Predicting Vegetable Prices in Sri Lanka Using Machine Learning Techniques

E. L. N. D. Madubhashini

2021

Predicting Vegetable Prices in Sri Lanka Using Machine Learning Techniques

A dissertation submitted for the Degree of Master of Business Analytics

E. L. N. D. Madubhashini
University of Colombo School of Computing
2021

ABSTRACT

Sri Lanka is an agriculture-based country, nearly 33.7% of households are farm dependent. As farmers they are contributing to the Sri Lankan economy. The major problem that the farmers face, is when the vegetables are not worth the price and farmers are unaware of the marketing price. Vegetable price prediction is very challenging due to many reasons such as climate changes, demand, supply etc. But, predicting the vegetable prices are essential for the Sri Lankan economy, agriculture sector, farmers as well as consumers to make effective decisions and to prevent the loss of social welfare due to excess supply and excess demand.

During the last decade, couple of studies were used traditional statistical techniques like ARIMA to forecast the crop prices in Sri Lanka. However, no study was utilized machine learning techniques like novel based approach to predict the vegetable prices in the Sri Lankan context.

This study was presented different machine learning techniques such as gradient boost, XG boost, random forest regression and stack regression techniques which were used to predict the vegetable prices in Sri Lanka. Utilized models for each vegetable were assessed based on the performance matrices and the best performing model for each vegetable was suggested for the future vegetable price prediction. As the source of data, daily price reports of vegetables published in the Central Bank of Sri Lanka from 2016-2022 was used.

Keywords: Vegetable price prediction, Machine learning techniques, Time series data

DECLARATION

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: E. L. N. D. Madubhashini
Registration Number: 2019/BA/014
Index Number: 19880146

Signature: Date: 25/02/2023

This is to certify that this thesis is based on the work of
~~Mr~~./Ms. E. L. N. D. Madubhashini
under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: R. A. B. Abeygunawardhana

Signature: Date:

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all those who provided support to make my research on “Predicting vegetable prices in Sri Lanka using machine learning techniques” successful.

First, I would like to express my gratitude towards to the project supervisor Dr. R. A. B. Abeygunawardhana, Senior Lecturer, Department of Statistics, University of Colombo. I am highly indebted to him for his guidance and constant supervision as well as for providing necessary information regarding the project and for his support in completing this project successfully. His motivation and guidance have inspired me throughout this journey.

As the course coordinator, my special thanks go to Dr. T. Halloluwa for the support given me throughout this course. Especially, I would like to thank Mrs. P. L. Rupasinghe from the Central Bank of Sri Lanka, who helped me to collect the data set without any delay.

Finally, I wish to thank to every person who helped me to make this research success.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

ARIMA	Auto-Regressive Integrated Moving Average
AR	Auto Regression
MA	Moving Average
XGBoost	Extreme Gradient Boost
GB	Gradient Boost
RF	Random Forest
AI	Artificial Intelligence
GDP	Gross Domestic Product
SARIMA	Seasonal Auto-Regressive Integrated Moving Average
BPNN	Back Propagation Neural Network
RBNN	Radial-Based Neural Network
LSTM	Long Short-Term Memory
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
BSTS	Bayesian Structural Time Series
GANN	Genetic Algorithm Based Neural Network
LSTM	Long Short-Term Memory
ML	Machine Learning
SONN	Self-Organized Neural Network
SVR	Support Vector Regression
EDA	Exploratory Data Analysis
ACF	Auto Corelation Function
PACF	Partial Auto Corelation Function
MSE	Mean Squared Error
MAPE	Mean Absolute Percentage Error
TSS	Total Sum of Squares
RSS	Residual Sum of Squares

CHAPTER 1

INTRODUCTION

1.1 Project Overview

Machine learning is a sub-area of Artificial Intelligence (AI). It is concerned with the design and development of algorithms and techniques that allow computers to learn. Machine learning is a novel-based approach that can be used in the Sri Lankan agriculture sector to improve the existing procedures effectively.

Agriculture plays a vital role in the Sri Lankan economy. Most families in rural areas are dependent on agriculture. The agricultural sector contributes about 7.4% to the national Gross Domestic Product (GDP) of the Sri Lankan Economy. Among the total Sri Lankan population, 30% of them are employed in the agricultural sector. Among all the agricultural products, vegetables have become one of the most important commodities in Sri Lankans' daily lives ("Sri Lanka - Agricultural Sector", n.d.).

Currently, Sri Lankan government sectors collect and present the daily, weekly, monthly and yearly price details for vegetables for selected markets like Pettah and Dambulla. But there is no proper process for analysing such information using scientific approaches. So, the main objective of this thesis is to identify the hidden patterns behind the history of vegetable prices for different markets and early prediction of the vegetable prices using machine learning techniques.

Using machine learning techniques, it provides the ability to transform this data into useful information for decision making. So that the farmers, consumers, agricultural departments and government can be made aware of it and throughout that, they can make better decisions to reduce their financial losses and hence can increase the demand.

1.2 Proposed Solution

The vegetable market plays an important role in the agriculture sector of Sri Lanka. There are several large vegetable markets in Sri Lanka such as Pettah (recently moved to Peliyagoda), Dambulla, Meegoda, Kandy, Norochchole, Kappetipola, Nuwara Eliya and Thambuththegama. Since wholesales of vegetables are mainly conducted in Pettah and Dambulla, vegetable prices in Pettah and Dambulla markets were considered for this research.

The prices of the vegetable market depend on the demand and supply of vegetables over time. Due to price fluctuations in the current market, prices of vegetables have more impact on the cost of living. Most of the countries have established early price detection systems to detect and evaluate the prices of vegetables so that, the prices can be adjusted based on the time and controlled when it is in an abnormal state. Since Sri Lanka is a developing country, we are lacking such early price prediction systems in the agricultural sector (Li et al., 2014). But it is necessary to modernize agricultural practices based on technological advancements to meet the demanding requirements.

In this research, wholesale price and retail price of selected vegetables in the Pettah market and Dambulla market in Sri Lanka were predicted using machine learning techniques by applying univariate time series forecasting models. For that, the daily price reports of vegetables published by the Central Bank of Sri Lanka from 2015-2022 were used as the source of data ("Daily Price Report, Central Bank of Sri Lanka", n.d.).

This proposed solution was predicted the retail and wholesale vegetable prices in the mentioned markets. Farmers are complaining that a huge amount of profit goes to the third party, not to the farmer or the seller. By taking the wholesale price difference in the Pettah market and the wholesale price difference in the Dambulla market, we can check the amount that goes to the third party. Using this solution we can identify the seasonal periods of each vegetable, compare Pettah market prices with Dambulla market prices, compare wholesale prices with retail prices. Likewise, valuable predictions and comparisons can be made. Based on the results, the best-performing model was proposed to use for vegetable price prediction.

Suddenly, Sri Lanka is suffered through one of the world's worst economic crises staring from last January. Due to that, there was a huge increase of vegetable prices during this period. This effect was not addressed from our initial solution. Therefore, as an additional step, vegetable prices were predicted after the economic crisis separately by collecting daily price reports of vegetables from April of 2022 – October of 2022. Where the retail and wholesale vegetable prices were predicted in the mentioned markets.

1.3 Motivation

Sri Lanka is an agriculture-based country. The vegetable market plays a major role in Sri Lankan agricultural sector. Cultivation of vegetables is mainly a labour-intensive task and because of that opens lots of employment opportunities in rural areas. In Sri Lanka, there is no proper system to forecast the prices of vegetables. Farmers plan their cultivations based on their previous experience and knowledge. They require more support from the IT industry for their development and improvements to plan their cultivations. Currently, we are lacking such technological advancement systems in the agricultural sector (Rakhra, 2020).

Vegetable prices are varying fast and are unstable. That makes a great impact on our daily lives. Their prices are affected by many causes such as climate changes, supply, demand, population, national policies, area of arable land, international financial markets, price of alternatives, economic growth, international trade, political situations festivals, etc. Due to that, price prediction is more difficult than ordinary commercial products. It is very challenging to collect vegetable prices based on these factors (Luo et al., 2011).

Apart from these factors forecasting vegetable prices is challenging due to data quality issues (manually recorded data), unreliability present in future weather forecasting and high fluctuation experienced in the past vegetable prices as well (Jain et al., 2020). Over periods, the oversupply of vegetables leads to vegetable prices plummeting. Which causes financial losses to agricultural households. Similarly, undersupply of agricultural products leads to increasing prices putting a burden on consumers (Yin et al., 2020). This imbalance factor in the supply and demand of vegetables affects both customers and farmers. Therefore, the government is lacking in making decisions to balance those factors.

To overcome such problems, an accurate and reliable price prediction system is required for the Sri Lankan vegetable market. By using predicted future prices, farmers can produce fewer vegetables beforehand the excess of supply in the market, where the price is expected to drop and detect undersupply periods where the demand is high (Shukla and Jharkharia, 2011). Based on that, they can plan their crop cultivations. Likewise, this research leads to reducing the risk factors for rural farmers who cultivate vegetables and based on that, decisions can be made for the more profitable cultivations. Accurate price prediction of vegetables also helps farmers to decide the best time to sell their harvest and based on that, they can achieve maximum profits.

Government can use predicted prices for planning agricultural development, post-harvest storage and management of harvest programs to stabilize market price volatility throughout the year. Consumers can also use this price prediction to plan their daily lifestyles. Therefore, this innovative application is not only useful for farmers and consumers. But this is also useful for all the decision-makers such as the Sri Lankan economy, agricultural planning, market planning and farming sectors as well (Nasira and Hemageetha, 2012).

1.4 Objectives

Predicting vegetable prices is a challenging problem in the local community. The main objective of this research is to predict the prices of selected vegetables in Sri Lanka.

Specific objectives of this research are,

- To predict the wholesale prices of selected vegetables in different markets in Sri Lanka.
- To predict the retail prices of selected vegetables in different markets in Sri Lanka.

1.5 Background

Agriculture is the most important sector of the Sri Lankan economy. The vegetable market plays a vital role in the agricultural sector. The main issue that the farmers in the rural area are facing is, that they cultivate the vegetables based on their previous experiences. Once they get the harvest and when they bring them to the market, they face difficulties since they do not have reasonable prices for the vegetables. This huge price volatility has been a major issue during the past few years for all parties including farmers, consumers and agricultural sectors. Due to that reason, most of the rural farmers are affected economically by losing their profits, even their capital.

A reliable vegetable price prediction system is required to help such farmers. By having such a system that helps farmers for more profitable cultivations, manage price risks and make informed decisions, throughout that they can achieve more benefits. This system is not only helpful for farmers; it is helpful for customers, agricultural sectors and government as well. As a developing country like Sri Lanka, proper planning by using such kind of efficient forecasting system is very important to achieve the sustainable growth of the country.

In this research, daily vegetable prices of selected vegetables in Pettah and Dambulla markets were used as the source of data. Which was collected from the Central Bank of Sri Lanka. Time-series forecasting is one of the main active research areas during the past couple of years and that has been studied under various categories such as electricity price prediction, forecasting crop prices, stock price predictions (Lu and Zhang, 2004), energy load forecasting (Amarasinghe et al., 2017), etc. Vegetable price prediction is one such field of time series forecasting problem that was considered in this study. By applying several machine learning techniques such as novel-based time series forecasting methods, wholesale vegetable prices and retail vegetable prices were predicted based on the collected data set.

1.6 Scope of the research

This study was focused on different machine learning techniques which can be used for vegetable price prediction and based on the results of selected models, the best performing model was proposed as the vegetable price prediction. As the data source, daily price reports of Sri Lankan vegetables published by the Central Bank were collected from 2015 to 2022. In these reports, daily wholesale prices and retail prices of a set of regular vegetables such as beans, carrots, cabbage, tomato, brinjal, pumpkin, green chili and lime for Pettah and Dambulla markets were available. Among them, daily wholesale prices and retail prices of beans and carrots in Pettah and Dambulla markets were selected for this research.

In addition to the vegetable price prediction, predicting the market level vegetable price differences, wholesale and retail price differences, seasonal periods of each vegetable, likewise valuable predictions can be made. Although the data were available from 2015-2022, there was an economic crisis in early 2022 and vegetable prices were drastically increased. In this research economic crisis effect also considered and vegetable prices were predicted after the economic crisis and before the economic crisis effect. Initially, such effect was not expected therefore vegetable prices during the inflation were considered as a separate price prediction task except to the main scope.

1.7 Feasibility Study

Agriculture has a significant contribution to the primary sector of Sri Lanka's economy. Vegetables play a major role among them. At present, farmers are having massive losses due to the unawareness of vegetable prices. There are several studies related to this solution and they have used traditional statistical approaches for crop price prediction in the past and machine learning approaches for crop price prediction in recent years.

As statistical methods, they have used Auto-Regressive Integrated Moving Average Method (ARIMA), Seasonal Auto-Regressive Integrated Moving Average Method (SARIMA), naïve model, exponential smoothing, etc. Machine learning is a novel-based approach that is used for vegetable price prediction. Machine learning has proved that it is better than traditional time series methods for price prediction since it has many linear and nonlinear forecasting models.

As machine learning and neural network techniques decision tree algorithm, Back Propagation Neural Network (BPNN), Radial-Based Neural Network (RBNN) were used for the crop price prediction (Subhasree and Priya, 2016). Long Short-Term Memory (LSTM), multilayer feedforward neural network algorithm, genetic-based neural network, improved version of neural network methods, a hybrid version of neural network methods and regression models were also used for forecasting the crop prices (Li et al., 2014).

In this study, vegetable price data set was collected from the Central Bank of Sri Lanka. The Central Bank publishes daily price reports for a set of vegetables. Among them, wholesale prices and retail prices of a set of vegetables in Pettah and Dambulla markets were selected for this analysis.

This study was focused on different time series forecasting machine learning techniques for vegetable price prediction. Data set was applied to each of the selected models and based on the results, the most accurate model will be selected and proposed for the price prediction.

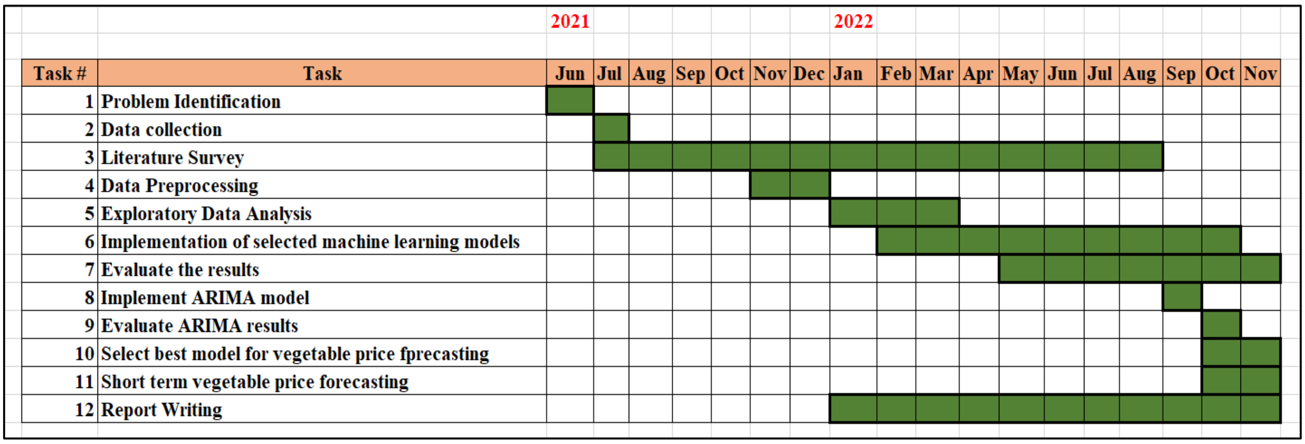
1.8 Structure of The Dissertation

Table 1.1 depicts the structure of the research project which provides the chapter-wise summary. The content and the overall expectation of each chapter were explained there. Having a proper structure of the thesis like this helps anyone to easily understand the research problem and the proposed solution.

Table 1.1: Structure of The Research Project

Chapter	Content	Description
Chapter 1	Introduction	Explain the motivation, project overview, proposed solution, objective and scope of the research.
Chapter 2	Background and Related Work	Explain the work done by others in the past and what kind of methods they have used related to the vegetable price prediction.
Chapter 3	Design and Methodology	Explain the methods and the design were applied for the vegetable price prediction including data set preparation (Pre-processing)
Chapter 4	Results and Evaluation	Apply the data set for the selected methods/ models and generate the results and discuss the deliverables.
Chapter 5	Conclusion and Future Work	Out of the selected models conclude which model or method provides more accurate and reliable results and based on that suggest the best model for the vegetable price prediction and mention about the future enhancements as well.

1.9 Refined Timeline



		2021							2022									
Task #	Task	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov
1	Problem Identification	🟩																	
2	Data collection		🟩																
3	Literature Survey		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			
4	Data Preprocessing						🟩	🟩											
5	Exploratory Data Analysis								🟩	🟩	🟩								
6	Implementation of selected machine learning models									🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	
7	Evaluate the results												🟩	🟩	🟩	🟩	🟩	🟩	🟩
8	Implement ARIMA model																🟩		
9	Evaluate ARIMA results																	🟩	
10	Select best model for vegetable price fprecasting																	🟩	🟩
11	Short term vegetable price forecasting																	🟩	🟩
12	Report Writing								🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Figure 1.1: Refined Timeline

Figure 1.1 indicates the refined timeline involved for this study.

1.10 Summary

This chapter was provided an introduction of the study. It was focused on the background of the vegetable price prediction, existing problems and the proposed solution for the vegetable price prediction. This chapter was further discussed with the objectives, scope and feasibility study of the research. Then the structure of the dissertation was provided to identify how the chapters are ordered in this dissertation. Refined timeline also shown to get an idea about the milestones achieved during each stage. Next chapter will discuss about the related work for this research.

CHAPTER 2
BACKGROUND AND RELATED WORK

Since the agricultural industry plays an important role in the world and vegetables are the main part of it, predicting vegetable prices is very important for farmers, consumers, the economy, and the agricultural industry. Although the prediction of vegetable prices is very challenging, during the past few years, vegetable price prediction was done using different approaches all over the world. In this chapter, the previous work related to the vegetable price prediction was presented and explained the methodologies, algorithms and techniques they have used and identified their gaps.

2.1 Predicting Crop Prices Using Traditional Statistical Methods

Traditional statistical time series forecasting methods are commonly used for forecasting crop prices in the early decades. Several studies were analysed and reviewed related to the crop price prediction using statistical methods under this section.

The prices in agricultural markets are determined based on the demand and supply of agricultural products. As a developing country like Sri Lanka, they lack of having a proper system to measure the real impacts of the demand and supply forces on market prices. (Cyril, 1988) has analysed rice's retail and wholesale prices in Colombo markets, Sri Lanka using the Box Jenkins ARIMA method. Based on the results, stated that both retail and wholesale market prices exhibit seasonality in prices. However, (Jadhav et al., 2017) have concluded that, the ARIMA model serves as a good technique for predicting the magnitude of any variable. As a limitation, it has mentioned requirements for a large sample size. Where the prices were predicted for Paddy, Ragi and Maize cereal crops which are cultivated in Karnataka. But (Jadhav et al., n.d.) have stated that, the mixed ARIMA/ GARCH model outperforms other models such as ARIMA, exponential smoothing, Generalized Autoregressive Conditional Heteroskedasticity (GARCH) when forecasting cocoa bean prices in Malaysia.

Moving onto vegetable price prediction, where farmers do not have much knowledge about the market trends and the price fluctuations of vegetables. Due to that, there is a huge mismatch between demand and supply for vegetables. This causes either waste of excess produce or unsatisfied customers in the end. As a solution to this problem, (Shukla and Jharkharia, 2011) have developed the ARIMA price forecasting model. Based on the results, concluded that, the model is highly efficient in predicting the demand for vegetables on a day-to-day basis. Although (Yoo, 2016) has stated that, reliable price prediction is very important in the vegetable market to prevent the loss of social welfare caused by excess supply or excess demand. Farmers can refer to the predicted prices and accordingly, they can plan their cultivations. For example, farmers can produce less vegetables beforehand in the excess supply market, where the price is expected to drop. Bayesian Structural Time Series (BSTS) Model is applied for the vegetable price prediction.

Based on the results states that, by introducing a typical index into the BSTS models, prediction power for vegetable prices can be improved for the selected vegetables such as dried red pepper, garlic, and onion. However, (Dieng, 2018) has stated that, the ARIMA model appears to be the best forecaster of the prices of the three crops namely tomato, potato and onion out of the naive model, exponential smoothing, Box and Jenkins Autoregressive Integrated Moving Average (ARIMA) model and spectral analysis techniques.

2.2 Predicting Crop Prices Using Deep Learning and Neural Network Methods

Forecasting crop prices using deep learning and neural network methods is a popular method in recent years. Several studies related to crop price prediction which are done using neural networks and deep learning methods are analysed and reviewed in this section.

(Subhasree and Priya, 2016) have stated that, predicting vegetable prices are very important in the agricultural sector for effective decision-making. But that is trickier than predicting usual commercial product prices. Therefore, most perishable vegetables like tomato, ladies fingers, broad beans, small onion and brinjal have been taken as the experimental data. Three machine learning methods were applied and concluded that the Genetic Algorithm Based Neural Network (GANN) has more advantages over the other two models namely Radial Basis Neural Network (RBNN), Back Propagation Neural Network (BPNN) for vegetable price prediction. To analyse the monthly price data for selected five types of vegetables such as cucumbers, peppers, tomatoes, green beans and cabbages (Li et al., 2014) have developed a hybrid approach by combining H-P filtering and neural network model. To compare with the results of this hybrid approach, traditional forecasting models such as the Auto-Regressive Integrated Moving Average method (ARIMA) and Back Propagation Artificial Neural Network (BPANN) method were implemented. Based on the accuracy of each of the methods, concluded that the hybrid model has the best performance in predicting the future prices of the vegetables.

There is a saying called, food is the god of people. Vegetables are half of it. Vegetable price is unstable and changing fast. So, predicting vegetable prices are complicated than commercial products but the prediction is very important. (Luo et al., 2011) have presented four models to predict the vegetable market price which are the Back Propagation Neural Network (BPNN), a neural network model based on the genetic algorithm, Radial Basis Neural Network (RBNN) and an integrated prediction model based on the above three models. Based on the performance of each model, concluded that the integrated prediction model is performed better than the other three models for the Lentinus Edodes' price prediction. However, (Nasira and Hemageetha, 2012) have developed a BP neural network model to predict the price of tomatoes and then performance was measured based on accuracy. Finally concluded that, BPNN is one way of predicting vegetable prices with non-linear time series. To predict the prices of five crops such as cabbage, radishes, onion, hot peppers, and garlic (Yin et al., 2020) have developed STLATTLSTM (STL-Attentionbased LSTM) model which integrated the seasonal trend decomposition using the Loess (STL) pre-processing method and attention mechanism based on long short-term memory (LSTM) methods. This proposed ATTLSTM model achieved the best performance than the other three benchmark models.

2.3 Predicting Crop Prices Using Machine Learning Techniques

Forecasting crop prices using Machine Learning (ML) techniques has been a topic of interest in the recent few years. Recent studies have proven that using ML methods, which produces more accurate results than traditional statistical methods. In this section, several studies were analysed and reviewed related to crop price prediction using ML techniques.

Agriculture is the backbone of every economy. There is no system in place to estimate costs to advise farmers about what crops to grow. Farmers are unaware of the demand expected in the agricultural economy. Due to that, they end up with losses. Price prediction is a challenging and important agricultural problem to address such issue. (Mulla and Quadri, 2020) have presented an approach to creating a user-friendly interface for farmers which gives the analysis of several crop production based on available data. To achieve the prediction, a decision tree algorithm was used and stated that it has produced good accuracy. But (Samuel, 2020) has applied several machine learning techniques and neural network methods such as decision trees, logistic regression, XGBoost, neural networks, and clustering algorithms to predict the crop based on multiple factors like area planted, area harvested, etc. They have concluded that XGBoost provided the best performance among them. However, (Yin et al., 2020) have implemented a crop price forecasting system that provides price predictions and profit predictions for the crop. Crop price prediction was done using the Naïve Bayes algorithm. Profit prediction was done using the K Nearest Neighbor technique. This study has stated that the proposed solution helps the farmers to make better-informed decisions and manage the price risk.

Farmers are an important part of the agricultural industry. Some of the major problems, farmers are facing today are commodity price prediction, yield prediction and profit prediction. Among those, predicting commodity prices is challenging due to frequent price changes. This price prediction is a big issue for farmers because they are unaware of the market prices. Forecasting the price of commodities helps farmers to aware of prices as well as the government to make better decisions. (Varun et al., 2019) have presented a new model based on the support vector machine, neural network and extended Kalman filter method to predict the prices of commodities. This study has stated that the proposed model provides good accuracy hence farmers can sell their crops without third-party involvement. However, (Lavanya and Raguchander, 2013) have developed a support vector regression (SVR) based price prediction method to predict the crop prices which were extracted using a self-organized neural network (SONN) through a website. This paper stated that the proposed method achieved more accurate predictions than other traditional methods like regression techniques.

2.4 Summary of previous studies related to the crop price prediction

Above discussed previous studies related to the crop price prediction in all over the world can be grouped into different categories such as statistical techniques, deep learning and neural network techniques and machine learning techniques. Previous work related to each category were summarized as shown in following tables. Each study presents the data set it has used, techniques they have applied for the price prediction and the final conclusion. In earlier decades, most of the studies have applied traditional statistical techniques and later they tried deep learning, neural network and machine learning techniques.

Table 2.1 depicts the previous studies related to the crop price prediction which were implemented using traditional statistical techniques. Autoregressive Integrated Moving Average (ARIMA) method was the most commonly applied method over the other methods. In addition to the ARIMA method, combination of multiple statistical methods, BSTS, naïve method, exponential smoothing and spectral analysis techniques were applied there.

Table 2.1: Summary of previous studies related to the crop price prediction using statistical techniques.

Reference	Title	Dataset	Statistical Techniques/ Algorithms	Outputs & Conclusion
(Kamu et al., 2008)	Univariate time series model for forecasting of Tawau Cocoa Bean price	Cocoa bean prices data set from the official website of the Malaysian Cocoa Board from 1992-2006	Exponential smoothing, ARIMA, generalized autoregressive conditional heteroskedasticity (GARCH), mixed ARIMA /GARCH model	The mixed ARIMA/ GARCH model outperforms other models.
(Bogahawatta, 1988)	Seasonal variations in retail and wholesale prices of rice in Colombo markets	Weekly retail price dataset of rice from Department of Agricultural Economics, University of Peradeniya	Box Jenkins ARIMA method	Both retail and wholesale prices have seasonality variations. That is more prominent in retail than wholesale prices.
(Dieng, 2018)	Alternative forecasting techniques for vegetable prices in Senegal	Monthly average consumer prices for tomato, potato and onion from 1980-2003	naive model, the exponential smoothing, ARIMA model and spectral analysis techniques	ARIMA model appears to be the best forecaster of the prices of the three crops. The Naive model ranks second to the ARIMA model
(Jadhav et al., 2017)	Application of ARIMA model for forecasting agricultural prices	Prices of paddy, ragi and maize dataset collected from 2002-2006 from the website of Agricultural Produce Market Committee	Univariate Time Series Forecasting ARIMA Technique	ARIMA model serves as a good technique for predicting the magnitude of any variable. As a limitation, it mentioned requirements for long time series (large sample size).
(Shukla and Jharkharia, 2011)	Applicability of ARIMA models in the wholesale vegetable market	Vegetable prices data set collected from Ahmedabad wholesales market over twenty-five months	ARIMA method	This model is highly efficient in forecasting the demand for vegetables on a day-to-day basis.

(Yoo, 2016)	Vegetable price prediction using a typical web-search data	Monthly prices of three vegetables of dried red pepper, garlic, and onion are considered from 2007-2016 in the Korean wholesale market	Bayesian Structural Time Series (BSTS) Model	By introducing a typical index into the Bayesian structural time series models, prediction power for vegetable prices can be improved.

Table 2.2 depicts the previous studies related to the crop price prediction which were implemented using deep learning and neural network methods. Among them, Radial Basis Neural Network, Back Propagation Neural Network, Genetic Algorithm-Based Neural Network, hybrid methods and STL-Attention- based LSTM methods are the mainly used neural network and deep learning methods for the crop price prediction.

Table 2.2: Summary of previous studies related to the crop price prediction using deep learning and neural network method.

Reference	**Title**	**Dataset**	**Deep Learning and Neural Network Methods**	**Outputs & Conclusion**
(Subhasree and Priya, 2016)	Forecasting vegetable price using time series data	Vegetable prices of time series data were manually collected from the Ulavar market in Tamilnadu	Radial Basis Neural Network, Back Propagation Neural Network, Genetic Algorithm-Based Neural Network (GANN)	GANN model has more advantages over the other two models for vegetable price prediction.
(Zheng, et al., 2014)	A hybrid neural network and H-P filter model for short-term vegetable price forecasting	Monthly price data for five types of vegetables such as cabbages, peppers, cucumbers, green beans and tomatoes from 2012 - 2013	A hybrid approach of combining H-P filtering and neural network model, Back Propagation Neural Network	The hybrid model has the best performance in predicting the future prices for the vegetables.
(Luo et al., 2011)	Prediction of vegetable price based on neural network and genetic algorithm	The daily price of Lentinus edodes was collected from Beijing Xinfadi wholesale market from 2003- 2009	Back Propagation Neural Network (BPNN), neural network model based on the genetic algorithm, Radial Basis Neural Network (RBNN) and an integrated prediction model based on the above	The integrated prediction model is performed better than the other three models for the price prediction.

			three models	
(Nasira and Hemageetha, 2012)	Vegetable price prediction using data mining classification technique	Three years of tomato price data from the Coimbatore market from 2009 to 2011	Back Propagation Neural Network (BPNN)	BPNN is one way of predicting vegetable prices with non-linear time series.
(Yin et al., 2020)	Vegetable price forecasting using STL and attention mechanism based LSTM	Monthly prices of five crops, cabbage, radishes, onion, hot peppers, and garlic, using vegetable prices, weather information about the main production areas, and import/export data of vegetables from January 2012 to December 2019	STL-ATTLSTM (STL-Attention-based LSTM) model which integrated the seasonal trend decomposition using the Loess (STL) pre-processing method and attention mechanism based on long short-term memory (LSTM) methods	STL-ATTLSTM model achieved the best performance than the other three benchmark models.

Table 2.3 depicts the previous studies related to the crop price prediction which were implemented using machine learning techniques. Logistic regression, decision tree algorithm, XG boost, clustering algorithms, naïve bayes algorithm and the K nearest neighbors were the mainly applied ML techniques for the crop price prediction.

Table 2.3: Summary of previous studies related to the crop price prediction using machine learning technique.

Reference	**Title**	**Dataset**	**Machine Learning Techniques**	**Outputs & Conclusion**
(Mulla and Quadri, 2020)	Crop-yield and price forecasting using machine learning	Few rabi and Kharif season crops like paddy, arhar, bajra, barley data set	Decision tree algorithm	Decision tree method has produced good accuracy.
(Samuel, 2020)	Crop Prediction System using Machine learning Algorithms	Data were integrated from different data sources	Logistic Regression, Decision Trees, XG Boost, Neural Nets, and Clustering algorithms	XGBoost provided the best performance among them.

(Rachana et al., 2020)	Crop price forecasting system using supervised machine learning algorithms	Crop prices data set of 2012	Naïve Bayes algorithm, K Nearest Neighbors	The proposed solution helps the farmers to make better informed decisions and manage the price risk.
(Varun et al., 2019)	Agriculture commodity price forecasting using ML techniques	Crop prices data set of 2018	support vector machine and extended Kalman filter method	This system helps farmers to sell their crops without third party involvement. So that no loss for both customers and the farmers.
(Lavanya and Raguchander, 2013)	Price forecasting & Anomaly detection for agricultural commodities in India	Market-wise support price for crops was extracted from a website	support vector regression (SVR) based price prediction method	The proposed method achieved more accurate predictions than other traditional methods like regression techniques.

2.5 Summary

This chapter was summarized the existing research approaches used for the vegetable price prediction. Gaps and the limitations of the discussed approaches were identified and discussed. Some of them will be addressed during this research in the upcoming chapters.

CHAPTER 3

DESIGN AND METHODOLOGY

This chapter is mainly focused on the methodological aspects and the design of the proposed methodology connected with the vegetable price prediction using the univariate time series forecasting approach. The proposed approach was broken into the different sections to show more clearly about how the research was conducted. It provides the detailed description of the overall architecture including the utilized dataset, pre-processing steps, exploratory data analysis, proposed prediction models, proposed evaluation methods and finally the short-term price forecasting function used for the vegetable prices modelling task.

3.1 Systematic Approach

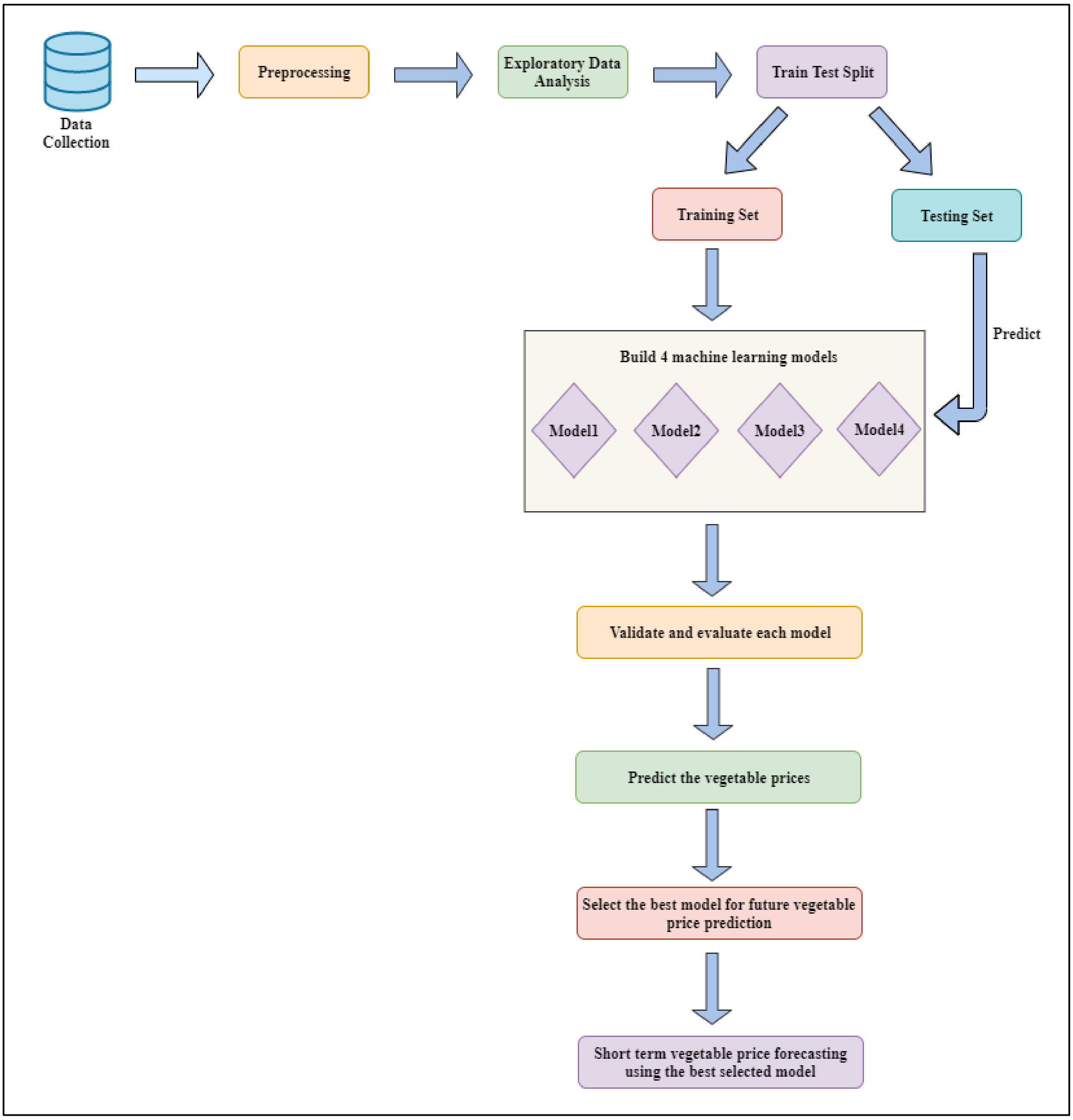
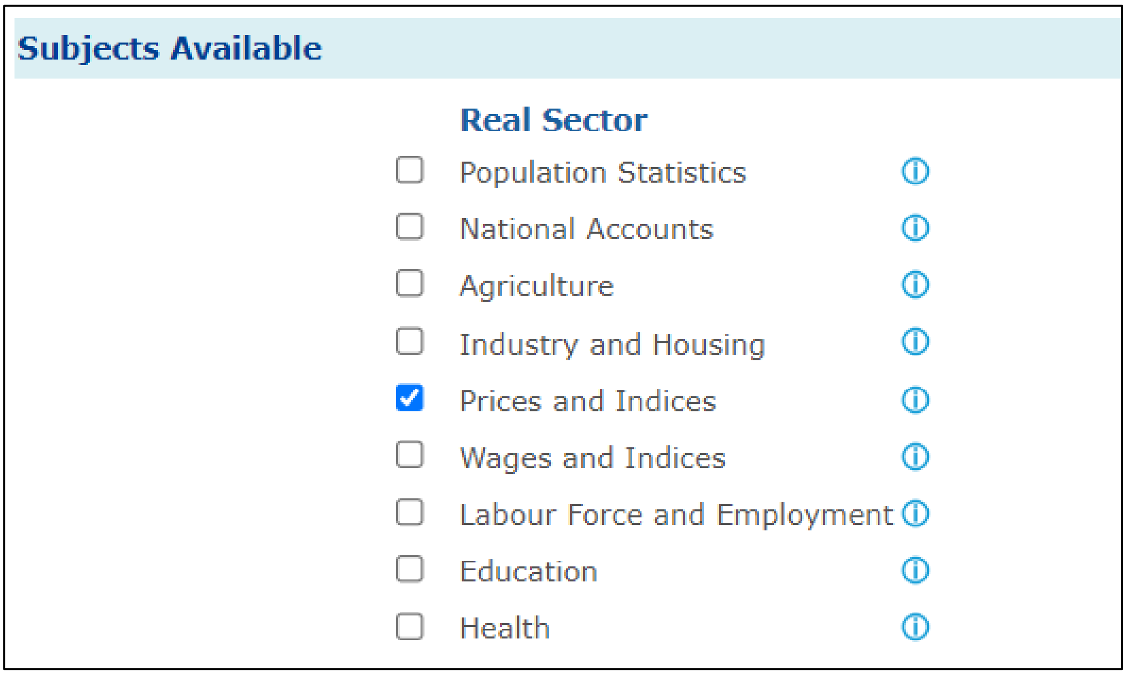

Data Collection
Preprocessing
Exploratory Data Analysis
Train Test Split
Training Set
Testing Set
Predict
Build 4 machine learning models
Model1
Model2
Model3
Model4
Validate and evaluate each model
Predict the vegetable prices
Select the best model for future vegetable price prediction
Short term vegetable price forecasting using the best selected model

Figure 3.1: Systematic Approach

The systematic approach of this research is described as shown in Figure 3.1. Each phase of this approach is discussed in more detail level in the following sections.

3.2 Data Collection

This study was used the daily price reports of vegetables published in the Central Bank of Sri Lanka from 2015-2022 (“Daily Price Report, Central Bank of Sri Lanka”, n.d.). Two data sets were collected to predict the vegetable prices before the crisis and after the crisis. The Central Bank has created the price reports by collecting daily vegetable prices. These price reports consist of daily retail prices and daily wholesale prices of a set of regular vegetables such as beans, carrots, cabbage, tomato, brinjal, pumpkin, green chili and lime for Pettah and the Dambulla markets. Among them, beans and carrot were selected for this analysis. The steps followed to collect the required data set were discussed during next stages.



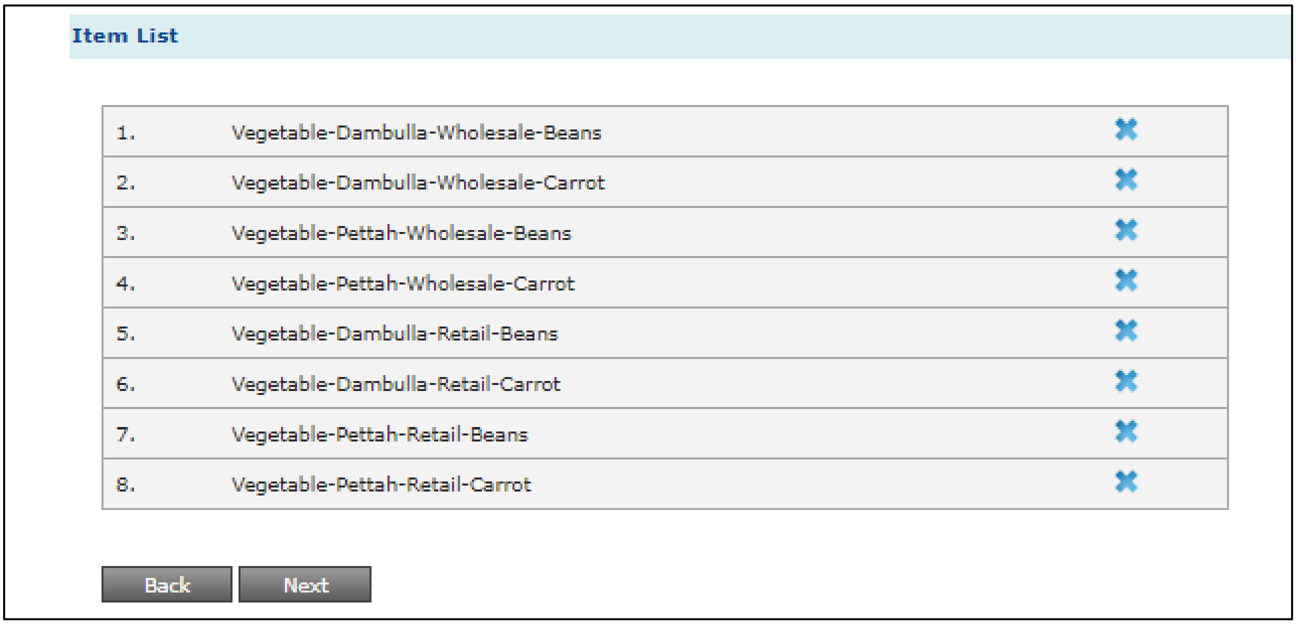
Subjects Available

Real Sector

- □ Population Statistics
- □ National Accounts
- □ Agriculture
- □ Industry and Housing
- ☑ Prices and Indices
- □ Wages and Indices
- □ Labour Force and Employment
- □ Education
- □ Health

Figure 3.2: Select the data subject

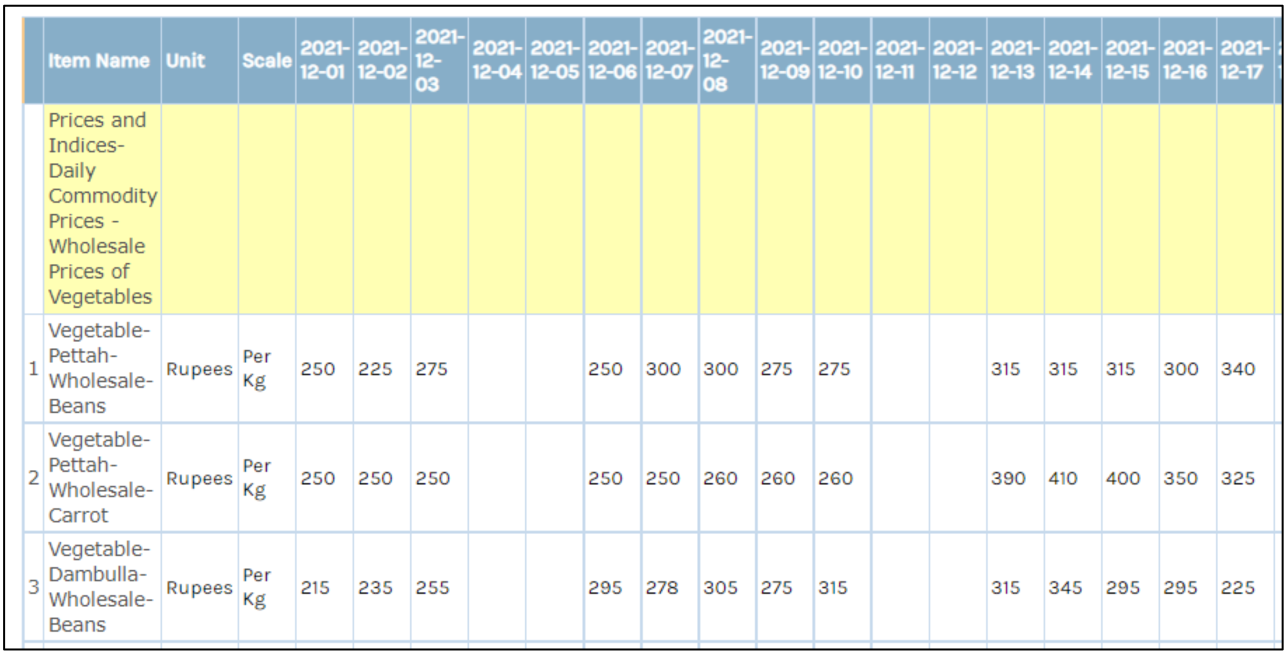
Initial step was to login to the data library of Central Bank of Sri Lanka and select the relevant sector where we want to collect the data. Since we were looking for the vegetable prices data set, ‘Prices and Indices’ sector was selected as the data subject as shown in Figure 3.2. Where we must specify the data frequency and the time period (monthly, daily like that) before moving to the next step.


Item List

1.	Vegetable-Dambulla-Wholesale-Beans
2.	Vegetable-Dambulla-Wholesale-Carrot
3.	Vegetable-Pettah-Wholesale-Beans
4.	Vegetable-Pettah-Wholesale-Carrot
5.	Vegetable-Dambulla-Retail-Beans
6.	Vegetable-Dambulla-Retail-Carrot
7.	Vegetable-Pettah-Retail-Beans
8.	Vegetable-Pettah-Retail-Carrot

Figure 3.3: Select the vegetable list

As the second step, wholesale and retail prices of set of vegetables were selected as the data set as shown in Figure 3.3. The list of selected vegetables was included Pettah wholesale carrot, Pettah wholesale beans, Pettah retail carrot, Pettah retail beans, Dambulla wholesale carrot, Dambulla wholesale beans, Dambulla retail carrot and Dambulla retail beans. Those selected vegetables were utilized as the data set for this study.



	Item Name	Unit	Scale	2021-12-01	2021-12-02	2021-12-03	2021-12-04	2021-12-05	2021-12-06	2021-12-07	2021-12-08	2021-12-09	2021-12-10	2021-12-11	2021-12-12	2021-12-13	2021-12-14	2021-12-15	2021-12-16	2021-12-17
	Prices and Indices-Daily Commodity Prices - Wholesale Prices of Vegetables																			
1	Vegetable-Pettah-Wholesale-Beans	Rupees	Per Kg	250	225	275			250	300	300	275	275			315	315	315	300	340
2	Vegetable-Pettah-Wholesale-Carrot	Rupees	Per Kg	250	250	250			250	250	260	260	260			390	410	400	350	325
3	Vegetable-Dambulla-Wholesale-Beans	Rupees	Per Kg	215	235	255			295	278	305	275	315			315	345	295	295	225

Figure 3.4: Selected data set

As the last step, daily vegetable price report was generated as shown in Figure 3.4 and which can be downloaded using different versions like xlsx, pdf etc. This price report consists of set of retail and wholesale vegetable prices for Pettah and Dambulla market with the relevant dates for the requested time period.

