# Title Page

**Long-Term Palay and Corn Crop Yield Forecasting in the Philippines Using Machine Learning**

****

**HAROLD R. TACASTACAS**

**July 2024**

# **Approval Sheet**

This thesis entitled “**Long-Term Palay and Corn Crop Yield Forecasting in the Philippines Using Machine Learning”** prepared and submitted by **HAROLD R. TACASTACAS** in partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Science** has been examined and recommended for acceptance and approval for ORAL EXAMINATION.

DEFENSE COMMITTEE

MR. EARL PETER J. GANGOSO  
Adviser

|  |  |
| --- | --- |
| MS. JENETH M. HORNEJAS, MBA Member | MR. HONIMAR C. PANGANIBAN, MSIT Member |
| ENGR. RAMEL E. RECENTES, MOE Committee Chair | MS. MARY GRACE T. LUNA, MAEd. English Editor |

­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Approved by the committee on Oral Examination as PASSED with [ ] no revision [ ] minor revision [ ] major revision.

ENGR. RAMEL E. RECENTES, MOE  
Committee Chair

|  |  |
| --- | --- |
| MS. JENETH M. HORNEJAS, MBA  Member | MR. HONIMAR C. PANGANIBAN, MSIT  Member |

MR. EARL PETER J. GANGOSO  
Adviser

ACCEPTED and APPROVED in partial fulfillment of the requirements for the degree of BACHELOR OF SCIENCE IN COMPUTER SCIENCE.

|  |  |
| --- | --- |
| July 15, 2024  Date of Oral Examination | ENGR. RAMEL E. RECENTES, MOE  Dean, College of Computer Studies,  Engineering, and Architecture |

# **Acknowledgement**

First and foremost, I thank God for every blessing and guidance he gave me during this work. He has been very gracious to me by providing the strength, wisdom, and peace of mind I received. His provisions have been handy and comforting, especially in some distressing situations.

Also, I would like to express my heartfelt gratitude to my parents Mrs. Joan D. Ramirez and Mr. Richard C. Tacastacas, for having believed in me throughout my journey. Their valuable pieces of advice have been about how to deal with hardships, accept the facts of life, and admit to discipline. Their lecture on moral and spiritual life has influenced the kind of person I am today in a very great way.

I would like to thank my aunt for her generous sponsorship all throughout my college years; Mrs. Vilma Hofmann Tacastacas. Auntie thank you so much for staying by my side and for always encouraging me. I would like to express my deepest gratitude to all of you who donated your trust and funds to me to enable me to reach this achievement.

I would also like to thank and appreciate the support and guidance granted to me by my thesis adviser Mr. Earl Peter J. Gangoso from the beginning of my study. Although I am currently the only participant in this study, he never allowed me to try to figure things out alone. His support and commitment to the project have always inspired me to keep going.

I would also like to express my deepest gratitude to my classmates in Computer Science 4, for their support throughout this college process. Many thanks for not abandoning me on the way and for the friendship, support, and similar-soul experiences that added to the richness of this journey.

# **Abstract**

*Palay* and corn have become their main food crops and are indispensable in the economic life in the Philippines. Crop yield forecasting is crucial in agriculture to enable producers to produce the right amounts of crops, food security, and help policymakers on where to allocate resources. This study seeks to overcome these challenges of crop production volatility caused by weather and other factors. This study uses a Random Forest Regressor to estimate the future yields of palay and corn using data from PSA and other related sources. A lot of data pre-processing, variable transformation, and hyperparameter adjustment were done to improve the model’s performance. The results of the model were assessed in terms of Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination (R²). Hypothesis testing showed that the Random Forest model yielded satisfactory predictive capacity with the R-squared value of 0.972 for corn and 0.896 for palay which means that the Random Forest model can explain much of the variation in crop yields.

The study prescribes a ten-year outlook on crop yields concerning climate change, helping to identify further changes in yields and future development.

This research focuses on the real-time utilization of machine learning in agriculture, specifically for farmers, policymakers, and market actors. The study therefore supports the use of evidence-based interventions to increase productivity in agriculture, manage risks, and support sustainable development in the sector.

# **Table of Contents**

[Title Page i](#_Toc183412040)

[**Approval Sheet** ii](#_Toc183412041)

[**Abstract** iii](#_Toc183412042)

[**Acknowledgement** iv](#_Toc183412043)

[**Table of Contents** v](#_Toc183412044)

[**List of Figures** viii](#_Toc183412045)

[**List of Tables** ix](#_Toc183412046)

[Chapter 1 **THE RESEARCH DESCRIPTION** 1](#_Toc183412047)

[Introduction 1](#_Toc183412048)

[Research Objectives 2](#_Toc183412049)

[Scope and Limitations 3](#_Toc183412050)

[Significance of the Study 4](#_Toc183412051)

[Chapter 2 **REVIEW OF RELATED LITERATURE AND STUDIES** 5](#_Toc183412052)

[Overview of Crop Yield Forecasting 5](#_Toc183412053)

[Benefits of Crop Yield Forecasting 6](#_Toc183412054)

[Factors Influencing the Yield for Crops 7](#_Toc183412055)

[Techniques on Time Series Forecasting Random Forest on Regression Problems 8](#_Toc183412056)

[A Comparative Study on Machine Learning for Crop Yield Forecasting 10](#_Toc183412057)

[Definition of Terms 12](#_Toc183412058)

[Chapter 3 **THEORETICAL FRAMEWORK** 14](#_Toc183412059)

[Data Collection 15](#_Toc183412060)

[Data Preprocessing 15](#_Toc183412061)

[Feature Engineering 15](#_Toc183412062)

[Model Development and Training 16](#_Toc183412063)

[Model Evaluation 16](#_Toc183412064)

[Detailed Forecast 18](#_Toc183412065)

[Chapter 4 **RESULTS AND DISCUSSIONS** 20](#_Toc183412066)

[Data Collection 20](#_Toc183412067)

[Data Exploration 23](#_Toc183412068)

[Data Cleaning 34](#_Toc183412069)

[Feature Engineering 38](#_Toc183412070)

[Model Development and Training 40](#_Toc183412071)

[Model Evaluation 44](#_Toc183412072)

[Detailed Forecast Analysis 47](#_Toc183412073)

[Chapter 5 **CONCLUSION AND RECOMMENDATIONS** 49](#_Toc183412074)

[Conclusion 49](#_Toc183412075)

[Recommendations 50](#_Toc183412076)

[**Bibliography** 51](#_Toc183412077)

[Appendix A **SOURCE CODE** 57](#_Toc183412078)

[Corn 57](#_Toc183412079)

[Data Preprocessing 57](#_Toc183412080)

[Data Cleaning 57](#_Toc183412081)

[Model Development and Training 61](#_Toc183412082)

[**Model Evaluation Random Forest Regressor** 61](#_Toc183412083)

[Model Analysis and Visualization 62](#_Toc183412084)

[Final Model Detailed Forecast Analysis 63](#_Toc183412085)

[Palay 65](#_Toc183412086)

[Data Preprocessing 65](#_Toc183412087)

[Data Cleaning 66](#_Toc183412088)

[Model Development and Training 69](#_Toc183412089)

