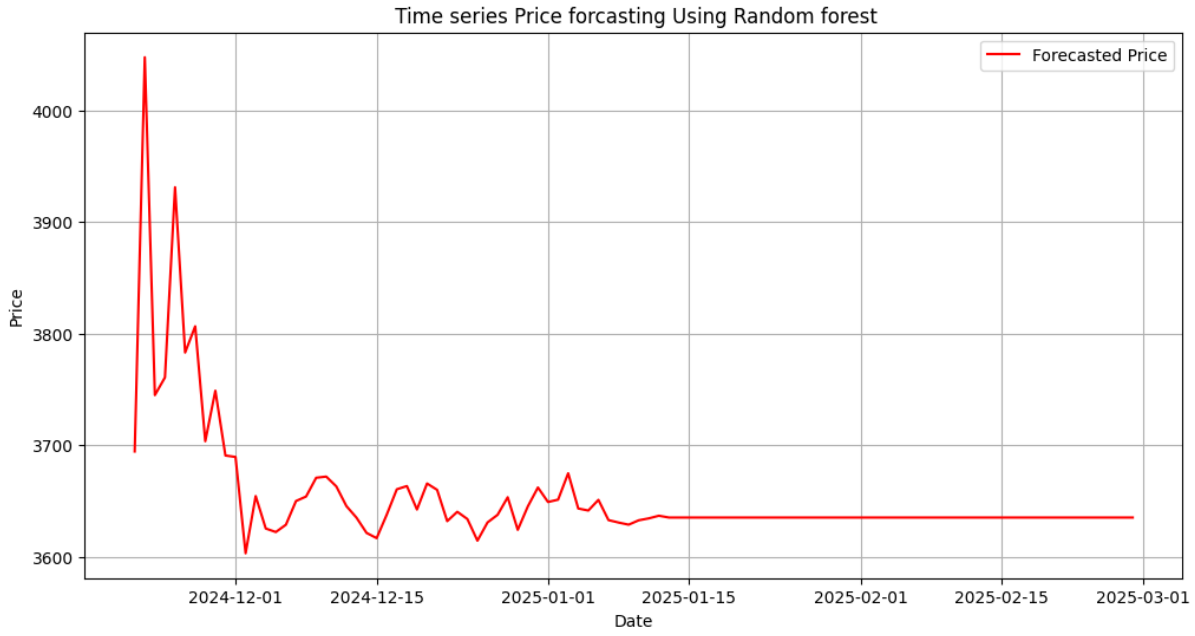

Figure 21: Random Forest Time Series Forecasting

8.9. LSTM:

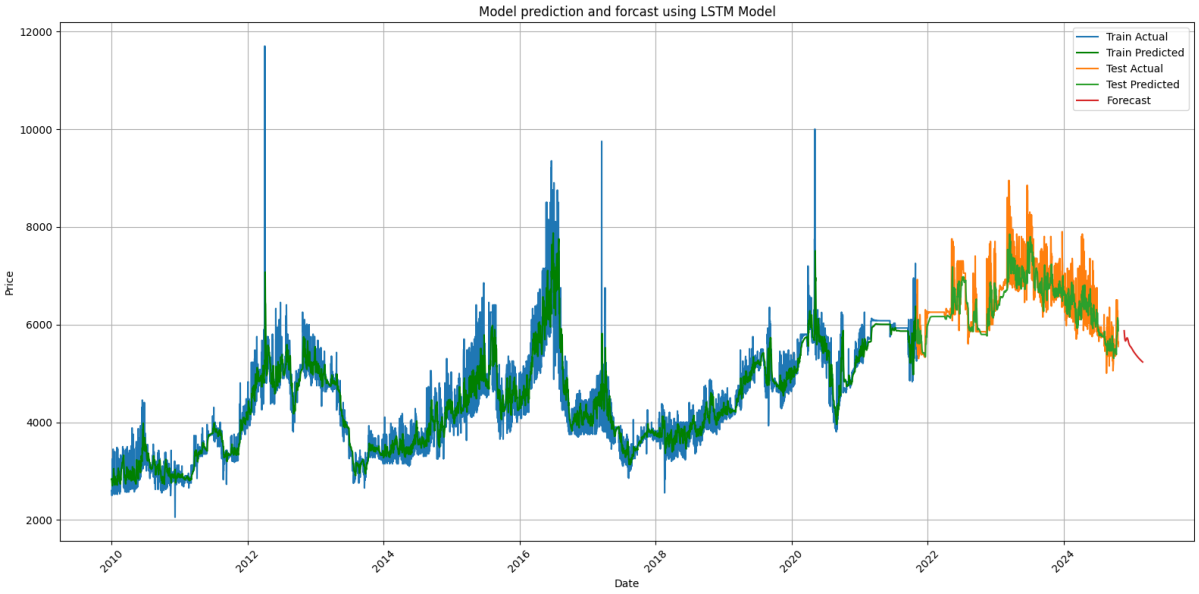

Figure 22: LSTM model Time Series Prediction

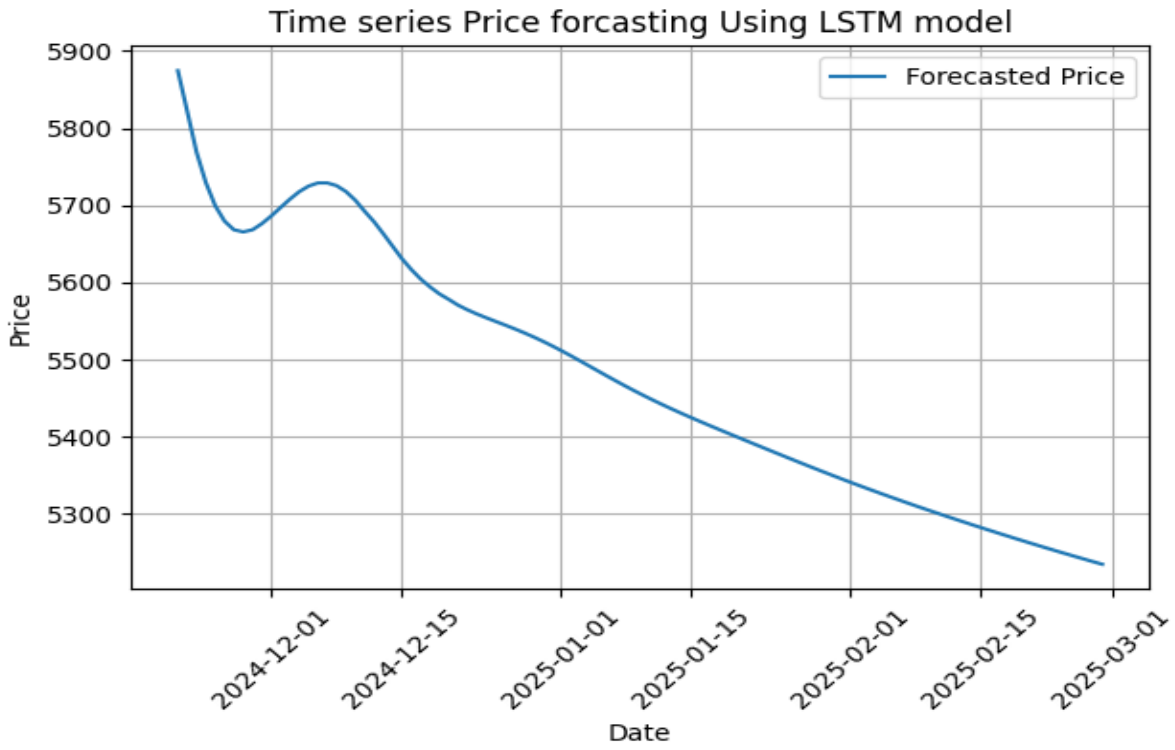

Figure 23: Random Forest Time Series Forecasting

8.10. GRU:

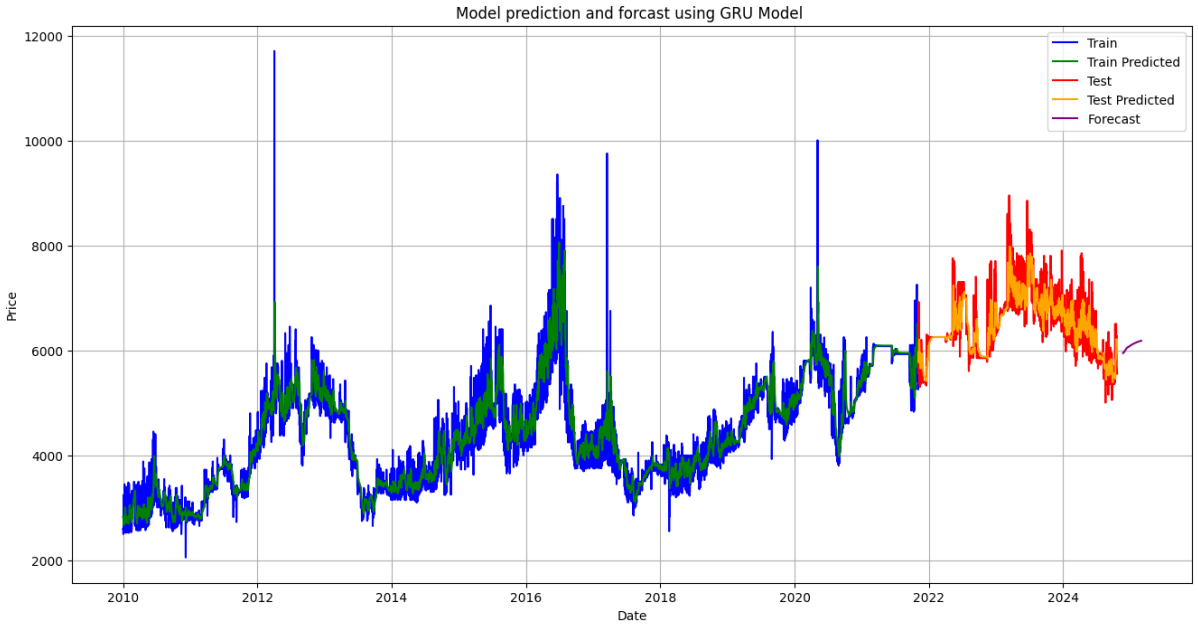

Figure 24: GRU Time Series Prediction

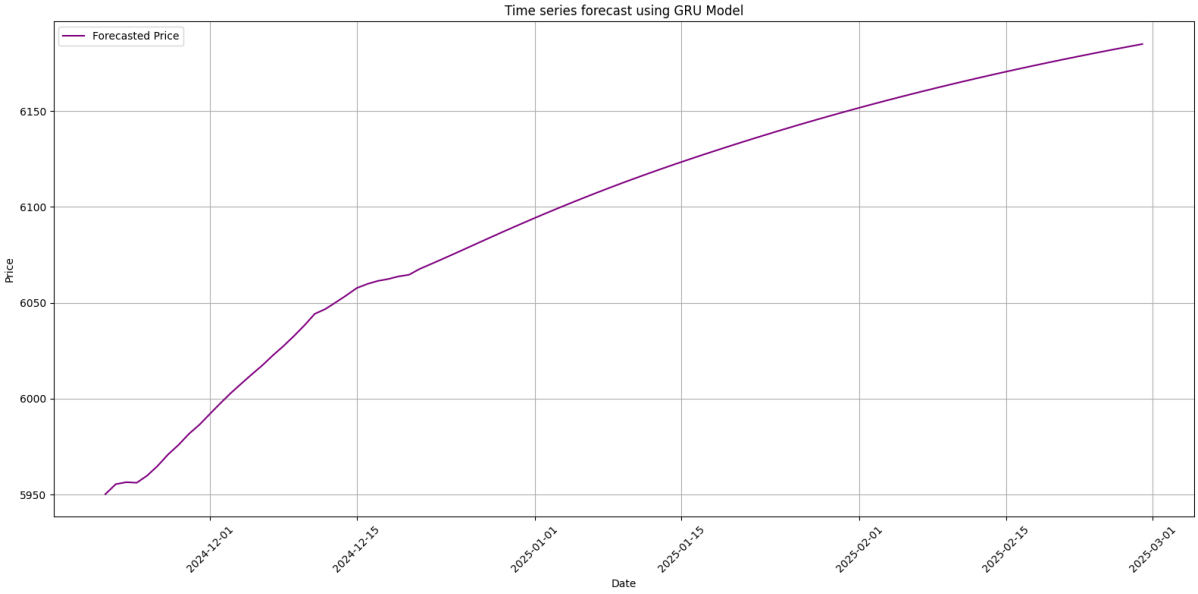

Figure 25: GRU Time Series Forecasting

9. Model Comparison

Table 2: Model Comparison

Sr. No	MODEL	AIC	BIC	MSE	RMSE	RMSE as a percentage of the range
1	ARIMA	65764.11	65821.69	539660.41	734.62	18.6
2	SARIMA	48765.87	48802.41	4680426.58	2163.43	34.59
3	ARIMAX	40937.39	40984.65	31120891.99	5578.61	74.8
4	SARIMAX	53255.67	53298.85	13694583.28	3700.62	59.69
5	ARCH	49217.42	49284.38	525864.69	725.17	11.59