3.2.1 Data Set Description

The detailed description about the collected data sets can be explained as given in the Table 3.1.

Table 3.1: Data Set Description.

Data Set	Period (dd/mm/yyyy)	No of Records	No of Columns and the description
Vegetable prices data before the crisis	06/03/2015 – 30/11/2021 (Daily Data)	1612	29 Contains dates, daily prices (per kg) of different vegetables in different markets
Vegetable prices data after the crisis	01/04/2022 – 07/10/2022 (Daily Data)	126	29 Contains dates, daily prices (per kg) of different vegetables in different markets

3.3 Data Pre-processing

Data pre-processing is a data mining technique, which is used to transform the raw data into a useful information. Data cleansing is the main stage performed under the data pre-processing stage which ensures error-free data and eliminates unnecessary data. This step includes handling missing values, removing duplicate records and removing garbage values. In this study, missing values were handled properly by using linear interpolation and data was enriched by renaming the column names as the pre-processing steps.

3.3.1 Liner Interpolation

Handling missing values properly, is very important during the pre-processing stages for time series analysis. Linear interpolation method was used to deal with missing data points during this analysis. This linear interpolation method estimates the missing values by assuming the linear relationship within a range of data points. It uses non null values available to compute the missing data points by checking the past and future data points from the missing value.

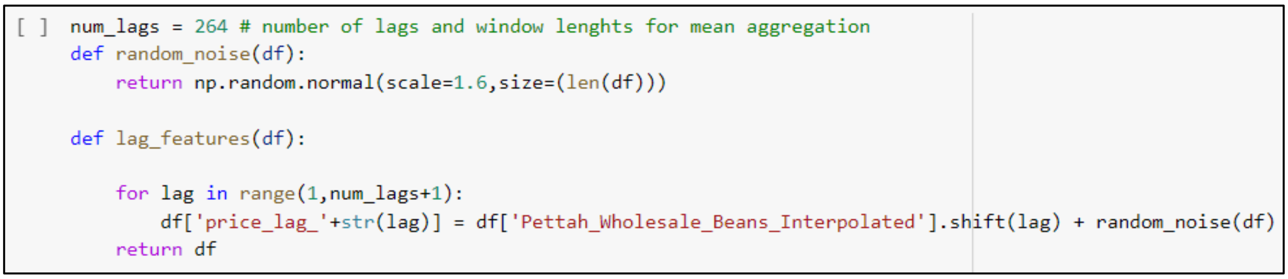
3.4 Exploratory Data Analysis (EDA)

Before training the models from the collected dataset, an exploratory data analysis was conducted to get more insights about the available dataset. Here different visualization techniques were carried out on the dataset to investigate and summarize the main characteristics of the dataset to discover patterns. Since this study was conducted the time series forecasting; time plot, decomposing time series plot, rolling window, ACF and PACF plot were drawn during the initial steps.

EDA is an approach used to understand the data set by analysing and investigating the data set to summarize its main characteristics by using visual elements. This helps to identify the trends and patterns in the data more accurately and perform the analysis effectively. It is a good practice to use EDA to understand the data set first and throughout that, we can gather many insights from the data.

3.5 Create Lag Features for Univariate Time Series Forecasting

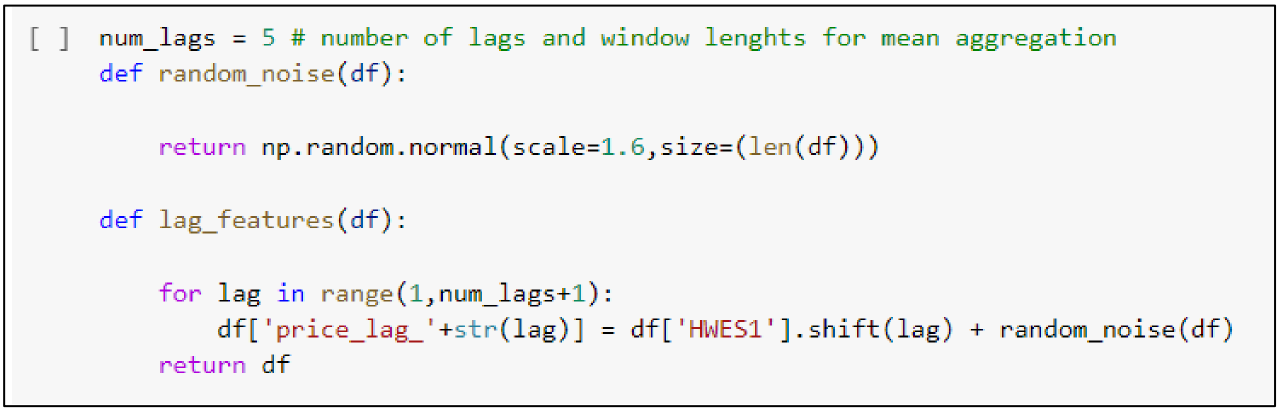
Creating lag features is the most common way of transforming the time series problem into a supervised learning problem. Since univariate time series forecasting approach was conducted during this analysis, to predict the next day vegetable price($P_{(t+1)}$), prices of previous days (P_t, $P_{(t-1)}$, $P_{(t-2)}$ …) were taken into the consideration.



```
[ ] num_lags = 264 # number of lags and window lenghts for mean aggregation
    def random_noise(df):
        return np.random.normal(scale=1.6,size=(len(df)))

    def lag_features(df):

        for lag in range(1,num_lags+1):
            df['price_lag_'+str(lag)] = df['Pettah_Wholesale_Beans_Interpolated'].shift(lag) + random_noise(df)
        return df
```

Figure 3.5: Create lag features and add noise for the prices before the crisis

To predict the vegetable prices before the crisis, number of lags of 264 (22*12) was considered by assuming that each month has approximately 22 prices (business days only) which can be shown in Figure 3.5. Where random noise was added when creating the lag features because, lag features really make sense for statistical time series models, but here machine learning models were utilized. When generating the lag features from the price variable for machine learning models, which causing a problem called data leakage. The reason for the data leakage problem is that generally, features are not generated using target variable in machine learning problems. Because it leads to overfitting to the training data. To avoid such overfitting situation, in this study, random gaussian noise was added to the lag features.



```
[ ] num_lags = 5 # number of lags and window lenghts for mean aggregation
    def random_noise(df):

        return np.random.normal(scale=1.6,size=(len(df)))

    def lag_features(df):

        for lag in range(1,num_lags+1):
            df['price_lag_'+str(lag)] = df['HWES1'].shift(lag) + random_noise(df)
        return df
```

Figure 3.6: Create lag features and add noise for the prices after the crisis

To predict the vegetable prices after the crisis, number of lags of 5 (one week with business days only) was considered since the data set was contained only 126 data points after the crisis as shown in Figure 3.6. For this scenario also, random noise was added when creating the lag features in order to avoid overfitting problem.

3.6 Train Test Split

Train test split is a technique used for evaluating the performance of machine learning models. Where pre-processed data set was split into training and testing sets based on the predefined percentages. Generally, we put 80% of the data in the training set and 20% of the data in the testing set. The training set was used to train the model and the testing set was used to validate the model. In this study, 1st 80% of the data points were used as the training set and the rest of the 20% of the data points were used as the testing set.

3.7 Building Forecasting Models

Different kinds of machine learning models were proposed for vegetable price prediction. Four machine learning time series foresting models were selected to predict the vegetable prices. For the carrot prices data set after the crisis, machine learning models did not provide accurate predictions due to that data set not fitting to the ML models. Hence ARIMA model also utilized for the carrot price prediction after the crisis. Each of the proposed models is further explained in upcoming sections.

3.7.1 Gradient Boosting Regression

Gradient boosting is one of the ensemble learning methods. Its prediction model is formed by creating multiple weak models and combining them together. Gradient boosting can be used for both regression and classification problems. When the target column is continuous, we use the gradient boosting regressor and otherwise we use the gradient boosting classifier. This study was utilized the gradient boosting regression as one of the forecasting models.

The gradient boosting algorithm involves three components.

- Loss Function

The loss function needs to be optimized. It may differ based on the type of problem we are going to solve. For example, regression uses square error and classification uses the logarithmic loss as loss functions.

- Week Learner

As week learners in gradient boosting, decision trees are used.

- Additive Model

Existing trees in the model are remaining the same. Trees are added to the model one by one. When adding more trees to the model, the gradient descent function will minimize the losses.

3.7.1.1 Advantages and disadvantages of gradient boosting

Advantages:

- Provide good predictive accuracy compared to other models
- Flexibility
- Can work with both categorical and numerical values as it is
- Handle missing values natively

Disadvantages:

- High computation cost. Takes much time to train the models
- Cause for overfitting and outliers
- Hard to interpret the final models

3.7.2 XG Boost Regression

XG boost stands for extreme gradient boosting, which is an open-source library that provides an efficient implementation of the gradient boosting algorithm. It is powerful for building regression models. It is a class of ensemble machine learning algorithm which can be used for the predictive modelling. Model is fit using a loss function and gradient descent optimization techniques. XG boost is performed well over other models due to execution speed and the model performance.

3.7.2.1 Advantages and disadvantages of XG boost regression

Advantages:

- It has a good predictive ability
- Effective method to handle large data sets
- Supports for regularization

Disadvantages:

- leads to overfitting when there are noisy data
- Does not work well with sparse data and unstructured data
- Kind of black box model

3.7.3 Random Forest Regression

Random forest is one of the ensemble learning methods used for both classification and regression problems. Based on the different sample sizes, it builds decision trees and finally takes the average vote for regression and the majority vote for classification. When the target column is continuous, we use the random forest regressor and when it is categorical, we use the random forest classifier. This study was utilized the random forest regression as one of the forecasting models.

Random forest algorithm steps:

- K number of records are selected from the total n number of records in the data set.
- For each sample separate decision trees are constructed.
- Out is generated using each decision tree.
- For classification problems final output is considered based on majority vote and for regression problems, the output is performed based on average votes.

3.7.3.1 Advantages and disadvantages of random forest

Advantages:

- Achieve better accuracy by reducing the overfitting in decision trees
- Highly stable since many trees are involved
- Can handle missing values automatically
- Feature scaling is not needed

Disadvantages:

- Complexity since involving a large no of trees
- Requires much time to train the model
- Relatively expensive

3.7.4 Stacking Regression

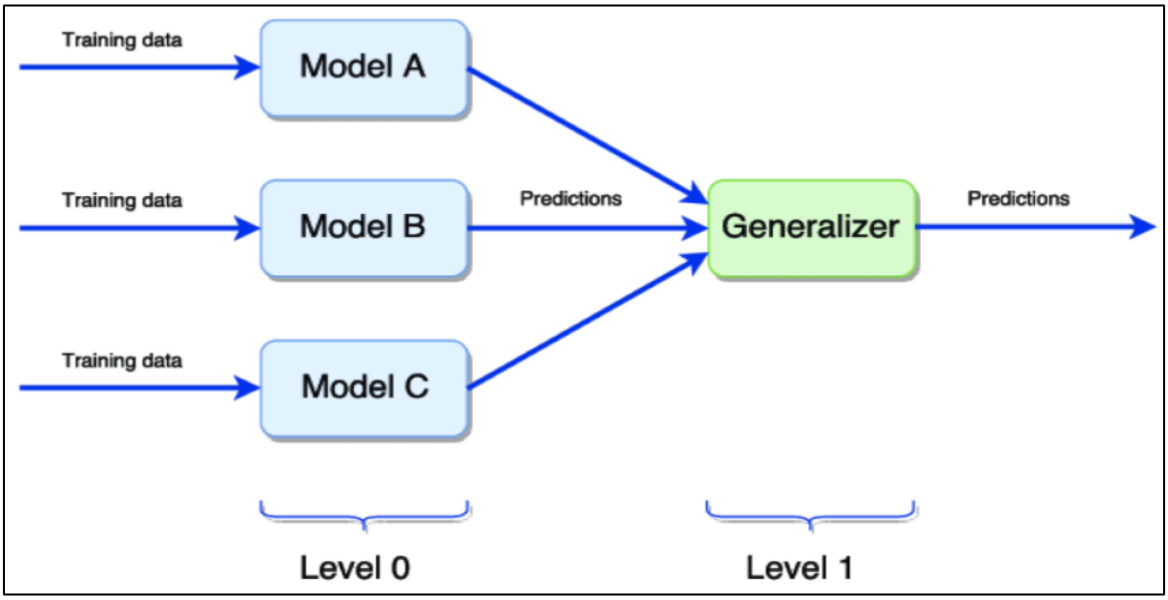


Figure 3.7: Stacking Regressor

Stacking regression is an ensemble learning technique which is used to combine multiple regression models via a generalizer as shown in Figure 3.7. Each individual model is trained based on the entire training data set and then the outputs of them applied to the generalizer to generate the final output. In this study, Random Forest, XG Boost, and Gradient Boost regression algorithms were applied as the individual models of the stacking regressor. Stacking regression model was applied as the one of the forecasting models for this analysis.

Level 0 - Different models are trained using the same training data set and then make predictions
Level 1 - Get the final output by generalizing the predictions made on each model.

3.7.4.1 Advantages and disadvantages of stacking regression

Advantage:

- Helps to produce a better performing model rather than individual models

Disadvantages:

- Less robust than a single model
- Time consuming to train the data

3.7.5 Auto-Regressive Moving Average Model (ARIMA)

ARIMA is the most common traditional statistical technique used for time series forecasting. This model uses the time series data and numerical data to clarify the data and predict the future values. It has 3 components.

1. Auto-Regression (AR)

   - Future values of Y are dependent of previous lagged values of Y.
   - Regression of y_t depends on $y(_{t-1})$, $y(_{t-2})$, … and etc
   - p = order of AR; current value of y is dependent on how many previous lagged values of current Y. if p=2 that means y_t is dependent on $y_{(t-1)}$ and $y_{(t-2).}$
   - p can be decided from the Partial Auto Correlation Function (PACF) or using auto ARIMA.

2. Moving Average (MA)

   - Future values of Y are dependent of previous lagged values of white noise ie the irregular component. White noise is just the error. Error is the difference between the actual value and the predicted value. Here error also taken into the consideration to predict the future value.
   - Autocorrelation between the errors.
   - The irregular component is captured in MA.
   - q is the order of MA.
   - q can be decided from the Auto Correlation Function (ACF) or using auto ARIMA.

3. Integrated (I)

   - Integrated implies the data is static. Differencing step (d) is involved to generate stationary time series data by removing the seasonal and trend components.

3.7.5.1 Advantages and disadvantages of ARIMA

Advantages:

- To perform the forecast, only the prior data of the time series is required.
- Provide better performance for short term forecasts.
- It models the non-stationary time series data.

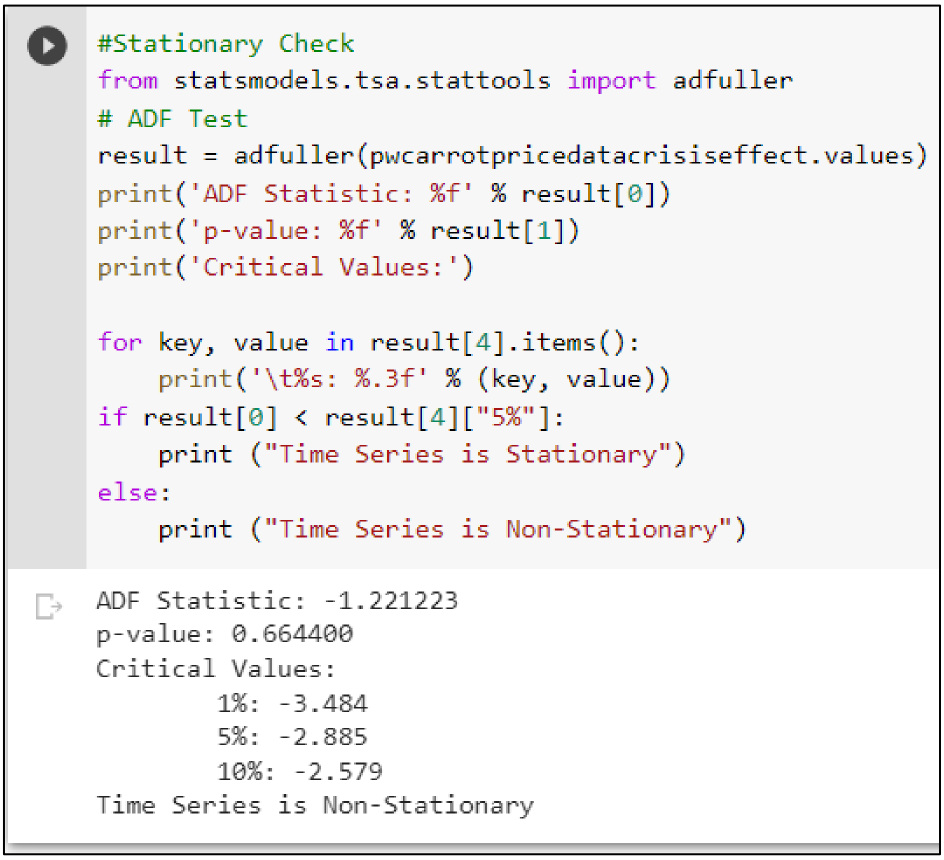
Disadvantages:

- Poor performance for long term forecasts.
- Not suitable for seasonal time series analysis.
- Computationally expensive
- Tricky to determine the order (p, q, d) of the model.

In this study, ARIMA model was applied for the carrot price data set after the crisis, since machine learning models did not perform well for that data. ACF, PACF plots and auto ARIMA function were used to select best possible p, q, d parameters for the ARIMA model.

3.7.5.2 Augmented Dickey Fuller test (ADF Test)

The Augmented Dickey-Fuller test (ADF test) is a typical measurable test used to test if a given Time series is stationary or not. It is kind of a statistical significance test which provide results with null and alternative hypothesis. It will result the p value, using that can decide whether time series is stationary or not.



```
#Stationary Check
from statsmodels.tsa.stattools import adfuller
# ADF Test
result = adfuller(pwcarrotpricedatacrisiseffect.values)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')

for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))
if result[0] < result[4]["5%"]:
    print ("Time Series is Stationary")
else:
    print ("Time Series is Non-Stationary")
```

```
ADF Statistic: -1.221223
p-value: 0.664400
Critical Values:
        1%: -3.484
        5%: -2.885
        10%: -2.579
Time Series is Non-Stationary
```

Figure 3.8: ADF test to check stationary

In this study, ADF test was conducted to check whether the vegetable price data is stationary or not before applying to the ARIMA model as shown in Figure 3.8. Where Pettah wholesale carrot prices data set after the crisis was applied for the ADF test and obtained results concluded that prices data is nonstationary.

3.8 Validate and evaluate each model

3.8.1 Model Validation

The ultimate target of any machine learning model is to achieve its intended goal. Under model validation, the trained model is validated with the testing data set. This testing data set can be a sample of the same data set that is used for training which is referred to as ‘in sample validation’. If the testing set is not a sample from the same data set as used for training that is referred to as ‘out of sample validation’. The model should perform well in both cases. The trained model should have the generalization capability where it should perform well for the real data which has never been seen before. In this study hold-out cross validation method was conducted to validate the selected machine learning models.

3.8.1.1 Hold-out Cross Validation

The cross-validation is a resampling approach that divides the entire dataset into two sections as training set and the testing set. The training data set is used to create the model, while the unseen testing data set is used for the prediction purposes. Typical ratios used for the split data set include 80% for training data and 20% for the testing data. Using the hold-out cross validation, we can reduce the model from overfitting the training data set. Throughout that, the model achieves good generalization power.

3.8.2 Model Evaluation

Once the models are fitted, we must assess their performance. There are several performance evaluation matrices available to evaluate the machine learning regression problems. In this study, statistical-based evaluation methods were used such as mean squared error, root mean squared error, coefficient of determination, mean absolute error and mean absolute percentage error which can be applied to assess the goodness of the ML regression models.

3.8.2.1 Mean Squared Error (MSE)

Mean square error is calculated by taking the average of the square of the difference between the actual values and the predicted values of the data. Lower MSE indicates the best fit for the model. This MSE is one of the good indicators of how correctly the model predicts the response. The formula used to calculate MSE can be expressed below.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\text{actual values} - \text{ predicted values})^2$$

Where n = number of data points.

3.8.2.2 Root Mean Squared Error (RMSE)

RMSE is calculated as the square root of mean square error and is an extension of MSE. Lower RMSE indicates the best fit for the model. The formula used to calculate RMSE can be expressed below.

$$RMSE = \sqrt{MSE}$$

3.8.2.3 Mean Absolute Error (MAE)

MAE is calculated by taking the average of the absolute difference between forecasted and actual values. The smaller the MAE value better the forecast is. The formula to calculate MAE can be expressed as below.

$$MAE = \sum_{t=1}^{n} |\frac{A_t - F_t}{n}|$$

Where A_t = actual value, F_t = forecast value and n = number of fitted points respectively.

3.8.2.4 Mean Absolute Percentage Error (MAPE)

MAPE is a measure of how accurate the predicted model is. This measures the accuracy of the model as a percentage. Error is defined as the actual value minus the predicted value. This measure is easy to understand since the error is shown in terms of percentages. The smaller the MAPE value better the forecast is. The formula to calculate MAPE can be expressed below.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} |\frac{A_t - F_t}{A_t}|$$

Where A_t = actual value, F_t = forecast value and n = number of fitted points respectively.

3.8.2.5 Coefficient of Determination (R^2)

This metric is also indicated how well the model fits a given data set. R^2 value lies between 0 and 1, where 0 indicates that the model does not fit the given data set well and 1 indicates that the model fits well for the given data set. Therefore, if the R^2 value is close to 1 means the best fit to the model. The formula to calculate R^2 can be expressed below.

$$R^2 = 1 - \frac{RSS}{TSS}$$

Where RSS = Residual Sum of Squares and TSS = Total Sum of Squares respectively.

$$TSS = \sum_{i=1}^{n}(y_i - \overline{y})^2$$

Where n = number of observations, y_i = value in a sample and $\overline{y}$ = mean value of a sample.

$$RSS = \sum_{i=1}^{n}(y_i - f(x_i))^2$$

Where n = upper limit of summation, y_i = i^{th} value of the variable to be predicted, $f(x_i)$ = predicted value of y_i

3.8.2.6 Public Data Set

Same vegetable prices data set (as used for the research), was collected from 1st of December 2021 – 7th of December 2021 before the economic crisis and 10th of October 2022 – 14th of October 2022 after the economic crisis from the Central Bank. They were used as the new data set to evaluate the best forecasting model.

3.9 Select the best model

For each model, the above discussed performance metrics were calculated. Based on their values (with the highest accuracy and the lowest errors), the best model for the vegetable price prediction is selected. Based on the best model, wholesale and retail prices are forecasted for the selected vegetables.

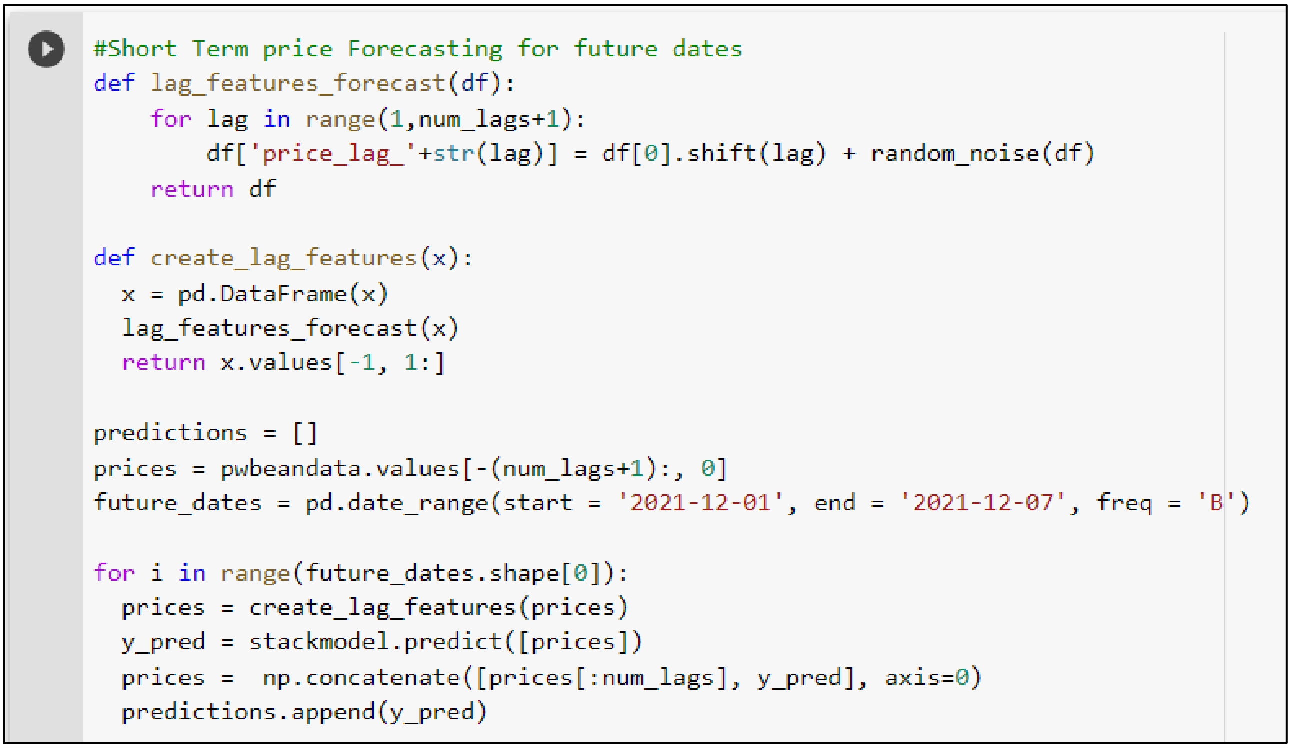
3.10 Hyper parameter tuning for the best model using Hyperopt

Hyper parameter tuning is the process of selecting the best optimal parameter set for the machine learning model. Hyperopt is an open-source python library which uses a Bayesian approach to find the best optimal values for the hyperparameters. Hyperopt method incorporates past information during the search process therefore which is more efficient than the grid search and the random search methods since they did not incorporate the past results during the search process. Hyperopt provides more advantage over grid search and random search methods because, they are more exhaustive and time consuming but hyperopt do not have such issues.

Hyperopt needs 5 major components to optimize the parameters such as search space, loss function, optimization algorithm, score and the configuration. In this study, hyperopt method was utilized to tune the parameters of the best model for each vegetable price.

3.11 Short term price forecasting using the best selected model

In this study, short term price forecasting refers to the forecasting prices of vegetables in future days only for the short time period by using its historical prices.



```
#Short Term price Forecasting for future dates
def lag_features_forecast(df):
    for lag in range(1,num_lags+1):
        df['price_lag_'+str(lag)] = df[0].shift(lag) + random_noise(df)
    return df

def create_lag_features(x):
  x = pd.DataFrame(x)
  lag_features_forecast(x)
  return x.values[-1, 1:]

predictions = []
prices = pwbeandata.values[-(num_lags+1):, 0]
future_dates = pd.date_range(start = '2021-12-01', end = '2021-12-07', freq = 'B')

for i in range(future_dates.shape[0]):
  prices = create_lag_features(prices)
  y_pred = stackmodel.predict([prices])
  prices =  np.concatenate([prices[:num_lags], y_pred], axis=0)
  predictions.append(y_pred)
```

Figure 3.9: Short term price forecasting

Here, last sample of the testing data set was used to create the lag variables in order to forecast the future prices. From that, prices were forecasted for the next 5 days as shown in Figure 3.9.

3.12 Summary

This chapter describes the proposed design and the methodology to perform this research work and it was explained the steps followed during each stage using a systematic approach. Next chapter will focus on the results generated from this proposed approach.

CHAPTER 4

RESULTS AND EVALUATION

This chapter presents the results obtained for the proposed approaches including the performance for individual machine learning models. Based on the present results, best optimal machine learning models to predict the vegetable prices were selected. Overall, there were 16 prices were predicted and forecasted such as Pettah wholesale bean, Pettah wholesale carrot, Pettah retail bean, Pettah retail carrot, Dambulla wholesale bean, Dambulla wholesale carrot, Dambulla retail bean and Dambulla retail carrot before and after crisis. End to end process of the results were discussed only for 4 prices out of them such as Pettah wholesale bean before the crisis, Pettah wholesale bean after the crisis, Pettah wholesale carrot before the crisis and Pettah wholesale carrot after the crisis. For the rest of the prices, same process was followed and only the prediction results and the forecasted results were discussed.

4.1 Pettah Wholesale Bean Prices Results

Results obtained for the Pettah wholesale bean prices before the crisis and after the crisis were explained in the following sections. Exploratory data analysis was conducted at initial steps and results were interpreted. Then proposed machine learning techniques were applied and their results were discussed. Then results were evaluated using evaluation methods and selected the best model for wholesale bean price forecasting. Finally, short term price forecasting was done using the selected best model.

4.1.1 Time Plot

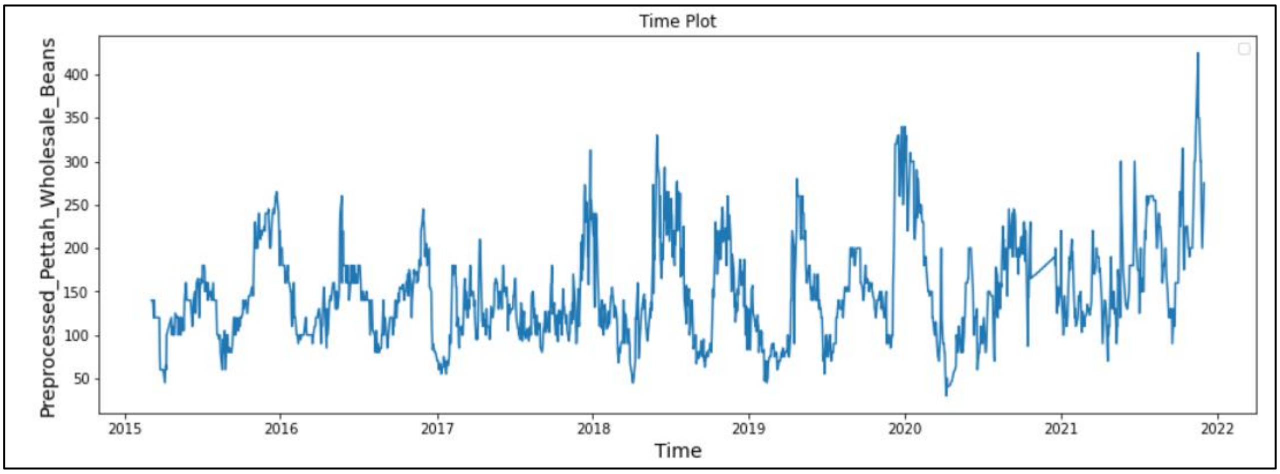


Figure 4.1: Time Plot

At initial stages, time series plot for Pettah wholesale beans was developed. Based on the time plot as shown in Figure 4.1, It can be concluded that, seasonality presents but overall, there is no trend. During the year end of each year, there is an upward trend for the wholesale bean prices.

4.1.2 Decomposing vegetable price data

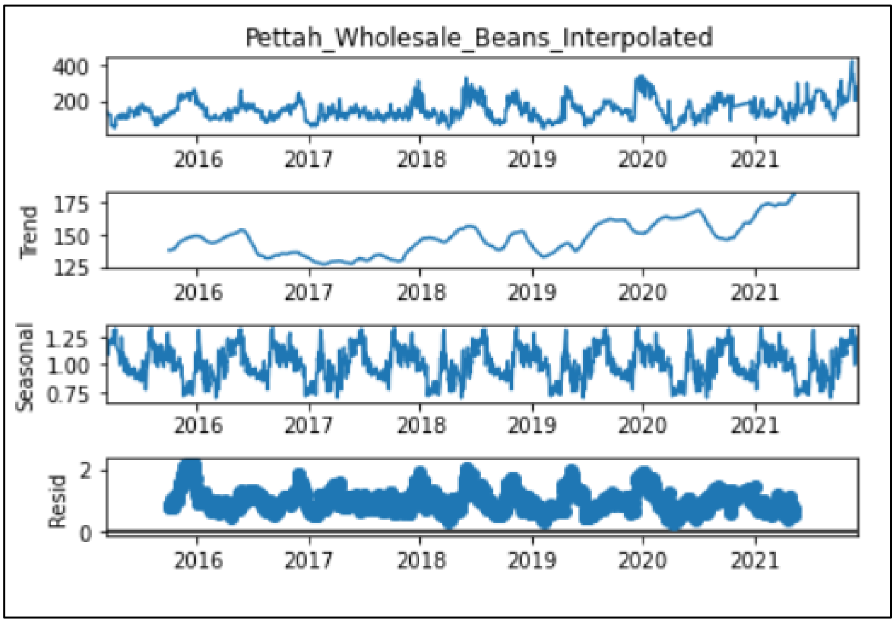


Figure 4.2: Decomposing the time series

Based on the Figure 4.2, there is no clear trend for prices, seasonality was presented, and outliers were there due to variability in macroeconomics. Please note that macroeconomics effect was not addressed due to applying univariate timeseries forecasting approach for the bean price prediction.

4.1.3 Rolling Window

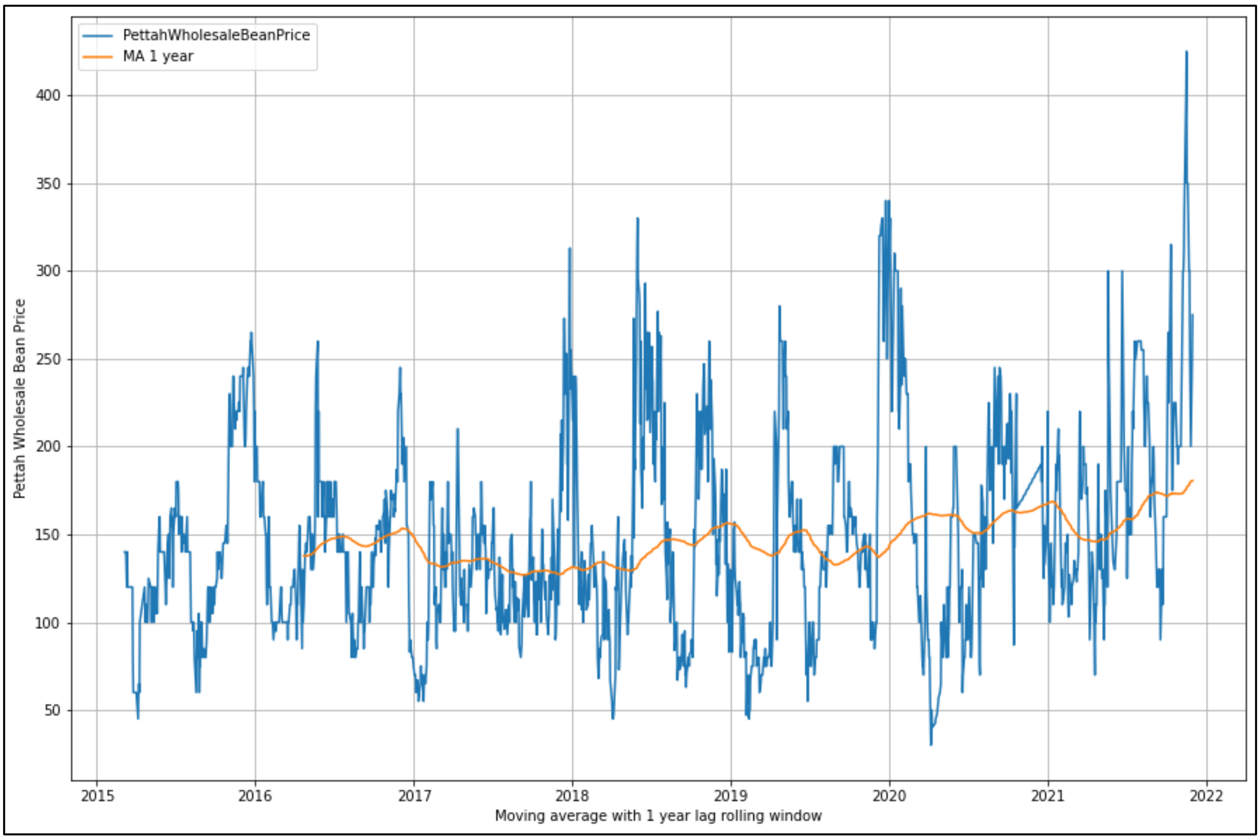


Figure 4.3: Rolling Window

Figure 4.3 depicts the rolling window with moving average of 264 days lag for Pettah wholesale bean price.

4.1.4 Autocorrelation and partial autocorrelation plot

The conduct of the ACF and PACF was inspected to get valuable insights into the behaviour of time series data.

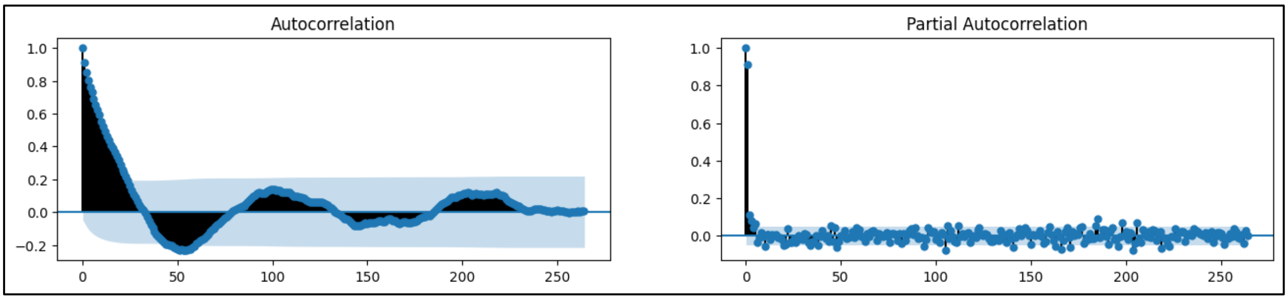


Figure 4.4: ACF and PACF Plot

Based on the ACF plot we can identify a pattern in the bean price data as shown in Figure 4.4. The ACF plot has multiple significant corelations, but higher spikes present at lag 0 and lag 1. PACF has 2 significant correlations at lag 0, and 1.

4.1.5 Training and Testing data sets plot

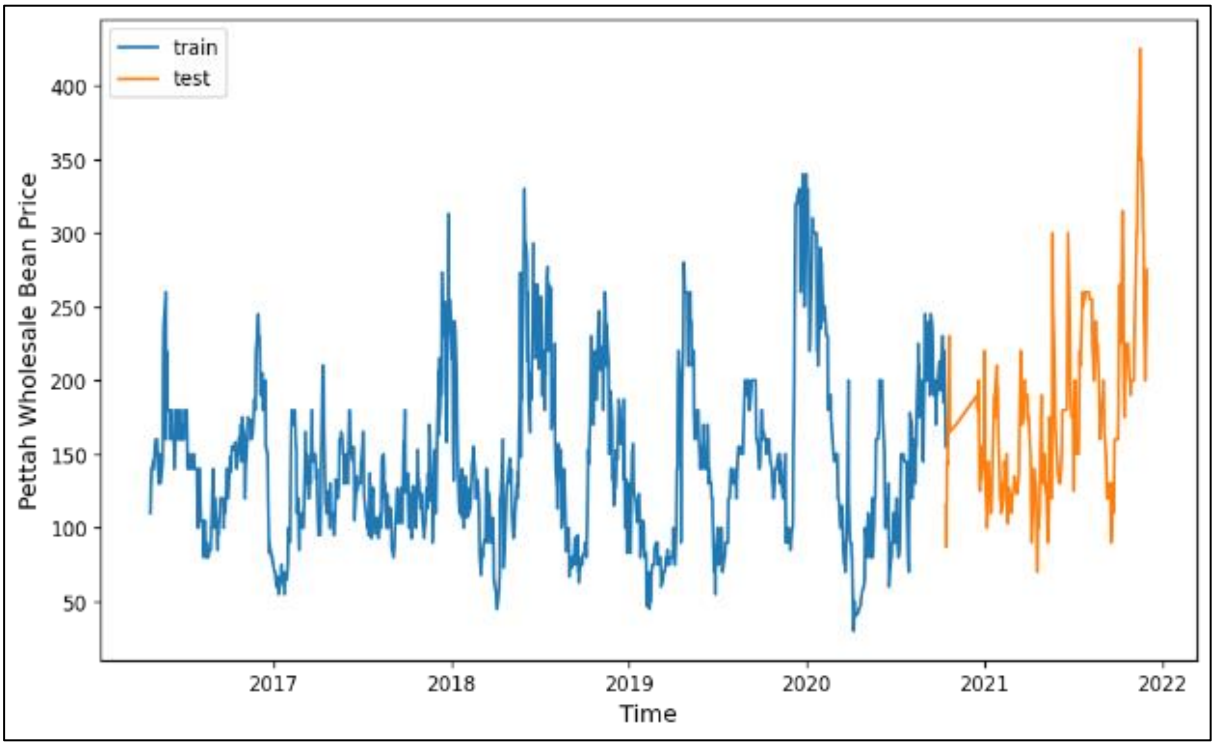


Figure 4.5: Train Set and Test Set Distribution

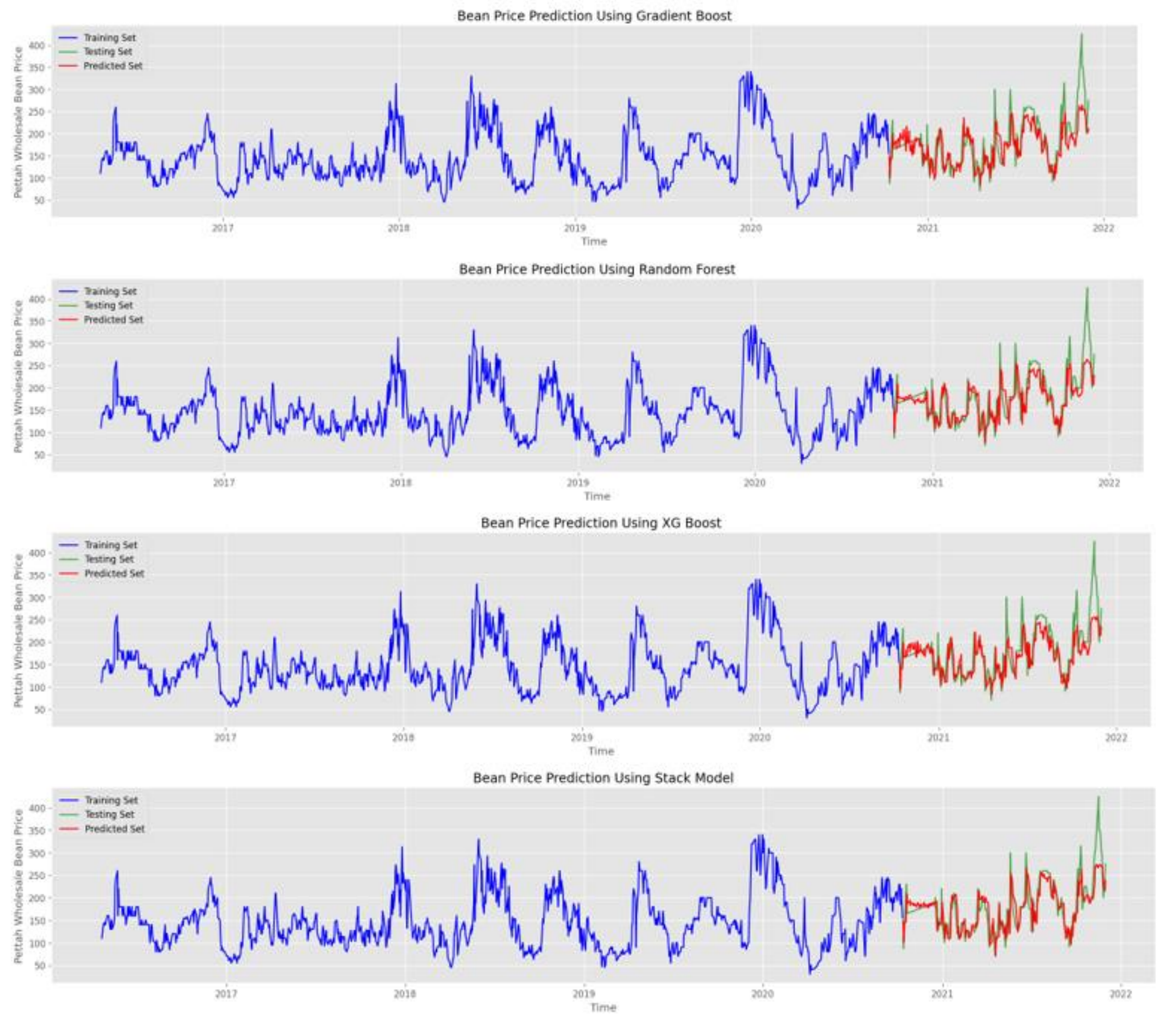
Figure 4.5 depicts the training and testing data sets distribution of Pettah wholesale bean prices data. First 80% of the entire dataset was used as the training set and rest of the 20% of the data set was used as the testing set.

4.1.6 Predicted prices of each machine learning model

Gradient Boost, Random Forest, XG Boost and Stack Regressor models were applied for the collected vegetable prices data set from 2015-2021 before the economic crisis and prices were predicted for the testing data set and subset of the results are shown in the Table 4.1 as below. It shows the predicted prices using XG boost, gradient boost, random forest and stack model with the actual prices. From that, how the predicted prices deviate from the original prices can be identified.

Table 4.1: Predicted prices using each of the model.

Date (mm/dd/yyyy)	Actual values in the testing data set	XGB Predictions	GB Predictions	RF Predictions	Stack Model Predictions
11/01/2021	200	188.78	186.48	185.88	191.99
11/02/2021	200	184.18	180.57	185.01	187.08
11/03/2021	200	184.40	177.94	188.15	187.50
11/05/2021	240	180.95	180.54	190.42	193.37
11/08/2021	300	217.11	219.95	235.82	238.07
11/09/2021	300	255.12	264.40	265.85	284.90
11/10/2021	310	259.33	261.50	265.40	276.58
11/11/2021	335	269.91	270.42	270.92	278.52
11/12/2021	350	267.45	272.69	266.49	287.61
11/15/2021	400	266.31	265.08	261.76	276.71


Bean Price Prediction Using Gradient Boost

Bean Price Prediction Using Random Forest

Bean Price Prediction Using XG Boost

Bean Price Prediction Using Stack Model

Figure 4.6: Graphical Representation of training set, testing set and the predicted set

Graphical representation of the training prices, testing prices and the predicted prices of each model can be shown using blue, green and red colours respectively as shown in Figure 4.6. All 4 models were predicted the prices for the same testing data set. Overall, there are no huge fluctuations between the testing data and predicted data but in some cases prediction results were overestimated the actual data and some cases prediction results were underestimated the actual data. Especially, prediction results were more biased for overestimating the last actual data points. But overall performance was good for all models.