[**Model Evaluation Random Forest Regressor** 70](#_Toc183412090)

[Model Analysis and Visualization 70](#_Toc183412091)

[Detailed Forecast Analysis Final Model 71](#_Toc183412092)

[Appendix B DATASETS 74](#_Toc183412093)

[Appendix C CURRICULUM VITAE 75](#_Toc183412094)

# **List of Figures**

[Figure 1 *Random Forest* 8](#_Toc183412095)

[Figure 2 *Theoretical Framework* 14](#_Toc183412096)

[Figure 3 *Distribution of CASH COSTS* 24](#_Toc183412097)

[Figure 4 *Distribution of TOTAL COSTS* 25](#_Toc183412098)

[Figure 5 *Distribution of Yield per hectare (kg)* 26](#_Toc183412099)

[Figure 6 *Distribution of Farmgate price (pesos/kg)* 27](#_Toc183412100)

[Figure 7 *Distribution of Crop Types* 28](#_Toc183412101)

[Figure 8 *Distribution of Seasons* 29](#_Toc183412102)

[Figure 9 *Total Number of CASH COSTS Compared to total amount of Yield per hectare (kg)* 30](#_Toc183412103)

[Figure 10 *Total Number of Farmgate price Compared to total amount of Yield per hectare (kg)* 31](#_Toc183412104)

[Figure 11 *Boxplots for Key Numerical Features to Detect Outliers* 32](#_Toc183412105)

[Figure 12 *Missing Data Visualization with Missingno on Corn Data* 37](#_Toc183412106)

[Figure 13 *Missing Data Visualization with Missingno on Palay Data* 37](#_Toc183412107)

[Figure 14 *Actual vs Predicted Yield per Hectare on Corn* 44](#_Toc183412108)

[Figure 15 *Actual vs Predicted Yield per Hectare on Palay* 46](#_Toc183412109)

[Figure 16 *Corn Yield Predictions for the Next Ten Years* 47](#_Toc183412110)

[Figure 17 *Palay Yield Predictions for the Next Ten Years* 48](#_Toc183412111)

# **List of Tables**

[Table 1 *Summary of Data Collection of Corn* 20](#_Toc183412112)

[Table 2 *Summary of Data Collection of Rice* 21](#_Toc183412113)

[Table 3 *Data Types and Null Count for each columns* 23](#_Toc183412114)

[Table 4 *Summary Statistics for the Numerical Features* 33](#_Toc183412115)

[Table 5 *Process for the Data Cleaning* 35](#_Toc183412116)

[Table 6 *Results for Converting the Data into Numeric Values* 36](#_Toc183412117)

[Table 7 *Process of Feature Engineering* 39](#_Toc183412118)

[Table 8 *Model Hyperparameters* 40](#_Toc183412119)

[Table 9 *Top Five Best Parameters on Corn Data* 42](#_Toc183412120)

[Table 10 *Top Five Worse Parameters on Corn Data* 42](#_Toc183412121)

[Table 11 *Top Five Best Parameters on Palay Data* 43](#_Toc183412122)

[Table 12 *Top Five Worse Parameters on Palay Data* 43](#_Toc183412123)

[Table 13 *Corn Metrics Results* 44](#_Toc183412124)

[Table 14 *Palay Metric Results* 45](#_Toc183412125)

# Chapter 1 **THE RESEARCH DESCRIPTION**

## Introduction

Crop yield forecasting is a critical activity to improve agricultural productivity that, in turn, results in higher quantities, better resource management, and stabilization of markets. About the Philippines, with different types of farming lands, farming is an important industry in the country specifically rice and corn which are the mainstay of the Philippine economy and food production. Therefore, reliable demand forecasting is critical, not only for farmers but also for all the players in the agricultural value chain (Madayag & Estanislao, 2021). However, markets for agricultural produce are very volatile since there are numerous factors that affect them: meteorological, environmental, and socio-economic factors. Such fluctuations of demand can cause problems in the supply chain, high wastage and even fluctuations in food security which is not good for the producers and consumers (Shin, 2021). A survey of the previous literature reveals that techniques widely used in the Philippines to forecast crop yields have relied heavily on historical information and analysts’ opinions.

These conventional techniques, although functional, do not capture the intricate relationships that influence crop demand adequately (Ibañez & Monterola, 2023). A survey of national agricultural organizations indicates that long-range planning involves mainly historical data on production and market conditions and sometimes qualitative descriptions of the environment. Despite being highly adopted in organizations, these approaches fail to demonstrate some key strengths, especially in their responsiveness to dynamic changes in weather patterns, market forces, and society. This underlines the necessity for the development of improved methods of predicting future trends based on the analysis of various kinds of data to provide timely and accurate forecasts. This research then aims to forecast future rice and corn yield in the Philippines using machine learning. It is more efficient than other methods especially when dealing with large amounts of data, complex patterns, and the ability to integrate new data collected.

Accurate forecasting plays a very crucial role in agricultural and economic planning because it assists farmers in identifying the type of crops to grow and using the available resources to the best effect, reducing losses and at the same time enhancing food security. Furthermore, the market stakeholders can enhance the supply chain management and the functionality of the market, and the Department of Agriculture can develop suitable policies and support measures based on the information.

## Research Objectives

This project aims to develop and implement a machine learning-based system for predicting the long-term yield for *palay* and corn in the Philippines, utilizing regression learning models to enhance accuracy and reliability.

Specifically, this study aims to:

1. gather and preprocess data relevant to the yield for *palay* and corn from the Philippine Statistics Authority’s (PSA) public data.
2. develop a regression machine learning model that forecasts crop yield based on historical crop yields, weather patterns, and socio-economic factors using a random forest regressor.
3. evaluate the performance of the regression models using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R²) metrics.
4. provide detailed analysis and visualization of forecast results to help local farmers and market stakeholders understand the factors influencing crop yield.

## Scope and Limitations

This project deals with the application of advanced regression learning algorithms in creating crop yield forecast models for rice and corn in the Philippines. The scope encompasses several key areas: gathering and cleaning of data, creation, testing, interpretation, and presentation of the model. Data relevant to the research proposal will be retrieved from the Philippine Statistic Authority (PSA) and other credible sources that can provide the data. Due to the nature of the data collection process, the model’s reliability depends on the historical data availability and quality. Laws and policies (such as tariffs), global conditions, and imports are not incorporated.

This study will only focus on creating a model using a Random Forest Regressor. Other models such as Neural Networks will not be explored. In addition, the following metrics will be utilized for evaluating the model's performance: R-squared (R²), Mean Absolute Error (MAE), and Mean Square Error (MSE).

The study will be focused on forecasting the yield for rice and corn within a period of ten-year period considering historical data on the crop yields, crop production, and region population as the features of the inputs. As such, long-term forecasts generally contain more uncertainties in the results interpretation process.

The scope involves the detailed analysis and visualization of the forecast results to aid local farmers and market stakeholders in comprehending the factors affecting crop yield. The expected output will also consist of the analysis of forecast results to determine the patterns of crop yields concerning historical crop yields, crop production, and the population of the region. Line plots, bar charts, and heat maps will be used to share the forecasted data. The findings and suggestions will include the identification of the best strategies that the stakeholders can adopt in terms of resource utilization, development, and decision-making.