6	GARCH	41329.54	41394.50	961105.19	3100.17	41.61
7	VAR	133079.62	133367.53	6560.29	2031.94	30.97
8	VARMAX	133079.62	133367.53	1192.85	587.35	49.24
9	RANDOM FOREST	-	-	253781.44	503.77	6.76
10	LSTM	-	-	172456.83	415.28	10.51
11	GRU	-	-	158914.62	398.64	10.09

10. Conclusion

- **Best Model for Accuracy**:
  GRU emerges as the best-performing model with the lowest MSE and RMSE (398.64 and 10.09% of the range), making it highly suitable for predicting groundnut prices accurately.
- **Stable and Reliable Models**:
  Random Forest and GRU are highly stable models, demonstrating low RMSE percentages (6.76% and 10.09%, respectively) and strong performance in capturing data patterns effectively.
- **ARCH and GARCH Models**:
  These models exhibit significantly lower MSE and RMSE than ARIMA-based methods, highlighting their strength in capturing data volatility. However, GARCH (41.61% RMSE) has relatively higher error compared to ARCH (11.59%).
- **Deep Learning Models (LSTM and GRU)**:
  Both models outperform traditional statistical models in accuracy. GRU slightly edges out LSTM with a lower RMSE, making it a robust choice for forecasting complex time-series data.
- Random Forest has the lowest MSE and RMSE, indicating excellent pattern recognition.
- ARIMA is suitable for baseline forecasts and capturing linear patterns in time-series data.

Reference

[1] J. A. Duke, *CRC Handbook of Nuts*, 1st ed. CRC Press, 2018. doi: 10.1201/9781351071130.

[2] "EFFECT OF PHOSPHOGYPSUM ON NUTRIENT UPTAKE, YIELD AND QUALITY OF SUMMER GROUNDNUT", [Online]. Available: https://krishikosh.egranth.ac.in/server/api/core/bitstreams/e5ec2e2d-216d-4767-a14e-65e3e01f3e55/content

[3] "Agricultural Commodities Price Prediction", [Online]. Available: https://www.researchgate.net/profile/Samyak-Kamble/publication/379696260_Agricultural_Commodities_Price_Prediction/links/661632a7f7d3fc28743f95f5/Agricultural-Commodities-Price-Prediction.pdf

[4] "Price Volatility in Food and Agricultural Markets: Policy Responses", [Online]. Available: https://www.alimenterre.org/system/files/ressources/pdf/86_g20_foodpricevolatility_en.pdf

[5] "Macroeconomic impacts of oil price volatility: mitigation and resilience", [Online]. Available: https://link.springer.com/article/10.1007/s11708-014-0303-0

[6] R. K. Bansal, V. K. Gondaliya, and A. S. Shaikh, "A Review of the Status of the Groundnut Production and Export of India," *Indi. Jour. of Econ. and Develop.*, vol. 13, no. 2, p. 369, 2017, doi: 10.5958/2322-0430.2017.00190.1.