4.1.7 Evaluating the models

To forecast the future wholesale bean prices in Sri Lanka, performance of each model was evaluated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Coefficient of Determination (R^2) metrics and their values can be shown as in Table 4.2. Based on the values presented in the table, it can be concluded that stack model can be selected as the best model for the wholesale beans price prediction.

Table 4.2: MSE, RMSE, MAE, MAPE and R^2 values of each model.

Model	MSE	RMSE	MAE	MAPE (%)	R^2 (%)
XGB	1,086.55	32.96	22.64	12.28	65.68
GB	1,134.76	33.68	23.76	12.96	64.16
Random Forest	1,033.17	32.14	21.25	11.66	67.37
Stack Model	993.19	31.51	22.11	12.49	68.63

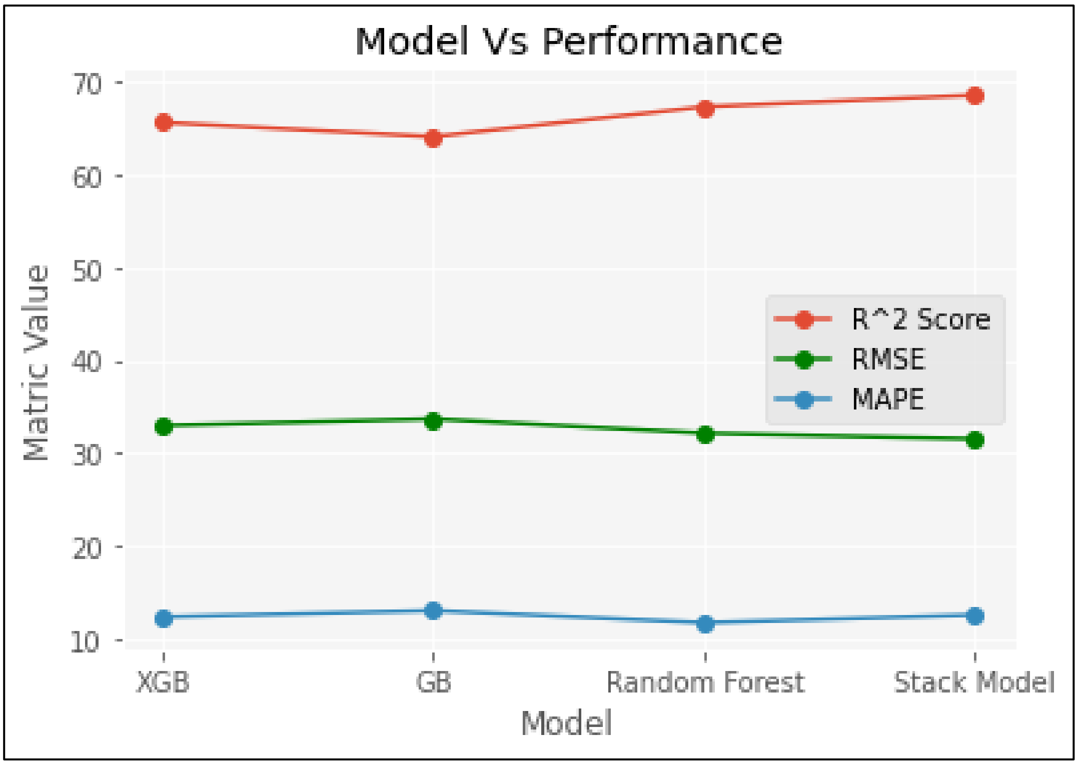


Figure 4.7: Model Vs Performance

Based on the values of the performance matrices as shown in Figure 4.7, stack model has the higher R^2 value as 68.63%, lower RMSE as 31.51 and MAPE as 12.49% over the other models. Therefore, it can be concluded that, the stacking regressor performs better than the other models. Therefore, the stacking regressor can be selected as the best model to predict the wholesale prices of beans in the Pettah market.

4.1.8 Short term price forecasting for Pettah wholesale bean price

Stacking regressor was suggested as the best model for the Pettah wholesale bean price prediction. Using the stacking regressor, prices were forecasted for the next 5 days. Since the actual data also currently available for the forecasting period, actual data was collected for the forecasting period as well. Both actual and forecasted prices for Pettah wholesale beans can be shown as in Table 4.3.

Table 4.3 Actual Prices Vs Forecasted Prices.

Date (mm/dd/yyyy)	Actual Data	Forecasted Prices
12/01/2021	250	231.12
12/02/2021	225	214.79
12/03/2021	275	229.22
12/06/2021	250	215.26
12/07/2021	300	230.39

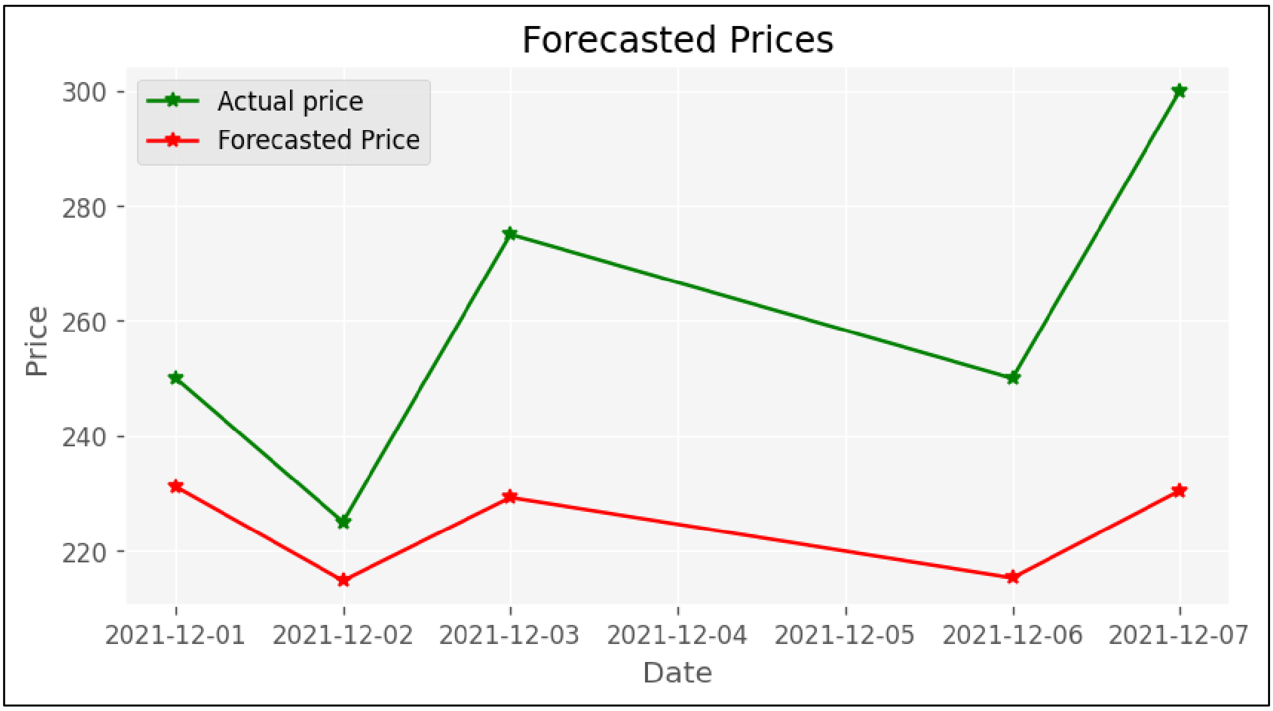


Figure 4.8: Actual Prices Vs Forecasted Prices

Figure 4.8 depicts the graphical representation of the forecasted prices and the actual prices as shown above. Stack regressor was used to forecast the wholesale bean prices for next 5 business days, but they were slightly underestimated the actual prices.

4.1.9 Additional Step – Predicting wholesale bean prices after economic crisis in Sri Lanka

During the inflation period, bean prices raised unevenly, and that effect could not be addressed from the above scenario. Therefore, bean prices during that time were predicted separately by collecting the bean price reports after crisis from the period of April of 2022 – October of 2022.

4.1.9.1 Time Plot

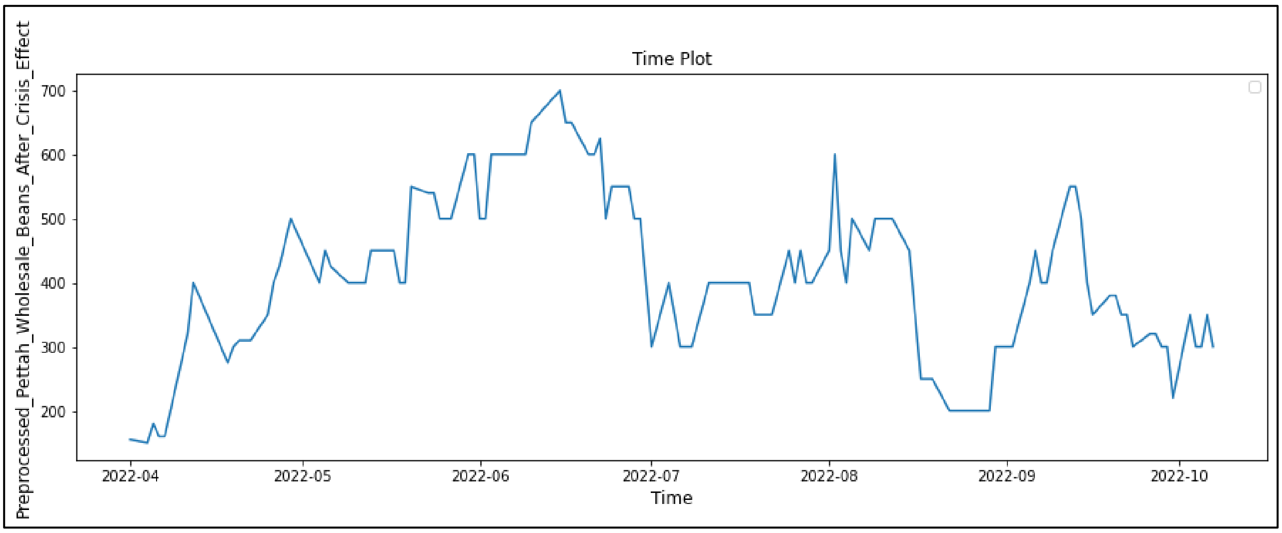


Figure 4.9: Time Plot of Prices During Crisis

At initial stages, time series plot for Pettah wholesale beans was analysed. Based on the time plot as shown in Figure 4.9, it can be concluded that, there is no seasonality or trend. During the crisis period, Pettah wholesale bean prices were increased drastically and most of the prices were varied between 300-600 (per kg).

4.1.9.2 Decomposing bean price data

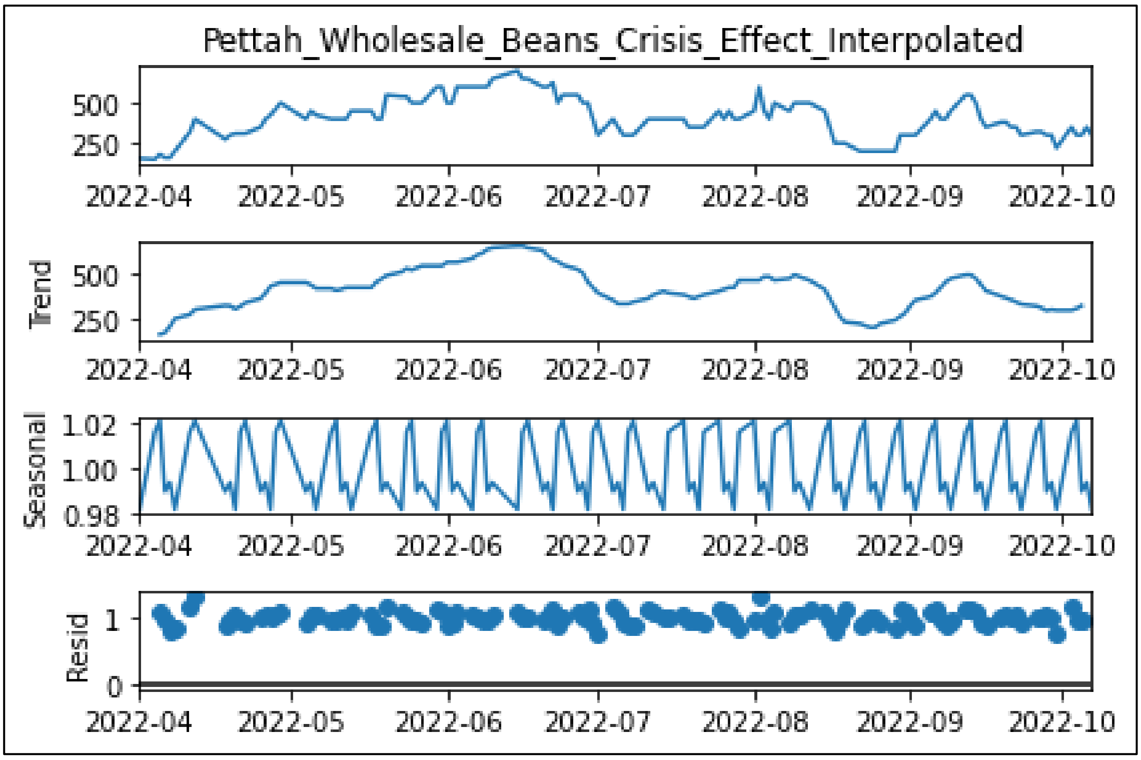


Figure 4.10: Decomposing time series

Based on the Figure 4.10, there is no clear trend for prices, seasonality was presented, and outliers were there due to variability in macroeconomics. Average prices were varied between 300-600.

4.1.9.3 Smoothing using Holt Winters Single Exponential Smoothing

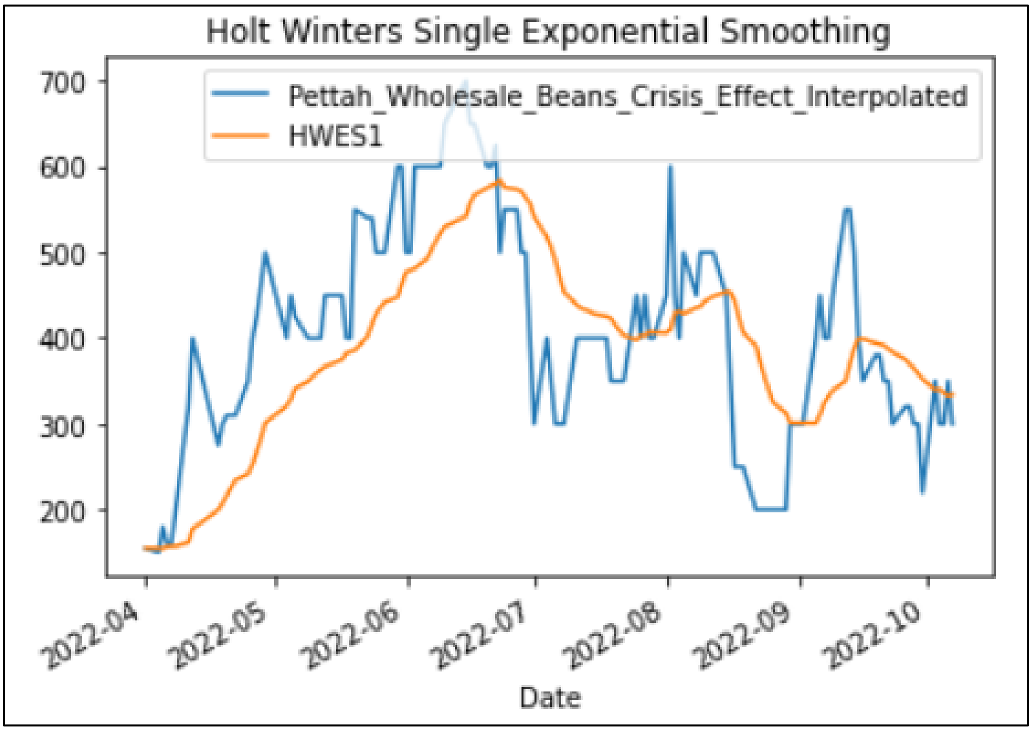


Figure 4.11: Smoothing time series

Since we have limited number of data points (126 records) for this scenario, there is no clear trend identified during decomposition. By applying smoothing, it removes irregular roughness to see a clear trend or pattern in data. Holt winters single exponential smoothing was applied to see a clear pattern in prices as shown in Figure 4.11.

4.1.9.4 Rolling Window

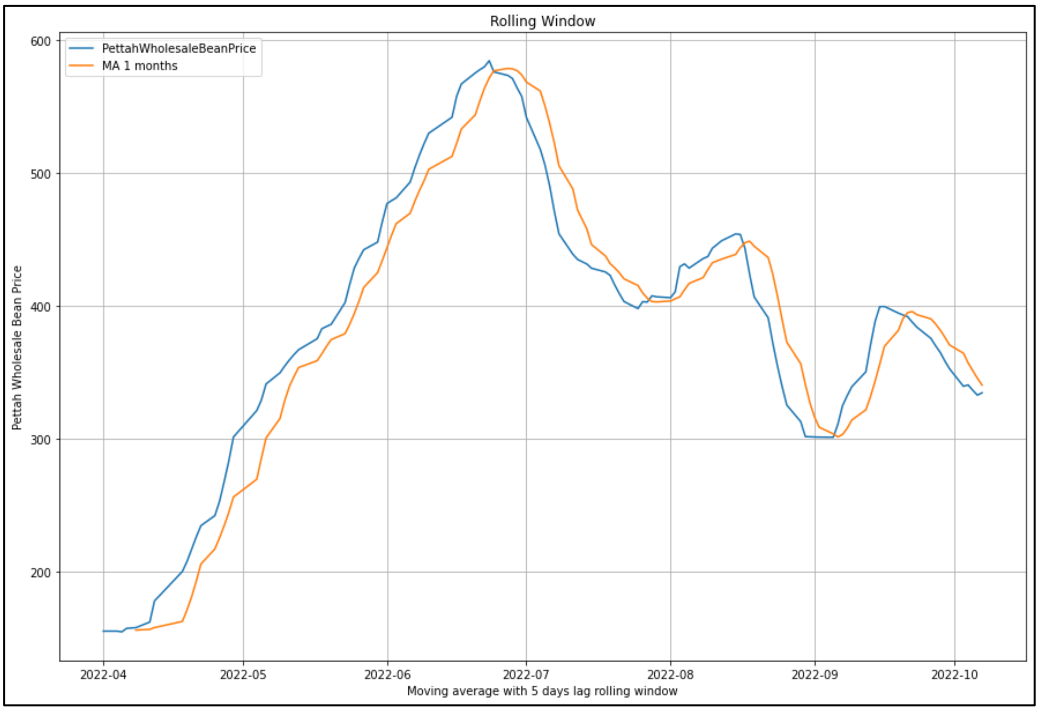


Figure 4.12: Rolling Window

Figure 4.12 depicts the rolling window with moving average of 5 days lag for Pettah wholesale bean price during crisis period.

4.1.9.5 Autocorrelation and partial autocorrelation plot

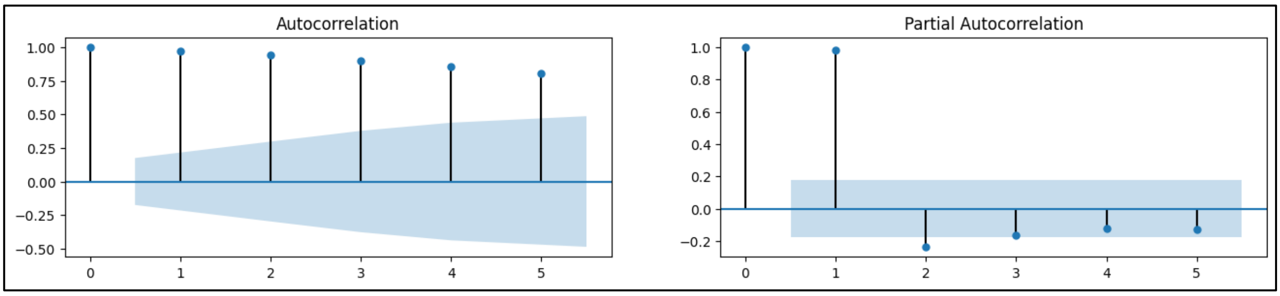


Figure 4.13: ACF and PACF Plot

ACF and PACF plots of Pettah wholesale bean price data are shown in

Figure 4.13. The ACF plot has multiple significant corelations, where higher spikes were presented at lag 1, 2, 3, 4 and 5. PACF has 2 significant correlations at lag 1, and 2.

4.1.9.6 Training and Testing data sets plot

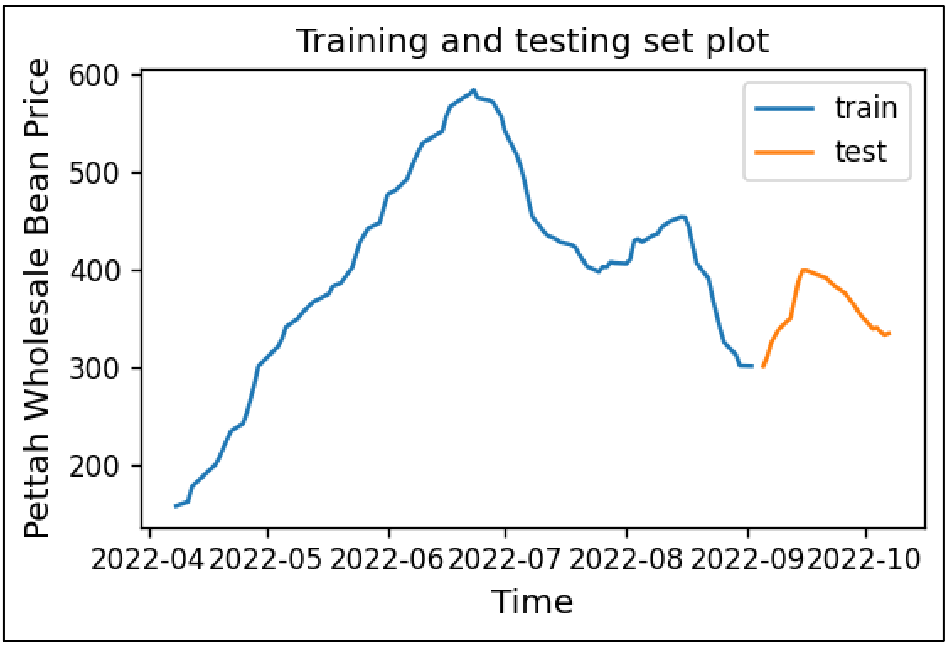


Figure 4.14: Train Set and Test Set Distribution

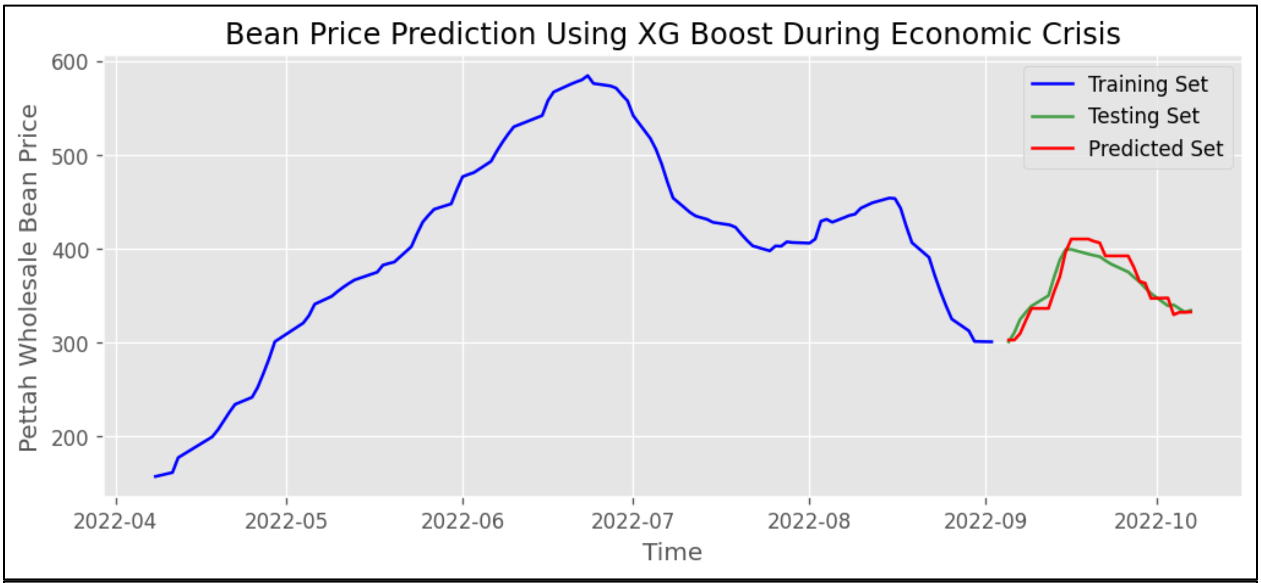
Figure 4.14 depicts the training and testing data sets distribution of Pettah wholesale bean prices data during the crisis period. From the entire data set, first 80% of the data were used as the training set and the rest of the 20% of the data were used as the testing set.

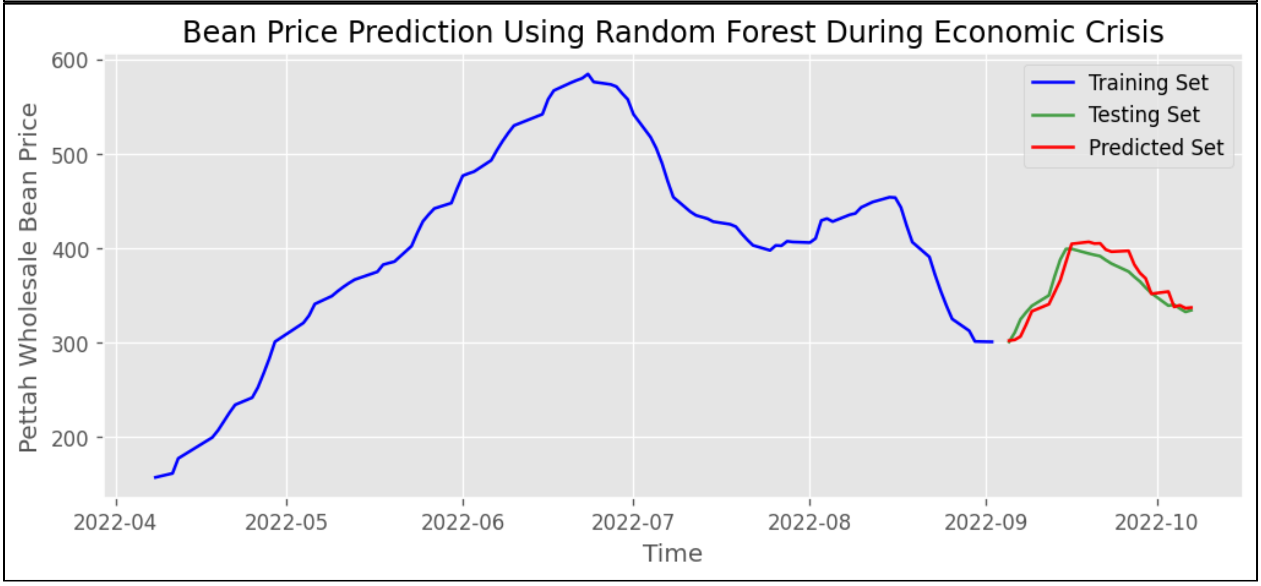
4.1.9.7 Predicted prices of each machine learning model

Gradient Boost, Random Forest, XG Boost and Stack Regressor models were applied for the collected vegetable prices data set from April 2022-September 2022 and prices were predicted for the testing data set and the results can be shown in the Table 4.4.

Table 4.4: Predicted prices using each of the model.

Date (mm/dd/yyyy)	Actual values in testing data	XGB Predictions	GB Predictions	RF Predictions	Stack Model Predictions
09/05/2022	300.95	302.93	300.88	302.28	257.15
09/06/2022	310.85	302.93	301.39	303.16	257.68
09/07/2022	324.77	309.71	307.78	306.65	267.38
09/08/2022	332.29	323.58	321.85	319.31	286.05
09/09/2022	339.06	336.56	335.29	333.33	307.05
09/12/2022	350.16	336.56	335.29	340.84	304.88
09/13/2022	370.14	354.49	355.00	353.37	332.33
09/14/2022	388.13	370.41	371.04	366.07	357.77
09/15/2022	399.31	396.44	397.67	385.55	401.16
09/16/2022	399.38	410.40	410.74	404.67	414.80





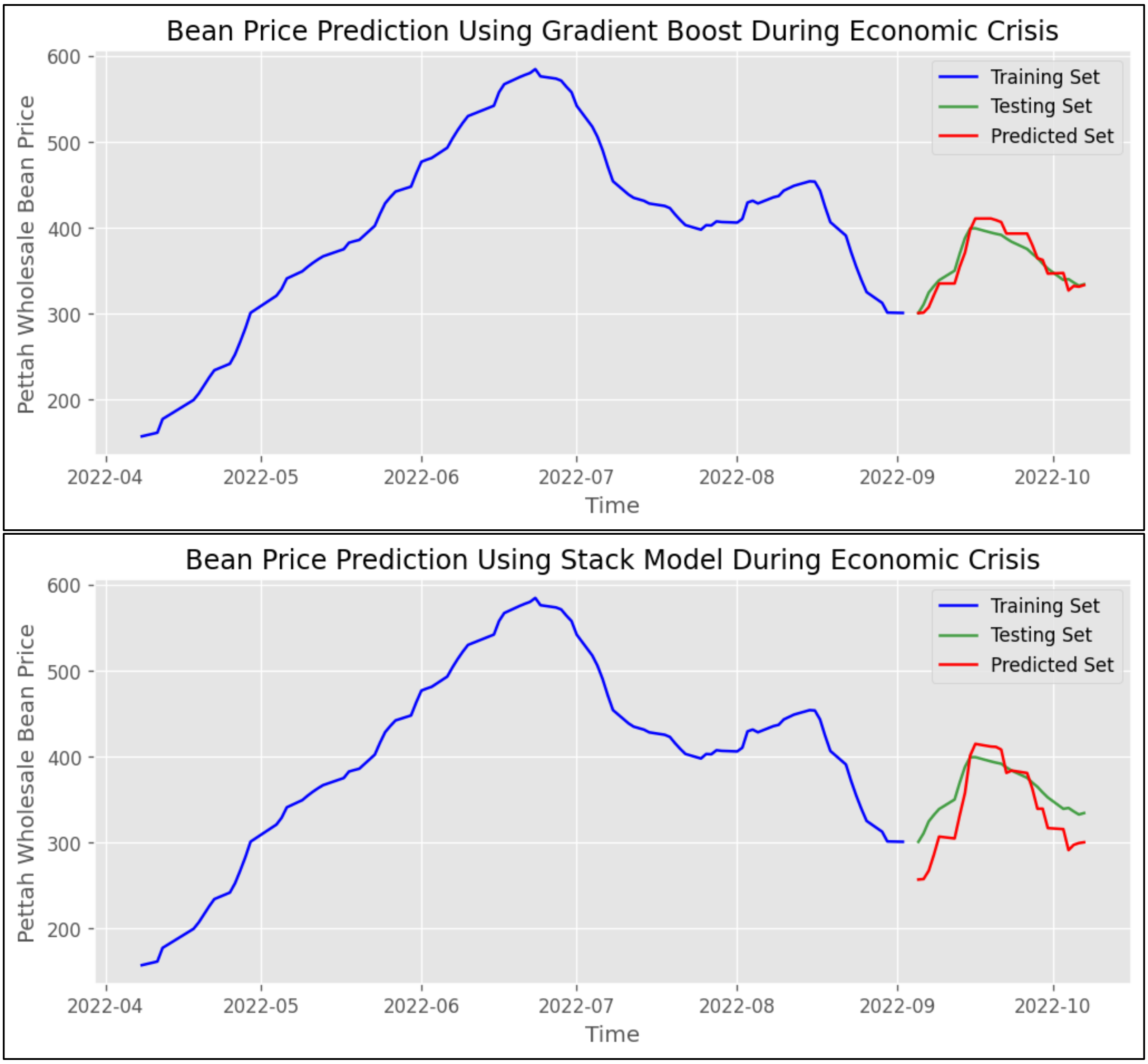


Figure 4.15: Graphical Representation of training set, testing set and the predicted set of each model

Graphical representations of the training prices, testing prices and predicted prices for Pettah wholesale bean after crisis were implemented using each model can be shown in Figure 4.15 using blue, green and red colours respectively as above. Although the prediction results were slightly fluctuated with the actual data in some cases, overall, all 4 models were performed well.

4.1.9.8 Evaluating the models

To forecast the future wholesale bean prices in Sri Lanka after crisis, performance of each model was evaluated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Coefficient of Determination (R^2) metrics as shown in Table 4.5. Based on the results shown there, XGB was presented the higher R^2 value as 86.63% with the lowest errors such that MAPE as 2.40% and RMSE as 10.39. The stack regressor model was provided the worst results over the other models where R^2 value was -28.12%, MAPE as 8.06% and RMSE as 32.17. Therefore, it can be concluded that, stack regressor model is not fitting to this data set.

Table 4.5: MSE, RMSE, MAE, MAPE and R^2 values of each model.

Model	**MSE**	**RMSE**	**MAE**	**MAPE (%)**	**R^2 (%)**
XGB	107.97	10.39	8.77	2.40	86.63
GB	119.26	10.92	9.17	2.52	85.23
Random Forest	140.57	11.85	10.26	2.81	82.59
Stack Model	1,034.96	32.17	27.71	8.06	-28.12

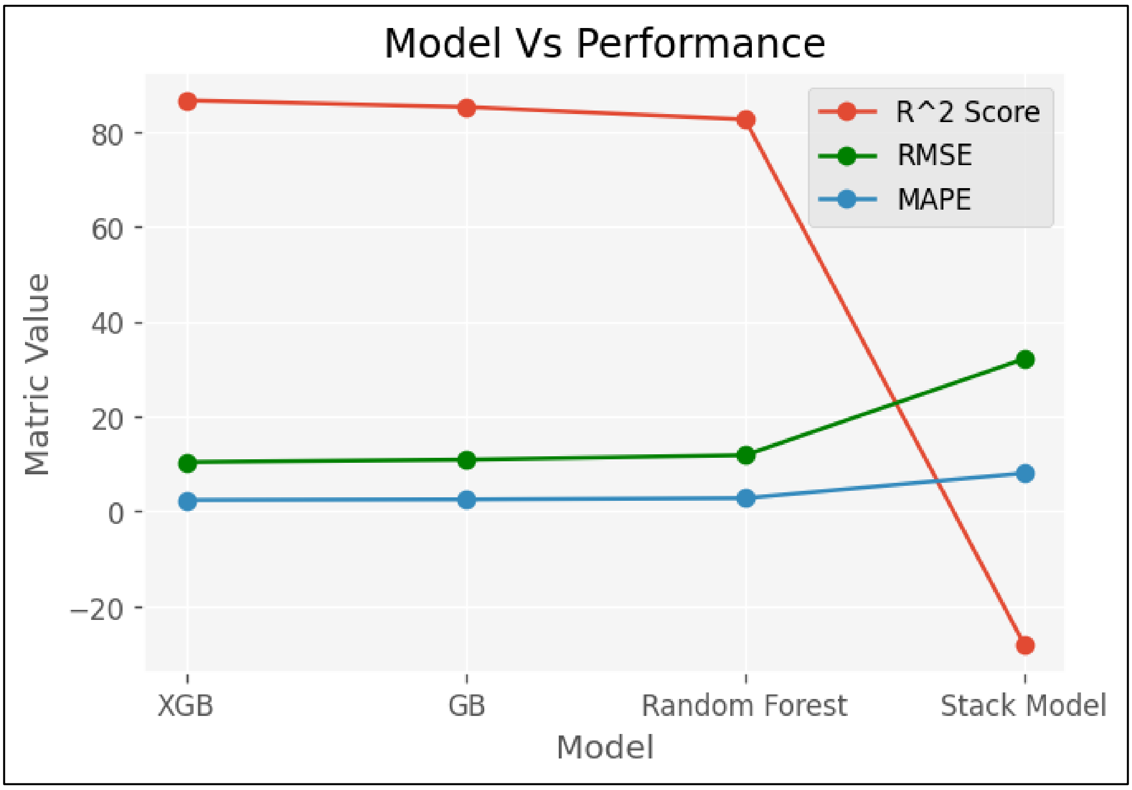


Figure 4.16: Model Vs Performance

Based on the values of the performance matrices as shown in Figure 4.16 and discussed as above, it can be concluded that, the XG Boost regressor performs better than the other models. Therefore, the XG Boost regressor can be selected as the best model to predict the wholesale prices of beans in the Pettah market during the inflation period.

4.1.9.9 Short term price forecasting for Pettah wholesale bean price

XG Boost regressor was suggested as the best model for the Pettah wholesale bean price prediction after the crisis effect. Using the XG Boost regressor, prices were forecasted for the next 5 days. Since the actual data also currently available for the forecasting period, actual data was collected for the forecasting period as well. Both actual and forecasted prices for Pettah wholesale beans after crisis can be shown as in Table 4.6.

Table 4.6: Actual prices Vs Forecasted prices.

Date (mm/dd/yyyy)	Actual Prices	Forecasted Prices
10/10/2022	350	333.43
10/11/2022	300	353.19
10/12/2022	350	333.43
10/13/2022	250	352.01
10/14/2022	300	333.43

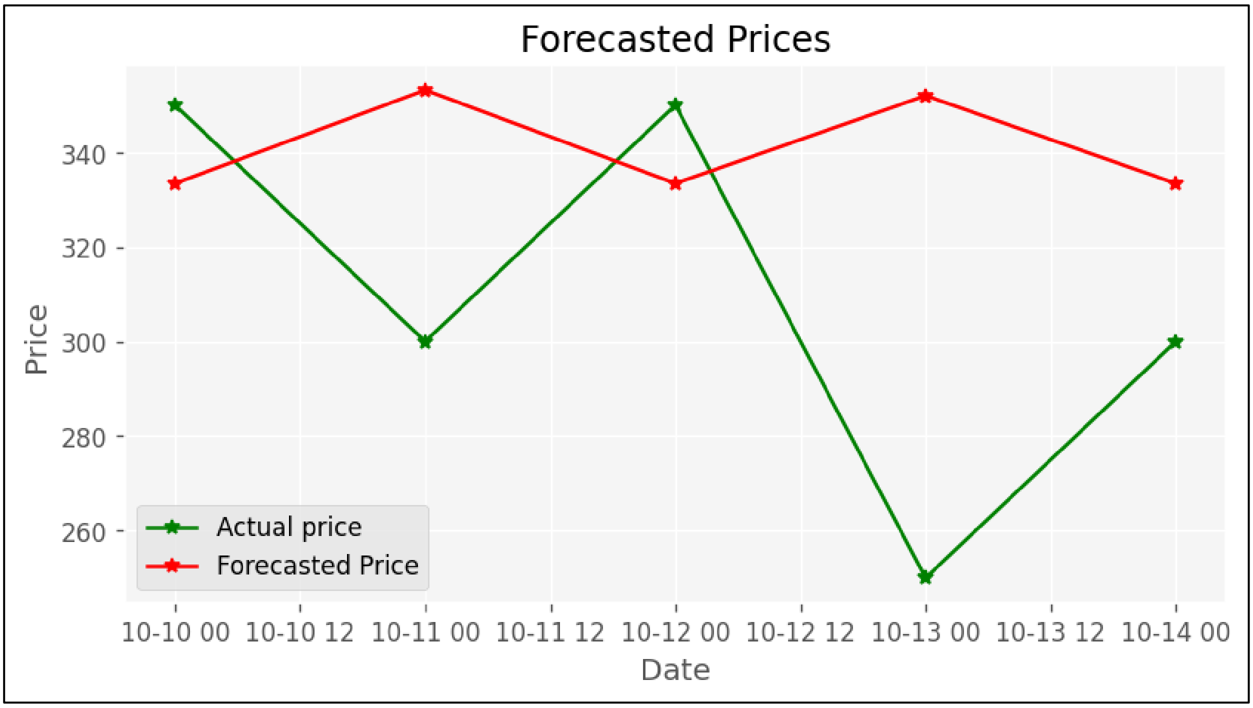


Figure 4.17: Graphical representation of actual prices Vs forecasted prices

Based on the results shown in Figure 4.17, forecasted prices were slightly deviated from the actual prices. Among them, 2nd day and the 4th day forecasted prices were more deviated with the actual prices than other 3 days.

4.2 Pettah Wholesale Carrot Prices Results

Results obtained for the Pettah wholesale carrot prices before the crisis and after the crisis were explained in following sections. Exploratory data analysis was conducted at initial steps and results were interpreted. Then proposed machine learning techniques were applied and their results were discussed. Then results were evaluated using evaluation methods and selected the best model for wholesale carrot price forecasting. Finally, short term price forecasting was done using the selected best model.

4.2.1 Time Plot

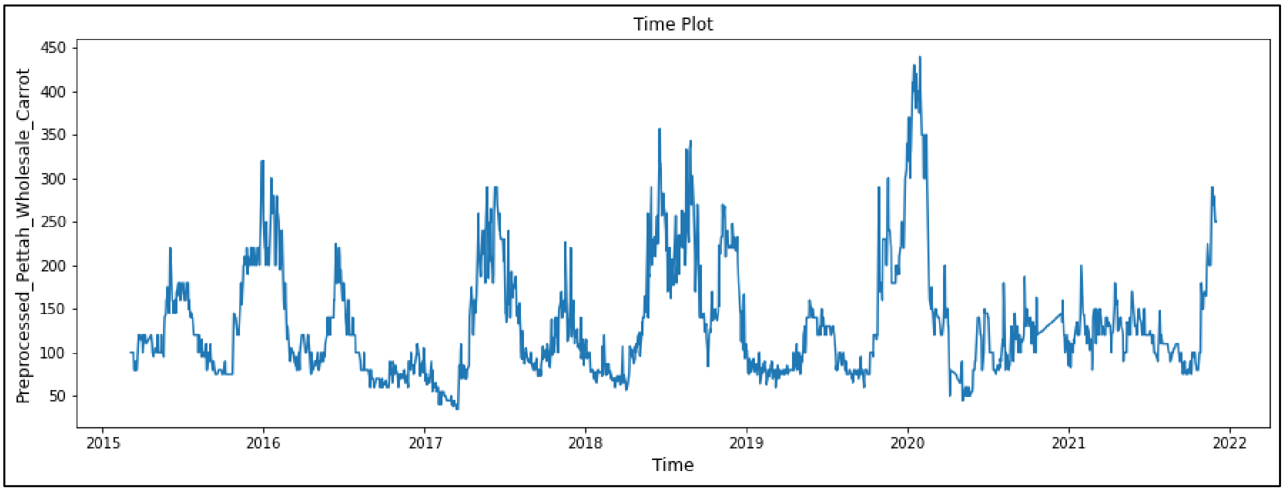


Figure 4.18: Time Plot

At initial stages, time series plot for Pettah wholesale carrot was analysed. Based on the time plot as shown in Figure 4.18, It can be concluded that, seasonality presents but overall, there is no trend. During the year end of most of the years, there is an upward trend for the wholesale carrot prices.

4.2.2 Decomposing carrot price data

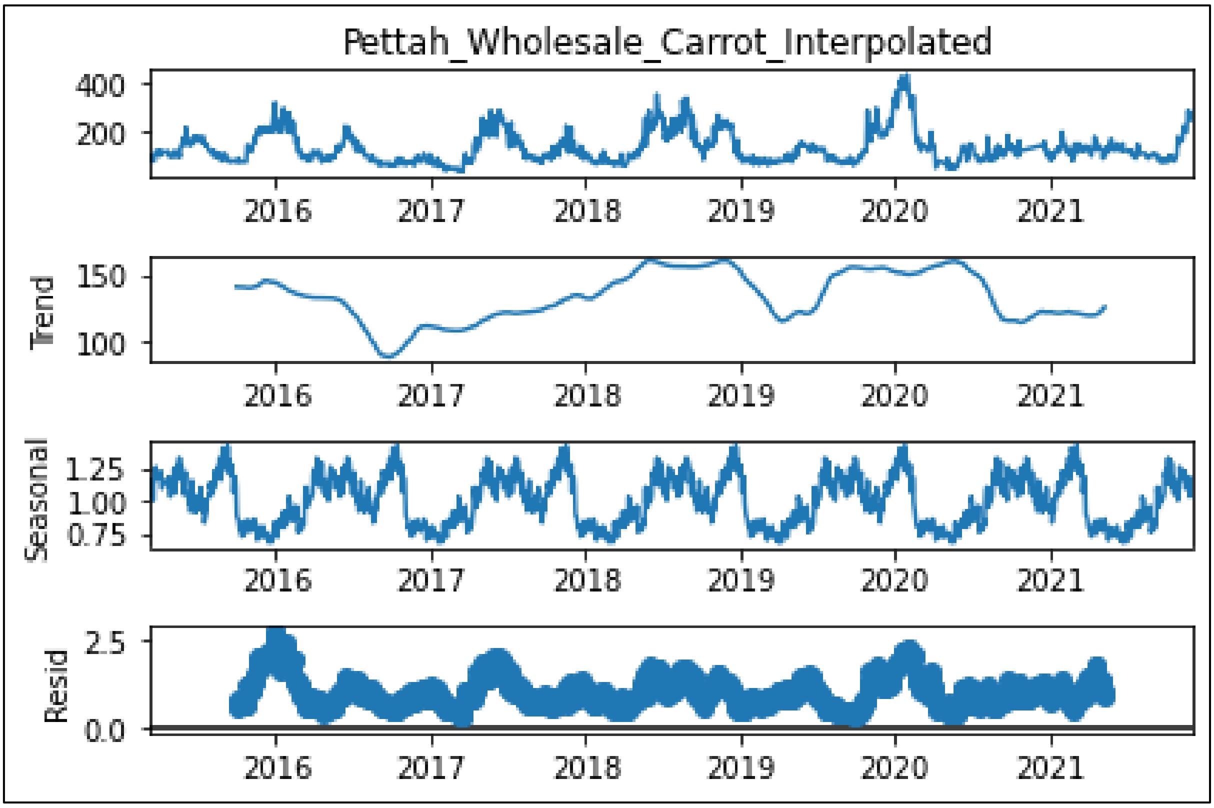


Figure 4.19: Decomposing the time series

Based on the Figure 4.19, there is no clear trend for carrot prices, seasonality was presented, and outliers were there due to variability in macroeconomics.