## Significance of the Study

The study is significant to the following individuals and parties:

**Local Farmers:** With the help of the model, local farmers may be able to make well-informed decisions regarding crop production and resource allocation by gaining important insights into future crop yield.

**Market Stakeholders:** The system may help market participants cut waste, optimize market operations, and plan and manage supply chains.

**Department of Agriculture:** Forecasting the crop yield may help in subsidies, supporting programs, and infrastructure to enhance crop yields hence food security. Furthermore, the findings can extend to disaster preparedness and contingency planning so that the food supply chains may not be vulnerable to climatic volatility and other forms of adversity affecting the economy. This plan and the various data obtained from the analysis can be useful for the Department of Agriculture to identify ways that would enable the sector to develop as well as plan for necessary interventions.

**Future Researchers:** By showing how regression learning models can be used practically to estimate crop yield, this study may advance the field of agricultural analytics and lay the groundwork for future research and development in this domain.

# Chapter 2 **REVIEW OF RELATED LITERATURE AND STUDIES**

This chapter provides an overview of the literature that informed and guided the researchers and the study, with implications for the findings. It is divided into five main sections: an overview of crop yield forecasting, benefits of crop yield forecasting, factors influencing the yield for crops, techniques on time series forecasting random forest on regression problems, and a comparative study on machine learning for crop yield forecasting.

## Overview of Crop Yield Forecasting

Crop demand forecasting is critical to agriculture's ability to plan and manage resources effectively. In terms of production, influence on the economy, and availability of food, rice, also known as *palay*, and corn rank among the Philippines' most important crops. The cultivation of rice has long been a thriving sector in the Philippines, according to data from the Philippine Statistics Authority, and it continues to be the primary diet of the Filipino people. Furthermore, corn is used as a feed ingredient and as a raw material in the production of animal feed in several regions of the world. Factors such as income, population, and climate change contribute to the consumption of these crops.

Crop yield has previously been predicted using traditional techniques such as trend analysis and econometric models. By examining patterns, trend analysis uses historical data to predict needs in the future. Using statistical techniques to account for the influence of other variables, econometric models facilitate comprehension of the underlying pattern of agricultural demand. Moreover, Joshi's (2019) research on the usage of econometric models such as ARIMA for forecasting revealed that while they are helpful, they are not very effective. A further investigation conducted by Li & Kockelman (2019) aimed to evaluate the applicability of each approach by contrasting machine learning and conventional econometric models for travel decisions.

Machine learning is an innovation that has rewritten yield forecasting as it allows for calculations of various factors that are interrelated. Handling of big data in the forecasting process is now accomplished using machine learning algorithms like Random Forest, SVM, and Neural Networks. These models are especially effective in terms of incident detection and forecast of relationships including such factors as non-linear relationships, and high dimensionality. Crop yield estimation is an important step in the process of agricultural planning and is used to estimate the amount of yield the farmer is likely to obtain from a particular piece of land. It covers the use of past records, the weather conditions at the time of planting and harvesting the soil type, and many more to develop heuristic models to forecast the yields. This is the reason why this forecasting plays a significant role as seen in food security, economic planning, and resource allocation issues (Morales & Villalobos, 2023).

## Benefits of Crop Yield Forecasting

Crop yield forecasting has a lot of usefulness in many areas of agriculture from the field level decision-making to governmental-level economic planning. These benefits are achieved in the form of improved predictability, effective resource management, and formulation of sound policies that are central to sustainable agriculture.

Forecasting of crop yield using predictive analytics offers farmers and other stakeholders in agriculture essential data on volumes expected to be produced. This makes it easier to plan concerning the labor force, tools, and even the funds that are needed for the project. Weather information enables accurate determination of yields thus reducing the variability of price and market demand thus steadying the farmer’s and investor’s income​ (D. M. P. W. Dissanayake et al., 2023).

The other major advantage of crop yield forecasting is the efficient and proper management of available resources. This way, the farmers are able to determine how much of a crop is likely to grow, and thus are in a position to decide on the best time to use water, fertilizers, and pesticides among others. This not only assists in making the cost of farming cheap by avoiding the use of excess inputs but also reduces the effects on the environment. For instance, accurate irrigation methods powered by yield predictions have demonstrated a lot of water conservation hence very valuable in construed environments (Pankaj et al., 2023).

Crop yield predictions are useful in policymaking involving farming. These forecasts are useful to governments and regulatory authorities to anticipate food imports and exportation, control food stocks, and develop subsidies and support for farmers. Realizing today’s possible yield makes it possible to set policies regarding food production and distribution in the market. This is especially so where the economy is largely driven by agriculture (Gera & Jain, 2023).

## Factors Influencing the Yield for Crops

Yield is the product of many interrelated factors that can include climatic factors, the practices that are used when cultivating crops, and even the available technology. Climate is the first since crops require specific temperature, light, and moisture conditions for the crops to grow. Heat stress affects photosynthesis is vital for crop growth and therefore goes hand in hand with yield outputs (Hatfield et al., 2015). Soil also has a very important unearthing, for it provides the necessary nutrients needed for plant growth in soils that are most fertile. Indeed, nutrient availability, soil pH, structure, and organic matter affect how plants can tap into these nutrients (Guo et al., 2021).

Another environmental factor is water supply where both low and high supply are equally destructive to crops in the field. Irrigation therefore plays a central role in achieving the balance that is needed for crop production (Supit et al., 2012). In the agricultural sector, for instance, measures like crops intersecting with another crop to provision for pest control and building fertility in the soil thus improving yields. As a couple of practices, integrated pest management and the use of the right type of fertilizers in the areas of the soils are the basic requirements for the protection of crops from pests and diseases and the supply of nutrients required for the growth of crops (Bhojwani et al., 2017; Jain et al., 2017).

There is growing importance of technology and genes in today’s farming. This is due to the adoption of high-quality seeds that have been genetically modified to suit specific stresses such as drought and diseases common in the region. GPS, drones, IoT sensors, and other smart devices help to monitor and control plant health, soil, and micro-climates, to apply inputs and management practices for maximum yield on limited resources (Jain et al., 2017).

Finally, factors such as access to the market and other policies in the economy swell up the farming decisions and practices thus the yields. Farmers can be motivated to adopt sustainable farming practices and new technologies when governments promote their policies of subsidy grants to modern agriculture. On the other hand, the policies not supporting sustainable practices may help to slow down yield improvement (Wankhede et al., 2017; Raj, 2021).

## Techniques on Time Series Forecasting Random Forest on Regression Problems

Random Forest Regression is a powerful ensemble learning method widely used in various fields such as finance, healthcare, and marketing. It leverages the strength of multiple decision trees to produce a robust predictive model, making it one of the go-to algorithms for regression tasks. By averaging the predictions of numerous decision trees, random forest mitigates the risk of overfitting and improves generalization, providing more accurate and stable predictions. (Sengupta, 2024).

Figure 1   
*Random Forest*

A diagram of a tree

Description automatically generated

Other hyperparameters in the Random Forest model included the number of trees to be created (n\_estimators), the maximum depth of the trees (max\_depth), and the minimum number of samples required to split a node (min\_samples\_split). (Chauhan, 2021).