[7] "https://www.fao.org/fileadmin/user_upload/inpho/docs/Post_Harvest_Compendium_-_Groundnut.pdf".

[8] "The world groundnut economy Facts, trends, and outlook", [Online]. Available: https://oar.icrisat.org/1074/

[9] "Forecasting of Arrival and Price of Groundnut in Rajasthan by ARIMA Model for Livelihood of Farmers", [Online]. Available: http://ebooks.manu2sent.com/id/eprint/1519

[10] "Forecasting Oilseeds Prices in India: Case of Groundnut", [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3237483

[11] "[No title found]," *Int. J. Stat. Appl. Math.*.

[12] "BANNORANDSHARMA1-20.pdf."

[13] “https://aaj.net.in/wp-content/uploads/2023/12/70-2-018.pdf.”

[14] “Modelling and Forecasting Wholesale Groundnut Prices in Bikaner District of Rajasthan for Marketing Intelligence.” [Online]. Available: https://www.researchgate.net/publication/315826972_Modelling_and_Forecasting_Wholesale_Groundnut_Prices_in_Bikaner_District_of_Rajasthan_for_Marketing_Intelligence

[15] “Forecasting the dynamics of cumulative COVID-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-Regressive Integrated Moving Average (ARIMA) and Seasonal Auto-Regressive Integrated Moving Average (SARIMA)”, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621000843

[16] F.-M. Tseng and G.-H. Tzeng, “A fuzzy seasonal ARIMA model for forecasting,” *Fuzzy Sets and Systems*, vol. 126, no. 3, pp. 367–376, Mar. 2002, doi: 10.1016/S0165-0114(01)00047-1.

[17] “ARIMAX: Time Series Forecasting with External Variables”, [Online]. Available: https://www.researchgate.net/profile/Marcin-Majka-2/publication/384196976_ARIMAX_Time_Series_Forecasting_with_External_Variables/links/66eddc636b101f6fa4f88253/ARIMAX-Time-Series-Forecasting-with-External-Variables.pdf

[18] “ARIMAX”, [Online]. Available: https://www.geeksforgeeks.org/what-is-an-arimax-model/

[19] F. R. Alharbi and D. Csala, “A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach,” *Inventions*, vol. 7, no. 4, p. 94, Oct. 2022, doi: 10.3390/inventions7040094.

[20] S. Degiannakis and E. Xekalaki, “Autoregressive Conditional Heteroscedasticity (ARCH) Models: A Review,” *Quality Technology & Quantitative Management*, vol. 1, no. 2, pp. 271–324, Jan. 2004, doi: 10.1080/16843703.2004.11673078.

[21] T. Bollerslev, “Glossary to ARCH (GARCH),” *SSRN Journal*, 2008, doi: 10.2139/ssrn.1263250.

[22] H. Lütkepohl, “Vector autoregressive models,” in *Handbook of Research Methods and Applications in Empirical Macroeconomics*, N. Hashimzade and M. A. Thornton, Eds., Edward Elgar Publishing, 2013. doi: 10.4337/9780857931023.00012.

[23] “VARMAX and Transfer Function Models”, [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-20790-8_3

Appendix

Date	Price	Arrivals
01-01-2010	2725	460
02-01-2010	2900	410
03-01-2010	2812.5	435
04-01-2010	2650	500
05-01-2010	2675	420
06-01-2010	2650	1000
07-01-2010	2800	350
08-01-2010	2575	310
09-01-2010	2750	280
10-01-2010	2683.333	476.6667
11-01-2010	2550	200
12-01-2010	2665	428
13-01-2010	2550	250
14-01-2010	2606.25	260
15-01-2010	2650	170
16-01-2010	2650	160
17-01-2010	2600	195
18-01-2010	2700	600
19-01-2010	2700	320
20-01-2010	2600	240
21-01-2010	2900	300
22-01-2010	2600	480
23-01-2010	2625	350
24-01-2010	2687.5	381.6667
25-01-2010	2750	400