4.2.3 Rolling Window

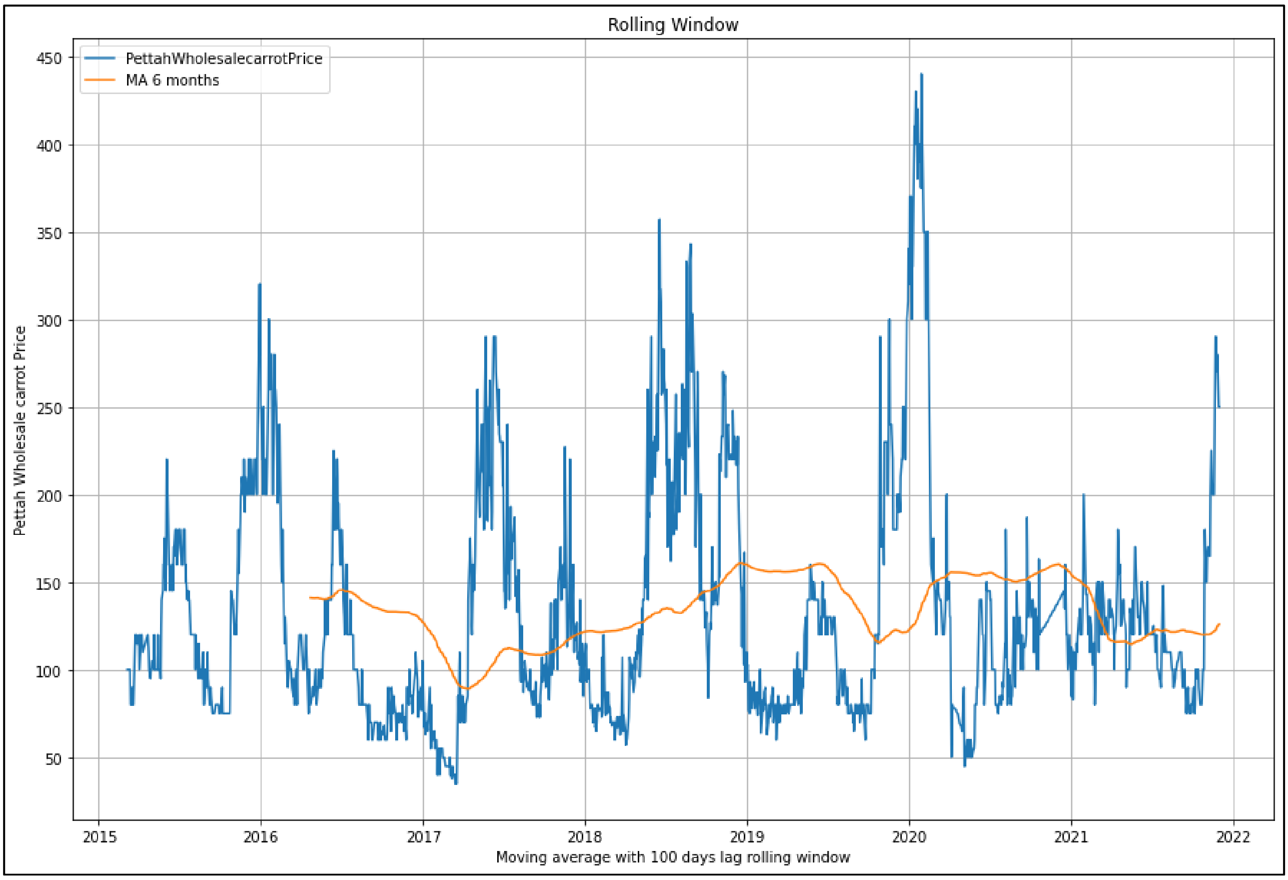


Figure 4.20: Rolling Window

Figure 4.20 depicts the rolling window with moving average of 264 days lag for Pettah wholesale carrot price.

4.2.4 Autocorrelation and partial autocorrelation plot

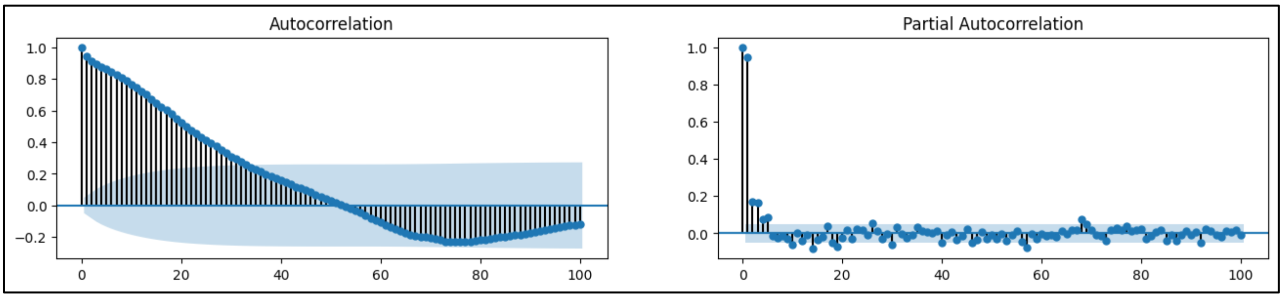


Figure 4.21: ACF and PACF Plot

Based on the ACF plot we can identify a pattern in the carrot price data as shown in

Figure 4.21. The ACF plot has multiple significant corelations, but higher spikes present at lag 0 and lag 1. PACF has 4 significant correlations at lag 0, 1, 2 and 3.

4.2.5 Training and Testing data sets plot

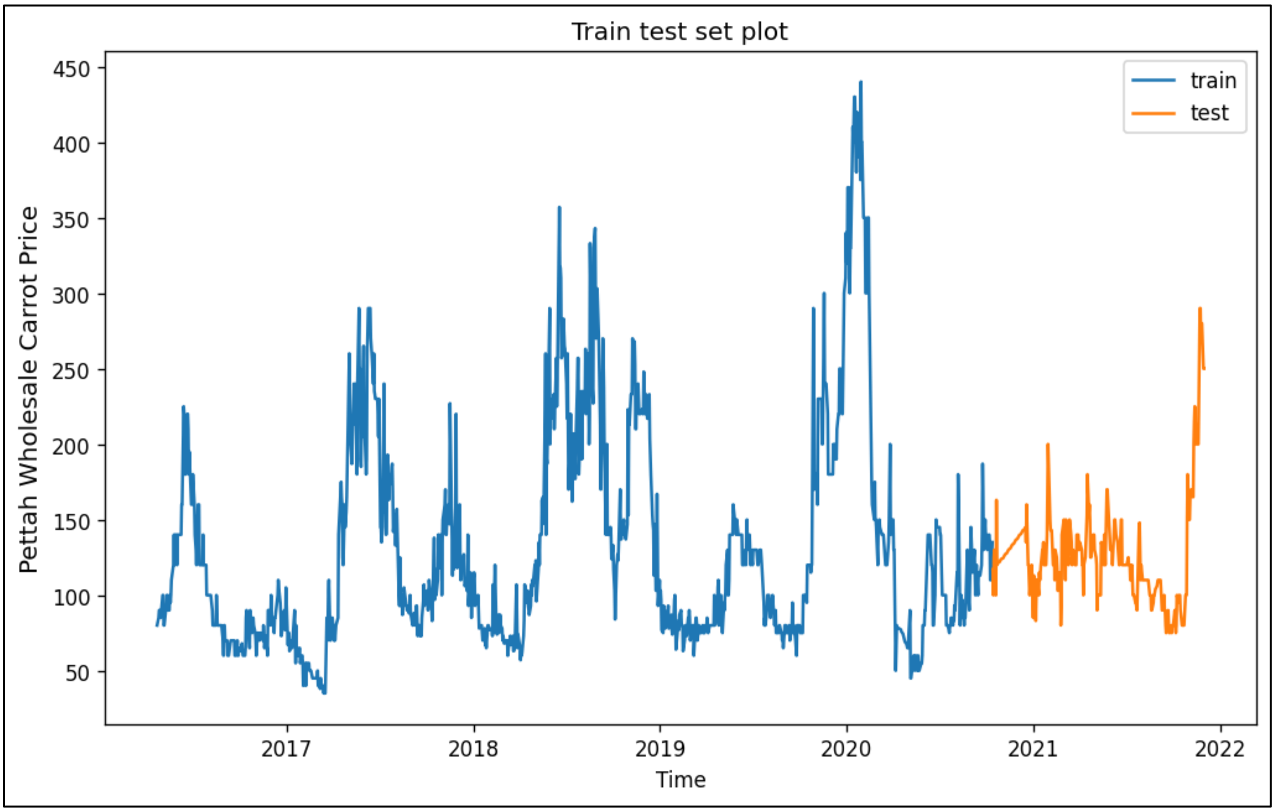


Figure 4.22: Training and testing set

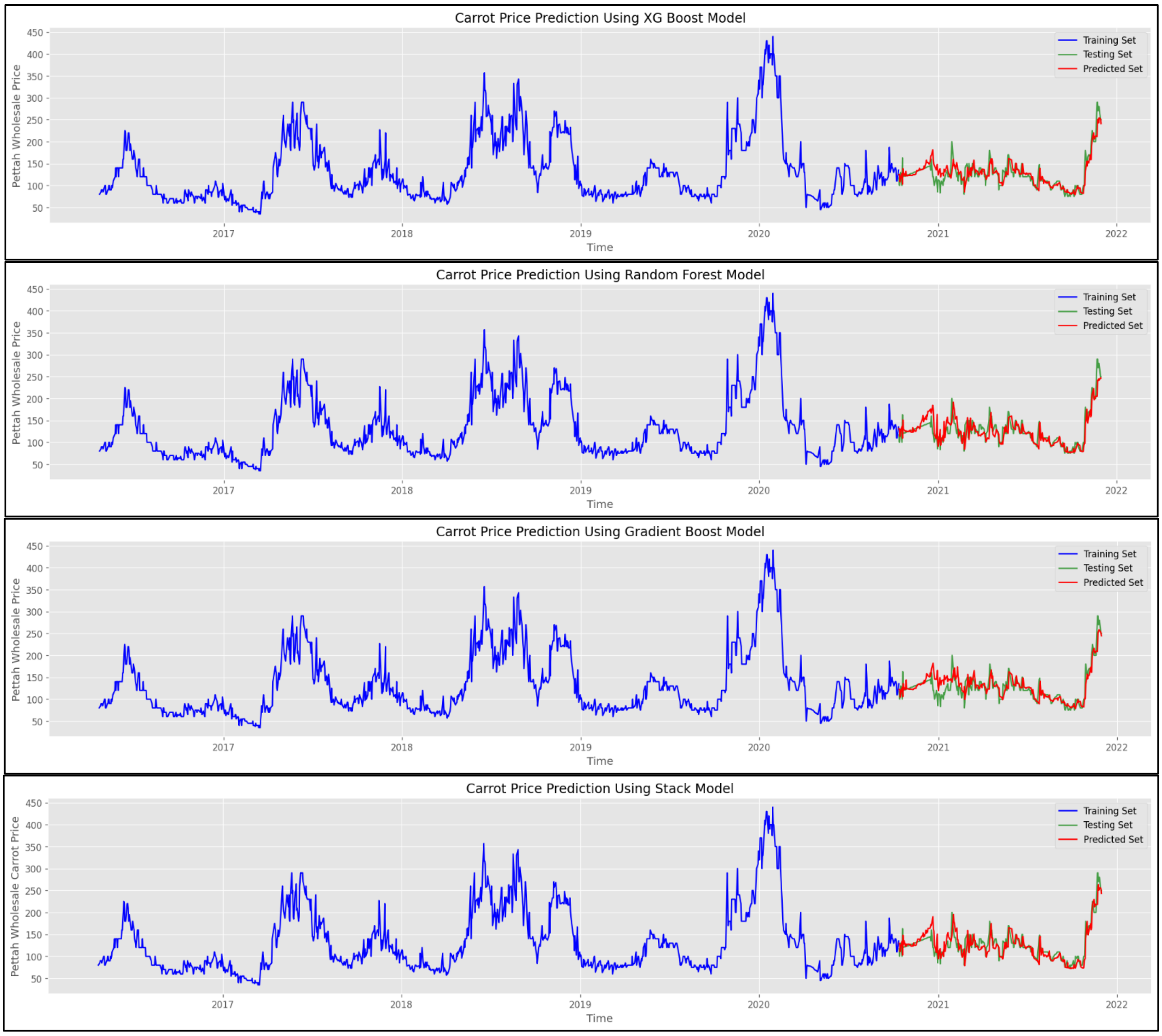
Figure 4.22 depicts the training and testing data sets distribution of Pettah wholesale carrot prices data.

4.2.6 Predicted prices of each machine learning model

Gradient Boost, Random Forest, XG Boost and Stack Regressor models were applied for the collected carrot prices data set from 2015-2021 before the economic crisis and prices were predicted for the testing data set and subset of results can be shown in Table 4.7. It shows the predicted prices using XG boost, gradient boost, random forest and stack model with the actual prices. From that, how the predicted prices deviate from the original prices can be identified.

Table 4.7: Predicted prices using each of the model.

Date (mm/dd/yyyy)	Actual values in the testing Data	XGB Predictions	GB Predictions	RF Predictions	Stack Model Predictions
11/01/2021	150	154.86	157.23	175.36	179.72
11/02/2021	150	144.55	146.46	147.93	151.70
11/03/2021	165	155.49	153.65	144.00	148.74
11/05/2021	170	151.47	154.35	157.77	161.25
11/08/2021	165	170.12	173.16	173.73	174.49
11/09/2021	165	158.83	161.52	157.95	160.68
11/10/2021	200	159.65	161.33	161.27	165.28
11/11/2021	215	189.89	192.38	190.10	200.77
11/12/2021	225	204.50	206.86	209.57	218.76
11/15/2021	200	220.12	216.09	222.69	229.64


Carrot Price Prediction Using XG Boost Model

Carrot Price Prediction Using Random Forest Model

Carrot Price Prediction Using Gradient Boost Model

Carrot Price Prediction Using Stack Model

Figure 4.23: Graphical Representation of training set, testing set and the predicted set of each model

Graphical representation of the training carrot prices, testing carrot prices and the predicted carrot prices of each model can be shown using blue, green and red colours respectively as shown in Figure 4.23. All 4 models were predicted the carrot prices for the same testing data set. Overall, there are no huge fluctuations between the testing data and predicted data, but in some cases predicted results were underestimated the actual results and in some cases prediction results were overestimated the actual results. But overall performance was good for all models for the Pettah carrot price data.

4.2.7 Evaluating the models

To forecast the future wholesale carrot prices in Sri Lanka, performance of each model was evaluated similarly as discussed during the Pettah bean price prediction. For that, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Coefficient of Determination (R^2) metrics were used, and their values can be shown as in Table 4.8. Based on the values presented in the table, it can be concluded that XGB can be selected as the best model for the wholesale carrot price prediction since it presented the higher R^2 value as 74.45% and lower errors such that MAPE as 9.4% and RMSE as 17.83 respectively.

Table 4.8: MSE, RMSE, MAE, MAPE and R^2 values of each model.

Model	MSE	RMSE	MAE	MAPE (%)	R^2 (%)
XGB	317.98	17.83	11.80	9.40	74.45
GB	374.11	19.34	13.04	10.58	69.94
Random Forest	379.14	19.47	13.25	10.21	69.54
Stack Model	345.66	18.59	12.98	10.14	72.23

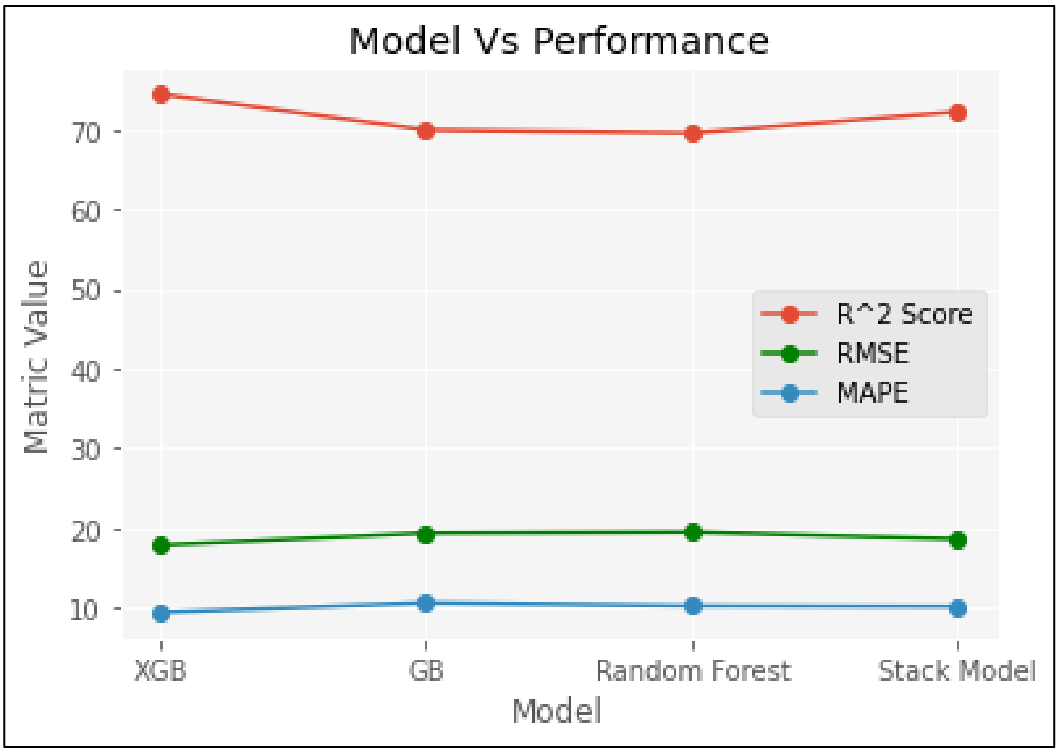


Figure 4.24: Model Vs Performance

Based on the values of the performance matrices as shown in Figure 4.24, it can be concluded that XG Boost model performed better than the other models. Therefore, the XG Boost regressor can be chosen as the best model to predict the wholesale prices of carrot in the Pettah market.

4.2.8 Short term price forecasting for Pettah wholesale carrot price

XG Boost regressor was suggested as the best model for the Pettah wholesale carrot price prediction. Using the XG Boost regressor, prices were forecasted for the next 5 days. Since the actual data also currently available for the forecasting period, actual data was collected for the forecasting period as well. Both actual and forecasted prices for Pettah wholesale carrot can be shown as in Table 4.9 as below.

Table 4.9: Actual prices Vs Forecasted prices.

Date (mm/dd/yyyy)	Actual Prices	Forecasted Prices
12/01/2021	250	244.65
12/02/2021	250	156.44
12/03/2021	250	245.56
12/06/2021	250	154.38
12/07/2021	250	248.52

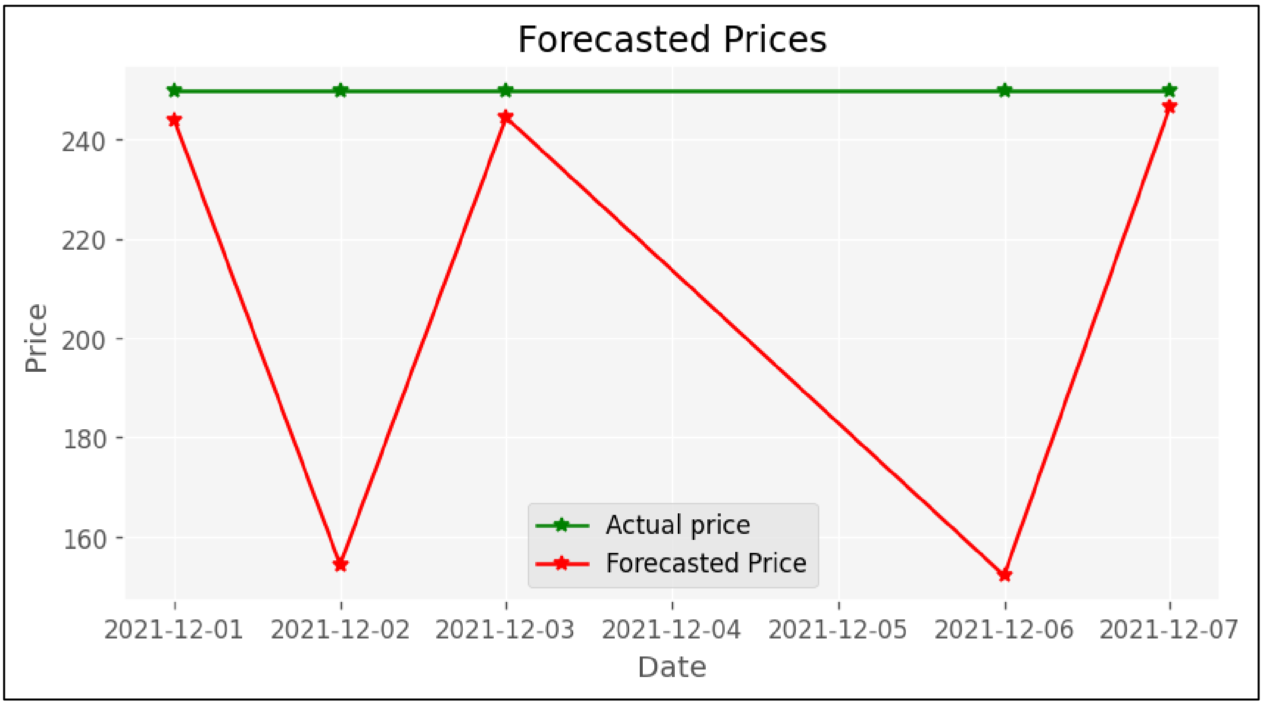


Figure 4.25: Graphical representation of actual prices Vs forecasted prices

Figure 4.25 depicts the graphical representation of the forecasted carrot prices and the actual carrot prices as shown above. XG boost regressor was used to forecast the wholesale carrot prices for the next 5 business days, but they were slightly underestimated the actual prices.

4.2.9 Additional Step – Predicting wholesale carrot prices after economic crisis in Sri Lanka

During the inflation period, carrot prices were increased, and that effect could not be addressed from the above scenario. Therefore, carrot prices during that time were predicted separately by collecting the carrot price reports after crisis from the period of April of 2022 – October of 2022.

4.2.9.1 Time Plot

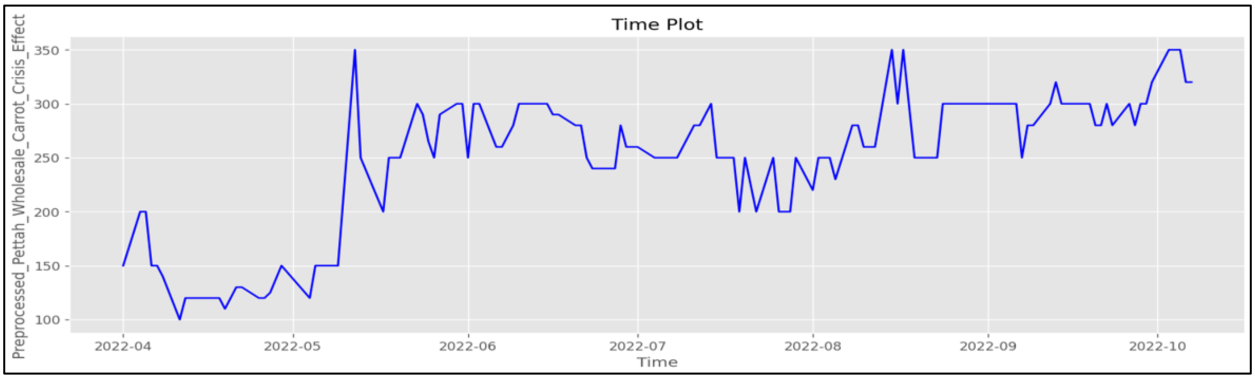


Figure 4.26: Time Plot of prices during the crisis

At initial stages, time series plot for Pettah wholesale carrot was analysed. Based on the time plot as shown in Figure 4.26, it can be concluded that, there is no seasonality or trend. During the crisis period, Pettah wholesale carrot prices were increased drastically and recently most of the prices were fluctuated between 200-350 (per kg).

4.2.9.2 Decomposing carrot price data

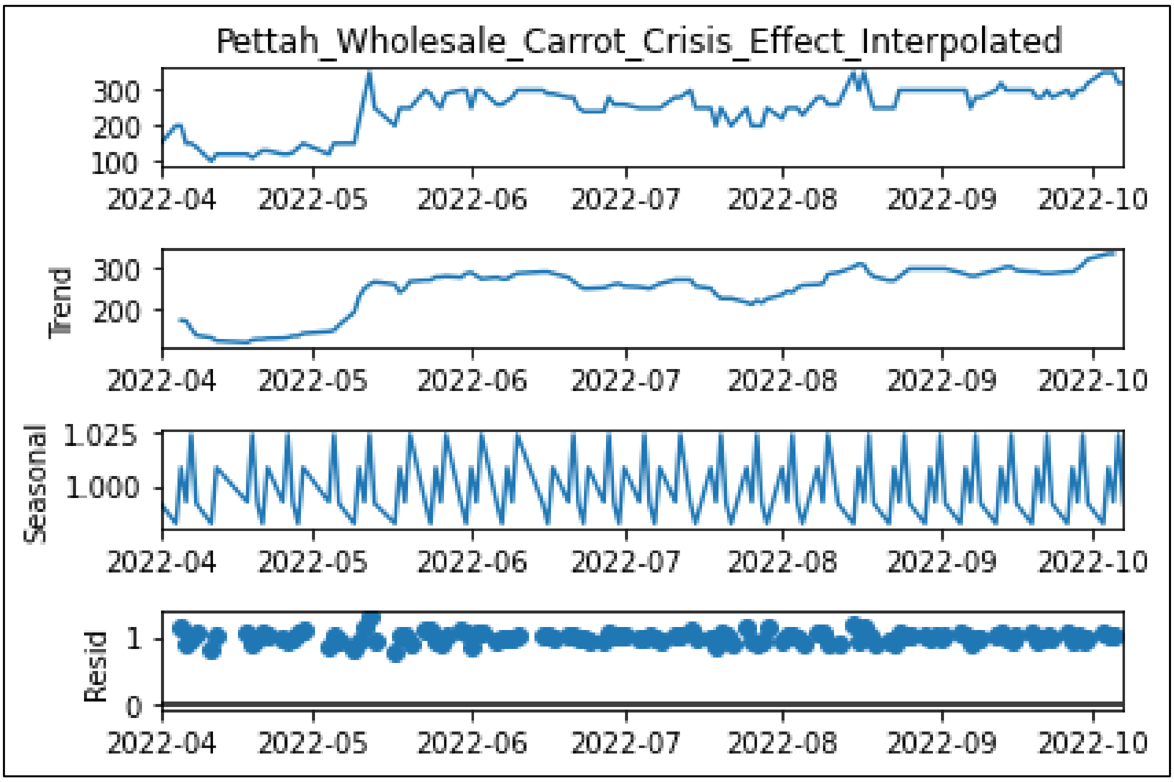


Figure 4.27: Decomposing time series

Based on the Figure 4.27 results, there is no clear trend for the prices, seasonality was presented, and outliers were there due to variability in macroeconomics. Average prices were varied between 200-350(per kg).

4.2.9.3 Smoothing using Holt Winters Single Exponential Smoothing

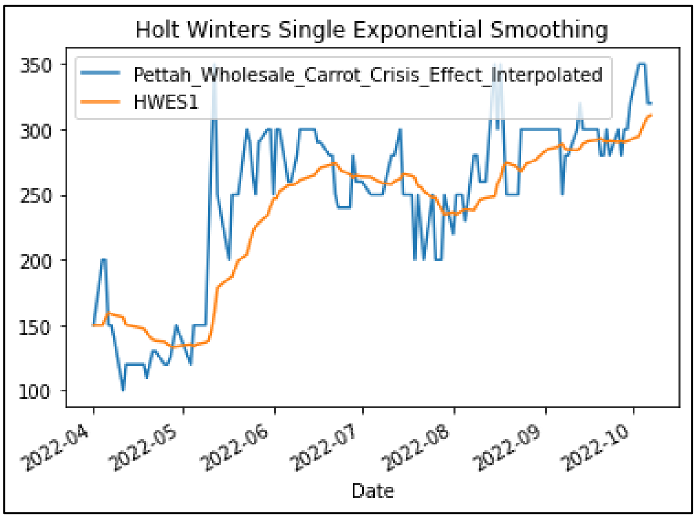


Figure 4.28: Smoothing the data set

Since we had limited number of data points (126 records) for this scenario, there is no clear trend identified during decomposition. By applying smoothing, it removes irregular roughness to see a clear trend or pattern in data. Holt winters single exponential smoothing was applied to see a clear pattern in prices as shown in Figure 4.28. But smoothing was not applied for the vegetable price predictions before the crisis since the data set contained 1612 records and also it does not involve irregular roughness in the data.

4.2.9.4 Rolling Window

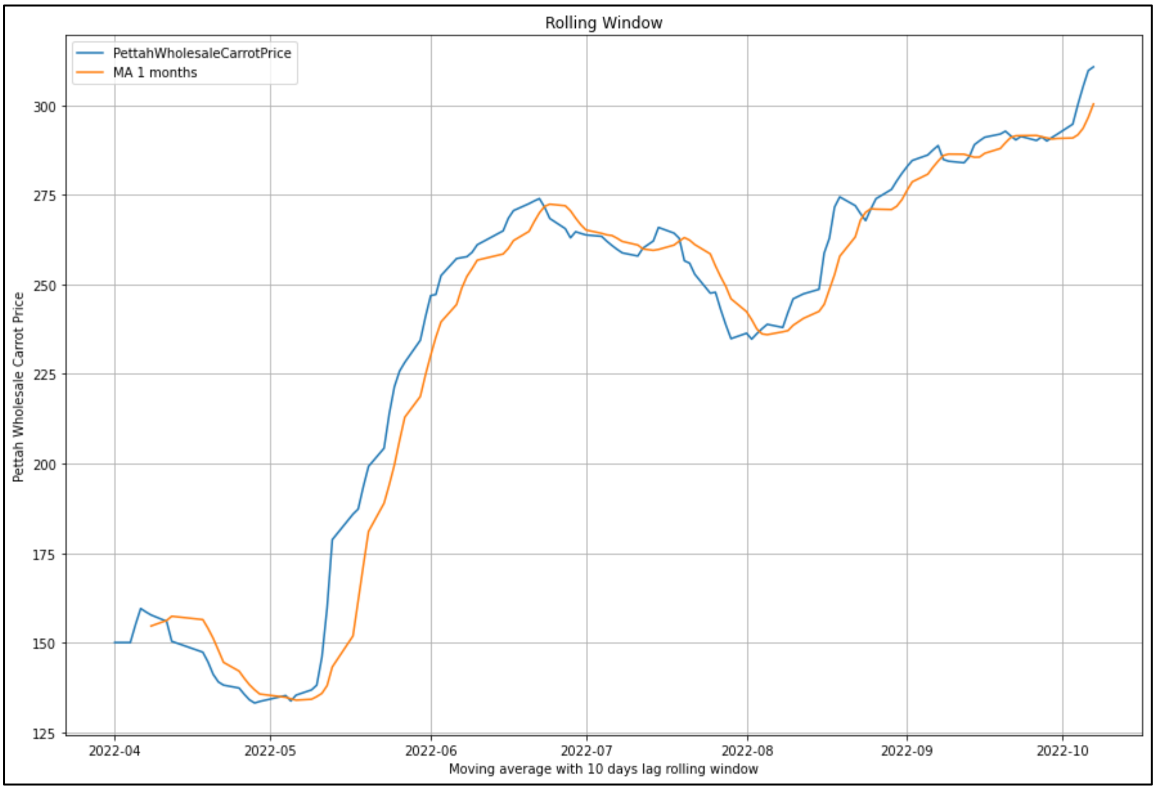


Figure 4.29: Rolling Window

Figure 4.29 depicts the rolling window with moving average of 5 days lag for Pettah wholesale carrot price during crisis period.

4.2.9.5 Autocorrelation and partial autocorrelation plot

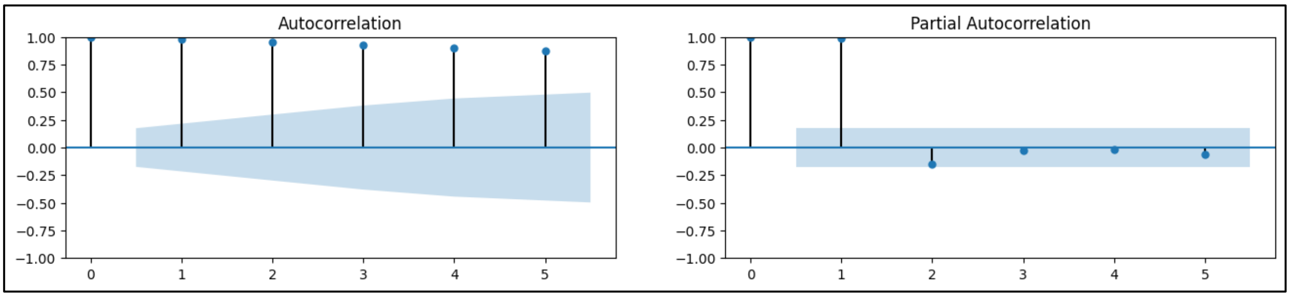


Figure 4.30: ACF and PACF Plot

ACF and PACF plots of Pettah wholesale carrot price data are shown in Figure 4.30. The ACF plot has multiple significant corelations, where higher spikes were presented at lag 1, 2, 3, 4 and 5. PACF has 1 significant correlation at lag 1.

4.2.9.6 Training and Testing data sets plot

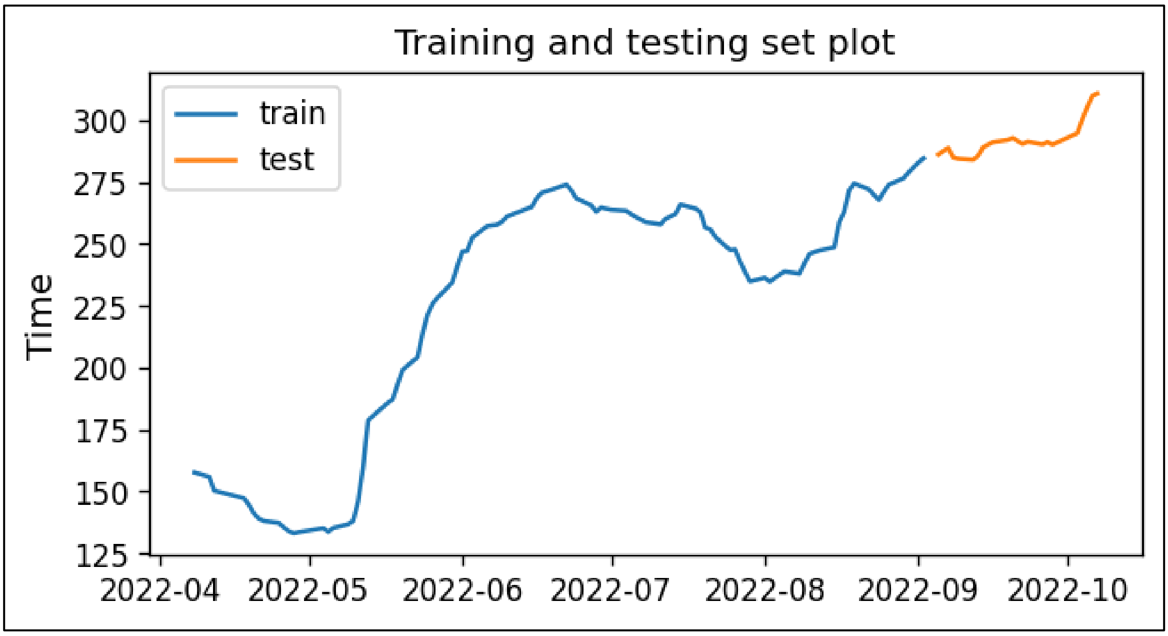


Figure 4.31: Training and testing set distribution

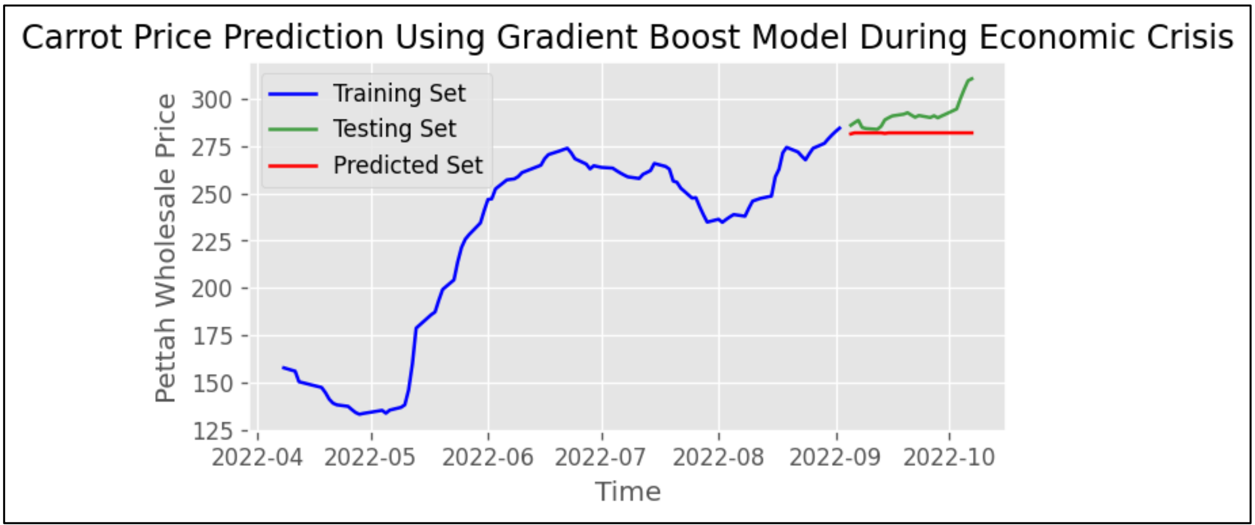
Figure 4.31 depicts the training and testing data sets distribution of Pettah wholesale carrot prices data during the crisis period.

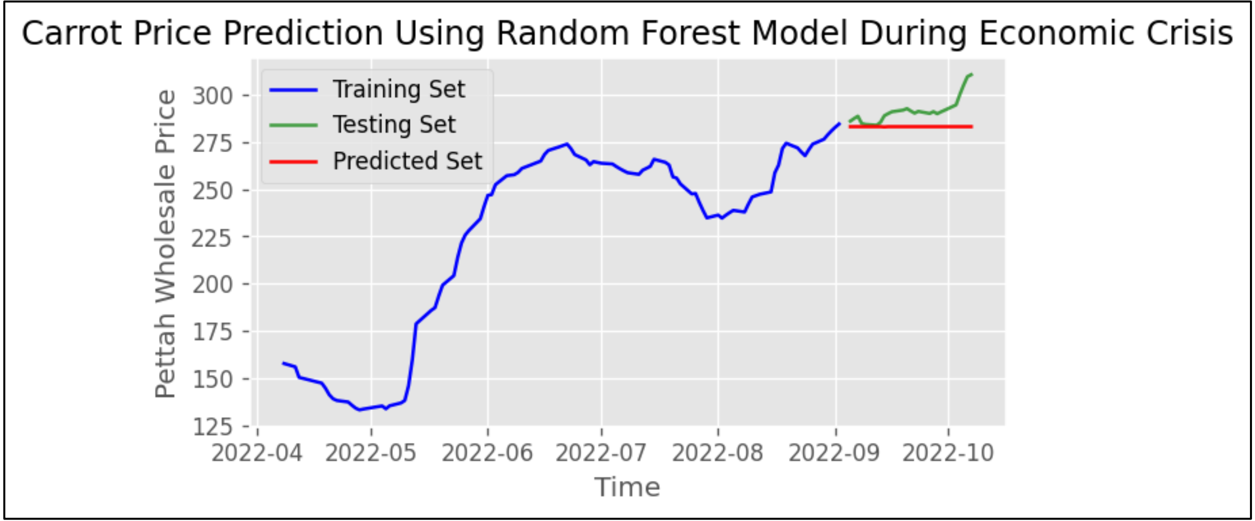
4.2.9.7 Predicted prices of each machine learning model

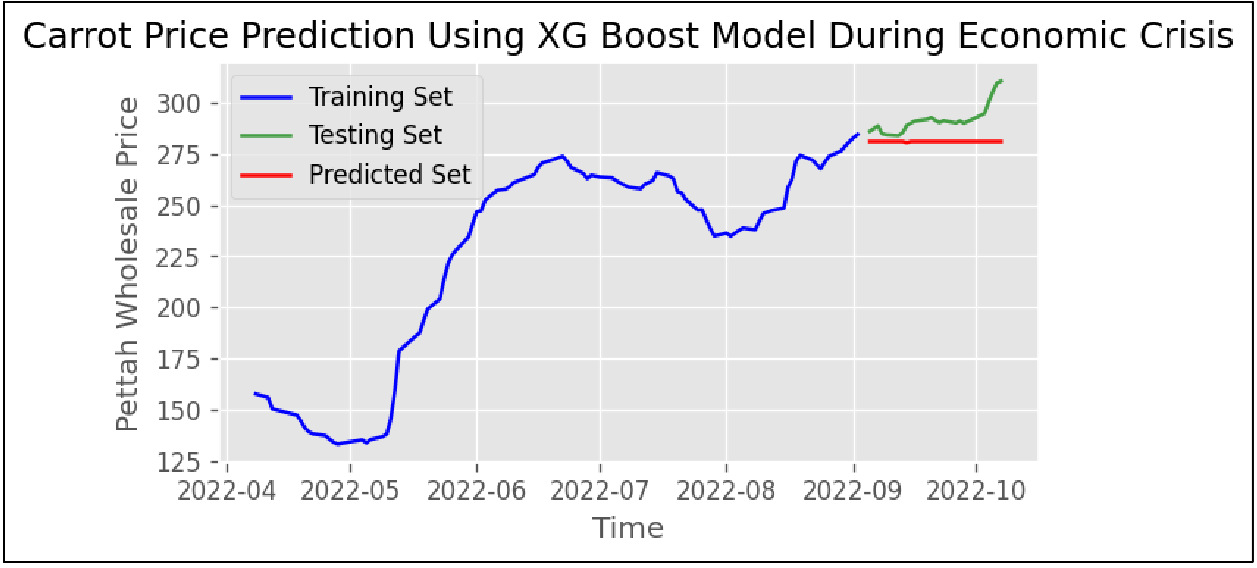
Gradient Boost, Random Forest, XG Boost, Stack Regressor and ARIMA models were applied for the collected carrot prices data set from April 2022-September 2022 and prices were predicted for the testing data set and the results can be shown in the Table 4.10.

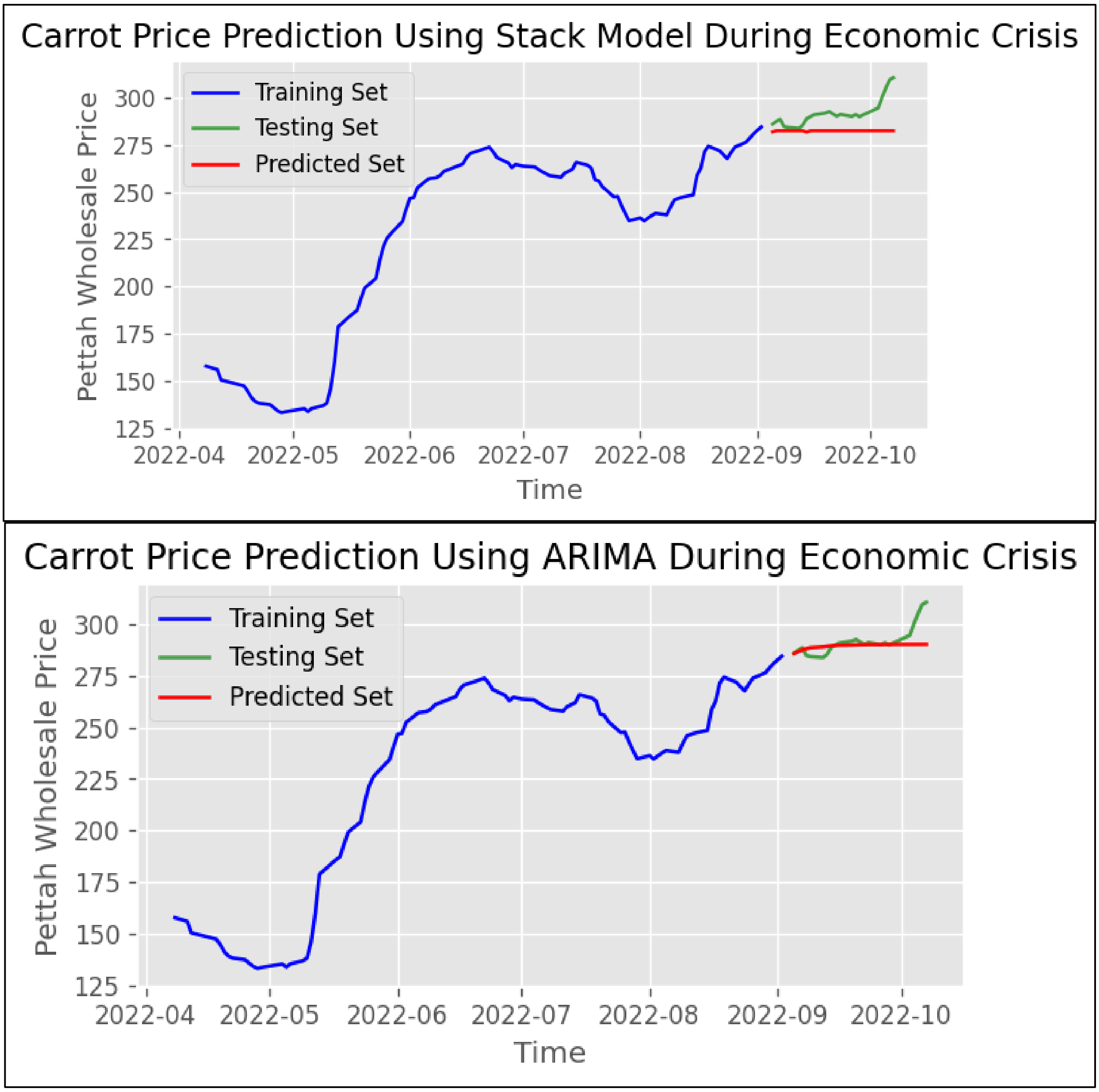
Table 4.10: Predicted prices using each of the model

Date (mm/dd/yyyy)	**Actual values in the testing set**	**XGB Predictions**	**GB Predictions**	**RF Predictions**	**Stack Model Predictions**	**ARIMA Predictions**
09/05/2022	286.15	281.07	281.58	283.25	282.07	286.90
09/06/2022	287.53	281.07	282.10	283.25	282.63	287.68
09/07/2022	288.78	281.07	282.10	283.25	282.63	288.29
09/08/2022	284.90	281.07	282.10	283.25	282.63	288.77
09/09/2022	284.41	281.07	282.10	283.25	282.63	289.14
09/12/2022	283.97	281.07	282.10	283.25	282.63	289.42
09/13/2022	285.57	281.07	282.10	283.25	282.63	289.65
09/14/2022	289.02	280.52	281.82	283.08	281.96	289.82
09/15/2022	290.11	281.07	282.10	283.25	282.63	289.96
09/16/2022	291.10	281.07	282.10	283.25	282.63	290.06


Carrot Price Prediction Using Gradient Boost Model During Economic Crisis
Training Set
Testing Set
Predicted Set
Pettah Wholesale Price
300
275
250
225
200
175
150
125
2022-04
2022-05
2022-06
2022-07
2022-08
2022-09
2022-10
Time


Carrot Price Prediction Using Random Forest Model During Economic Crisis
Training Set
Testing Set
Predicted Set
Pettah Wholesale Price
300
275
250
225
200
175
150
125
2022-04
2022-05
2022-06
2022-07
2022-08
2022-09
2022-10
Time


Carrot Price Prediction Using XG Boost Model During Economic Crisis
Training Set
Testing Set
Predicted Set
Pettah Wholesale Price
300
275
250
225
200
175
150
125
2022-04
2022-05
2022-06
2022-07
2022-08
2022-09
2022-10
Time



Carrot Price Prediction Using Stack Model During Economic Crisis

Carrot Price Prediction Using ARIMA During Economic Crisis

Figure 4.32: Graphical Representation of training set, testing set and the predicted set of each model

Graphical representations of the training prices, testing prices and predicted prices for Pettah wholesale carrot after the crisis were implemented using each model can be shown in Figure 4.32 using blue, green and red colours respectively as above. Machine learning models were not provided more accurate predictions for the carrot prices during the economic crisis. They were provided constant predictions for the test data set. Therefore, ARIMA model was also applied for the analysis. Where ARIMA model was provided better predictions than the ML models for carrot prices during the economic crisis.