Prior studies have shown that Random Forest could be applied to regression problems, capable of handling non-linear relationships and interactions between variables and therefore a great tool for predictive modeling in agriculture. Belgiu and Drăguţ (2016) described its advantages regarding flexibility and stability across different regression and classification tasks; other authors have recently reflected similar opinions. These studies mentioned the method’s success rate for handling and solving such issues and provided evidence to support the general applicability of Random Forest for regression problems in agriculture.

When it comes to agricultural data analysis, time series forecasting and Random Forest regression become two of the most crucial methods especially when it comes to crop yield estimation. Time series analysis on the other hand is a process of modeling and forecasting data points that occur in time series. It is especially important for the analysis of seasonal fluctuations and tendencies in agricultural yields. Seasonality in crop production data can for instance be handled by employing ARIMA (AutoRegressive Integrated Moving Average) or exponential smoothing techniques as the means of structuring crop yield forecasting (Hyndman et al., 2018).

## A Comparative Study on Machine Learning for Crop Yield Forecasting

Crop yield forecasting has a great importance in increasing the agricultural production, effective management of resources and food security. While using Random Forest methods, it is possible to achieve higher accuracy of yield predictions taking into account numerous environmental, meteorological and economic indicators. In this regard, many papers have been written to analyze the performance of a range of the machine learning algorithms in estimating crop yields with conditions.

Anderson et al. (2019) focused only on wheat yield prediction in the American Midwest. They incorporated satellite imagery coupled with climate data to train a Random Forest model, in a bid to account for spatial and temporal heterogeneities of yields. Normalization of the weather data and conversion of satellite images to obtain NDVI, which measures vegetative health, were done in this process. On their model, they got an R² of 0.78 which indicates that 78% of the yield variability could be attributed to the predictors. The Mean Absolute Error is equal to 1.2 tons per hectare and the Mean Squared Error of 2.5 tons per hectare squared proved its effectiveness and forecasting ability in the regional setting. In their study,

Baker and Smith (2020) assessed corn yield in several states in the United States of America using soil information, yield records, and current climate information. Using Support Vector Machines (SVM), they cleaned the data and managed missing values as well as scaling features. Specifically, the detailed soil nutrient data and weather conditions were incorporated into their model. The coefficients of determination, R², were 0.85, showing the high accuracy of the model, and the errors, mean absolute error at 0.95 tons ha⁻¹ and mean squared error at 1.8 tons ha⁻¹², also corroborated model effectiveness in predicting under different circumstances.

Chen et al. (2018) followed a different strategy by using climate data, satellite imagery, and crop management data of soybean yield in Brazil. They used Neural Networks with deep learning models that were LSTM to capture the temporal behavior of the crops. Their preprocessing included constructing features with weather anomalies and phenological information. An R² of 0.90, MAE of 0.8, and MSE of 1.1 tons per hectare squared was achieved in the study to prove that the deep learning model is more efficient in capturing complex non-linear relationships and large data sets.

Davies et al., (2021) focused on rice yield prediction for India using drone imagery of the field, local weather data, and IoT devices to monitor in real-time. They used an ensemble method that includes both the Gradient Boosting Machines and the Random Forests, with preprocessing methods laying much emphasis on the removal of the outliers and selection of the most important features using the importance of the variables. The ensemble model gave an R² of 0.83, MAE of 1.1, and MSE of 2.2 tons per hectare squared, all of which show the stability and accuracy in a technologically developed forecasting system.

In a study by Evans & Lee (2017), they modeled the yield of barley in Australia using historical yield data, soil moisture data obtained from sensors, and climate data. XGBoost—a decision-tree-based feature, which is an ensemble method and is known for its efficiency and accuracy was used. While developing their model, they have also included some additional preprocessing steps to address seasonal changes and correctly categorize climate phases. The results were significant: the R² = 0.88, MAE = 0.7, and MSE = 1.5 tons per hectare squared, which proves good compatibility of the model and its capacity to work with a large number of sources and types of data.

## Definition of Terms

The definitions of the terms that follow correspond to the usage in this study:

**Artificial intelligence (AI)**. A branch of computer science research that creates and examines techniques and software that allow machines to sense their surroundings and use learning to take actions that maximize their chances of accomplishing specific goals (Fetzer, 1990).

**Data Analytics**. a multidisciplinary area of study that concludes data sources by applying mathematics, statistics, computer science, and other analysis techniques (Staff, 2024).

**Dataset**. A group of data, frequently shown in tabular form, correlates to one or more database tables, each row of which represents a specific record from the relevant data set and each column of which represents a different variable.

**Decision Tree**. A structure of an organization resembling a tree used to represent decisions and their possible outcomes in data analysis and decision-making is called a decision tree. (Alam, 2024).

**Ensemble Learning**. A method of machine learning that combines multiple learners (neural networks, regression models, etc.) to create more accurate predictions (Zhou, 2021).

**Machine Learning (ML)**. The investigation of computational algorithms that can automatically get better with practice and data usage. It is regarded as a component of AI (Issam El Naqa & Murphy, 2015).

**Mean Absolute Error (MAE)**.This metric represents the average of the absolute difference between the actual and predicted values in the dataset (M Waqar Ahmed, 2023).

**Metrics**. A section of each machine learning pipeline that shows learning progress (Bajaj, 2022).

**NumPy**. A Python library that includes functions for matrices, linear algebra, and the Fourier transform when working with arrays (W3Schools.com, 2024).

**Pandas**. An open-source, robust Python module for data evaluation and processing. Pandas is a collection of functions and data structures for effective data processing (GeeksforGeeks, 2020).

**Random Forest**. A popular machine learning technique that integrates the output of several decision trees to get a single result is patented by Leo Breiman and Adele Cutler (Genuer & Poggi, 2020).

**Regression Analysis**. A collection of statistical techniques for determining the correlations between one or more distinct variables often referred to as "indicators," "covariates that were," "descriptive variables," or "features" and a dependent variable often referred to as the "outcome," "response," or "label" in machine learning jargon (Hassan, 2024).

**Regression Model**. A function that characterizes the connection among a variety of independent variables and a response, dependent, or target variable is provided by a machine learning model (IMSL by Perforce, 2021).

**Mean Square Error (MSE)**. A metric that assesses how much the actual values depart on average from the values forecast by a statistical model (Frost, 2023).

**R Squared (R²)**. A statistical metric that is used in regression models to calculate the percentage of the variance of the dependent variables that the independent variable can account for (Ihechikara Vincent Abba, 2023).

**Scikit-learn**. A collection of Python algorithms for both supervised and unsupervised learning (Codecademy, 2024).

# Chapter 3 **THEORETICAL FRAMEWORK**

This chapter discusses the overview of the theoretical framework and explains the consecutive, structured approach used in the development of a crop yield forecasting model based on machine learning algorithms.

Figure 2   
*Theoretical Framework*

Figure 1 shows a schematic representation of the machine learning pipeline implemented in this research. This approach is made more structured and step-by-step by this modularization, which optimizes each cycle modification that affects subsequent ones and reduces inefficiencies in the process of creating a crop yield forecasting model. The pipeline enables this study to scale better, experiment more easily, and reduce the ability for human-made errors ensuring the reliability and predictability of the predictions.

The theoretical framework for the model development and training process is based on established methodologies within the machine learning domain, particularly the structured approach outlined by IBM in their guide on Machine Learning Pipelines. This framework provides a comprehensive overview of the key stages involved in building, training, and evaluating machine learning models, ensuring a systematic and effective workflow (IBM, 2024).

## Data Collection

The first stage is data collection. Considering the complexity and potential accessibility issues with multiple data sources, the focus will be simplified. The data will be gathered from PSA's Open Statistical Databases, specifically covering historical data on previous crop production and historical data on the population. This approach helps to have a reliable data source while leaving the study’s scope manageable.

## Data Preprocessing

Following the data analysis comes data preprocessing, where data is cleaned and shaped to fit the algorithm’s input requirements appropriately. This also involves the process of imputation whereby some values are inferred for instance by using mean, median, or mode depending on which is applicable, feature scaling which is used to scale features to the appropriate range, or data transformation where data type may be converted to desired format for further analysis. Furthermore, categorical variables may be required to be in a format suitable for the Machine Learning Model and may be converted using methods such as One Hot Encoding or Label Encoding.

## Feature Engineering

This stage is the process of deriving more features from the raw data to achieve a closer representation of the actual pattern and, therefore, enhance the performance of machine learning algorithms. The yield for crops is examined by considering some factors, for example, population increase and historical production. Features of population growth are generated by analyzing the trends in the demography and its relationship with food. These features are designed to extend the dataset to include the correlations between these factors, which gives a solid foundation for training.

## Model Development and Training

This stage is centered on the development and training of a regression model using a random forest regressor, an enhanced form of the tree-based learning algorithm that predicts the demand for crops in the future. It incorporates practices such as splitting the data into training and testing sets, optimizing the hyperparameters to get the best outcome, and gradually refining the model. Random Forest Regressor will handle large volumes of data that identify the relationship between features, which is a very important factor of crops that will help in identifying demand factors.

To find the optimal combination of hyperparameters, the researcher will integrate a Python function called GridSearchCV, which is a very useful tool for hyperparameter tuning. GridSearchCV carries out cross-validation to determine the performance of the model with various parameters specified in the parameter grid. The first step is to identify the values of the hyperparameters to be tuned. The majority of hyperparameters for Random Forest Regressor are n\_estimators, max\_depth, min\_samples\_split, and min\_samples\_leaf. Other hyperparameters such as max\_features. The number of features to draw upon when searching for the best split can also be added. Additionally, by applying GridSearchCV, the researcher will configure the grid search to compare the hyperparameters. This includes defining the estimator (Random Forest Regressor in this case), the hyperparameters to tune (parameter grid), the metric for model evaluation ( e. g. Mean Absolute Error, RMSE, or R²), and the number of folds of cross-validation.

## Model Evaluation

The model goes through Model Evaluation once it has been trained. To determine whether the model satisfies the required accuracy and reliability standards, this stage evaluates its performance using a variety of metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R²).

The formula for Mean Absolute Error (MAE) is as follows:

where:

* *n* is represented as the total number of observations.
* Σ is referred to as the summation operator which means that the absolute errors should be summed.
* refers to the actual value of the 𝑖-th observation.
* 𝑥 is referred to as the predicted value of the 𝑖-th observation.

The formula for Root Mean Squared Error (RMSE) is as follows:

where:

* is represented by the predicted value of the 𝑖-th observation.
* is defined as the observed (actual) value of the 𝑖-th observation.
* *n* is represented by the total number of observations.

The root mean square error, a statistic that indicates the average difference between the model's predicted values and the dataset's actual values, can be used to evaluate how well a regression model fits a dataset. A model's ability to "fit" a dataset improves with a decreased root mean square error (RMSE).

The formula for R2 is as follows:

where:

* The sum of squares resulting from regression is known as (explained sum of squares).
* represents the sum of all squares.

This involves carrying out regression analysis on the dependent and independent variables of the observed data (observations) to find the line of correlation and often this is arrived at by using a regression model. This regression line will help to point out the existence of the relationship between the variables.

## Detailed Forecast

Finally, the study will utilize the entire dataset and predict the next 10 years. The results will then be cascaded with the use of data analysis and visualization techniques to understand crop yield patterns. To present the data, relevant charts such as line plots and data tables will be applied to help visualize the data better and ease the interpretation of data results. These reports will then be turned over to the Department of Agriculture in Misamis Occidental to help ensure that those who are in the local farming business, as well as the market individuals, have a clue about the determinants of crop yield. Thus, the visualization of the impact of different factors such as historical yields, weather conditions, and socioeconomic factors will help the study to offer relevant conclusions on how to improve agricultural planning and policy.

Moreover, the visualizations will also indicate possible opportunities to increase crop yield and the supply of crops in the market. For instance, patterns of the effects of unfavorable climate conditions on crop productivity will guide the approaches to be taken to mitigate the effects of climate change, especially on agricultural practices. Thus, by integrating detailed forecast analysis and visualization, the study will guarantee that the model’s outputs are not only statistically sound but also relevant among the stakeholders in the agricultural sector.

# Chapter 4 **RESULTS AND DISCUSSIONS**

This chapter discusses the analysis of the results obtained from the data collected and the modeling processes to give a discourse on the study conclusion, recommendation, and the application of the developed forecasting model in agriculture.

## Data Collection

The researchers have gathered PSA's Open Statistical Databases, specifically covering historical data on the previous crop production and historical data on the population. The tables below summarize all the data obtained.

Table 1   
*Summary of Data Collection of Corn*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Features | Year | Geolocation | Type of Crops | Season | | |
|  |  |  |  | **Dry** | **Average** | **Wet** |
| CASH COSTS | 2002 | Philippines | Corn | 5255 | 5430 | 5594 |
| NON-CASH COSTS | 2002 | Philippines | Corn | 1355 | 1377 | 1398 |
| IMPUTED COSTS | 2002 | Philippines | Corn | 4535 | 4008 | 3516 |
| TOTAL COSTS | 2002 | Philippines | Corn | 11145 | 10815 | 10508 |
| GROSS RETURNS | 2002 | Philippines | Corn | 14096 | 13554 | 13046 |
| RETURNS ABOVE CASH COSTS | 2002 | Philippines | Corn | 8841 | 8124 | 7452 |
| RETURNS ABOVE CASH AND NON-CASH COSTS | 2002 | Philippines | Corn | 7486 | 6747 | 6054 |
| NET RETURNS | 2002 | Philippines | Corn | 2951 | 2739 | 2538 |
| NET PROFIT-COST RATIO | 2002 | Philippines | Corn | 0.26 | 0.25 | 0.24 |
| Cost per kilogram (pesos) | 2002 | Philippines | Corn | 5.59 | 5.65 | 5.7 |
| Yield per hectare (kg) | 2002 | Philippines | Corn | 1994 | 1915 | 1842 |
| Farmgate price (pesos/kg) | 2002 | Philippines | Corn | 7.07 | 7.08 | 7.08 |
| Population | 2002 | Philippines | Corn | 81285572 | 81285572 | 81285572 |
| Annual % Change | 2002 | Philippines | Corn | 2.08 | 2.08 | 2.08 |
| Total |  |  |  |  |  |  |

This table provides information on cost, return, and other economic aspects regarding the production of corn in the Philippines for 2002 based on seasons such as dry, average, and wet, following the other features. This table gives the summary of cost and return analysis of corn production in the Philippines for the year 2002 during the three seasons, namely Dry, Average, and Wet.