4.2.9.8 Evaluating the models

To forecast the future wholesale carrot prices in Sri Lanka after the crisis, performance of each model was evaluated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) metrics as shown in Table 4.11. Based on the obtained results, all machine learning models have provided slightly higher errors. Therefore, it can be concluded that the machine learning models are not suitable to predict the Pettah wholesale carrot price data after crisis. Since the machine learning models were not performed well for this data set, ARIMA model (traditional statistical method) was also applied for this data set. ARIMA was presented the lowest errors such that MAPE as 1.64% and RMSE as 7.74 over the other machine learning models. Therefore, ARIMA was selected as the best model for the wholesale carrot price prediction after the economic crisis.

Table 4.11: MSE, RMSE, MAE and MAPE of each model.

Model	MSE	RMSE	MAE	MAPE (%)
XGB	172.81	13.14	11.14	3.75
GB	150.90	12.28	10.11	3.40
Random Forest	129.45	11.37	08.98	3.02
Stack Model	140.70	11.86	9.60	3.23
ARIMA	60.01	7.74	4.93	1.64

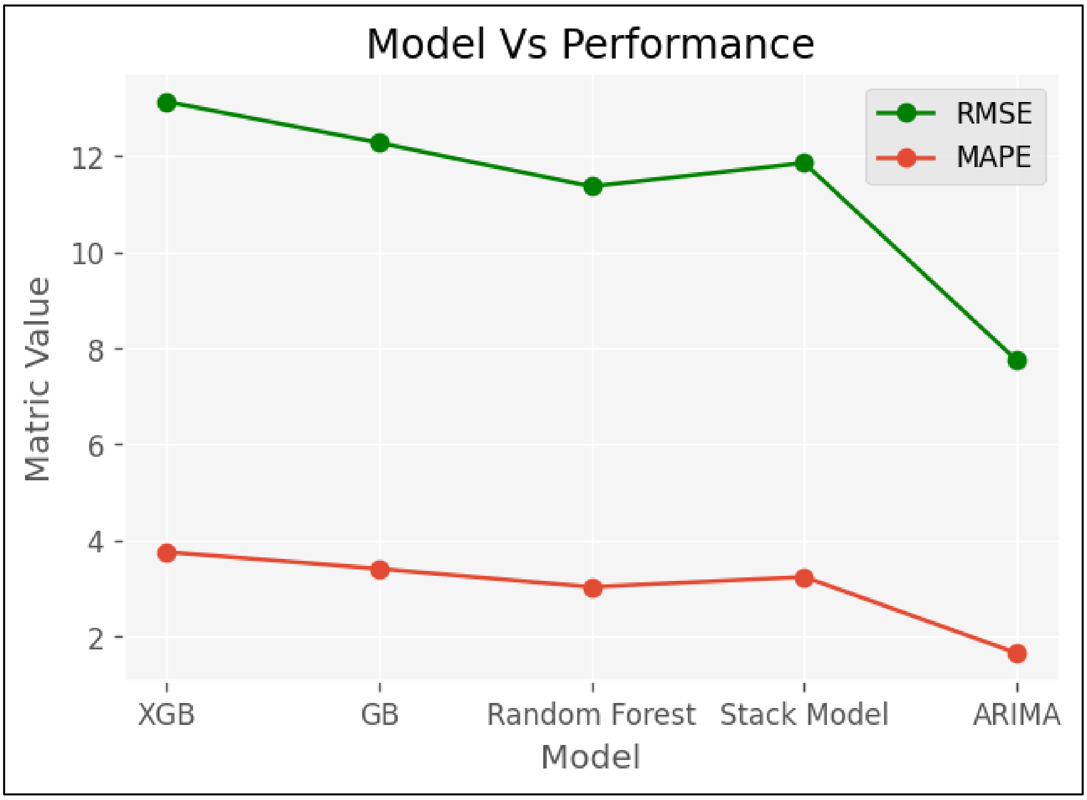


Figure 4.33:Model Vs Performance

Based on the values of the performance matrices as shown in Figure 4.33, it can be concluded that, the ARIMA model performs better than the other machine learning models. Therefore, the ARIMA model can be selected as the best model to predict the wholesale prices of carrot in the Pettah market during the inflation period.

4.2.9.9 Short term price forecasting for Pettah wholesale carrot price

ARIMA model was suggested as the best model for the Pettah wholesale carrot price prediction after the crisis effect. Using ARIMA model, prices were forecasted for the next 5 days. Since the actual data also currently available for the forecasting period, actual data was collected for the forecasting period as well. Both actual and forecasted prices for Pettah wholesale carrot after the crisis can be shown as in Table 4.12 as below.

Table 4.12: Actual prices Vs Forecasted prices.

Date (mm/dd/yyyy)	Actual Prices	Forecasted Prices
10/10/2022	300	285.51
10/11/2022	280	286.41
10/12/2022	300	287.30
10/13/2022	250	288.18
10/14/2022	300	289.06

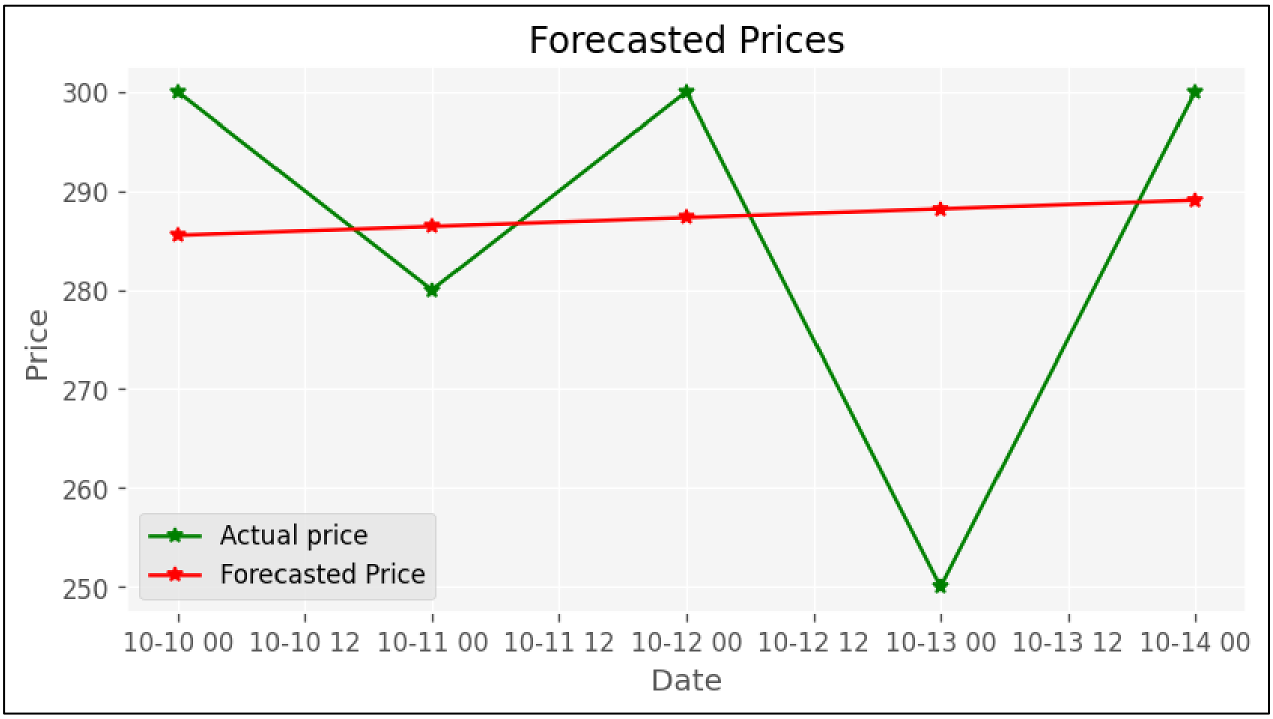


Figure 4.34: Graphical representation of the actual prices Vs forecasted prices

Based on the results shown in Figure 4.34, forecasted prices were slightly deviated from the actual prices. Among them, forecasted price of 4th day was more deviated with the actual prices than other days.

4.3 Prediction results of the rest of the vegetable prices

Out of the 16 price predictions (For wholesale and retail beans and carrot price predictions in Pettah and Dambulla markets before the crisis and after the crisis) were considered for the initial analysis, 4 price predictions were analysed and results were discussed during the previous stages. All the steps followed to forecast the bean prices before the crisis and after the crisis were attached in Appendices A and Appendices B respectively. Same steps were followed to forecast the rest of the prices as well.

Since the process followed to predict the prices is same for all 16 scenarios, prediction results of the rest of the 12 prices were summarised as shown in the Table 4.13 and Table 4.14.

Table 4.13 shows the evaluation results for each model applied for the prices before the crisis. For Pettah retail bean price prediction, stack model can be suggested as the best model over other models since it has higher R^2 score value of 77.87%, lower errors like RMSE of 31.24 and MAPE of 9.47%. XG Boost regression method was performed better than other models with R^2 score of 83.78%, RMSE of 15.37, MAPE of 6.5% for Pettah retail carrot prices. Therefore, XG Boost regression method can be proposed as the best model for the Pettah retail carrot price prediction.

Random forest model shows the R^2 score of 60.3%, RMSE of 35.64 and MAPE as 15.41% for Dambulla wholesale beans. Which presents the highest accuracy and lowest errors among other models. Therefore, random forest model can be suggested for the Dambulla wholesale beans price prediction. XGB model shows the highest accuracy as R^2 score of 57.8%, lowest errors as RMSE of 25.33 and MAPE of 13.51% over the other models for Dambulla wholesale carrot price prediction. Hence, XG Boost regression can be proposed for the Dambulla wholesale carrot price prediction.

Stack model presents the R^2 score of 66.44%, RMSE of 35.11 and MAPE as 12.88% for Dambulla retail beans. Which presents the highest accuracy and lowest errors among other models. Therefore, stack regression model can be suggested for the Dambulla retail beans price prediction. GB model shows the highest accuracy as R^2 score of 61.61%, lowest errors as RMSE of 24.69 and MAPE of 10.52% over the other models for Dambulla retail carrot price prediction. Hence, Gradient Boost regression can be proposed for the Dambulla retail carrot price prediction.

Table 4.13: Evaluation results for each model for rest of the prices before the crisis.

Price (Before the crisis)	Models Applied	MSE	RMSE	MAE	MAPE (%)	R^2 Score (%)
Pettah retail beans	XGB	1,054.85	32.47	20.02	9.08	76.09
	RF	1,058.81	32.53	20.83	9.58	76.01
	GB	1,051.53	32.42	20.34	9.22	76.17
	Stack model	976.41	31.24	20.43	9.47	77.87
Pettah retail carrot	XGB	236.46	15.37	10.37	6.50	83.78
	RF	268.47	16.38	10.98	6.96	81.59
	GB	250.35	15.82	10.48	6.60	82.83
	Stack model	246.10	15.68	10.60	6.64	83.12
Dambulla wholesale beans	XGB	1,489.62	38.59	27.20	16.04	53.47
	RF	1,270.87	35.64	25.33	15.41	60.30
	GB	1,407.16	37.51	26.38	15.90	56.05
	Stack model	1,321.92	36.36	25.56	15.52	58.71
Dambulla	XGB	641.94	25.33	17.05	13.51	57.80

wholesale carrot	RF	718.64	26.81	18.37	14.76	52.76
	GB	687.75	26.22	17.78	14.24	54.79
	Stack model	705.95	26.56	18.37	14.63	53.60
Dambulla retail beans	XGB	1,624.48	40.30	28.75	13.93	55.79
	RF	1,279.34	35.76	26.21	13.33	65.18
	GB	1,348.19	36.71	26.29	13.11	63.31
	Stack model	1,233.02	35.11	25.02	12.88	66.44
Dambulla retail carrot	XGB	615.64	24.81	15.70	10.41	61.24
	RF	666.89	25.82	16.66	11.07	58.02
	GB	609.77	24.69	15.98	10.52	61.61
	Stack model	697.72	26.41	17.64	11.80	56.07

Table 4.14 depicts the evaluation results for each model applied for the bean prices after the crisis. For Pettah retail bean price prediction, XGB model can be suggested as the best model over the other models since it has higher R^2 score value of 89.58%, lower errors like RMSE of 9.27 and MAPE of 1.88%. XG Boost regression method was performed better than other models with R^2 score of 89.94%, RMSE of 10.9, MAPE of 2.5% for Dambulla wholesale bean prices. Therefore, XG Boost regression method can be proposed as the best model for the Dambulla wholesale bean price prediction. XB Boost regression is suggested as the best model for Dambulla retail bean prices as well with the R^2 score of 91.11%, RMSE of 10.38, MAPE of 2.43%. Overall XG Boost model was the best model suggested for all retail and wholesale bean price predictions in both markets after the crisis.

Table 4.14: Evaluation results for each model for bean prices after the crisis.

Price (After the crisis)	**Models Applied**	**MSE**	**RMSE**	**MAE**	**MAPE (%)**	**R^2 Score (%)**
Pettah retail beans	XGB	85.88	9.27	7.83	1.88	89.58
	RF	135.28	1.63	9.49	2.28	83.59
	GB	98.92	9.94	8.32	2.00	88.00
	Stack model	651.39	25.52	21.25	5.41	20.99
Dambulla wholesale beans	XGB	118.91	10.90	8.39	2.50	89.94
	RF	154.87	12.44	9.82	3.02	86.89
	GB	202.94	14.24	10.79	3.27	82.83
	Stack model	650.99	25.51	21.55	6.49	44.93
Dambulla retail beans	XGB	107.92	10.38	8.69	2.43	91.11
	RF	155.20	12.45	9.89	2.79	87.21
	GB	166.59	12.90	10.33	2.90	86.28
	Stack model	433.12	20.81	18.43	5.03	64.33

Table 4.15 depicts the evaluation results for each model applied for the carrot prices after the crisis. For all carrot prices after the crisis, machine learning models did not provide accurate predictions. Hence machine learning models are not suitable to predict the retail and wholesale carrot prices in both markets after the crisis. Therefore, ARIMA model (traditional statistical technique) also applied for the carrot price predictions as well. ARIMA model was provided lower errors with better accuracy over the machine learning models for all the carrot price predictions after the crisis. Therefore, ARIMA model can be suggested as the best model for the retail and wholesale carrot price prediction in both markets after the crisis.

Table 4.15: Evaluation results for each model for carrot prices after the crisis.

Price (After the crisis)	**Models Applied**	**MSE**	**RMSE**	**MAE**	**MAPE (%)**
Pettah retail carrot	XGB	127.02	11.27	8.76	2.53
	RF	119.65	10.93	8.36	2.41
	GB	114.99	10.72	8.10	2.34
	Stack model	73.70	8.58	5.73	1.64
	ARIMA	53.81	7.33	4.40	1.26
Dambulla wholesale carrot	XGB	1,011.48	31.80	29.78	10.48
	RF	770.81	27.76	25.42	8.92
	GB	875.02	29.58	27.40	9.63
	Stack model	1,082.49	32.90	30.94	10.89
	ARIMA	390.94	19.77	17.49	6.12
Dambulla retail carrot	XGB	785.05	28.01	25.59	8.12
	RF	767.46	27.70	25.31	8.03
	GB	721.94	26.86	24.40	7.74
	Stack model	683.97	26.15	23.62	7.49
	ARIMA	372.68	19.30	17.09	5.41

Based on the results discussed during the previous stages, best performing model for each vegetable price prediction before the crisis and after the crisis can be concluded as given in the Table 4.16.

Table 4.16: Proposed Best Performing Model for Each Vegetable.

Before the crisis and after the crisis effect	**Vegetable Price**	**Best Performing Model**
Before crisis	Pettah Wholesale Beans	Stack Regression
	Pettah Wholesale Carrot	XG Boost Regression
	Pettah Retail Beans	Stack Regression
	Pettah Retail Carrot	XG Boost Regression
	Dambulla Wholesale Beans	Random Forest Regression
	Dambulla Wholesale Carrot	XG Boost Regression
	Dambulla Retail Beans	Stack Regression
	Dambulla Retail Carrot	Gradient Boost Regression
After crisis	Pettah Wholesale Beans	XG Boost Regression
	Pettah Wholesale Carrot	ARIMA Model
	Pettah Retail Beans	XG Boost Regression
	Pettah Retail Carrot	ARIMA Model
	Dambulla Wholesale Beans	XG Boost Regression
	Dambulla Wholesale Carrot	ARIMA Model

	Dambulla Retail Beans	XG Boost Regression
	Dambulla Retail Carrot	ARIMA Model

Based on the selected best performing models, short term price forecasting was conducted for each vegetable for the next 5 days. Prices were forecasted from 1st of December 2021 - 7th of December 2021 (business days only) before the crisis as shown in Table 4.17. Actual prices also collected during this period in order to validate the forecasted data points.

Table 4.17: Actual prices Vs forecasted prices for each vegetable before the crisis.

Vegetable Prices before the crisis	**Date (mm/dd/yyyy)**	**Actual Prices (per kg)**	**Forecasted Prices (per kg)**
Pettah retail beans	12/01/2021	300	307.09
	12/02/2021	325	244.40
	12/03/2021	350	308.18
	12/06/2021	300	245.93
	12/07/2021	350	305.10
Pettah retail carrot	12/01/2021	300	286.88
	12/02/2021	300	185.56
	12/03/2021	300	290.16
	12/06/2021	300	182.29
	12/07/2021	300	290.16
Dambulla wholesale beans	12/01/2021	215	193.90
	12/02/2021	235	160.85
	12/03/2021	255	193.55
	12/06/2021	295	161.22
	12/07/2021	278	193.68
Dambulla wholesale carrot	12/01/2021	200	224.52
	12/02/2021	195	111.50
	12/03/2021	225	224.01
	12/06/2021	238	117.21
	12/07/2021	278	224.01
Dambulla retail beans	12/01/2021	235	209.20
	12/02/2021	255	196.99
	12/03/2021	275	208.80
	12/06/2021	315	196.49
	12/07/2021	298	208.16
Dambulla retail carrot	12/01/2021	220	243.06
	12/02/2021	215	129.05
	12/03/2021	245	241.53
	12/06/2021	258	135.54
	12/07/2021	298	242.25

Prices were forecasted from 10th of October 2022 - 14th of October 2022 (business days only) after the crisis as shown in Table 4.18. Actual prices were collected during this period in order to validate the forecasted prices.

Table 4.18: Actual prices Vs forecasted prices for each vegetable after the crisis.

Vegetable Prices after the crisis	Date (mm/dd/yyyy)	Actual Prices (per kg)	Forecasted Prices (per kg)
Pettah retail beans	10/10/2022	400	384.37
	10/11/2022	350	401.25
	10/12/2022	400	384.37
	10/13/2022	300	401.26
	10/14/2022	350	381.88
Pettah retail carrot	10/10/2022	350	334.80
	10/11/2022	320	335.76
	10/12/2022	350	336.46
	10/13/2022	300	336.97
	10/14/2022	350	337.34
Dambulla wholesale beans	10/10/2022	245	293.59
	10/11/2022	280	320.13
	10/12/2022	225	293.59
	10/13/2022	275	320.13
	10/14/2022	250	293.59
Dambulla wholesale carrot	10/10/2022	270	258.23
	10/11/2022	290	258.88
	10/12/2022	270	259.47
	10/13/2022	225	260.02
	10/14/2022	215	260.52
Dambulla retail beans	10/10/2022	275	329.77
	10/11/2022	310	348.73
	10/12/2022	255	329.77
	10/13/2022	305	348.73
	10/14/2022	280	329.77
Dambulla retail carrot	10/10/2022	300	288.04
	10/11/2022	320	288.73
	10/12/2022	300	289.37
	10/13/2022	255	289.97
	10/14/2022	245	290.52

4.4 Summary

This chapter was explained the results generated using the proposed methods. Then results were evaluated using the evaluation matrices and the best performing model was selected for each vegetable price prediction. Short term price forecasting was implemented for each vegetable using the selected model.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

A definitive purpose of this study was to distinguish the best forecasting model from the utilized machine learning time series models which can be used to forecast the future vegetable prices in Sri Lanka.

This study was presented several machine learning models such as XG Boost Regression, Gradient Boost Regression, Random Forest Regression and Stack Regression for vegetable price prediction in Sri Lankan context. ARIMA model was applied for the carrot price prediction after the crisis since machine learning models were not fitted to that data set.

The results obtained in this study was shown that XG Bost regression and the stack regression models were mostly selected as the best model for the price predictions before the economic crisis. Among them XG boost model was proposed for both the markets such that Pettah and Dambulla for wholesale carrot price prediction before the crisis. But two different models such as XG Boost, and Gradient Boost models were suggested for retail carrot price prediction in both the markets before the crisis. For wholesale beans price prediction, stack model was suggested for the Pettah market. But Random Forest model suggested for the Dambulla market. Stack model was proposed for both the markets for retail beans price prediction before the crisis. Stack regression was the best performing model suggested for the wholesale and retail bean price prediction in the Pettah market. XG Boost regression was the best performing model suggested for the wholesale and retail carrot price prediction in the Pettah market. But two different models such as Random Forest model and Stack model were proposed to predict the wholesale and retail bean prices in Dambulla market. To predict the wholesale and retail carrot prices in the Dambulla market, XG Boost, and Gradient Boost methods were suggested.

Based on the results, study was also shown that XG Boost is the best model suggested for the wholesale and retail beans price prediction for both markets Pettah and Dambulla after the crisis. ARIMA was the best model suggested for the retail and wholesale carrot price prediction in both the markets after the economic crisis. Where machine learning models were not performed well for the carrot prices data after the crisis. Based on the results, this study has proven that best performing model can be varied from vegetable wise, marketwise, wholesale to retail and before the economic crisis effect to after the economic crisis effect as well.

Once the model was finalized for each vegetable price, short term price forecasting was implemented. Based on the forecasted prices, less deviation (with the actual data) was presented for the Pettah retail bean prices which were forecasted using the stack regression model over the other prices before the crisis. Slightly higher deviation was presented for the Dambulla retail bean prices which were forecasted using the same model (stack regression model) before the crisis. Forecasted prices of Pettah retail carrot were shown that less deviated results compared to the actual data over the other prices which were forecasted using ARIMA model after the economic crisis. Slightly higher deviation was presented for Dambulla retail bean prices which were forecasted using XG Boost model over the other prices after the crisis. Although the best performing model was used for the forecasting for the vegetable prices, sometimes forecasted results may get slightly deviated with the actual prices due to variability in macroeconomics.

Results were also stated that, Dambulla market having less wholesale and retail prices than Pettah market for all the selected vegetables. Since, there was a considerable amount of wholesale price difference between Dambulla and Pettah markets, third party have earned more profits than the farmers and the sellers during that period. Furthermore, an amount going for the third party, wholesale and retail vegetable price differences and market level price differences can also be calculated and then the obtained information can be used to get more insights about the vegetable prices.

5.2 Limitations, challenges and Future Work

5.2.1 Limitations and challenges

Accommodating sudden changes affected for the vegetable prices during this analysis was challenging. There was a huge impact on the vegetable prices data and prices were increased in larger amount due to the economic crisis. Initially, when we start this research, there was no such impact on the vegetable prices therefore one-time univariate time series forecasting was expected to do for the vegetable price prediction. But due to this inflation, we had to take that crisis effect into the consideration of our study as the separated task. This research would provide more accurate predictions for the after-crisis vegetable price predictions if the data set contained more records than it had.

5.2.2 Future Work

In this study, only the machine learning time series models were utilized. ARIMA was used for carrot price prediction after the crisis since machine learning models did not perform well for that data set. However, many other techniques can be used to predict the vegetable prices, such as neural network and deep learning methods, etc. An examination of such practices to recognize methods for the more precise forecast of vegetable prices might be helpful. This study was mainly focused on univariate vegetable price forecasting; however, multivariate forecast using other dependent variables would give a more accurate forecast of vegetable prices may be helpful. In this study, vegetable prices were predicted for different markets for before the crisis and after the crisis separately. This study can be further extended to combine both the effects together to perform one price prediction for each vegetable as well. Price forecasting was implemented for each vegetable separately for this analysis, we can think of having one single forecasting approach to apply all vegetables as a group.

5.3 Summary

This chapter was focused about the conclusion of the entire study, limitations, challenges and the future enhancements can be applied for the study.

REFERENCES

Amarasinghe, K., Marino, D.L., Manic, M., 2017. Deep neural networks for energy load forecasting, Presented at the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), IEEE, Edinburgh, United Kingdom, pp. 1483–1488. https://doi.org/10.1109/ISIE.2017.8001465.

Assis Kamu, Amran Ahmed, Remali Yusoff, 2008. Univariate Time Series Model for forecasting of Tawau Cocoa Bean Price.

Cyril Bogahawatte, 1988. Seasonal variations in retail and wholesale prices of rice in Colombo Markets, Sri Lanka.

Daily Price Report, Central Bank of Sri Lanka [WWW Document], n.d. URL https://www.cbsl.gov.lk/en/statistics/economic-indicators/price-report [accessed 29 April 2022].

Dieng, D.A., 2018. Alternative Forecasting Techniques for Vegetable Prices in Senegal 6.

Jadhav, V., Reddy, B.V.C., Gaddi, G.M., 2017. Application of ARIMA Model for Forecasting Agricultural Prices 12.

Jain, A., Marvaniya, S., Godbole, S., Munigala, V., 2020. A Framework for Crop Price Forecasting in Emerging Economies by Analyzing the Quality of Time-series Data. ArXiv200904171 Econ Stat.

Lavanya, K., Raguchander, T., 2013. SVM Regression and SONN based approach for seasonal crop price prediction. Sci. Technol. 6, 14.

Li, Y., Li, C., Zheng, M., 2014. A Hybrid Neural Network and H-P Filter Model for Short-Term Vegetable Price Forecasting. Math. Probl. Eng. 2014, 1–10. https://doi.org/10.1155/2014/135862.

Lu, Y., Zhang, J., 2004. Forecasting Stock Price by SVMs Regression.

Luo, C., Wei, Q., Zhou, L., Zhang, J., Sun, S., 2011. Prediction of Vegetable Price Based on Neural Network and Genetic Algorithm, in: Li, D., Liu, Y., Chen, Y. (Eds.), Computer and Computing Technologies in Agriculture IV, IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 672– 681. https://doi.org/10.1007/978-3-642-18354-6_79.

Mulla, S.A., Quadri, D.S.A., 2012. Crop-yield and Price Forecasting using Machine Learning.

Nasira, G M, Hemageetha, N., 2012. Forecasting Model for Vegetable Price Using Back Propagation Neural Network 2, 6.

Nasira, G. M., Hemageetha, N., 2012. Vegetable price prediction using data mining classification technique, in: International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012). Presented at the 2012 International Conference on Pattern

Recognition, Informatics and Medical Engineering (PRIME), IEEE, Salem, Tamilnadu, India, pp. 99–102. https://doi.org/10.1109/ICPRIME.2012.6208294.

Rachana P S, Rashmi G, Shravani D, Shruthi N, Seema Kousar R, 2019. Crop price forecasting system using supervised machine learning algorithms.

Rakhra, M., 2020 Crop Price Prediction Using Random Forest and Decision Tree Regression: -A Review. Mater. Today 5.

Samuel, P., 2020. Crop Price Prediction System using Machine learning Algorithms.

Shukla, M., Jharkharia, S., 2011. Applicability of ARIMA Models in Wholesale Vegetable Market: An Investigation 6.

Sri Lanka - Agricultural Sector [WWW Document], n.d. URL https://www.trade.gov/country-commercial-guides/sri-lanka-agricultural-sector [Accessed 29 April 2022].

Subhasree, M., Priya, M.C.A., 2016. Forecasting vegetable price using time series data. Int. J. Adv. Res. 3, 7.

Varun R, Neema N, Sahana, 2019. Agriculture Commodity Price Forecasting using Ml Techniques. Int. J. Innov. Technol. Explore. Eng. 9, 729–732. https://doi.org/10.35940/ijitee.B1226.1292S19.

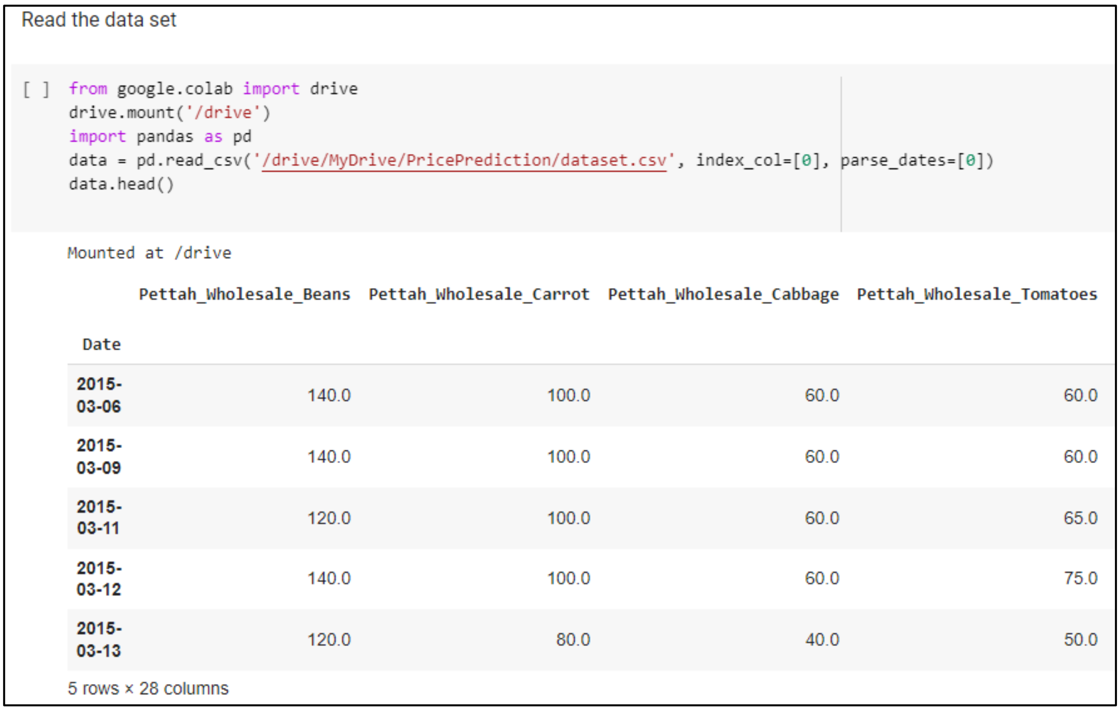
Yin, H., Jin, D., Gu, Y.H., Park, C.J., Han, S.K., Yoo, S.J., 2020. STL-ATTLSTM: Vegetable Price Forecasting Using STL and Attention Mechanism-Based LSTM.
Agriculture 10, 612. https://doi.org/10.3390/agriculture10120612.

Yoo, D., 2016. Vegetable Price Prediction Using A Typical Web-Search Data.

APPENDICES

Appendices A

Source codes and results generated for Pettah wholesale bean price prediction before the economic crisis using machine learning techniques.



Read the data set

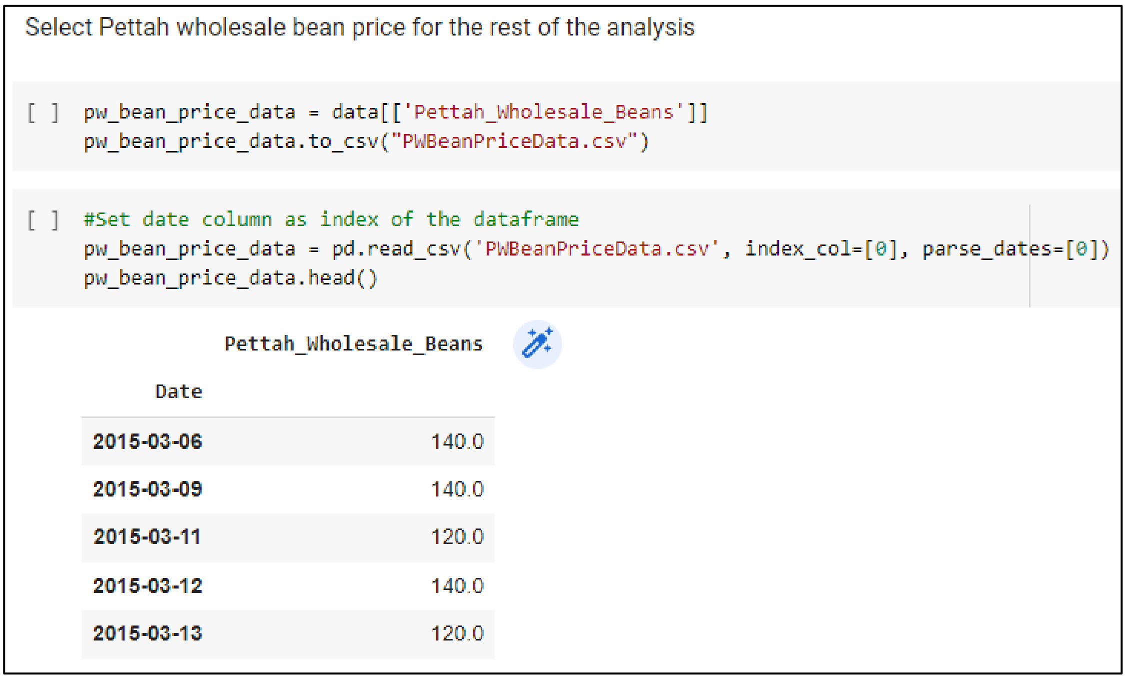
```
[ ] from google.colab import drive
    drive.mount('/drive')
    import pandas as pd
    data = pd.read_csv('/drive/MyDrive/PricePrediction/dataset.csv', index_col=[0], parse_dates=[0])
    data.head()
```

```
Mounted at /drive
```

Date	Pettah_Wholesale_Beans	Pettah_Wholesale_Carrot	Pettah_Wholesale_Cabbage	Pettah_Wholesale_Tomatoes
2015-03-06	140.0	100.0	60.0	60.0
2015-03-09	140.0	100.0	60.0	60.0
2015-03-11	120.0	100.0	60.0	65.0
2015-03-12	140.0	100.0	60.0	75.0
2015-03-13	120.0	80.0	40.0	50.0

5 rows × 28 columns

Appendix A.1: Read the data set



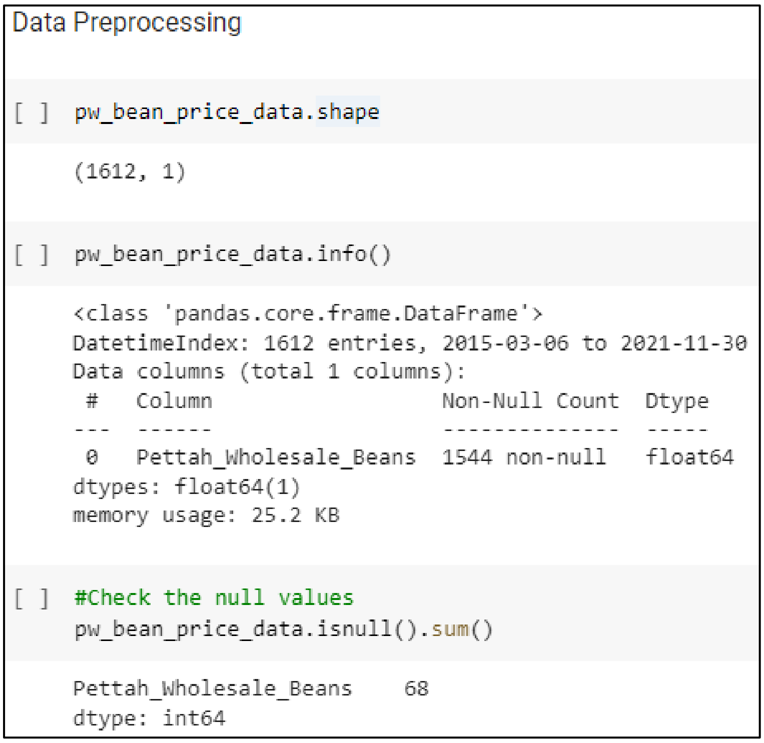
Select Pettah wholesale bean price for the rest of the analysis

```
[ ] pw_bean_price_data = data[['Pettah_Wholesale_Beans']]
    pw_bean_price_data.to_csv("PWBeanPriceData.csv")
```

```
[ ] #Set date column as index of the dataframe
    pw_bean_price_data = pd.read_csv('PWBeanPriceData.csv', index_col=[0], parse_dates=[0])
    pw_bean_price_data.head()
```

Date	Pettah_Wholesale_Beans
2015-03-06	140.0
2015-03-09	140.0
2015-03-11	120.0
2015-03-12	140.0
2015-03-13	120.0

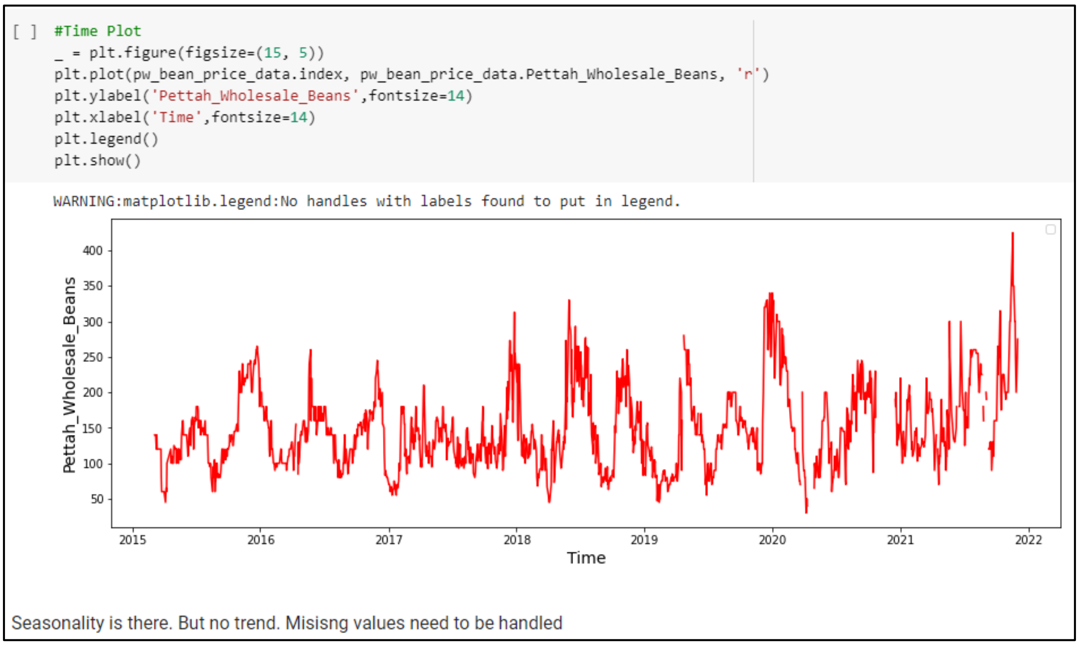
Appendix A.2: Select Pettah wholesale bean price



Data Preprocessing

```
[ ]  pw_bean_price_data.shape

     (1612, 1)

[ ]  pw_bean_price_data.info()

     <class 'pandas.core.frame.DataFrame'>
     DatetimeIndex: 1612 entries, 2015-03-06 to 2021-11-30
     Data columns (total 1 columns):
      #   Column                   Non-Null Count  Dtype  
     ---  ------                   --------------  -----  
      0   Pettah_Wholesale_Beans  1544 non-null   float64
     dtypes: float64(1)
     memory usage: 25.2 KB

[ ]  #Check the null values
     pw_bean_price_data.isnull().sum()

     Pettah_Wholesale_Beans    68
     dtype: int64
```

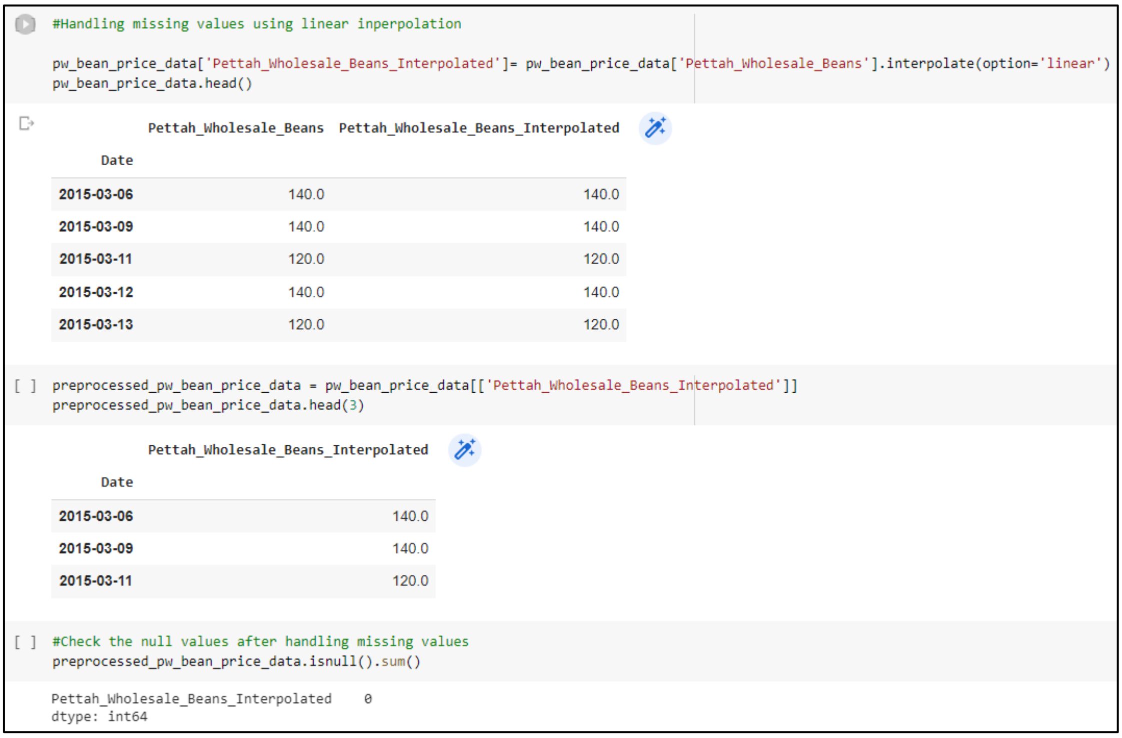
Appendix A.3: Pre-processing time series data



```
[ ]  #Time Plot
     _ = plt.figure(figsize=(15, 5))
     plt.plot(pw_bean_price_data.index, pw_bean_price_data.Pettah_Wholesale_Beans, 'r')
     plt.ylabel('Pettah_Wholesale_Beans',fontsize=14)
     plt.xlabel('Time',fontsize=14)
     plt.legend()
     plt.show()

     WARNING:matplotlib.legend:No handles with labels found to put in legend.
```

Seasonality is there. But no trend. Misisng values need to be handled

Appendix A.4: Time Plot for Pettah wholesale beans prices before crisis



```
#Handling missing values using linear inperpolation

pw_bean_price_data['Pettah_Wholesale_Beans_Interpolated']= pw_bean_price_data['Pettah_Wholesale_Beans'].interpolate(option='linear')
pw_bean_price_data.head()
```

	Pettah_Wholesale_Beans	Pettah_Wholesale_Beans_Interpolated
Date		
2015-03-06	140.0	140.0
2015-03-09	140.0	140.0
2015-03-11	120.0	120.0
2015-03-12	140.0	140.0
2015-03-13	120.0	120.0

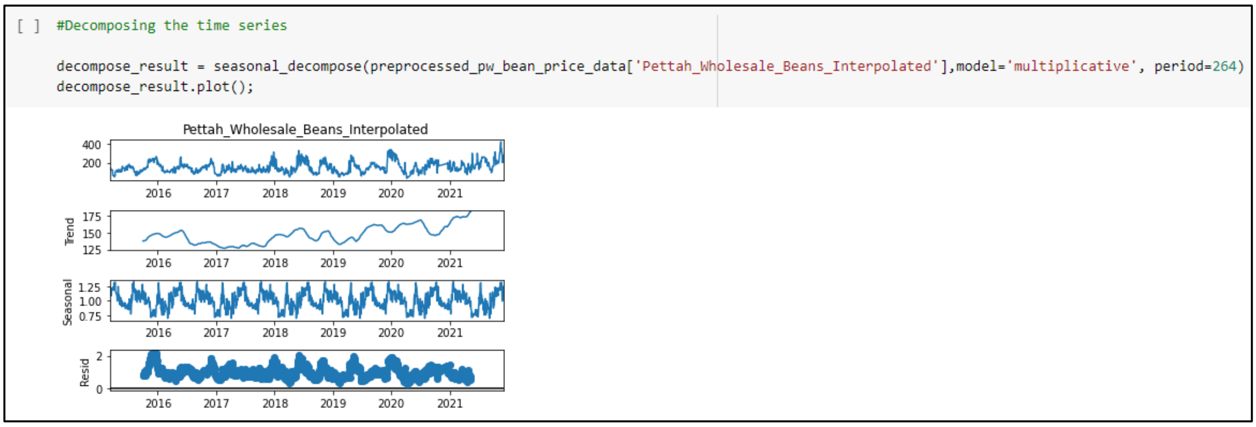
```
[ ] preprocessed_pw_bean_price_data = pw_bean_price_data[['Pettah_Wholesale_Beans_Interpolated']]
    preprocessed_pw_bean_price_data.head(3)
```

	Pettah_Wholesale_Beans_Interpolated
Date	
2015-03-06	140.0
2015-03-09	140.0
2015-03-11	120.0

```
[ ] #Check the null values after handling missing values
    preprocessed_pw_bean_price_data.isnull().sum()
```

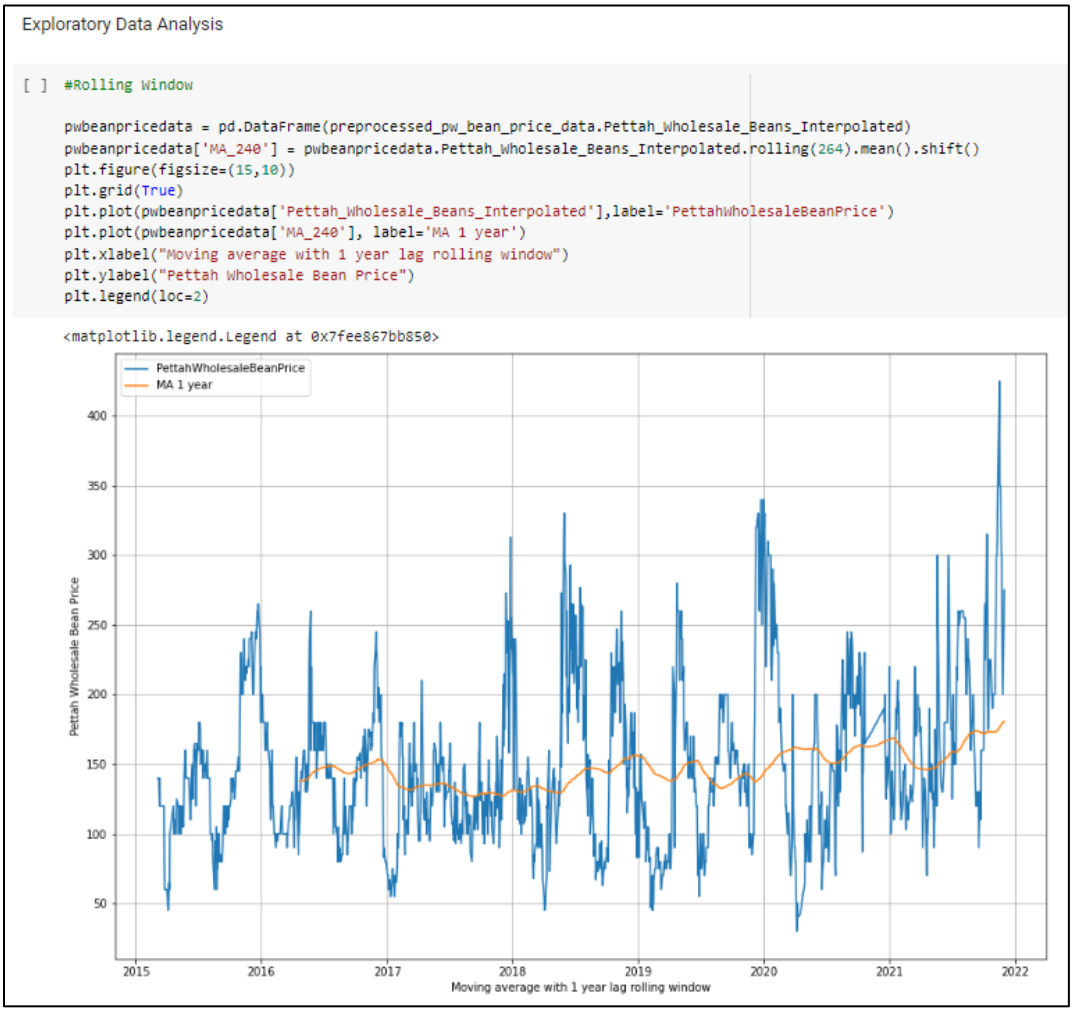
```
Pettah_Wholesale_Beans_Interpolated    0
dtype: int64
```

Appendix A.5: Handling missing values for Pettah wholesale beans prices before crisis



```
[ ] #Decomposing the time series

    decompose_result = seasonal_decompose(preprocessed_pw_bean_price_data['Pettah_Wholesale_Beans_Interpolated'],model='multiplicative', period=264)
    decompose_result.plot();
```

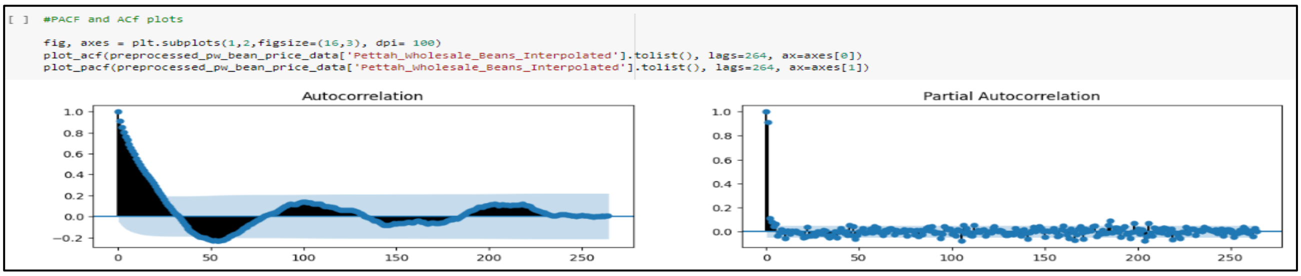
Appendix A.6: Decomposing time series for Pettah wholesale beans prices before crisis



Exploratory Data Analysis

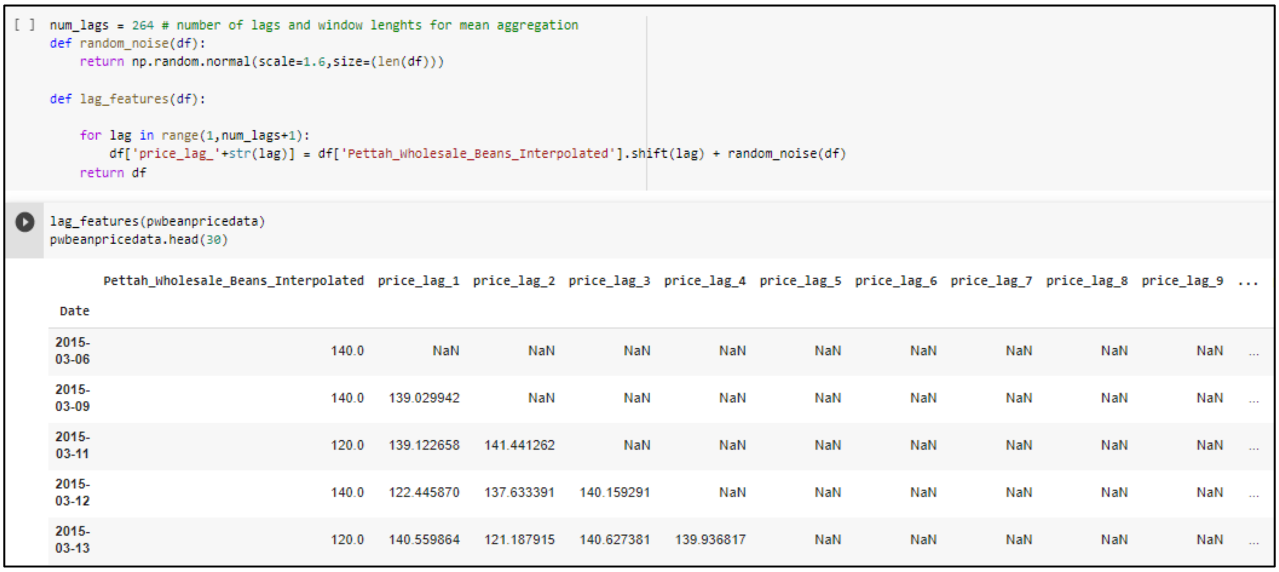
```
[ ]  #Rolling Window

     pwbeanpricedata = pd.DataFrame(preprocessed_pw_bean_price_data.Pettah_Wholesale_Beans_Interpolated)
     pwbeanpricedata['MA_240'] = pwbeanpricedata.Pettah_Wholesale_Beans_Interpolated.rolling(264).mean().shift()
     plt.figure(figsize=(15,10))
     plt.grid(True)
     plt.plot(pwbeanpricedata['Pettah_Wholesale_Beans_Interpolated'],label='PettahWholesaleBeanPrice')
     plt.plot(pwbeanpricedata['MA_240'], label='MA 1 year')
     plt.xlabel("Moving average with 1 year lag rolling window")
     plt.ylabel("Pettah Wholesale Bean Price")
     plt.legend(loc=2)

<matplotlib.legend.Legend at 0x7fee867bb850>
```

Appendix A.7: Rolling Window for Pettah wholesale beans prices before crisis



```
[ ]  #PACF and ACf plots

     fig, axes = plt.subplots(1,2,figsize=(16,3), dpi= 100)
     plot_acf(preprocessed_pw_bean_price_data['Pettah_Wholesale_Beans_Interpolated'].tolist(), lags=264, ax=axes[0])
     plot_pacf(preprocessed_pw_bean_price_data['Pettah_Wholesale_Beans_Interpolated'].tolist(), lags=264, ax=axes[1])
```

Appendix A.8: ACF and PACF Plots for Pettah wholesale beans prices before crisis

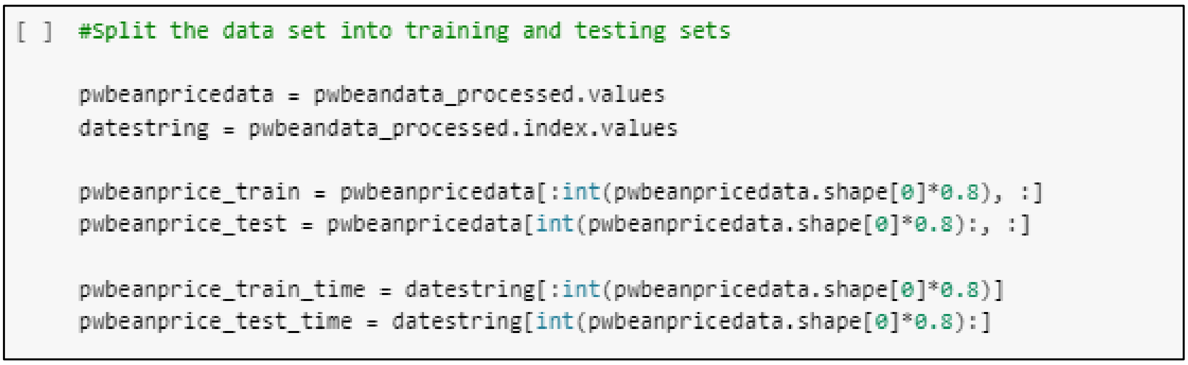


```
[ ] num_lags = 264 # number of lags and window lenghts for mean aggregation
    def random_noise(df):
        return np.random.normal(scale=1.6,size=(len(df)))

    def lag_features(df):

        for lag in range(1,num_lags+1):
            df['price_lag_'+str(lag)] = df['Pettah_Wholesale_Beans_Interpolated'].shift(lag) + random_noise(df)
        return df
```

```
lag_features(pwbeanpricedata)
pwbeanpricedata.head(30)
```

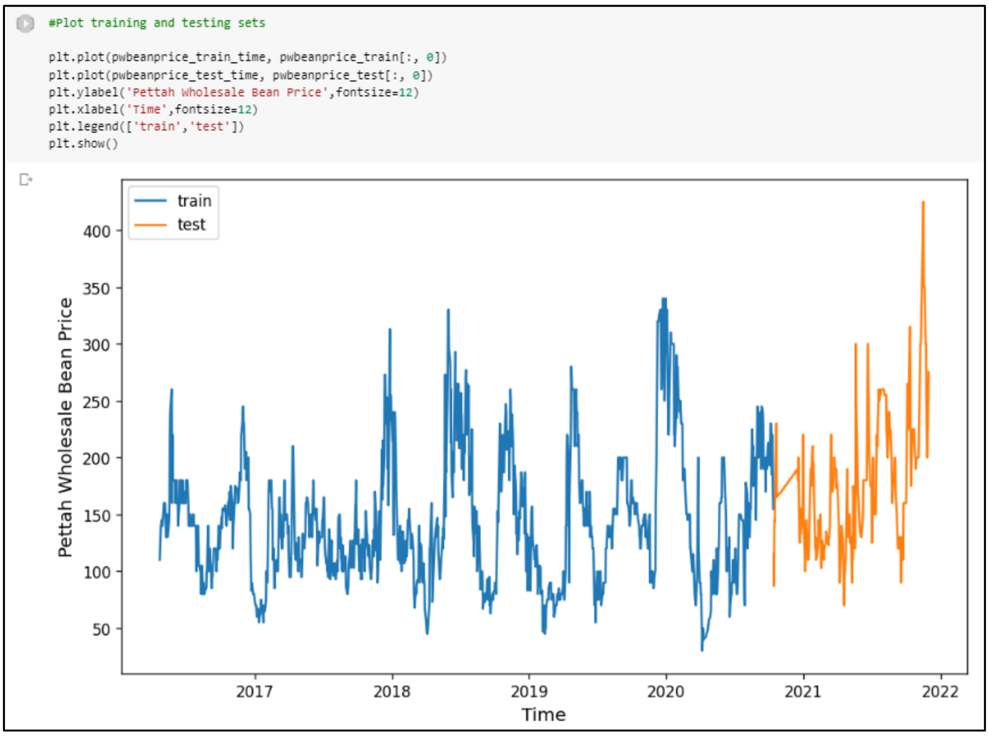
	Pettah_Wholesale_Beans_Interpolated	price_lag_1	price_lag_2	price_lag_3	price_lag_4	price_lag_5	price_lag_6	price_lag_7	price_lag_8	price_lag_9	...
Date											
2015-03-06	140.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
2015-03-09	140.0	139.029942	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
2015-03-11	120.0	139.122658	141.441262	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
2015-03-12	140.0	122.445870	137.633391	140.159291	NaN	NaN	NaN	NaN	NaN	NaN	...
2015-03-13	120.0	140.559864	121.187915	140.627381	139.936817	NaN	NaN	NaN	NaN	NaN	...

Appendix A.9: Create Lag Features Pettah wholesale beans prices before crisis



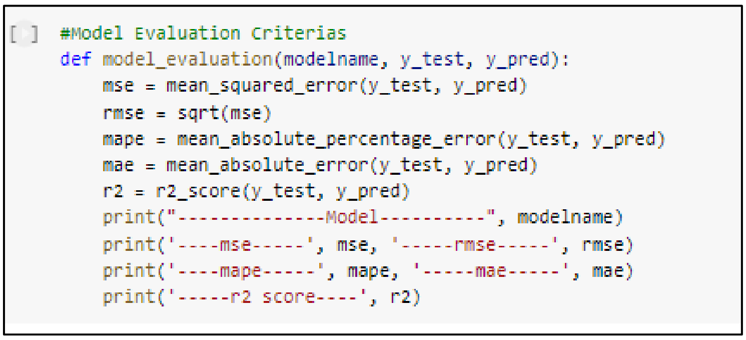
```
[ ] #Split the data set into training and testing sets

    pwbeanpricedata = pwbeandata_processed.values
    datestring = pwbeandata_processed.index.values

    pwbeanprice_train = pwbeanpricedata[:int(pwbeanpricedata.shape[0]*0.8), :]
    pwbeanprice_test = pwbeanpricedata[int(pwbeanpricedata.shape[0]*0.8):, :]

    pwbeanprice_train_time = datestring[:int(pwbeanpricedata.shape[0]*0.8)]
    pwbeanprice_test_time = datestring[int(pwbeanpricedata.shape[0]*0.8):]
```

Appendix A.10: Splitting the data set for Pettah wholesale beans prices before crisis



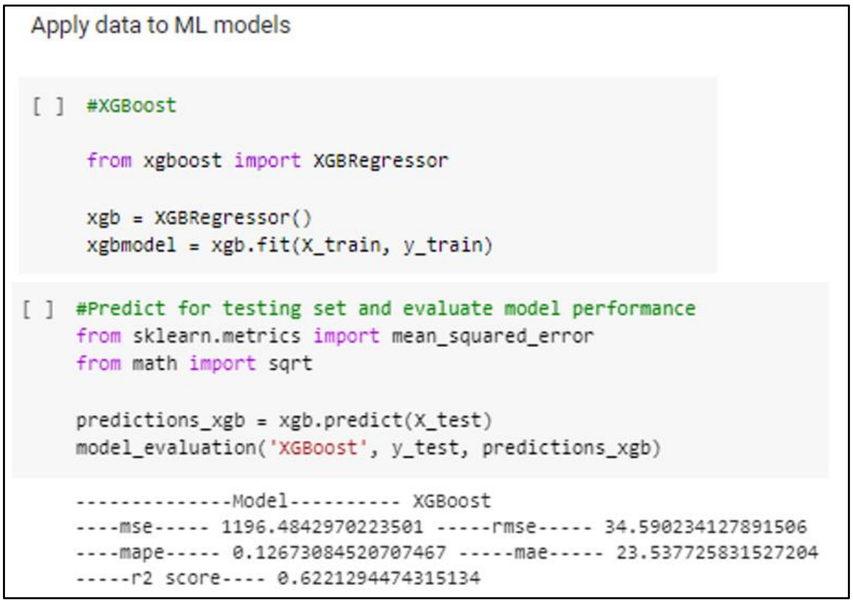
```
#Plot training and testing sets

plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0])
plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0])
plt.ylabel('Pettah Wholesale Bean Price',fontsize=12)
plt.xlabel('Time',fontsize=12)
plt.legend(['train','test'])
plt.show()
```

Appendix A.11: Training and testing sets plot



```
[ ]  #Model Evaluation Criterias
     def model_evaluation(modelname, y_test, y_pred):
         mse = mean_squared_error(y_test, y_pred)
         rmse = sqrt(mse)
         mape = mean_absolute_percentage_error(y_test, y_pred)
         mae = mean_absolute_error(y_test, y_pred)
         r2 = r2_score(y_test, y_pred)
         print("--------------Model----------", modelname)
         print('----mse-----', mse, '-----rmse-----', rmse)
         print('----mape-----', mape, '-----mae-----', mae)
         print('-----r2 score----', r2)
```

Appendix A.12: Evaluation matrices of machine learning regression models



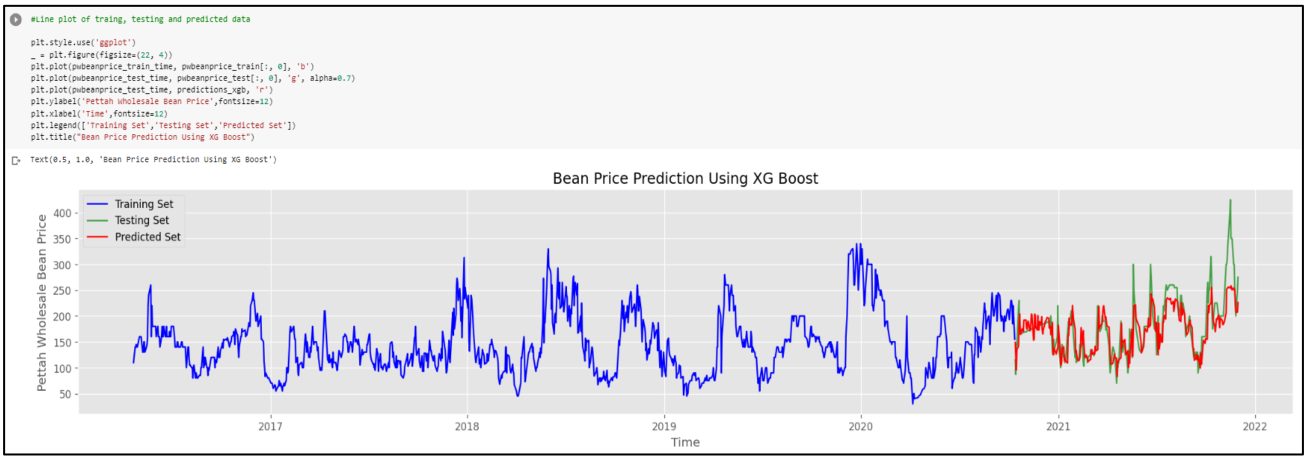
Apply data to ML models

```
[ ]  #XGBoost

     from xgboost import XGBRegressor

     xgb = XGBRegressor()
     xgbmodel = xgb.fit(X_train, y_train)
```

```
[ ]  #Predict for testing set and evaluate model performance
     from sklearn.metrics import mean_squared_error
     from math import sqrt

     predictions_xgb = xgb.predict(X_test)
     model_evaluation('XGBoost', y_test, predictions_xgb)
```

```
     --------------Model---------- XGBoost
     ----mse----- 1196.4842970223501 -----rmse----- 34.590234127891506
     ----mape----- 0.12673084520707467 -----mae----- 23.537725831527204
     -----r2 score---- 0.6221294474315134
```

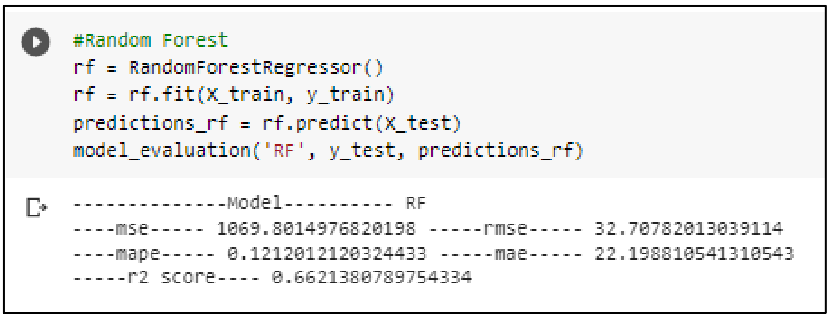
Appendix A.13: Apply XG Boost Model



```
#Line plot of traing, testing and predicted data

plt.style.use('ggplot')
_ = plt.figure(figsize=(22, 4))
plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0], 'b')
plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0], 'g', alpha=0.7)
plt.plot(pwbeanprice_test_time, predictions_xgb, 'r')
plt.ylabel('Pettah Wholesale Bean Price',fontsize=12)
plt.xlabel('Time',fontsize=12)
plt.legend(['Training Set','Testing Set','Predicted Set'])
plt.title("Bean Price Prediction Using XG Boost")
```

```
Text(0.5, 1.0, 'Bean Price Prediction Using XG Boost')
```

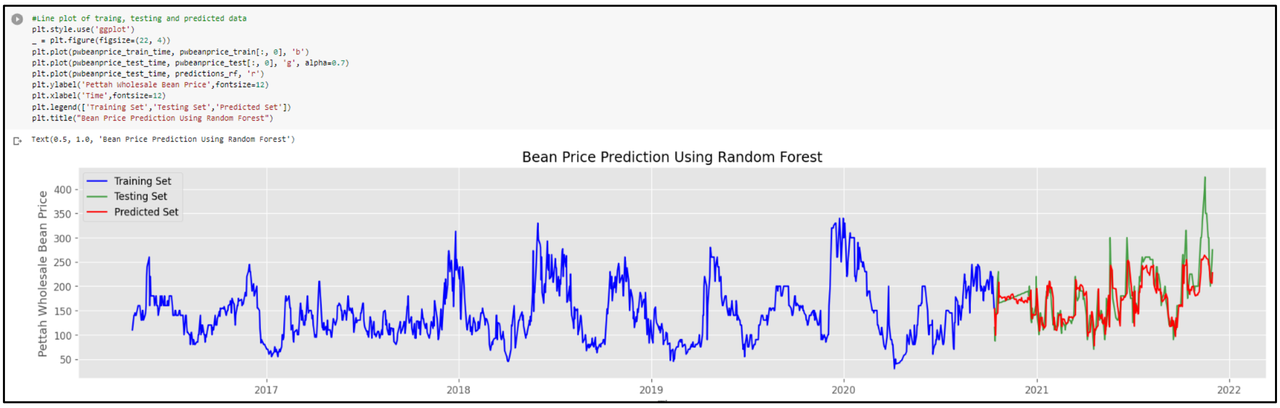
Appendix A.14: Graphical Representation of training set, testing set and the predicted set for XG boost



```
#Random Forest
rf = RandomForestRegressor()
rf = rf.fit(X_train, y_train)
predictions_rf = rf.predict(X_test)
model_evaluation('RF', y_test, predictions_rf)
```

```
--------------Model---------- RF
----mse----- 1069.8014976820198 -----rmse----- 32.70782013039114
----mape----- 0.1212012120324433 -----mae----- 22.198810541310543
-----r2 score---- 0.6621380789754334
```

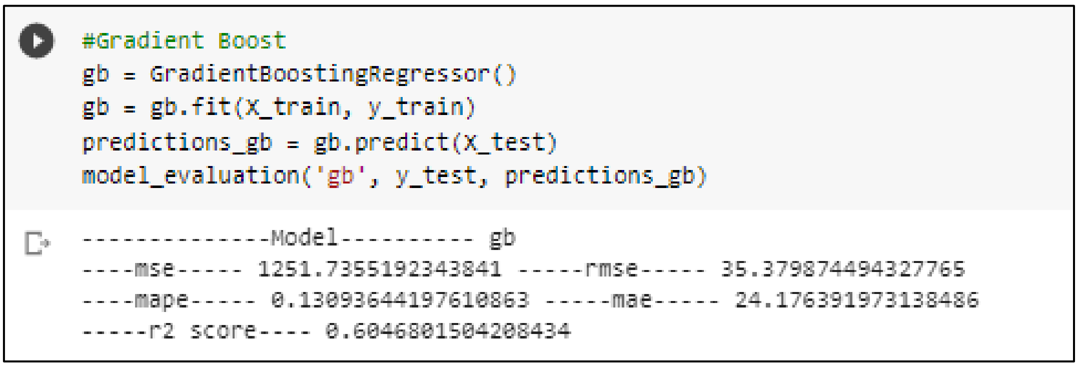
Appendix A.15: Apply Random Forest Model



```
#Line plot of traing, testing and predicted data
plt.style.use('ggplot')
_ = plt.figure(figsize=(22, 4))
plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0], 'b')
plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0], 'g', alpha=0.7)
plt.plot(pwbeanprice_test_time, predictions_rf, 'r')
plt.ylabel('Pettah Wholesale Bean Price',fontsize=12)
plt.xlabel('Time',fontsize=12)
plt.legend(['Training Set','Testing Set','Predicted Set'])
plt.title("Bean Price Prediction Using Random Forest")
```

```
Text(0.5, 1.0, 'Bean Price Prediction Using Random Forest')
```

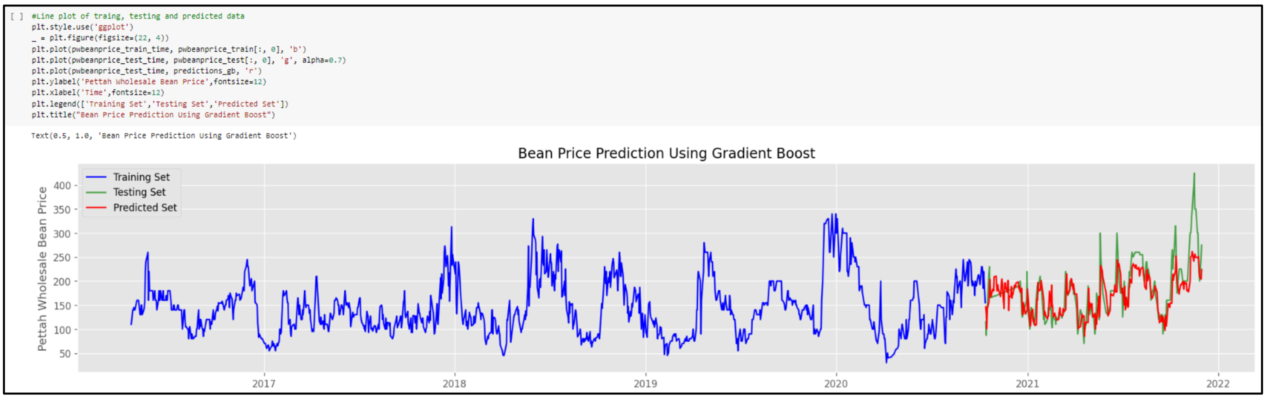
Appendix A.16: Graphical Representation of training set, testing set and the predicted set for Random Forest



```
#Gradient Boost
gb = GradientBoostingRegressor()
gb = gb.fit(X_train, y_train)
predictions_gb = gb.predict(X_test)
model_evaluation('gb', y_test, predictions_gb)
```

```
--------------Model---------- gb
----mse----- 1251.7355192343841 -----rmse----- 35.379874494327765
----mape----- 0.13093644197610863 -----mae----- 24.176391973138486
-----r2 score---- 0.6046801504208434
```

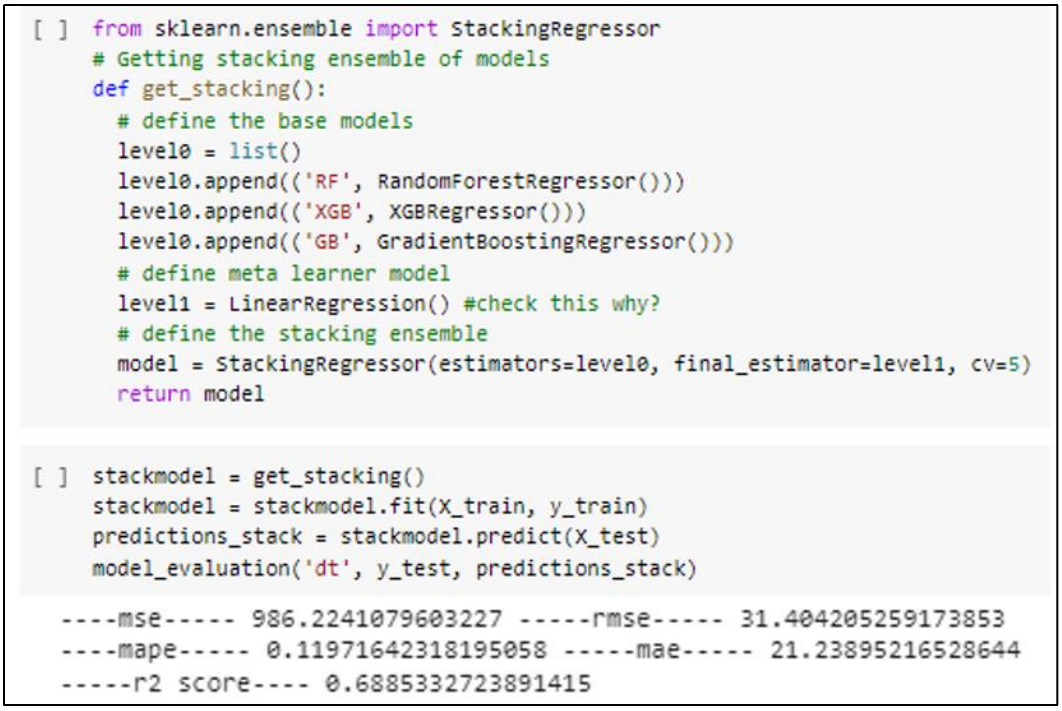
Appendix A.17: Apply Gradient Boost Model



```
#Line plot of traing, testing and predicted data
plt.style.use('ggplot')
_ = plt.figure(figsize=(22, 4))
plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0], 'b')
plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0], 'g', alpha=0.7)
plt.plot(pwbeanprice_test_time, predictions_gb, 'r')
plt.ylabel('Pettah Wholesale Bean Price',fontsize=12)
plt.xlabel('Time',fontsize=12)
plt.legend(['Training Set','Testing Set','Predicted Set'])
plt.title("Bean Price Prediction Using Gradient Boost")
```

```
Text(0.5, 1.0, 'Bean Price Prediction Using Gradient Boost')
```

Appendix A.18: Graphical Representation of training set, testing set and the predicted set for Gradient Boost

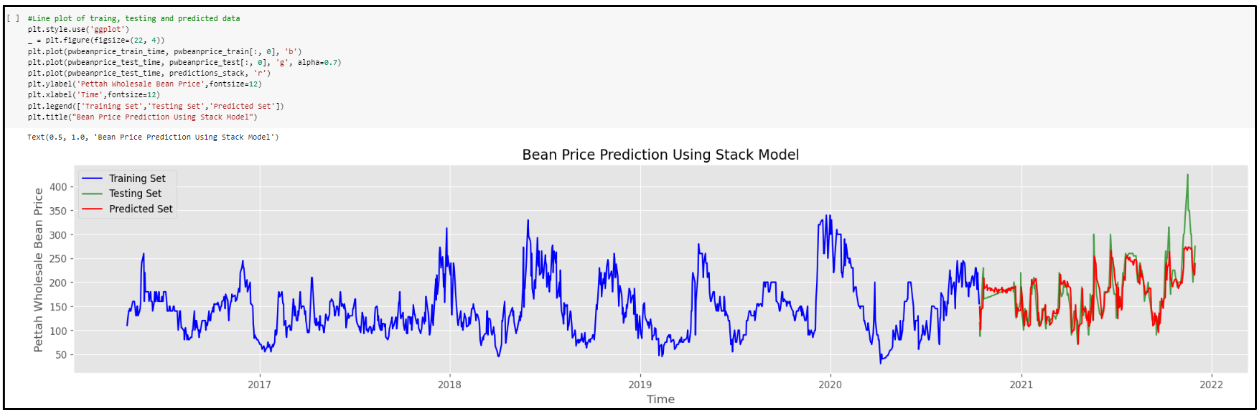


```
[ ] from sklearn.ensemble import StackingRegressor
    # Getting stacking ensemble of models
    def get_stacking():
      # define the base models
      level0 = list()
      level0.append(('RF', RandomForestRegressor()))
      level0.append(('XGB', XGBRegressor()))
      level0.append(('GB', GradientBoostingRegressor()))
      # define meta learner model
      level1 = LinearRegression() #check this why?
      # define the stacking ensemble
      model = StackingRegressor(estimators=level0, final_estimator=level1, cv=5)
      return model
```

```
[ ] stackmodel = get_stacking()
    stackmodel = stackmodel.fit(X_train, y_train)
    predictions_stack = stackmodel.predict(X_test)
    model_evaluation('dt', y_test, predictions_stack)
```

```
  ----mse----- 986.2241079603227 -----rmse----- 31.404205259173853
  ----mape----- 0.11971642318195058 -----mae----- 21.23895216528644
  -----r2 score---- 0.6885332723891415
```

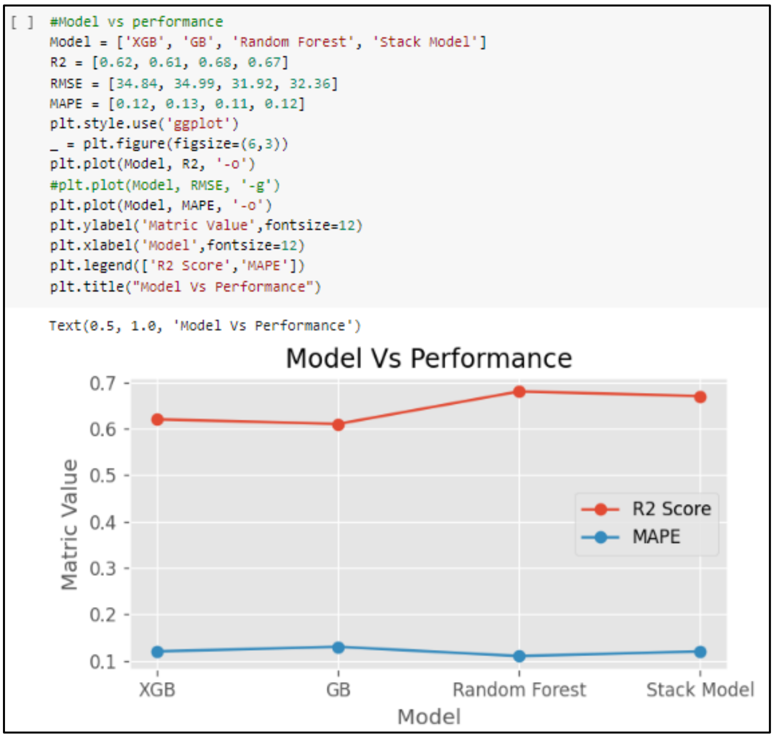
Appendix A.19: Apply stack regression model



```
[ ] #Line plot of traing, testing and predicted data
    plt.style.use('ggplot')
    _ = plt.figure(figsize=(22, 4))
    plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0], 'b')
    plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0], 'g', alpha=0.7)
    plt.plot(pwbeanprice_test_time, predictions_stack, 'r')
    plt.ylabel('Pettah Wholesale Bean Price',fontsize=12)
    plt.xlabel('Time',fontsize=12)
    plt.legend(['Training Set','Testing Set','Predicted Set'])
    plt.title("Bean Price Prediction Using Stack Model")
```

```
    Text(0.5, 1.0, 'Bean Price Prediction Using Stack Model')
```

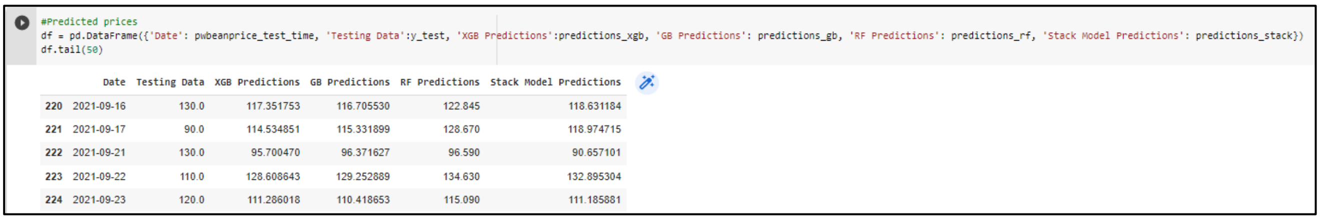
Appendix A.20: Graphical Representation of training set, testing set and the predicted set for stack model



```
#Model vs performance
Model = ['XGB', 'GB', 'Random Forest', 'Stack Model']
R2 = [0.62, 0.61, 0.68, 0.67]
RMSE = [34.84, 34.99, 31.92, 32.36]
MAPE = [0.12, 0.13, 0.11, 0.12]
plt.style.use('ggplot')
_ = plt.figure(figsize=(6,3))
plt.plot(Model, R2, '-o')
#plt.plot(Model, RMSE, '-g')
plt.plot(Model, MAPE, '-o')
plt.ylabel('Matric Value',fontsize=12)
plt.xlabel('Model',fontsize=12)
plt.legend(['R2 Score','MAPE'])
plt.title("Model Vs Performance")
```

```
Text(0.5, 1.0, 'Model Vs Performance')
```

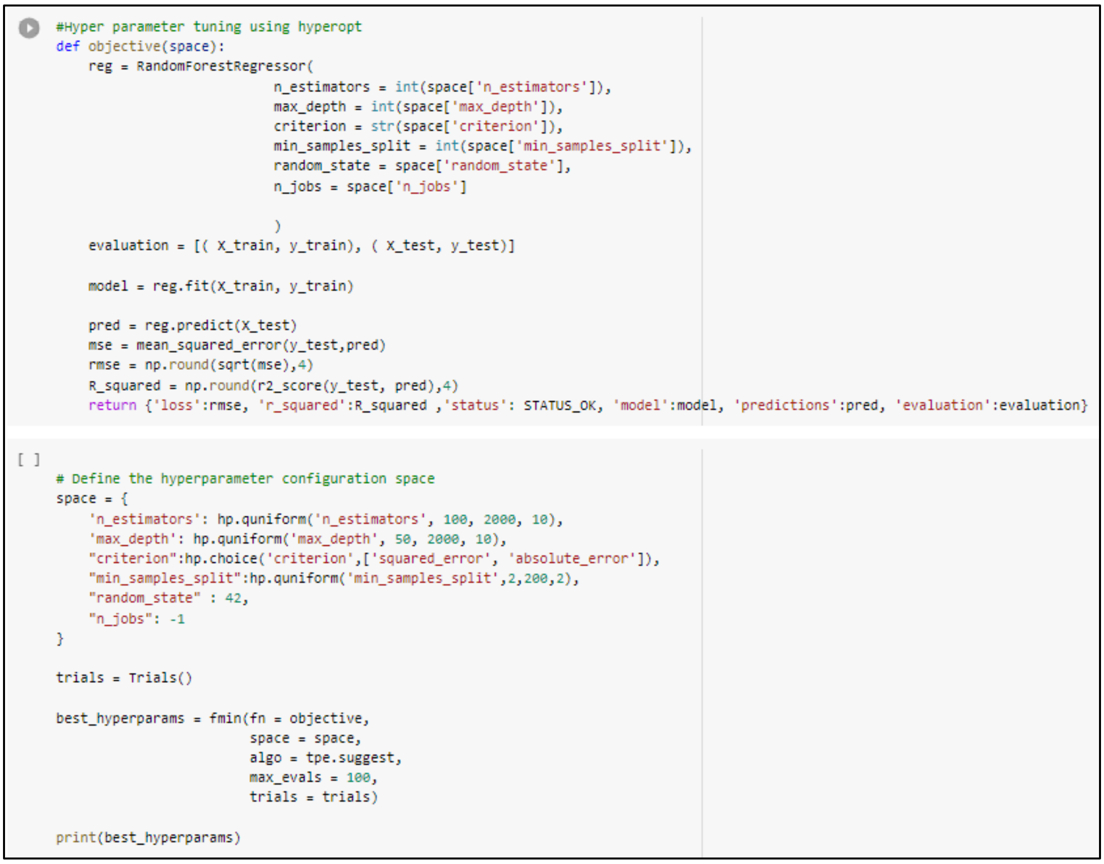
Appendix A.21: Model Vs Performance



```
#Predicted prices
df = pd.DataFrame({'Date': pwbeanprice_test_time, 'Testing Data':y_test, 'XGB Predictions':predictions_xgb, 'GB Predictions': predictions_gb, 'RF Predictions': predictions_rf, 'Stack Model Predictions': predictions_stack})
df.tail(50)
```

	Date	Testing Data	XGB Predictions	GB Predictions	RF Predictions	Stack Model Predictions
220	2021-09-16	130.0	117.351753	116.705530	122.845	118.631184
221	2021-09-17	90.0	114.534851	115.331899	128.670	118.974715
222	2021-09-21	130.0	95.700470	96.371627	96.590	90.657101
223	2021-09-22	110.0	128.608643	129.252889	134.630	132.895304
224	2021-09-23	120.0	111.286018	110.418653	115.090	111.185881

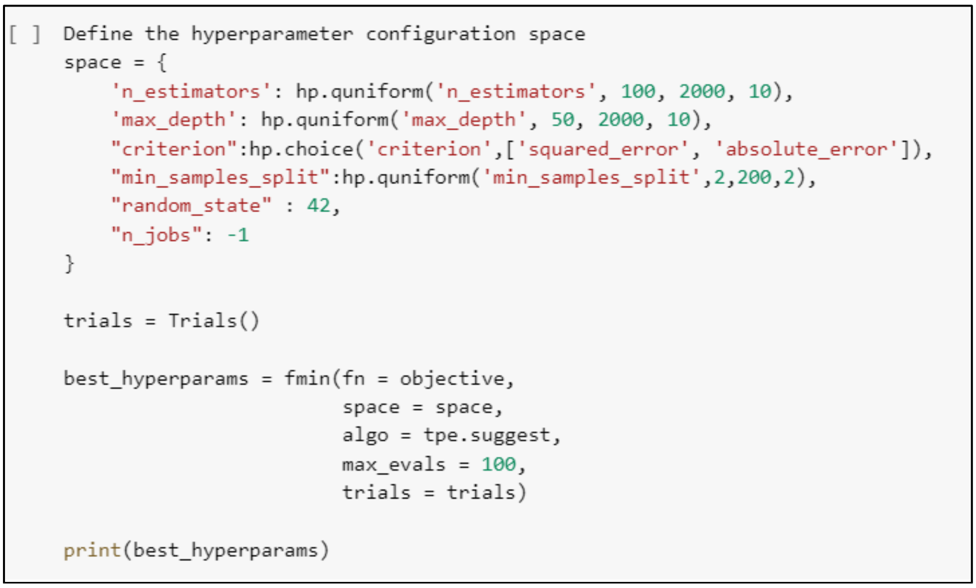
Appendix A.22: Predicted prices using each of the model



```
#Hyper parameter tuning using hyperopt
def objective(space):
    reg = RandomForestRegressor(
                                n_estimators = int(space['n_estimators']),
                                max_depth = int(space['max_depth']),
                                criterion = str(space['criterion']),
                                min_samples_split = int(space['min_samples_split']),
                                random_state = space['random_state'],
                                n_jobs = space['n_jobs']

                                )
    evaluation = [( X_train, y_train), ( X_test, y_test)]

    model = reg.fit(X_train, y_train)

    pred = reg.predict(X_test)
    mse = mean_squared_error(y_test,pred)
    rmse = np.round(sqrt(mse),4)
    R_squared = np.round(r2_score(y_test, pred),4)
    return {'loss':rmse, 'r_squared':R_squared ,'status': STATUS_OK, 'model':model, 'predictions':pred, 'evaluation':evaluation}
```

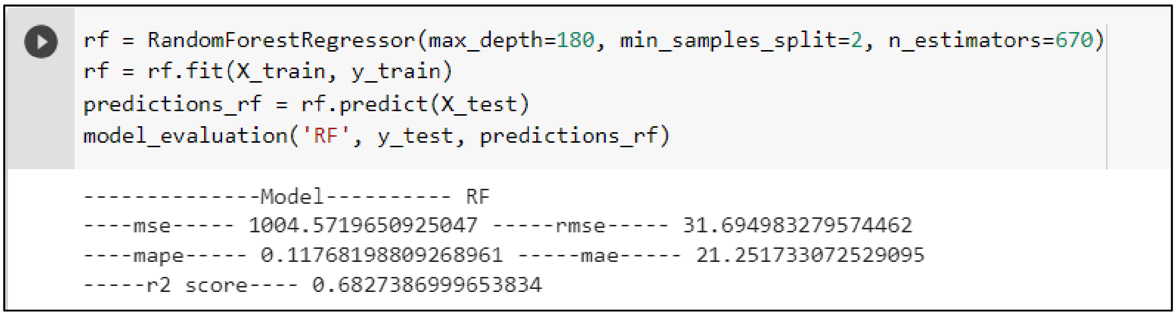
```
[ ]
    # Define the hyperparameter configuration space
    space = {
        'n_estimators': hp.quniform('n_estimators', 100, 2000, 10),
        'max_depth': hp.quniform('max_depth', 50, 2000, 10),
        "criterion":hp.choice('criterion',['squared_error', 'absolute_error']),
        "min_samples_split":hp.quniform('min_samples_split',2,200,2),
        "random_state" : 42,
        "n_jobs": -1
    }

    trials = Trials()

    best_hyperparams = fmin(fn = objective,
                            space = space,
                            algo = tpe.suggest,
                            max_evals = 100,
                            trials = trials)

    print(best_hyperparams)
```

Appendix A.23: Hyper parameter tuning



```
[ ]  Define the hyperparameter configuration space
     space = {
         'n_estimators': hp.quniform('n_estimators', 100, 2000, 10),
         'max_depth': hp.quniform('max_depth', 50, 2000, 10),
         "criterion":hp.choice('criterion',['squared_error', 'absolute_error']),
         "min_samples_split":hp.quniform('min_samples_split',2,200,2),
         "random_state" : 42,
         "n_jobs": -1
     }

     trials = Trials()

     best_hyperparams = fmin(fn = objective,
                             space = space,
                             algo = tpe.suggest,
                             max_evals = 100,
                             trials = trials)

     print(best_hyperparams)
```