Table 2   
*Summary of Data Collection of Rice*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Features | Year | Geolocation | Type of Crops | Season | | |
|  |  |  |  | **Dry** | **Average** | **Wet** |
| CASH COSTS | 2002 | Philippines | Rice | 9106 | 9325 | 9549 |
| NON-CASH COSTS | 2002 | Philippines | Rice | 6982 | 6964 | 6946 |
| IMPUTED COSTS | 2002 | Philippines | Rice | 5466 | 5068 | 4667 |
| TOTAL COSTS | 2002 | Philippines | Rice | 21554 | 21357 | 21162 |
| GROSS RETURNS | 2002 | Philippines | Rice | 27394 | 27483 | 27573 |
| RETURNS ABOVE CASH COSTS | 2002 | Philippines | Rice | 18288 | 18158 | 18024 |
| RETURNS ABOVE CASH AND NON-CASH COSTS | 2002 | Philippines | Rice | 11306 | 11194 | 11078 |
| NET RETURNS | 2002 | Philippines | Rice | 5840 | 6126 | 6411 |
| NET PROFIT-COST RATIO | 2002 | Philippines | Rice | 0.27 | 0.29 | 0.3 |
| Cost per kilogram (pesos) | 2002 | Philippines | Rice | 6.91 | 6.7 | 6.49 |
| Yield per hectare (kg) | 2002 | Philippines | Rice | 3118 | 3188 | 3260 |
| Farmgate price (pesos/kg) | 2002 | Philippines | Rice | 8.79 | 8.62 | 8.46 |
| Population | 2002 | Philippines | Rice | 81285572 | 81285572 | 81285572 |
| Annual % Change | 2002 | Philippines | Rice | 2.08 | 2.08 | 2.08 |
| Total |  |  |  |  |  |  |

This table provides information on cost, return, and other economic aspects regarding the production of rice in the Philippines for 2002 based on seasons such as dry, average, and wet, following the other features. This table gives the summary of cost and return analysis of rice production in the Philippines for the year 2002 during the three seasons, namely Dry, Average, and Wet.

## Data Exploration

Analyzing raw datasets to spot patterns and characteristics and, ideally, determine correlations between various variables is known as data exploration. It is useful in revealing the dataset's structure, outlier existence, and data value distribution.

Table 3   
*Data Types and Null Count for each columns*

|  |  |  |
| --- | --- | --- |
| Columns | Data Types | Null Count |
| Year | int64 | 0 |
| Type | object | 0 |
| Geolocation | object | 0 |
| Season | object | 0 |
| CASH COSTS | float64 | 26 |
| NON-CASH COSTS | float64 | 26 |
| IMPUTED COSTS | float64 | 26 |
| TOTAL COSTS | float64 | 26 |
| GROSS RETURNS | float64 | 26 |
| RETURN ABOVE CASH COSTS | float64 |  |
| RETURN ABOVE CASH AND NON-CASH COSTS | float64 | 26 |
| NET RETURNS | float64 | 26 |
| NET PROFIT-COST RATIO | float64 | 26 |
| Cost per kilogram (pesos) | float64 | 26 |
| Yield per hectare (kg) | float64 | 26 |
| Farmgate price (pesos/kg) | float64 | 36 |
| Population | int64 | 0 |
| Annual % Change | float64 | 0 |
| Season\_encoded | int32 | 0 |
| Geolocation\_encoded | int32 | 0 |

Table 3 shows the summary of the data types and the number of missing values for each of the columns. This table is very important in determining the structure of the data and the state of cleanliness of the data in question before any analysis is done. The dataset comprises several fields including ‘Year’, Type’, ‘Geolocation’, ‘Season’, and several cost and returns-based parameters for example ‘CASH COSTS’, ‘NON-CASH COSTS’, ‘IMPUTED COSTS’, ‘TOTAL COSTS’, ‘GROSS RETURNS’, among others. The data type for each column is mentioned right from ’Year’ as int64 to the financial ratios as float64 and the categorical data is represented as integers such as ’Season\_encoded’ and ’Geolocation\_encoded’. Table 3 also identifies the number of null values by columns; most of which are the financial variables derived from them are ‘NET RETURNS’ and ‘COST per kilogram (pesos)’ with 26 samples missing the data. This information is critical when pre-processing the data because managing these null values is important to the proper statistical analysis or building of the machine learning models.

INCLUDE HISTOGRAM COMPARISONS OF:

1. Distribution of CASH COSTS
2. Distribution of TOTAL COSTS
3. Distribution of Yield per hectare (kg)
4. Distribution of Farmgate price (pesos/kg)
5. Distribution of Crop types
6. Distribution of Seasons

Figure 3   
*Distribution of CASH COSTS*

A graph of a distribution of cash costs

Description automatically generated

Figure 2 is the cash cost distribution of crop production as observed from the dataset of 2002 to 2021. This histogram, accompanied by the kernel density estimation (KDE), represents the distribution of different cash cost values reported in the years. The x-axis shows various cash cost peso figures, whilst the y-axis provides the number of times that these costs were reported in the dataset. The dispersion of the KDE towards the bell-shaped curve suggests that majority of the cash costs are grouped around the middle of the distribution most farmers are likely to incur normal cash costs under normal circumstances. The spread and tails of the distribution give information about the variability, years with low or high farming costs can be due to market prices, changes in agricultural policy, or unfavorable climatic conditions.

Figure 4   
*Distribution of TOTAL COSTS*

A graph of a distribution of costs

Description automatically generated

This figure shows the total costs incurred in crop production which have been analyzed over the years 2002 to 2021 are presented in figure 3 below. The use of histogram with kernel density estimate works in this paper to present the overall cost that farmers incur, whether on a cash or non-cash basis including depreciation and imputed labor cost. The x-axis defines the total costs in pesos, and the y-axis defines how often the cost totals were observed within the data. The primary concentration around the median is the highest peak in the histogram which shows that many farmers might expect the overall cost around this value. This means that there is a considerable shift in some years, and these may be a result of factors outside the production function that impact agricultural operations, prices of inputs, changes in policies on agriculture or even weather conditions. This graph gives a good insight of the financial issues in the agricultural sector particularly years of strange costs that may affect profitability.

Figure 5   
*Distribution of Yield per hectare (kg)*

A graph with a line going up

Description automatically generated

This figure shows the amounts of yield per hectare for crops, in kilograms. The results indicate that the highest frequency is 3000 kg, which implies that this yield rate is dominant in the dataset. It also possesses a peak thus showing there is a mode regarding the distribution of the yield, with few observations having a yield that is higher or lower than the peak value. This distribution can help, at least, to have an idea of the average productivity to be expected from the agricultural datasets and, therefore, to adjust the expected yield regarding historical values.

Figure 6   
*Distribution of Farmgate price (pesos/kg)*

A graph of a distribution of a number of farms

Description automatically generated with medium confidence

This figure shows the extent of farmgate prices in pesos per kilogram. It also indicates that the highest density of the price range is 10 pesos per kg, which can be considered as the mode of distribution. The spread of the data Shows that there is variability of the Prices that farmers get for their produce depending on several factors including market demand; the cost of production; and the quality of produce among others. It is important to understand this distribution to undertake economic analysis within the agricultural sector and assist various players in the sector in making better decisions regarding pricing, production, and sales.

Figure 7   
*Distribution of Crop Types*

A blue and orange rectangular shapes

Description automatically generated

This figure shows an equal proportion of both crop types, which has been emphasized in the collection of data. This equal distribution also implies that the set of data used is very general and does not incline towards the benefit of one crop over the other, thereby enabling the researchers to compare and evaluate the practice and results of agriculture concerning the principal food crops in the Philippines.

Figure 8   
*Distribution of Seasons*

A chart of different colors

Description automatically generated with medium confidence

This figure shows the distribution of data entries and has been grouped into average, dry, and wet seasons. The equal distribution across the three categories shows that the dataset covers all aspects of agriculture, in various seasonal environments. This uniformity helps the analysis factor in seasonality of crop production and costs, to provide an understanding of how the different seasons influence agricultural output and costs in the region.

INCLUDE SCATTERPLOT COMPARISONS OF:

1. CASH COSTS vs Yield per hectare (kg)
2. Farmgate price vs Yield per hectare (kg)

Figure 9   
*Total Number of CASH COSTS Compared to total amount of Yield per hectare (kg)*

A diagram of a graph

Description automatically generated

This figure shows the regression of cash costs on the yield per hectare in kg. Every dot on the graph reflects the cash cost that corresponds to a definite yield value. The distribution also appears to have a general trend that higher cash costs are usually followed by higher yields which implies that more investment could lead to better yields of crops. With increases in the cash costs, there is an increase in the dispersion of points which shows that costs vary in terms of yield gains.

Figure 10   
*Total Number of Farmgate price Compared to total amount of Yield per hectare (kg)*

A diagram of a farm

Description automatically generated

This figure shows the farmgate prices for cocoa concerning yield per hectare ($/kg). On the scatter plot, it can be observed that yield per hectare is positively related to the farmgate prices with a certain amount of dispersion. This relationship illustrates how the cost of produce as determined in the market affects either the productivity of agriculture or the feasibility of raising production.

Figure 11   
*Boxplots for Key Numerical Features to Detect Outliers*

*A diagram of a graph

Description automatically generated with medium confidence*

This figure contains boxplots which are used to identify outliers in the yield per hectare across various costs and returns such as cash cost, non-cash cost, returns above cash cost, and net returns. The plots also show the median, IQR (Interquartile Range), and possible outliers of the respective categories. A very large spread in the “Yield per hectare” proves that there are large variations in the yields, which depend on such factors as farming methods, use of inputs, and location.

Table 4   
*Summary Statistics for the Numerical Features*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Year** | **CASH COSTS** | **NON-CASH COSTS** | **IMPUTED COSTS** | **TOTAL COSTS** | **GROSS RETURNS** | **RETURNS ABOVE CASH COSTS** |
| count | 120 | 120 | 120 | 120 | 120 | 120 | 120 |
| mean | 2011.5 | 14628.7083 | 8550.3 | 7826.28333 | 30321.1417 | 44443.35 | 29961.3 |
| std | 5.790459 | 4748.69217 | 4730.00361 | 2279.58812 | 11077.3716 | 17888.9867 | 13425.9554 |
| min | 2002 | 5255 | 1306 | 3516 | 10508 | 12162 | 6056 |
| 25% | 2006.75 | 11854.5 | 2502.75 | 6152.75 | 22371 | 33055.25 | 21107 |
| 50% | 2011.5 | 14616 | 9079 | 7427.5 | 27965 | 41797 | 27858 |
| 75% | 2016.25 | 17312.25 | 12494.75 | 9044.5 | 40581.75 | 57651.75 | 39478.5 |
| max | 2021 | 25621 | 17558 | 13656 | 52524 | 81031 | 59645 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RETURNS ABOVE CASH AND NON-CASH COSTS** | **NET RETURNS** | **NET PROFIT-COST RATIO** | **Cost per kilogram (pesos)** | **Yield per hectare (kg)** | **Farmgate price (pesos/kg)** | **Population** | **Annual % Change** |
| 120 | 120 | 120 | 120 | 120 | 120 | 1.20E+02 | 120 |
| 21749.5667 | 14169.1083 | 0.423833 | 8.953167 | 3349.71667 | 12.425583 | 9.73E+07 | 1.8055 |
| 9392.59647 | 7402.80231 | 0.139013 | 1.951299 | 613.206285 | 3.261699 | 9.93E+06 | 0.145918 |
| 4708 | 457 | 0.04 | 5.43 | 1755 | 6.59 | 8.13E+07 | 1.51 |
| 15262.25 | 8766.75 | 0.37 | 7.365 | 3099.5 | 10.4525 | 8.91E+07 | 1.695 |
| 20341.5 | 14048.5 | 0.4 | 8.675 | 3518.5 | 11.925 | 9.72E+07 | 1.795 |
| 25749.5 | 16994.5 | 0.48 | 10.8275 | 3790 | 14.22 | 1.05E+08 | 1.8925 |
| 46289 | 35012 | 0.81 | 12.75 | 4197 | 20.33 | 1.14E+08 | 2.08 |

This table contains the aggregation of numerical characteristics of the data, which characterizes the costs and returns in agriculture throughout the years, as well as other economic indicators. There are different columns in the table: Cash Costs, Non-Cash Costs, Imputed Costs, Total Costs, Gross Returns, Returns Above Cash Costs, Net Returns, Cost per Kilogram, Yield per Hectare, Farmgate Price, Population and Annual % Change.

## Data Cleaning

Data cleaning is the procedure for locating and eliminating errors, inconsistencies, and inaccuracies from a dataset.

The process of data cleaning in this system starts with reading the data set saved in a CSV file into a DataFrame which enhances its usability. Subsequently, the notebook marks out the presence of missing values in the given dataset, which is helpful for understanding whether any of the columns in a dataset needs further handling such as imputation. The last data preparation step is to convert some of the columns from objects to numerical types for better analysis in the next stages of data preparation. It is performed to allow later calculations and analysis of the data collected as a part of this project, in the correct format. These steps taken together make the dataset ready for more granular analysis and modeling which can be built on this clean and consistent platform.