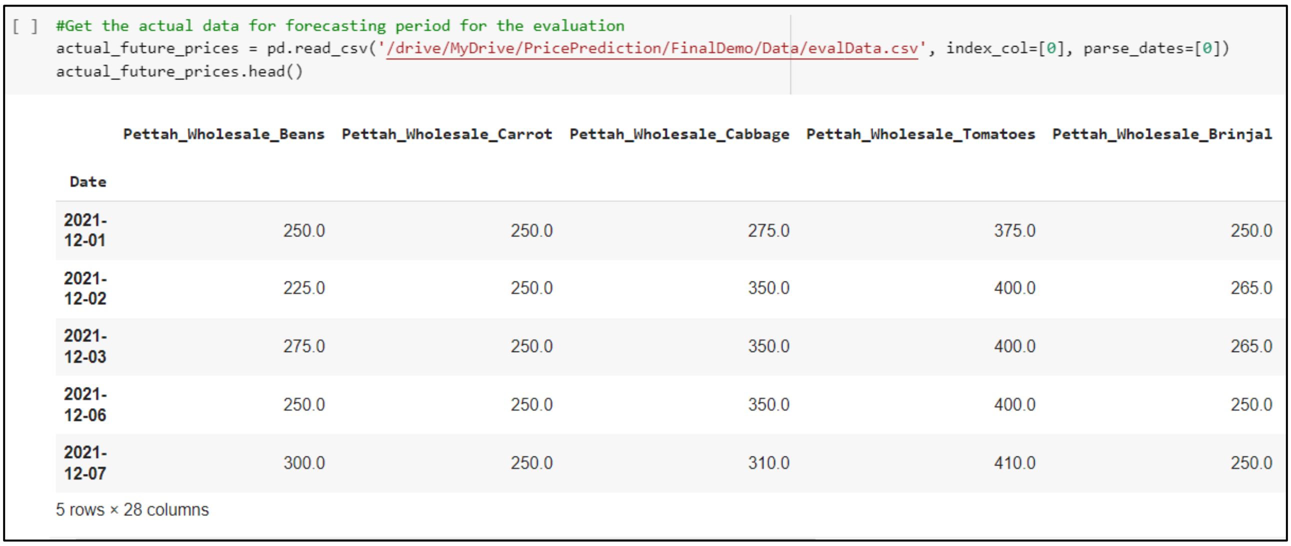
Appendix A.24: Hyper parameter configuration space



```
rf = RandomForestRegressor(max_depth=180, min_samples_split=2, n_estimators=670)
rf = rf.fit(X_train, y_train)
predictions_rf = rf.predict(X_test)
model_evaluation('RF', y_test, predictions_rf)
```

```
-------------Model---------- RF
----mse----- 1004.5719650925047 -----rmse----- 31.694983279574462
----mape----- 0.11768198809268961 -----mae----- 21.251733072529095
-----r2 score---- 0.6827386999653834
```

Appendix A.25: Random Forest model performance after tuning the parameters

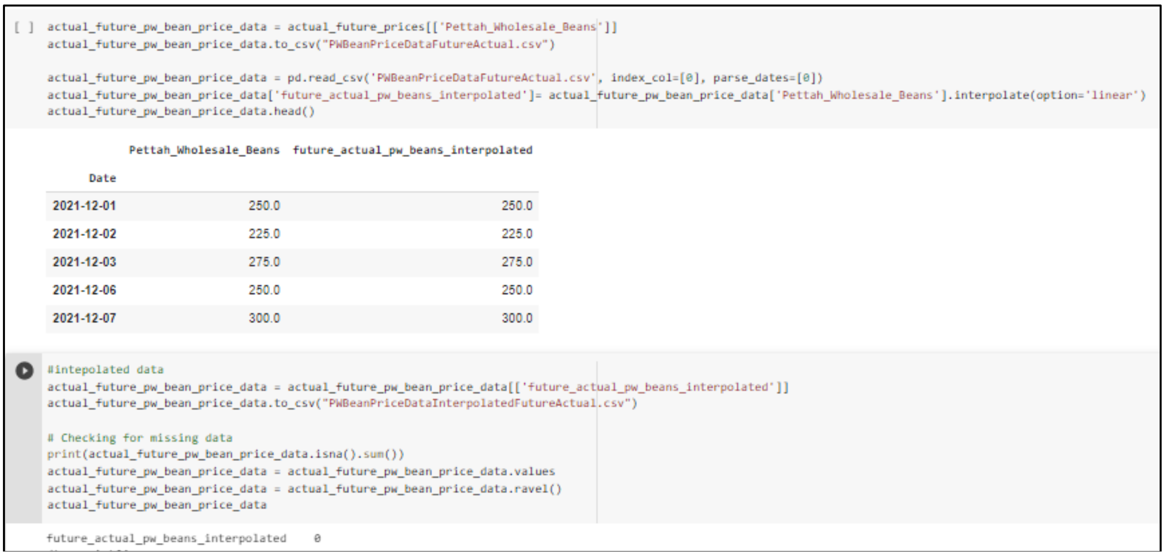


```
[ ] #Get the actual data for forecasting period for the evaluation
    actual_future_prices = pd.read_csv('/drive/MyDrive/PricePrediction/FinalDemo/Data/evalData.csv', index_col=[0], parse_dates=[0])
    actual_future_prices.head()
```

	Pettah_Wholesale_Beans	Pettah_Wholesale_Carrot	Pettah_Wholesale_Cabbage	Pettah_Wholesale_Tomatoes	Pettah_Wholesale_Brinjal
Date					
2021-12-01	250.0	250.0	275.0	375.0	250.0
2021-12-02	225.0	250.0	350.0	400.0	265.0
2021-12-03	275.0	250.0	350.0	400.0	265.0
2021-12-06	250.0	250.0	350.0	400.0	250.0
2021-12-07	300.0	250.0	310.0	410.0	250.0

5 rows × 28 columns

Appendix A.26: Actual data set for model evaluation



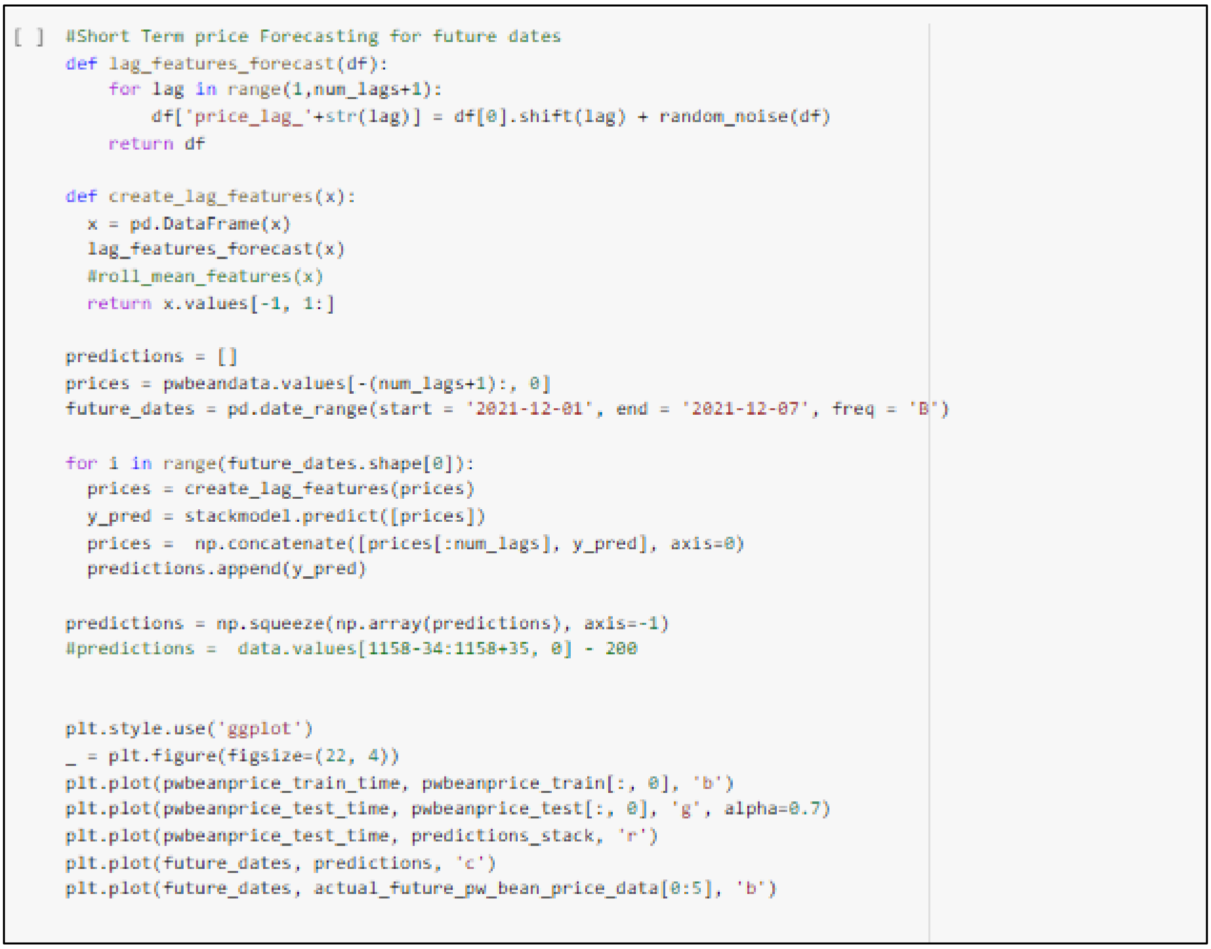
```
[ ] actual_future_pw_bean_price_data = actual_future_prices[['Pettah_Wholesale_Beans']]
    actual_future_pw_bean_price_data.to_csv("PWBeanPriceDataFutureActual.csv")

    actual_future_pw_bean_price_data = pd.read_csv('PWBeanPriceDataFutureActual.csv', index_col=[0], parse_dates=[0])
    actual_future_pw_bean_price_data['future_actual_pw_beans_interpolated']= actual_future_pw_bean_price_data['Pettah_Wholesale_Beans'].interpolate(option='linear')
    actual_future_pw_bean_price_data.head()
```

	Pettah_Wholesale_Beans	future_actual_pw_beans_interpolated
Date		
2021-12-01	250.0	250.0
2021-12-02	225.0	225.0
2021-12-03	275.0	275.0
2021-12-06	250.0	250.0
2021-12-07	300.0	300.0

```
#intepolated data
actual_future_pw_bean_price_data = actual_future_pw_bean_price_data[['future_actual_pw_beans_interpolated']]
actual_future_pw_bean_price_data.to_csv("PWBeanPriceDataInterpolatedFutureActual.csv")

# Checking for missing data
print(actual_future_pw_bean_price_data.isna().sum())
actual_future_pw_bean_price_data = actual_future_pw_bean_price_data.values
actual_future_pw_bean_price_data = actual_future_pw_bean_price_data.ravel()
actual_future_pw_bean_price_data
```

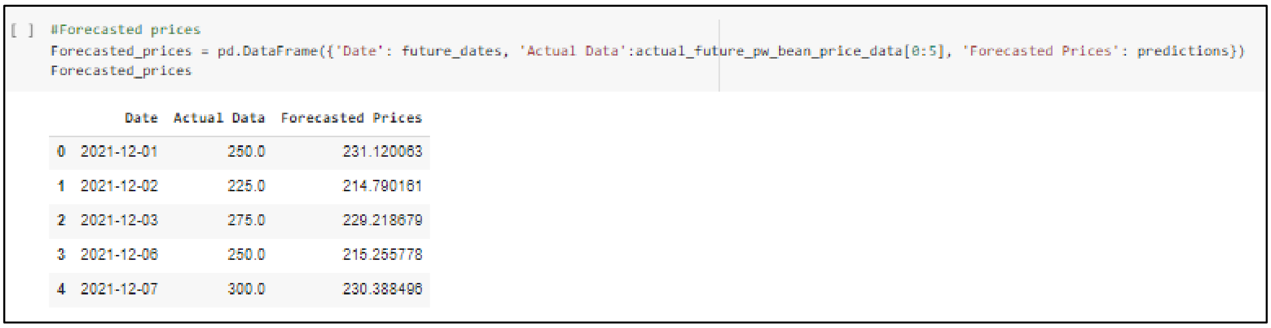
```
future_actual_pw_beans_interpolated    0
```

Appendix A.27: Pre-process actual data set



```
[ ]  #Short Term price Forecasting for future dates
     def lag_features_forecast(df):
         for lag in range(1,num_lags+1):
             df['price_lag_'+str(lag)] = df[0].shift(lag) + random_noise(df)
         return df

     def create_lag_features(x):
       x = pd.DataFrame(x)
       lag_features_forecast(x)
       #roll_mean_features(x)
       return x.values[-1, 1:]

     predictions = []
     prices = pwbeandata.values[-(num_lags+1):, 0]
     future_dates = pd.date_range(start = '2021-12-01', end = '2021-12-07', freq = 'B')

     for i in range(future_dates.shape[0]):
       prices = create_lag_features(prices)
       y_pred = stackmodel.predict([prices])
       prices =  np.concatenate([prices[:num_lags], y_pred], axis=0)
       predictions.append(y_pred)

     predictions = np.squeeze(np.array(predictions), axis=-1)
     #predictions =  data.values[1158-34:1158+35, 0] - 200


     plt.style.use('ggplot')
     _ = plt.figure(figsize=(22, 4))
     plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0], 'b')
     plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0], 'g', alpha=0.7)
     plt.plot(pwbeanprice_test_time, predictions_stack, 'r')
     plt.plot(future_dates, predictions, 'c')
     plt.plot(future_dates, actual_future_pw_bean_price_data[0:5], 'b')
```

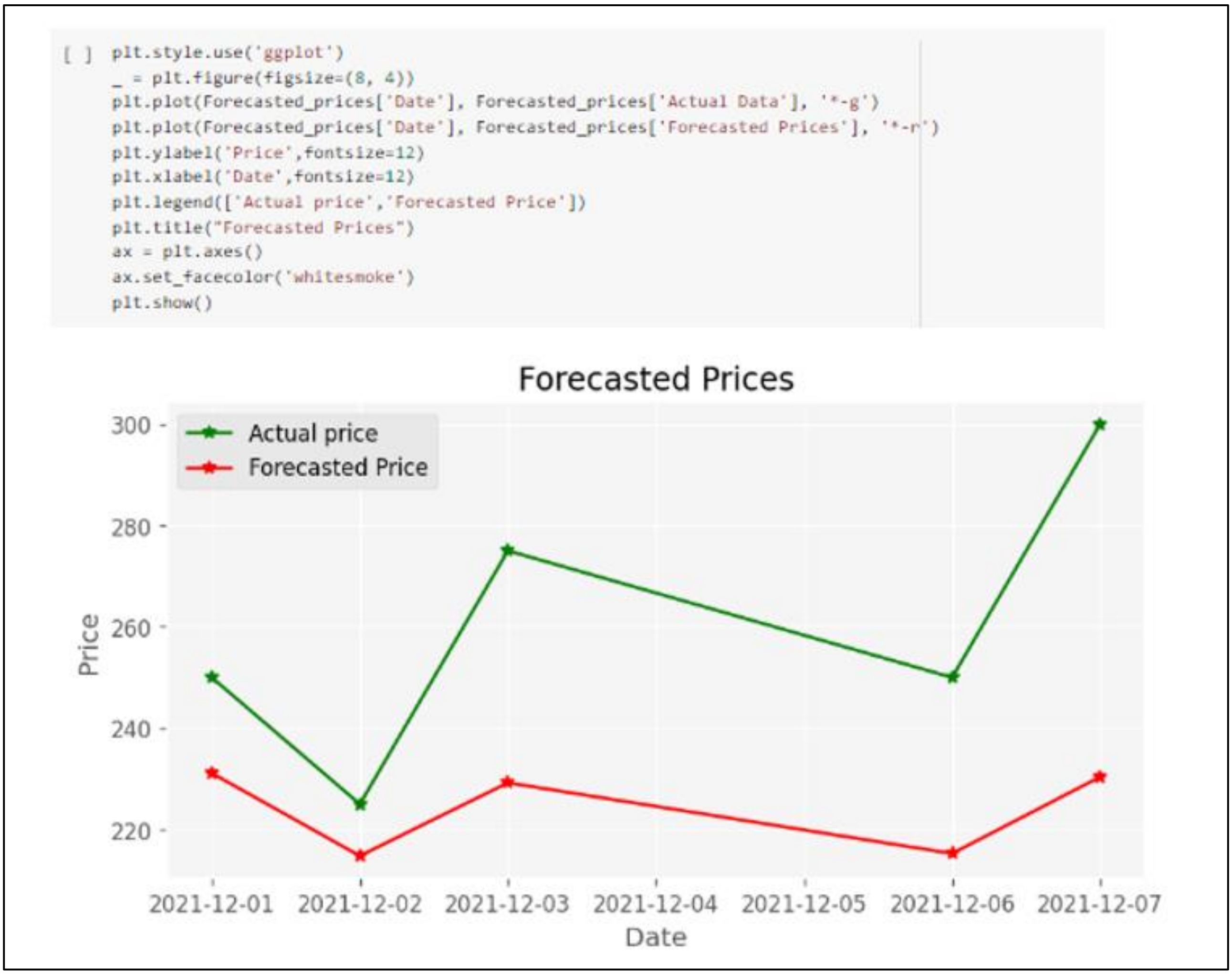
Appendix A.28: Short term price forecasting



```
[ ]  #Forecasted prices
     Forecasted_prices = pd.DataFrame({'Date': future_dates, 'Actual Data':actual_future_pw_bean_price_data[0:5], 'Forecasted Prices': predictions})
     Forecasted_prices
```

	Date	Actual Data	Forecasted Prices
0	2021-12-01	250.0	231.120063
1	2021-12-02	225.0	214.790161
2	2021-12-03	275.0	229.218679
3	2021-12-06	250.0	215.255778
4	2021-12-07	300.0	230.388496

Appendix A.29: Actual prices Vs forecasted prices

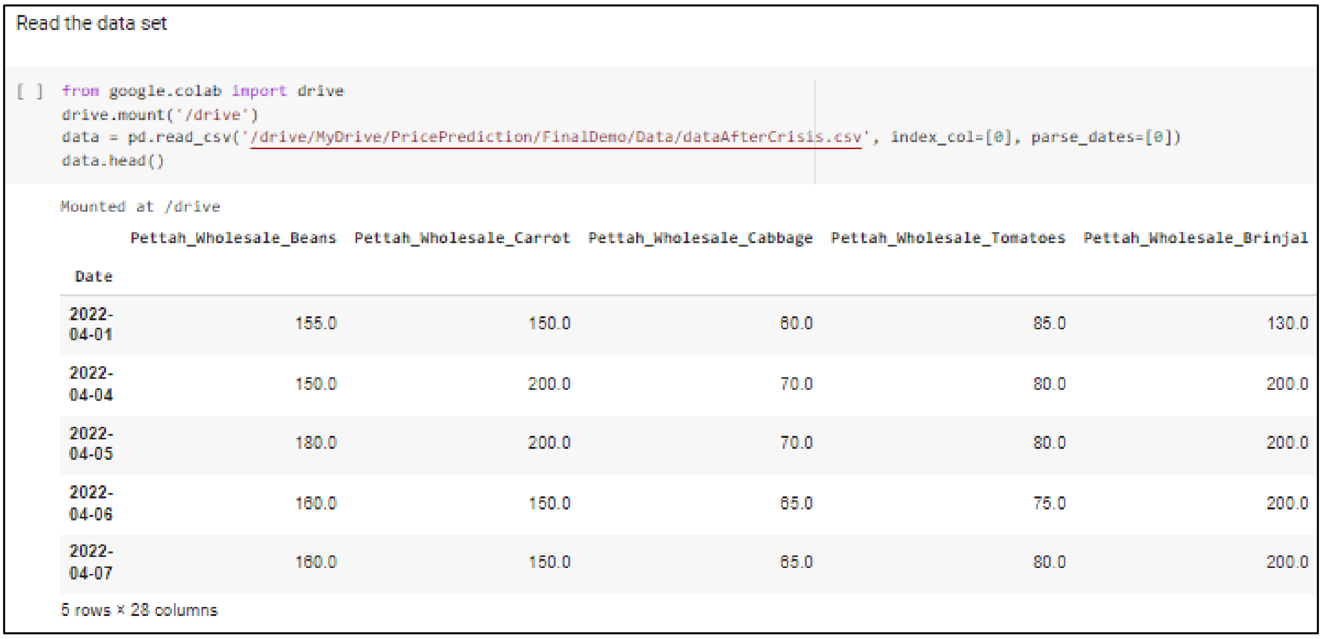


```
[ ] plt.style.use('ggplot')
    _ = plt.figure(figsize=(8, 4))
    plt.plot(Forecasted_prices['Date'], Forecasted_prices['Actual Data'], '*-g')
    plt.plot(Forecasted_prices['Date'], Forecasted_prices['Forecasted Prices'], '*-r')
    plt.ylabel('Price',fontsize=12)
    plt.xlabel('Date',fontsize=12)
    plt.legend(['Actual price','Forecasted Price'])
    plt.title("Forecasted Prices")
    ax = plt.axes()
    ax.set_facecolor('whitesmoke')
    plt.show()
```

Appendix A.30: Graphical representation of actual prices vs forecasted prices

Appendices B

Source codes and results generated for Pettah wholesale bean price prediction after the economic crisis using machine learning techniques.



Read the data set

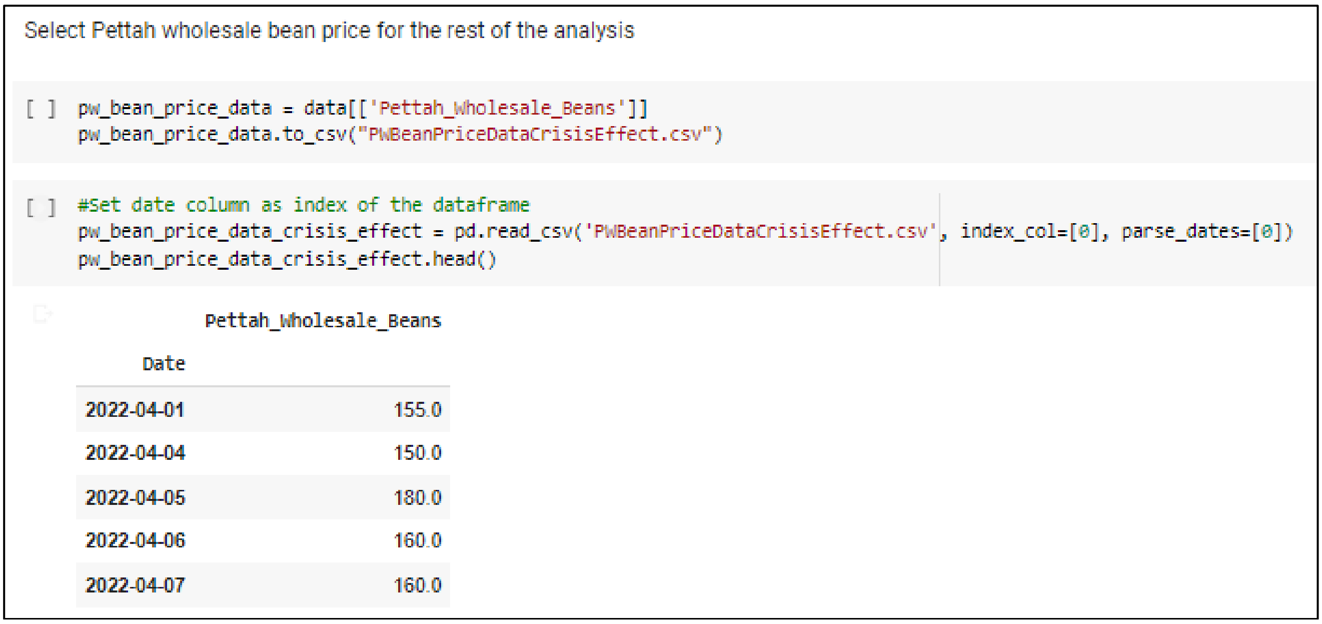
```
[ ] from google.colab import drive
    drive.mount('/drive')
    data = pd.read_csv('/drive/MyDrive/PricePrediction/FinalDemo/Data/dataAfterCrisis.csv', index_col=[0], parse_dates=[0])
    data.head()
```

```
Mounted at /drive
```

Date	Pettah_Wholesale_Beans	Pettah_Wholesale_Carrot	Pettah_Wholesale_Cabbage	Pettah_Wholesale_Tomatoes	Pettah_Wholesale_Brinjal
2022-04-01	155.0	150.0	60.0	85.0	130.0
2022-04-04	150.0	200.0	70.0	80.0	200.0
2022-04-05	180.0	200.0	70.0	80.0	200.0
2022-04-06	160.0	150.0	65.0	75.0	200.0
2022-04-07	160.0	150.0	65.0	80.0	200.0

5 rows × 28 columns

Appendix B.1: Read the data set



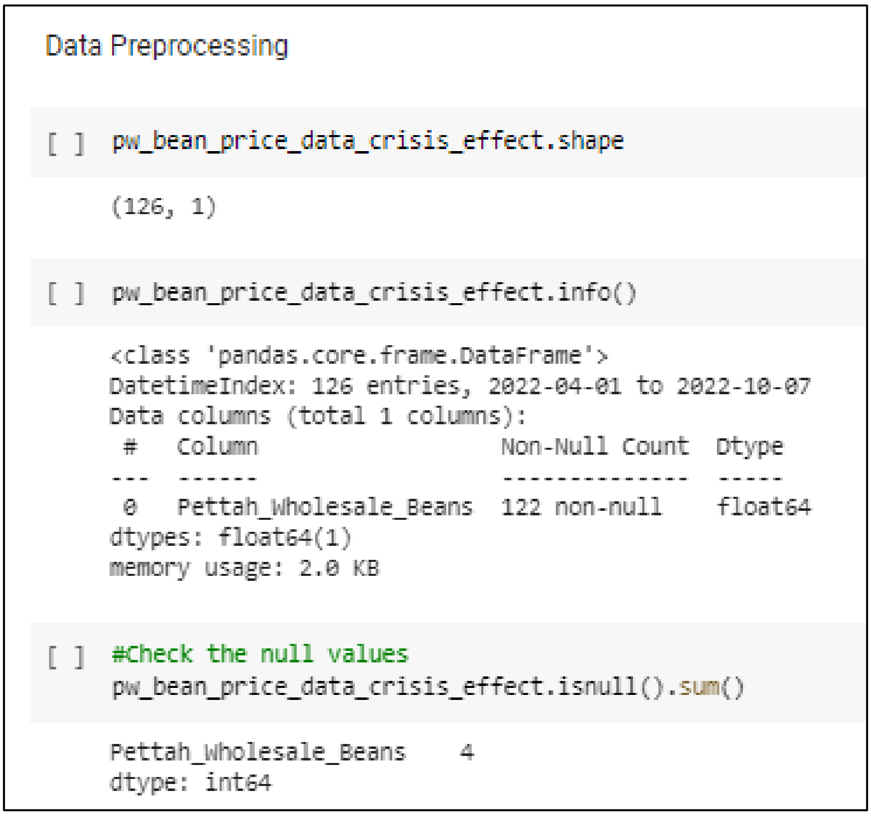
Select Pettah wholesale bean price for the rest of the analysis

```
[ ] pw_bean_price_data = data[['Pettah_Wholesale_Beans']]
    pw_bean_price_data.to_csv("PWBeanPriceDataCrisisEffect.csv")
```

```
[ ] #Set date column as index of the dataframe
    pw_bean_price_data_crisis_effect = pd.read_csv('PWBeanPriceDataCrisisEffect.csv', index_col=[0], parse_dates=[0])
    pw_bean_price_data_crisis_effect.head()
```

Date	Pettah_Wholesale_Beans
2022-04-01	155.0
2022-04-04	150.0
2022-04-05	180.0
2022-04-06	160.0
2022-04-07	160.0

Appendix B.2: Select Pettah wholesale bean price



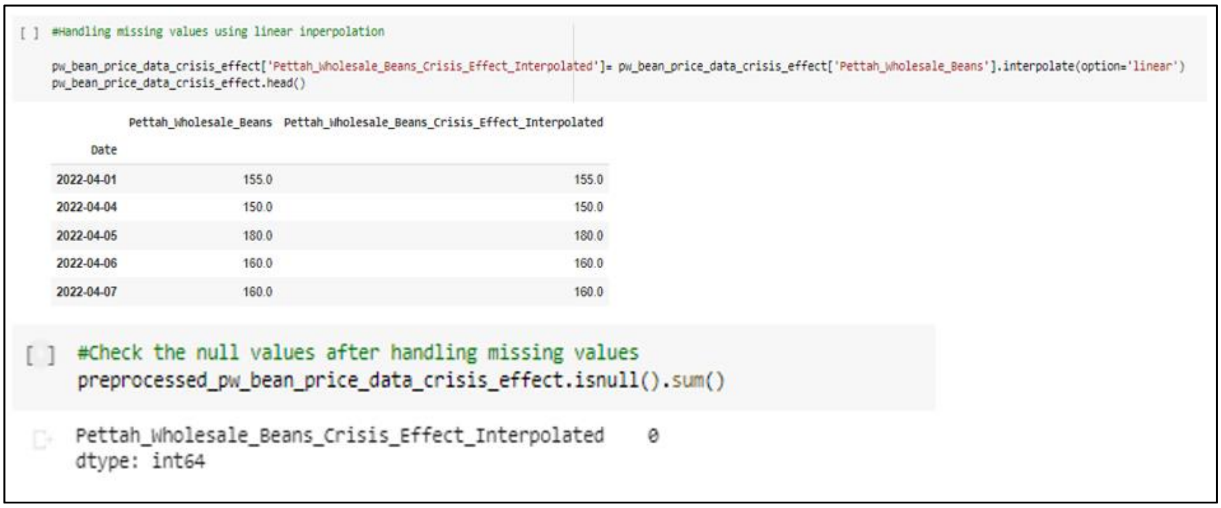
Data Preprocessing

```
[ ]  pw_bean_price_data_crisis_effect.shape

     (126, 1)
```

```
[ ]  pw_bean_price_data_crisis_effect.info()

     <class 'pandas.core.frame.DataFrame'>
     DatetimeIndex: 126 entries, 2022-04-01 to 2022-10-07
     Data columns (total 1 columns):
      #   Column                   Non-Null Count  Dtype  
     ---  ------                   --------------  -----  
      0   Pettah_Wholesale_Beans   122 non-null    float64
     dtypes: float64(1)
     memory usage: 2.0 KB
```

```
[ ]  #Check the null values
     pw_bean_price_data_crisis_effect.isnull().sum()

     Pettah_Wholesale_Beans    4
     dtype: int64
```

Appendix B.3: Data Pre-processing

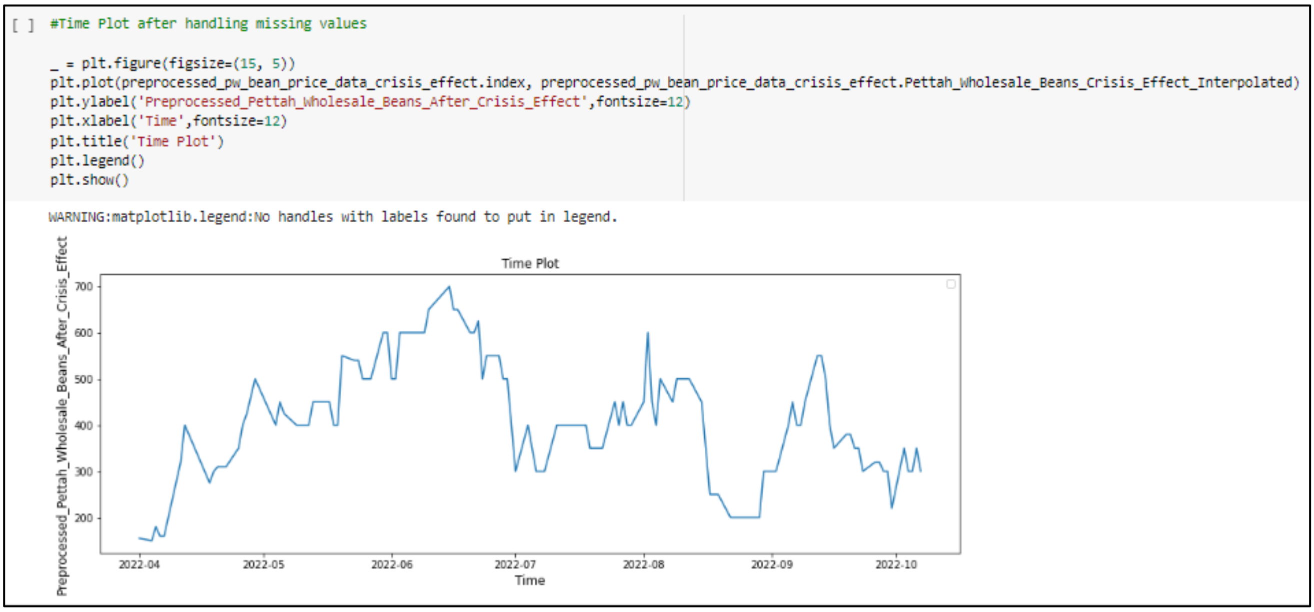


```
[ ]  #Handling missing values using linear inperpolation

     pw_bean_price_data_crisis_effect['Pettah_Wholesale_Beans_Crisis_Effect_Interpolated']= pw_bean_price_data_crisis_effect['Pettah_Wholesale_Beans'].interpolate(option='linear')
     pw_bean_price_data_crisis_effect.head()
```

Date	Pettah_Wholesale_Beans	Pettah_Wholesale_Beans_Crisis_Effect_Interpolated
2022-04-01	155.0	155.0
2022-04-04	150.0	150.0
2022-04-05	180.0	180.0
2022-04-06	160.0	160.0
2022-04-07	160.0	160.0

```
[ ]  #Check the null values after handling missing values
     preprocessed_pw_bean_price_data_crisis_effect.isnull().sum()

     Pettah_Wholesale_Beans_Crisis_Effect_Interpolated    0
     dtype: int64
```

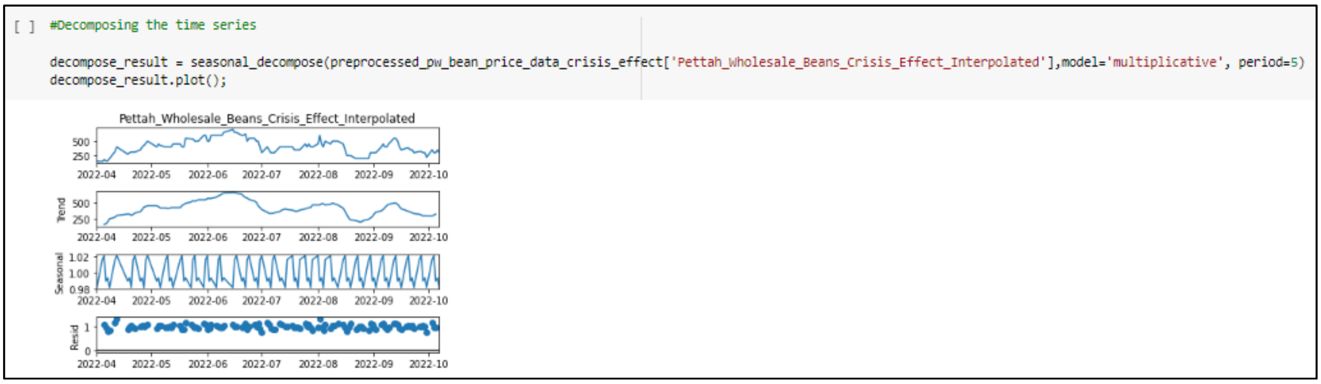
Appendix B.4: Handling missing values



```
#Time Plot after handling missing values

_ = plt.figure(figsize=(15, 5))
plt.plot(preprocessed_pw_bean_price_data_crisis_effect.index, preprocessed_pw_bean_price_data_crisis_effect.Pettah_Wholesale_Beans_Crisis_Effect_Interpolated)
plt.ylabel('Preprocessed_Pettah_Wholesale_Beans_After_Crisis_Effect',fontsize=12)
plt.xlabel('Time',fontsize=12)
plt.title('Time Plot')
plt.legend()
plt.show()
```

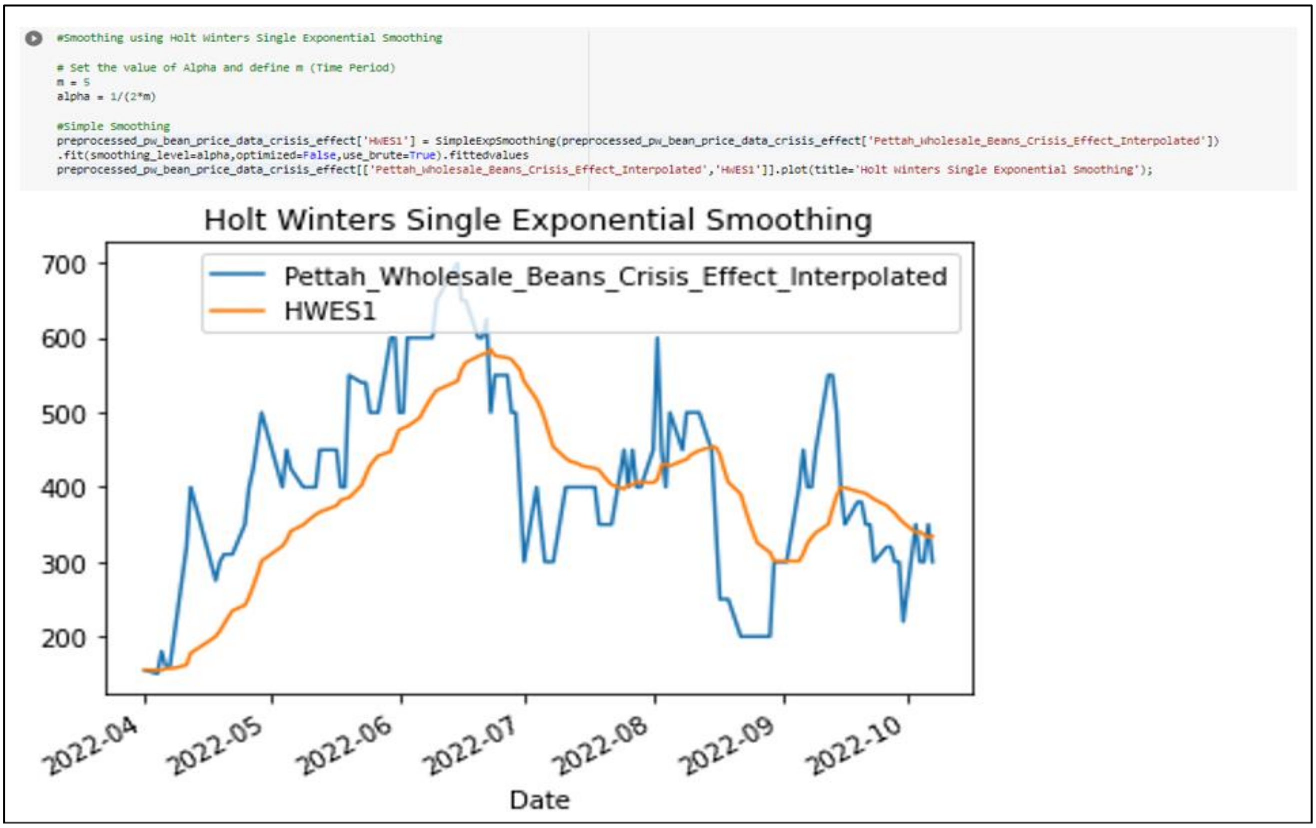
```
WARNING:matplotlib.legend:No handles with labels found to put in legend.
```

Appendix B.5: Time Plot



```
#Decomposing the time series

decompose_result = seasonal_decompose(preprocessed_pw_bean_price_data_crisis_effect['Pettah_Wholesale_Beans_Crisis_Effect_Interpolated'],model='multiplicative', period=5)
decompose_result.plot();
```

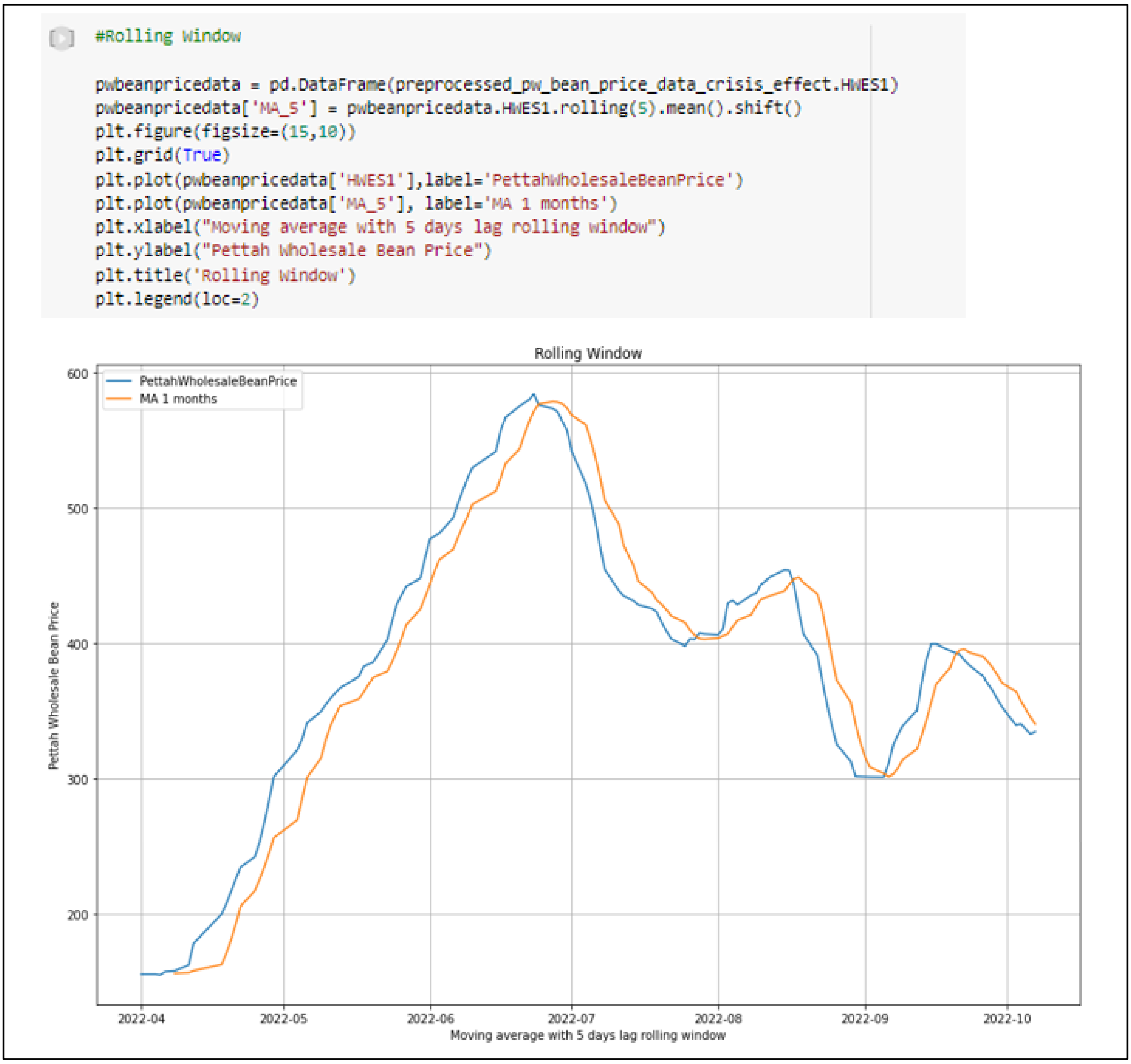
Appendix B.6: Decomposing time series



```
#Smoothing using Holt Winters Single Exponential Smoothing

# Set the value of Alpha and define m (Time Period)
m = 5
alpha = 1/(2*m)

#Simple Smoothing
preprocessed_pw_bean_price_data_crisis_effect['HWES1'] = SimpleExpSmoothing(preprocessed_pw_bean_price_data_crisis_effect['Pettah_Wholesale_Beans_Crisis_Effect_Interpolated'])
.fit(smoothing_level=alpha,optimized=False,use_brute=True).fittedvalues
preprocessed_pw_bean_price_data_crisis_effect[['Pettah_Wholesale_Beans_Crisis_Effect_Interpolated','HWES1']].plot(title='Holt Winters Single Exponential Smoothing');
```

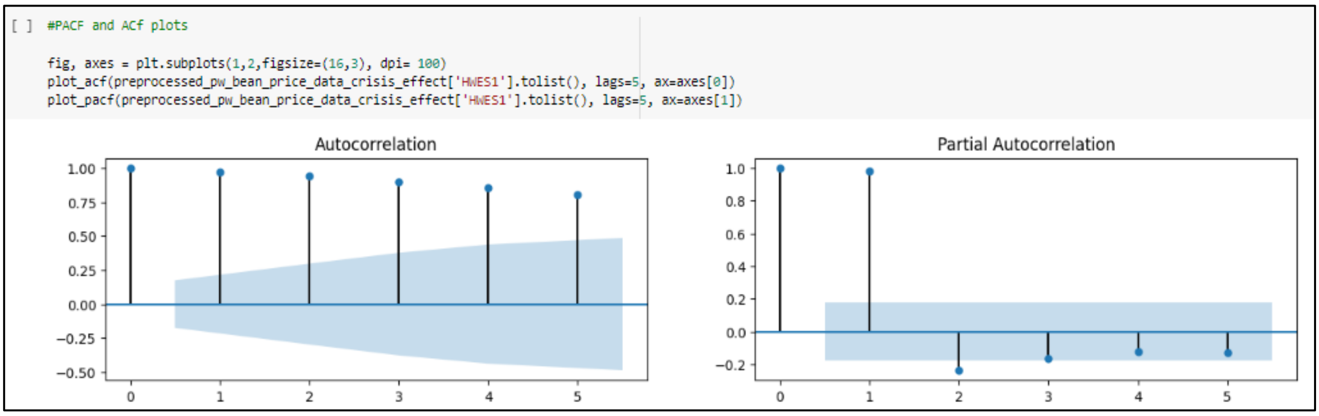
Holt Winters Single Exponential Smoothing
Pettah_Wholesale_Beans_Crisis_Effect_Interpolated
HWES1
700
600
500
400
300
200
2022-04
2022-05
2022-06
2022-07
2022-08
2022-09
2022-10
Date

Appendix B.7: Smoothing the data set



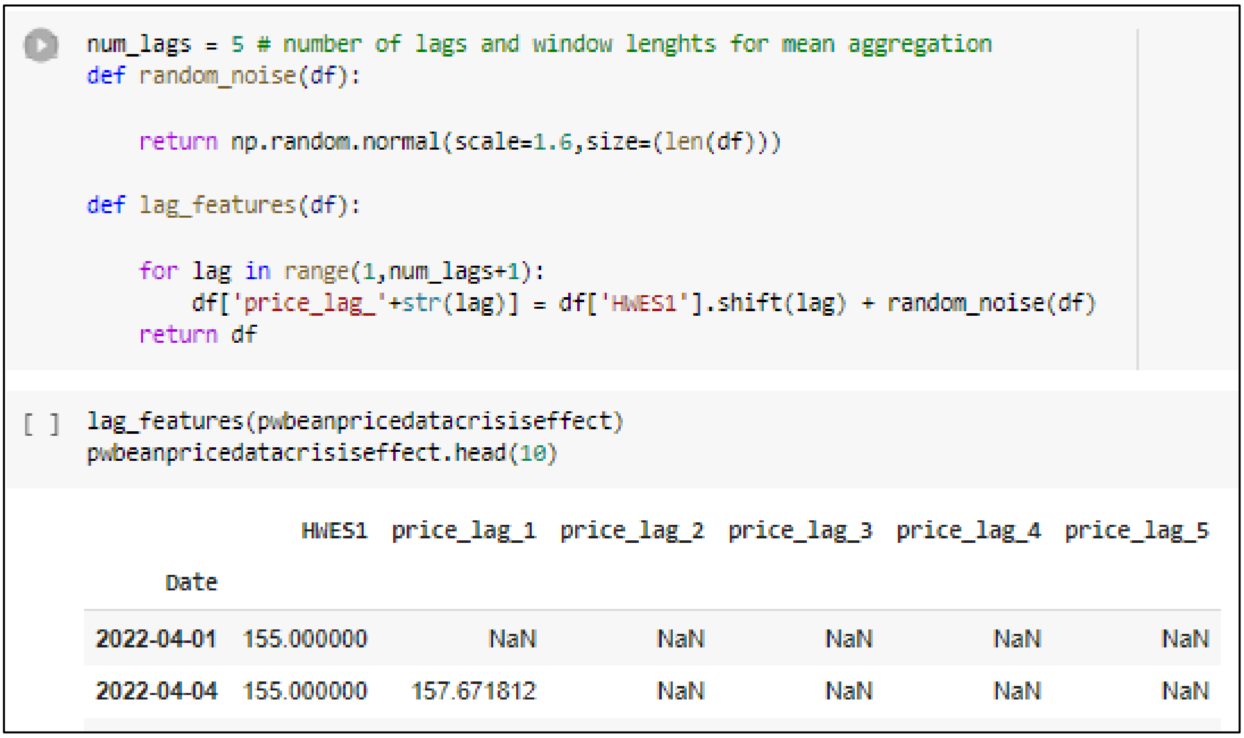
```
#Rolling Window

pwbeanpricedata = pd.DataFrame(preprocessed_pw_bean_price_data_crisis_effect.HWES1)
pwbeanpricedata['MA_5'] = pwbeanpricedata.HWES1.rolling(5).mean().shift()
plt.figure(figsize=(15,10))
plt.grid(True)
plt.plot(pwbeanpricedata['HWES1'],label='PettahWholesaleBeanPrice')
plt.plot(pwbeanpricedata['MA_5'], label='MA 1 months')
plt.xlabel("Moving average with 5 days lag rolling window")
plt.ylabel("Pettah Wholesale Bean Price")
plt.title('Rolling Window')
plt.legend(loc=2)
```

Appendix B.8: Rolling Window



```
[ ] #PACF and ACf plots

    fig, axes = plt.subplots(1,2,figsize=(16,3), dpi= 100)
    plot_acf(preprocessed_pw_bean_price_data_crisis_effect['HWES1'].tolist(), lags=5, ax=axes[0])
    plot_pacf(preprocessed_pw_bean_price_data_crisis_effect['HWES1'].tolist(), lags=5, ax=axes[1])
```

Appendix B.9: ACF and PACF plot

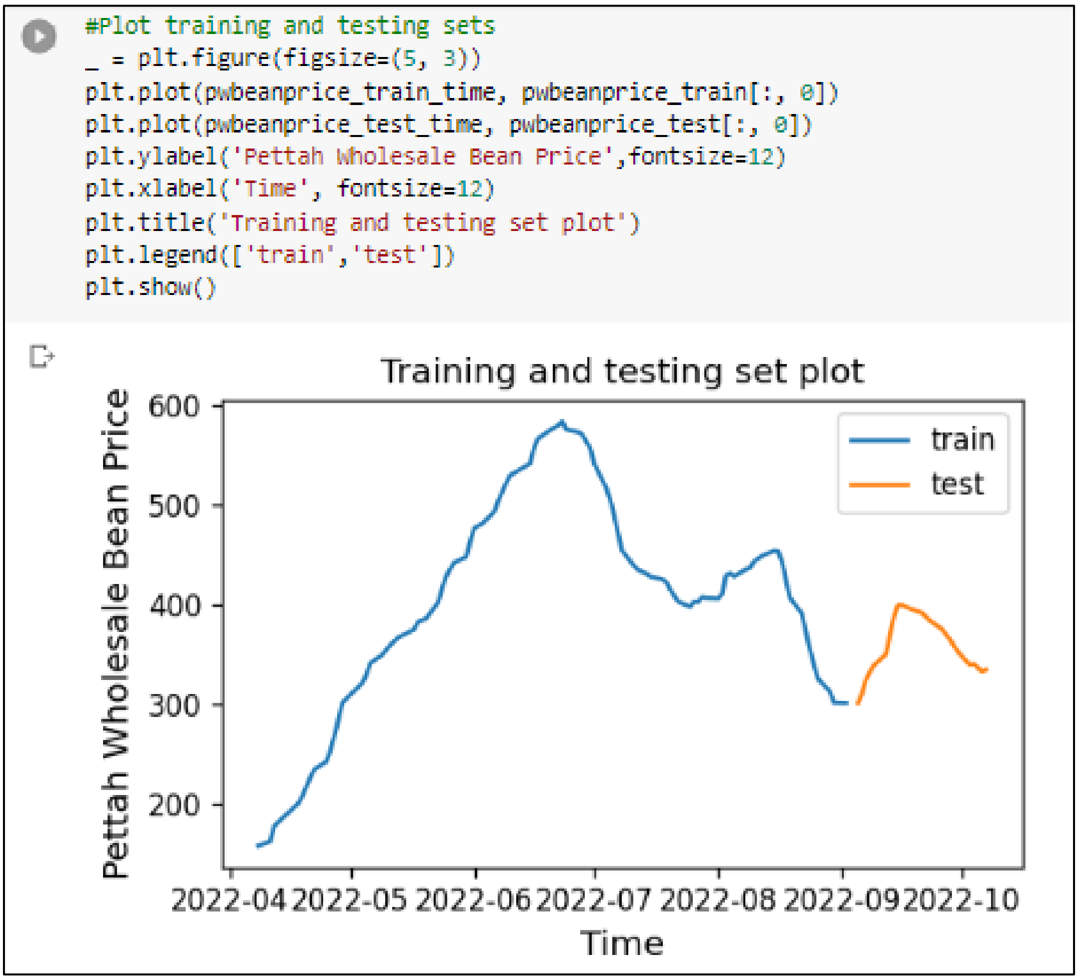


```
num_lags = 5 # number of lags and window lenghts for mean aggregation
def random_noise(df):

    return np.random.normal(scale=1.6,size=(len(df)))

def lag_features(df):

    for lag in range(1,num_lags+1):
        df['price_lag_'+str(lag)] = df['HWES1'].shift(lag) + random_noise(df)
    return df
```

```
[ ] lag_features(pwbeanpricedatacrisiseffect)
    pwbeanpricedatacrisiseffect.head(10)
```

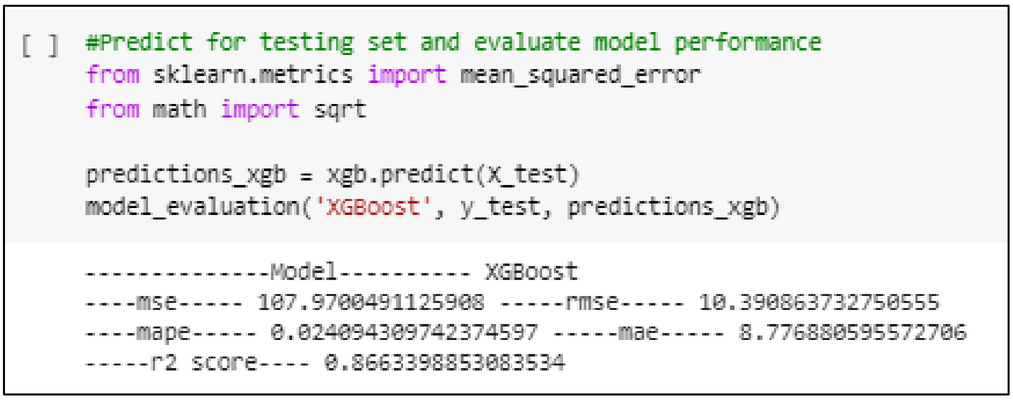
```
                 HWES1  price_lag_1  price_lag_2  price_lag_3  price_lag_4  price_lag_5

      Date
2022-04-01  155.000000          NaN          NaN          NaN          NaN          NaN
2022-04-04  155.000000   157.671812          NaN          NaN          NaN          NaN
```

Appendix B.10: Create lag variables



```
#Plot training and testing sets
_ = plt.figure(figsize=(5, 3))
plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0])
plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0])
plt.ylabel('Pettah Wholesale Bean Price',fontsize=12)
plt.xlabel('Time', fontsize=12)
plt.title('Training and testing set plot')
plt.legend(['train','test'])
plt.show()
```

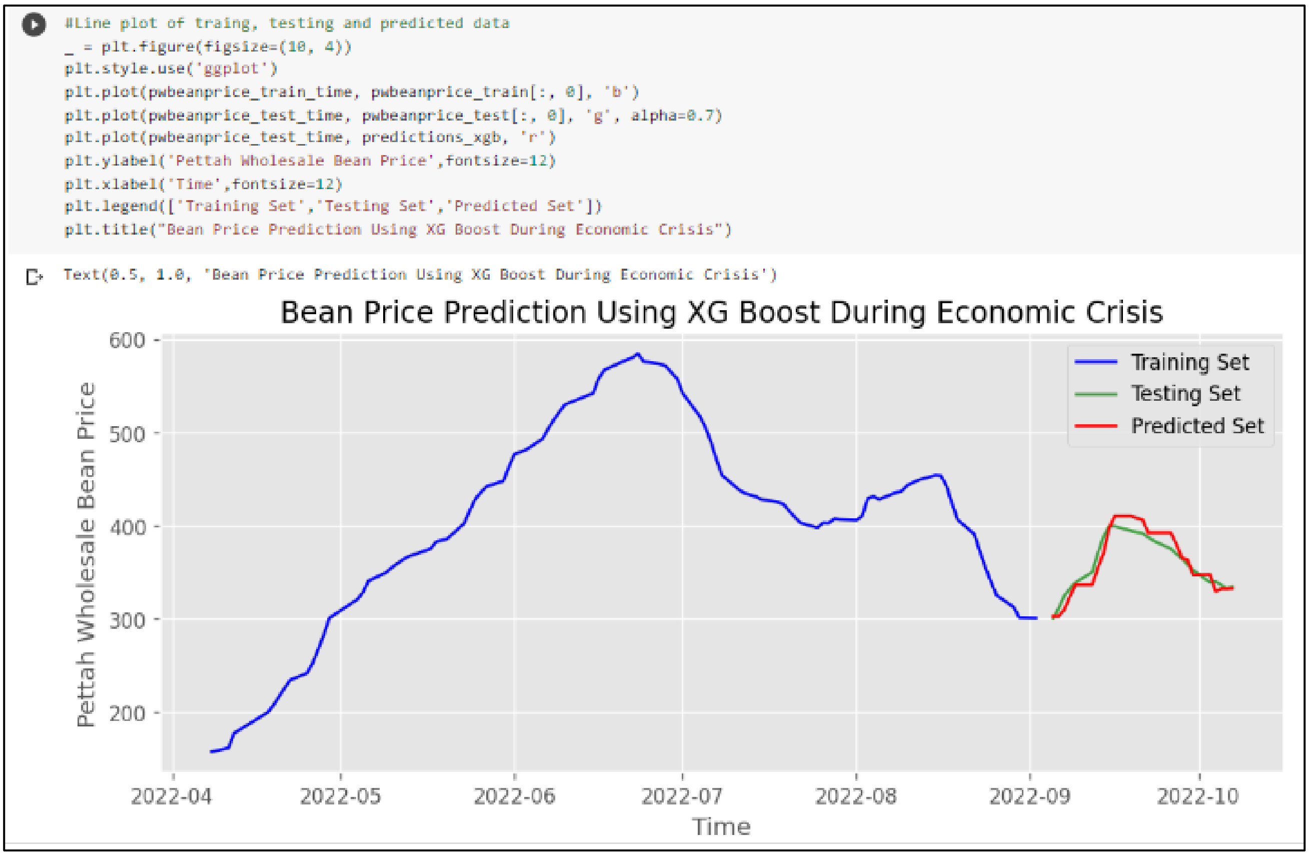
Appendix B.11: Training and testing set plot



```
[ ] #Predict for testing set and evaluate model performance
    from sklearn.metrics import mean_squared_error
    from math import sqrt

    predictions_xgb = xgb.predict(X_test)
    model_evaluation('XGBoost', y_test, predictions_xgb)
```

```
--------------Model---------- XGBoost
----mse----- 107.9700491125908 -----rmse----- 10.390863732750555
----mape----- 0.024094309742374597 -----mae----- 8.776880595572706
-----r2 score---- 0.8663398853083534
```

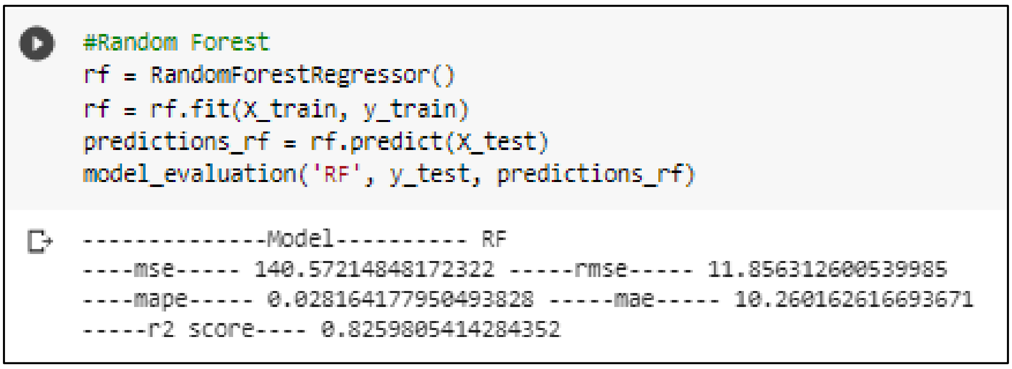
Appendix B.12: Performance of XGB model



```
#Line plot of traing, testing and predicted data
_ = plt.figure(figsize=(10, 4))
plt.style.use('ggplot')
plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0], 'b')
plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0], 'g', alpha=0.7)
plt.plot(pwbeanprice_test_time, predictions_xgb, 'r')
plt.ylabel('Pettah Wholesale Bean Price',fontsize=12)
plt.xlabel('Time',fontsize=12)
plt.legend(['Training Set','Testing Set','Predicted Set'])
plt.title("Bean Price Prediction Using XG Boost During Economic Crisis")
```

```
Text(0.5, 1.0, 'Bean Price Prediction Using XG Boost During Economic Crisis')
```

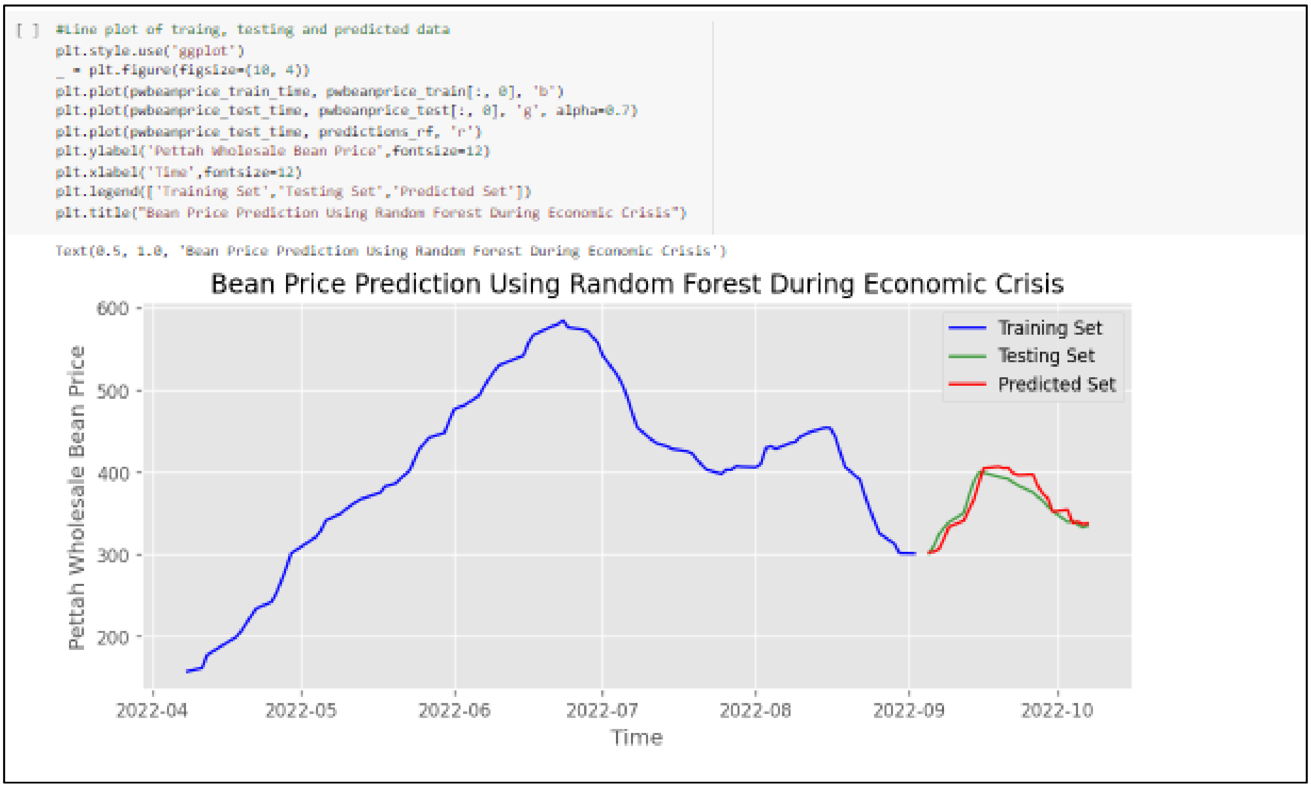
Appendix B.13: Training set, testing set, and prediction set plot for XGB



```
#Random Forest
rf = RandomForestRegressor()
rf = rf.fit(X_train, y_train)
predictions_rf = rf.predict(X_test)
model_evaluation('RF', y_test, predictions_rf)
```

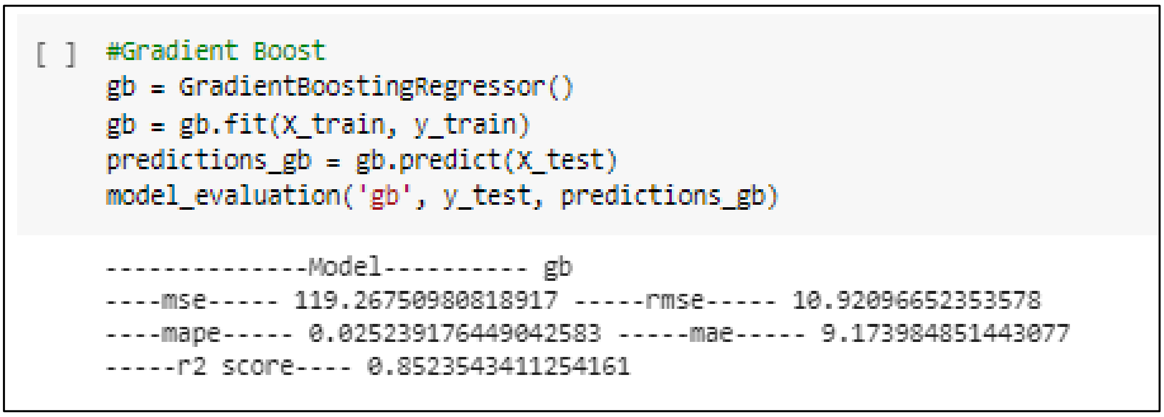
```
--------------Model---------- RF
----mse----- 140.57214848172322 -----rmse----- 11.856312600539985
----mape----- 0.028164177950493828 -----mae----- 10.260162616693671
-----r2 score---- 0.8259805414284352
```

Appendix B.14: Performance of RF model



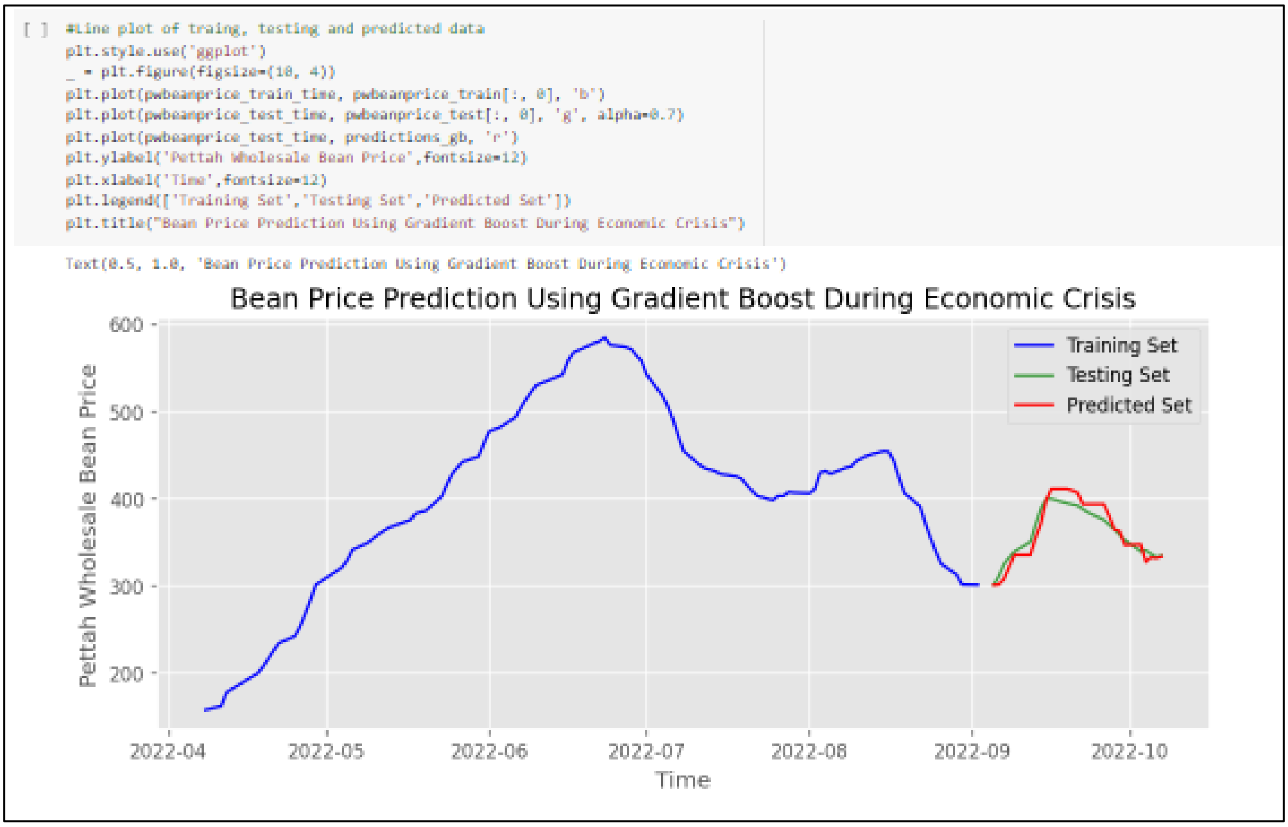
```
[ ]  #Line plot of traing, testing and predicted data
     plt.style.use('ggplot')
     _ = plt.figure(figsize=(10, 4))
     plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0], 'b')
     plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0], 'g', alpha=0.7)
     plt.plot(pwbeanprice_test_time, predictions_rf, 'r')
     plt.ylabel('Pettah Wholesale Bean Price',fontsize=12)
     plt.xlabel('Time',fontsize=12)
     plt.legend(['Training Set','Testing Set','Predicted Set'])
     plt.title("Bean Price Prediction Using Random Forest During Economic Crisis")

     Text(0.5, 1.0, 'Bean Price Prediction Using Random Forest During Economic Crisis')
```

Appendix B.15: Training set, testing set, and prediction set plot for RF model



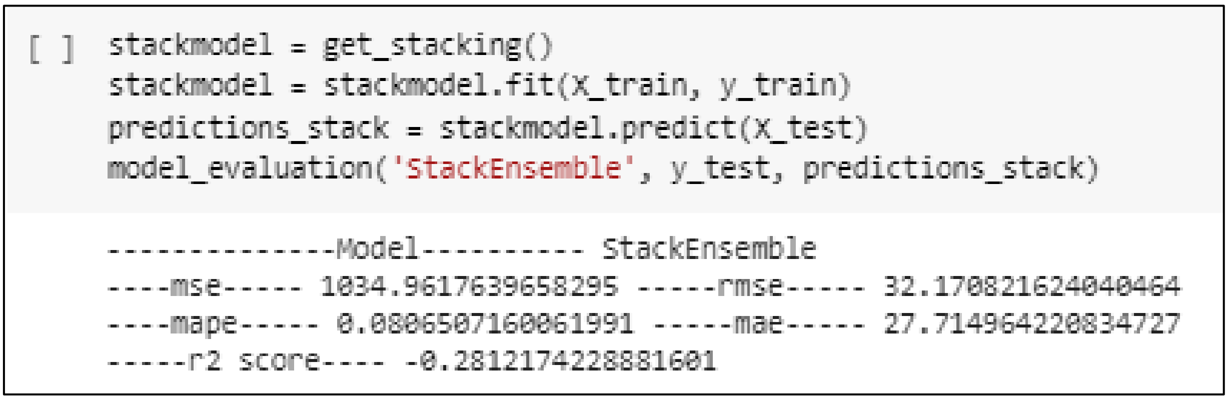
```
[ ]  #Gradient Boost
     gb = GradientBoostingRegressor()
     gb = gb.fit(X_train, y_train)
     predictions_gb = gb.predict(X_test)
     model_evaluation('gb', y_test, predictions_gb)

     --------------Model---------- gb
     ----mse----- 119.26750980818917 -----rmse----- 10.92096652353578
     ----mape----- 0.025239176449042583 -----mae----- 9.173984851443077
     -----r2 score---- 0.8523543411254161
```

Appendix B.16: Performance of GB model



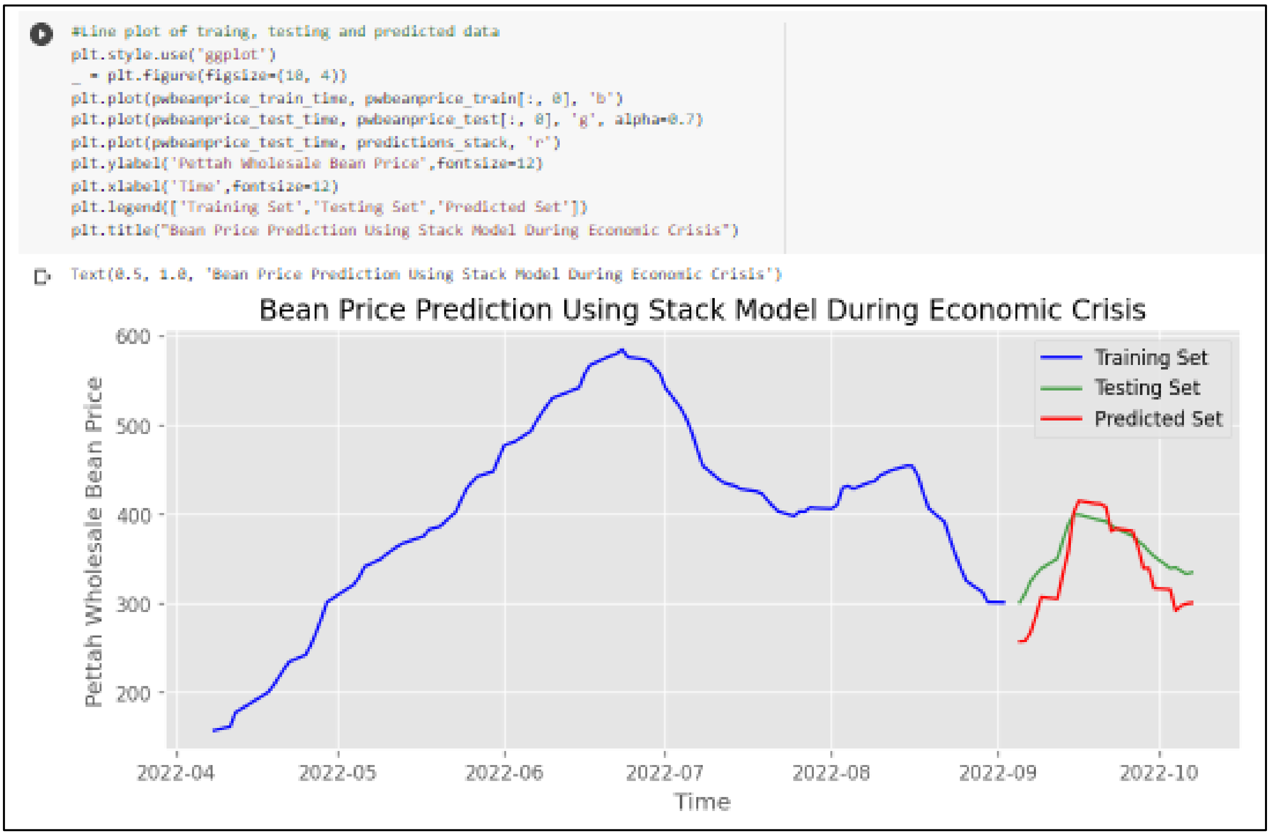
```
[ ]  #Line plot of traing, testing and predicted data
     plt.style.use('ggplot')
     _ = plt.figure(figsize=(10, 4))
     plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0], 'b')
     plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0], 'g', alpha=0.7)
     plt.plot(pwbeanprice_test_time, predictions_gb, 'r')
     plt.ylabel('Pettah Wholesale Bean Price',fontsize=12)
     plt.xlabel('Time',fontsize=12)
     plt.legend(['Training Set','Testing Set','Predicted Set'])
     plt.title("Bean Price Prediction Using Gradient Boost During Economic Crisis")

     Text(0.5, 1.0, 'Bean Price Prediction Using Gradient Boost During Economic Crisis')
```

Appendix B.17: Training set, testing set, and prediction set plot for GB model



```
[ ]  stackmodel = get_stacking()
     stackmodel = stackmodel.fit(X_train, y_train)
     predictions_stack = stackmodel.predict(X_test)
     model_evaluation('StackEnsemble', y_test, predictions_stack)

     --------------Model---------- StackEnsemble
     ----mse----- 1034.9617639658295 -----rmse----- 32.170821624040464
     ----mape----- 0.0806507160061991 -----mae----- 27.714964220834727
     -----r2 score---- -0.2812174228881601
```

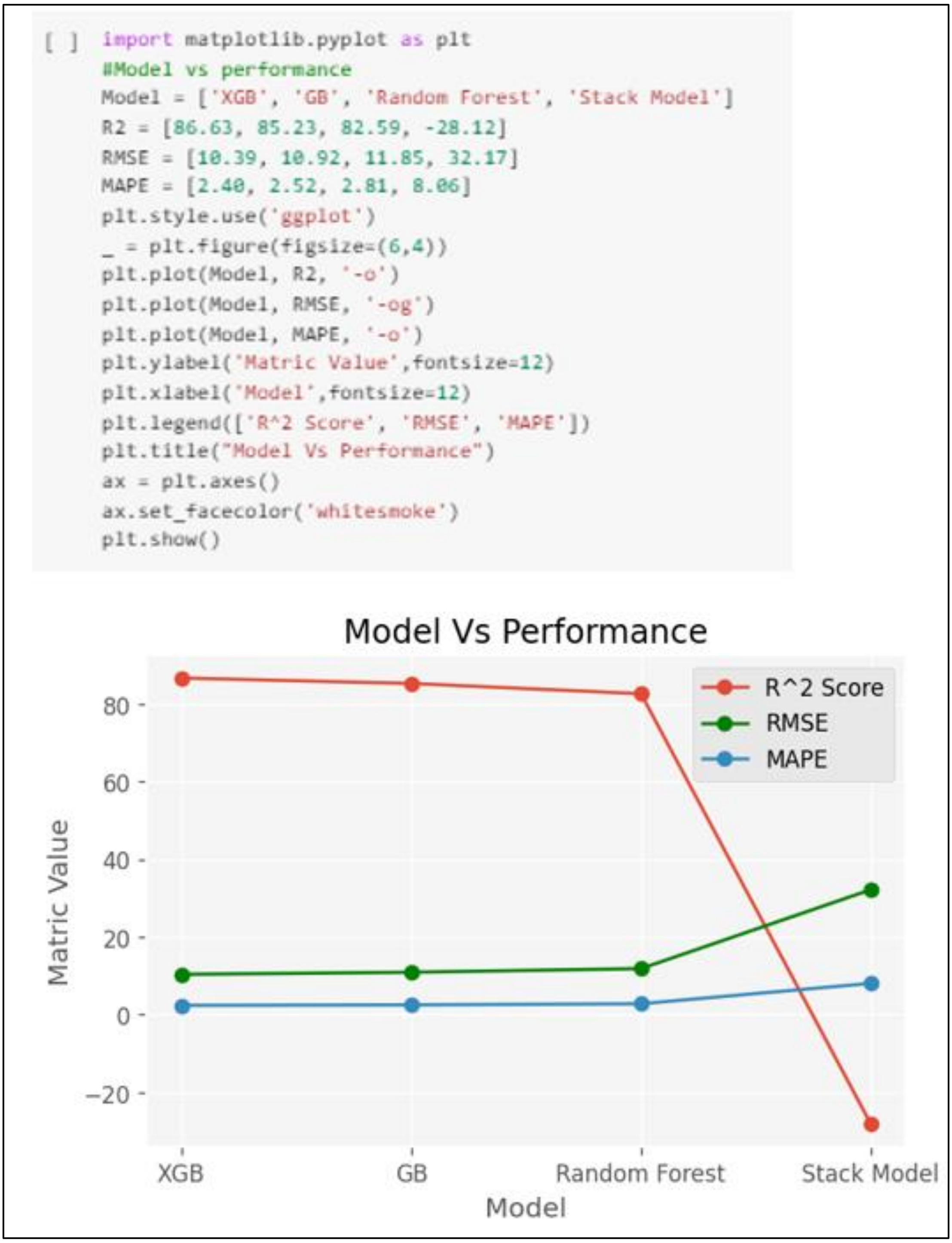
Appendix B.18: Performance of stack model



```
#Line plot of traing, testing and predicted data
plt.style.use('ggplot')
_ = plt.figure(figsize=(10, 4))
plt.plot(pwbeanprice_train_time, pwbeanprice_train[:, 0], 'b')
plt.plot(pwbeanprice_test_time, pwbeanprice_test[:, 0], 'g', alpha=0.7)
plt.plot(pwbeanprice_test_time, predictions_stack, 'r')
plt.ylabel('Pettah Wholesale Bean Price',fontsize=12)
plt.xlabel('Time',fontsize=12)
plt.legend(['Training Set','Testing Set','Predicted Set'])
plt.title("Bean Price Prediction Using Stack Model During Economic Crisis")
```

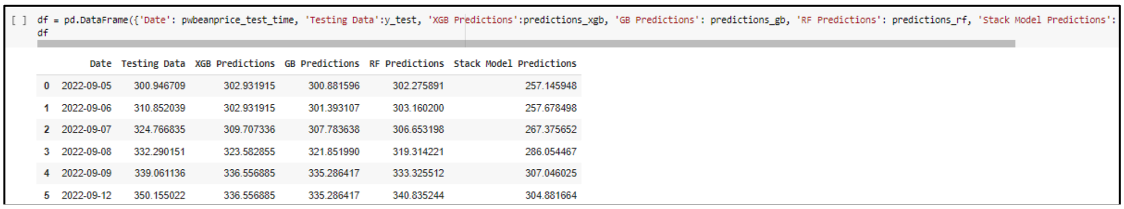
```
Text(0.5, 1.0, 'Bean Price Prediction Using Stack Model During Economic Crisis')
```

Appendix B.19: Training set, testing set, and prediction set plot for stack model



```
[ ] import matplotlib.pyplot as plt
    #Model vs performance
    Model = ['XGB', 'GB', 'Random Forest', 'Stack Model']
    R2 = [86.63, 85.23, 82.59, -28.12]
    RMSE = [10.39, 10.92, 11.85, 32.17]
    MAPE = [2.40, 2.52, 2.81, 8.06]
    plt.style.use('ggplot')
    _ = plt.figure(figsize=(6,4))
    plt.plot(Model, R2, '-o')
    plt.plot(Model, RMSE, '-og')
    plt.plot(Model, MAPE, '-o')
    plt.ylabel('Matric Value',fontsize=12)
    plt.xlabel('Model',fontsize=12)
    plt.legend(['R^2 Score', 'RMSE', 'MAPE'])
    plt.title("Model Vs Performance")
    ax = plt.axes()
    ax.set_facecolor('whitesmoke')
    plt.show()
```

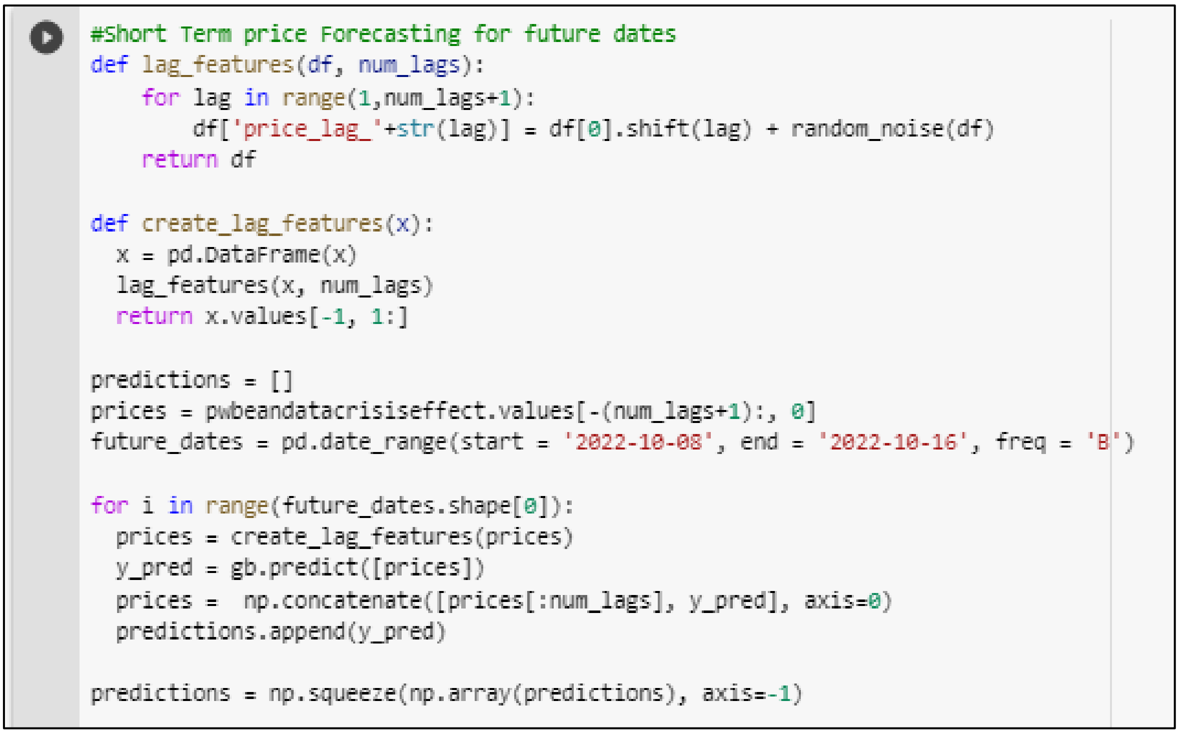
Appendix B.20: Model Vs Performance



```
[ ] df = pd.DataFrame({'Date': pwbeanprice_test_time, 'Testing Data':y_test, 'XGB Predictions':predictions_xgb, 'GB Predictions': predictions_gb, 'RF Predictions': predictions_rf, 'Stack Model Predictions':
    df
```

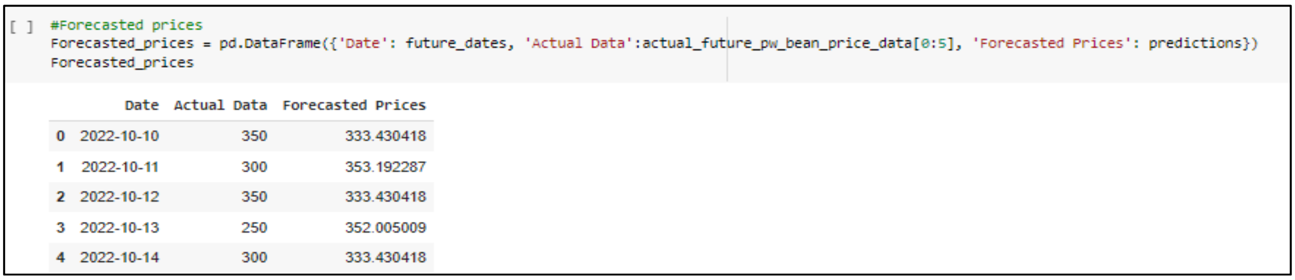
	Date	Testing Data	XGB Predictions	GB Predictions	RF Predictions	Stack Model Predictions
0	2022-09-05	300.946709	302.931915	300.881596	302.275891	257.145948
1	2022-09-06	310.852039	302.931915	301.393107	303.160200	257.678498
2	2022-09-07	324.766835	309.707336	307.783638	306.653198	267.375652
3	2022-09-08	332.290151	323.582855	321.851990	319.314221	286.054467
4	2022-09-09	339.061136	336.556885	335.286417	333.325512	307.046025
5	2022-09-12	350.155022	336.556885	335.286417	340.835244	304.881664

Appendix B.21: Prediction Results



```
#Short Term price Forecasting for future dates
def lag_features(df, num_lags):
    for lag in range(1,num_lags+1):
        df['price_lag_'+str(lag)] = df[0].shift(lag) + random_noise(df)
    return df

def create_lag_features(x):
  x = pd.DataFrame(x)
  lag_features(x, num_lags)
  return x.values[-1, 1:]

predictions = []
prices = pwbeandatacrisiseffect.values[-(num_lags+1):, 0]
future_dates = pd.date_range(start = '2022-10-08', end = '2022-10-16', freq = 'B')

for i in range(future_dates.shape[0]):
  prices = create_lag_features(prices)
  y_pred = gb.predict([prices])
  prices =  np.concatenate([prices[:num_lags], y_pred], axis=0)
  predictions.append(y_pred)

predictions = np.squeeze(np.array(predictions), axis=-1)
```

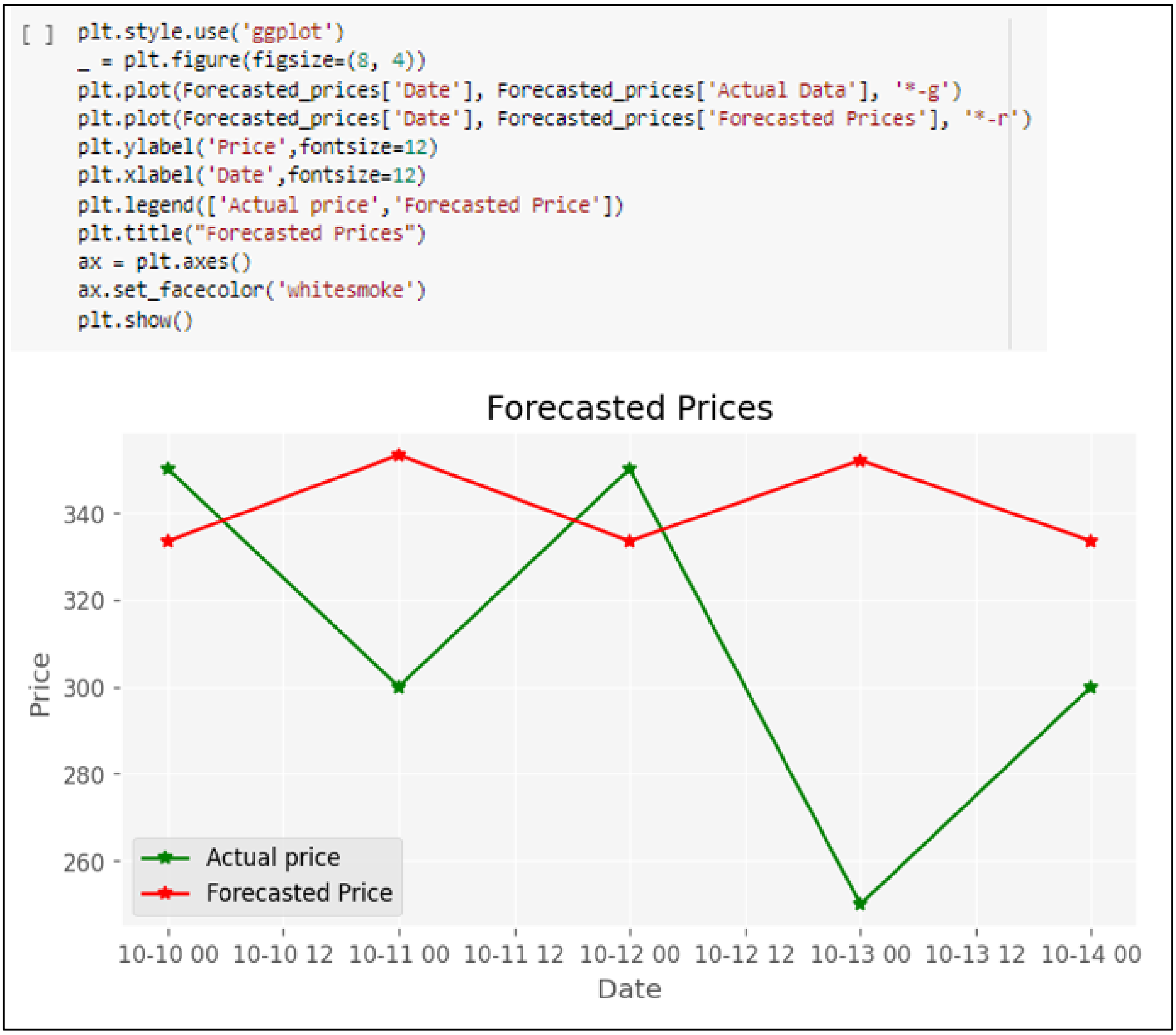
Appendix B.22: Short term forecasting



```
#Forecasted prices
Forecasted_prices = pd.DataFrame({'Date': future_dates, 'Actual Data':actual_future_pw_bean_price_data[0:5], 'Forecasted Prices': predictions})
Forecasted_prices
```

	Date	Actual Data	Forecasted Prices
0	2022-10-10	350	333.430418
1	2022-10-11	300	353.192287
2	2022-10-12	350	333.430418
3	2022-10-13	250	352.005009
4	2022-10-14	300	333.430418

Appendix B.23: Actual prices Vs forecasted prices



```
[ ]  plt.style.use('ggplot')
     _ = plt.figure(figsize=(8, 4))
     plt.plot(Forecasted_prices['Date'], Forecasted_prices['Actual Data'], '*-g')
     plt.plot(Forecasted_prices['Date'], Forecasted_prices['Forecasted Prices'], '*-r')
     plt.ylabel('Price',fontsize=12)
     plt.xlabel('Date',fontsize=12)
     plt.legend(['Actual price','Forecasted Price'])
     plt.title("Forecasted Prices")
     ax = plt.axes()
     ax.set_facecolor('whitesmoke')
     plt.show()
```

Appendix B.24: Graphical representation of actual prices Vs forecasted prices