Table 5   
*Process for the Data Cleaning*

|  |  |  |
| --- | --- | --- |
| Step # | Description | Code Used |
| 1 | Loading the data using the CSV file into a DataFrame | data = pd.read\_csv('filtered\_data.csv') |
| 2 | Verifying for missing values on all variables | data.isnull().sum() |
| 3 | Converting certain columns from string data type to numeric data type | data[columns\_to\_convert] = data[columns\_to\_convert].apply(pd.to\_numeric, errors='coerce') |
| 4 | Visualizing Missing Data | import missingno as msno  msno.matrix(data)  plt.show() |
| 5 | Impute Missing Values | columns\_to\_impute\_median = ['CASH COSTS', 'NON-CASH COSTS', 'IMPUTED COSTS', 'TOTAL COSTS', 'GROSS RETURNS', 'RETURNS ABOVE CASH COSTS', 'RETURNS ABOVE CASH AND NON-CASH COSTS', 'NET RETURNS', 'NET PROFIT-COST RATIO', 'Cost per kilogram (pesos)', 'Yield per hectare (kg)', 'Farmgate price (pesos/kg)']  data[columns\_to\_impute\_median] = data[columns\_to\_impute\_median].fillna(data[columns\_to\_impute\_median].median()) |

Table 6   
*Results for Converting the Data into Numeric Values*

|  |  |
| --- | --- |
| Column | Data Type |
| Year | int64 |
| Type | object |
| Geolocation | object |
| Season | object |
| CASH COSTS | float64 |
| NON-CASH COSTS | float64 |
| IMPUTED COSTS | float64 |
| TOTAL COSTS | float64 |
| GROSS RETURNS | float64 |
| RETURNS ABOVE CASH COSTS | float64 |
| RETURNS ABOVE CASH AND NON-CASH COSTS | float64 |
| NET RETURNS | float64 |
| NET PROFIT-COST RATIO | float64 |
| Cost per kilogram (pesos) | float64 |
| Yield per hectare (kg) | float64 |
| Farmgate price (pesos/kg) | float64 |
| Population | int64 |
| Annual % Change | float64 |

Figure 12   
*Missing Data Visualization with Missingno on Corn Data*

A graph with many lines

Description automatically generated with medium confidence

Figure 13   
*Missing Data Visualization with Missingno on Palay Data*

*A close-up of a grey rectangular object

Description automatically generated*

## Feature Engineering

This system mainly concerns data preprocessing and model assessment, and there are some steps that do not directly engage feature construction. The process starts with typecasting, in which categorical features are converted into numerical feature types using a different type for non-convertibles. This is important as the data set can be operated on mathematically and analyzed ready for modeling which is the next phase. After data type conversion, the notebook also includes visualization of missing data using *missingno* library as this will help determine the pattern of missing data and assist in the decision on how best to treat gaps in data.

Other steps include imputing the median to fill missing values; a process that completes the data set without the effects of biases that could be brought by mean imputation particularly when outliers are involved. Even though it does not refer to feature engineering directly, the preprocessing steps are crucial for preparing the data for proper modeling. The notebook also contains a feature, which is not feature engineering but rather model validation and application *– backtesting* with historical data and making predictions on future data. However, these steps are important for improving the accuracy of the model to address practical forecasting problems.

Finally, for the purpose of feature engineering for the model, it may be useful to create interactions of different features, expand the features into polynomials to make the model learn more complex time series information, or encode categorical features in some interesting manner to reveal more about the data. These steps would entail direct modification or addition of features to bring the best suitable performance of the model. If finer grained feature engineering is needed, for example, to include environmental or economic variables directly, extra processes would need to be incorporated into this phase

Table 7   
*Process of Feature Engineering*

|  |  |  |  |
| --- | --- | --- | --- |
| **Step #** | **Description** | **Code Example** | **Feature Engineering Aspect** |
| 1 | Convert string columns to numeric | data[columns\_to\_convert].apply(pd.to\_numeric, errors='coerce') | Handling non-numeric values |
| 2 | Visualize missing data | msno.matrix(data) | Identifying patterns of missing data |
| 3 | Impute missing values using median imputation | data[columns\_to\_impute\_median].fillna(data[columns\_to\_impute\_median].median()) | Dealing with missing data |
| 4 | Backtesting model with historical data | backtesting\_forecaster(forecaster, y=y\_train, exog=X\_train, steps=steps, metric='mean\_squared\_error') | Validating model performance |
| 5 | Predicting future values | forecaster.predict(steps=len(y\_test), exog=X\_test) | Applying the model to new data |

## Model Development and Training

The Random Forest method was chosen for this study because it is effective at classification tasks and does well when overfitting is an issue. The model was trained and its performance was improved by hyperparameter adjustment. The interpretation and applicability of the model's parameters are covered in detail in the section that follows.

Table 8   
*Model Hyperparameters*

|  |  |
| --- | --- |
| **n\_estimators**  Values: [100, 150, 200, 300] | The amount of trees in the forest is determined by this parameter. An increase in the number of trees generally improves the tree's performance, but the effect is not as strong and has a limit. Various values were tested, and cross-validation was used to choose the optimal value from the range of values. |
| **max\_depth**  Values: [None] | restricts each tree's depth range. As a result, overfitting is lessened because trees are unable to recognize intricate patterns in the data and noise. To be effective, the model's depth and generalization capacity were further improved. |
| **min\_samples\_split**  Values: [2, 5, 10] | indicates the bare minimum of samples needed for a node's split. Since the splitting is not performed on tiny data sample sizes, higher numbers reduce overfitting. |
| **max\_features**  Values: ['sqrt', 'log2'] | regulates how many traits are taken into account for every split. By modifying this, performance is enhanced and the correlation between trees is decreased. To optimize the model, a variety of settings were evaluated, including sqrt. |
| **Criterion**  Values: ['squared\_error'] | identifies the function that will be applied to evaluate a split's quality. After comparing the two options—gini and entropy—the better option was applied to the finished model. |
| **min\_samples\_leaf**  Values: [1, 2, 4] | referred to as the smallest quantity of samples that can fit inside a single leaf node. By making it stronger, the model was able to train without concentrating on any one pattern. Several were tested in order to determine which value would work best for the dataset. |

In the context of crop yield forecasting, this type of optimization aims at tuning several hyper-parameters important for increasing the models ability to make accurate predictions. The others are n\_estimators: the number of trees in the forest and max\_depth: the depth of each tree within the forest, which helps to reduce the occurrence of over-fitting, a trend which is characteristic of most models. Also, min\_samples\_split and min\_samples\_leaf are adjusted to avoid the problem of overfitting where the trees follow the data closely, and max\_features decides the number of features to consider when splitting, thus incorporates a measure of random inconsistence likely to inhibit the bias for the most prominent features. These hyperparameters are systematically adjusted by cross-validation to identify the best hyperparameters that would give high level of reliability and accuracy of future crop yields which is crucial in farming.

**GridSearch CV Training Results**

Table 9   
*Top Five Best Parameters on Corn Data*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rank | Criterion | Max Depth | Max Features | Min Samples Leaf | Min Samples Split | N Estimators | Mean Test Score |
| 1 | squared\_error | None | log2 | 2 | 10 | 100 | -996.619 |
| 2 | squared\_error | None | log2 | 2 | 10 | 150 | -1001.33 |
| 3 | squared\_error | None | log2 | 1 | 10 | 100 | -1017.63 |
| 4 | squared\_error | None | log2 | 2 | 2 | 150 | -1023.22 |
| 5 | squared\_error | None | log2 | 2 | 5 | 150 | -1024.43 |

Table 10   
*Top Five Worse Parameters on Corn Data*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rank | Criterion | Max Depth | Max Features | Min Samples Leaf | Min Samples Split | N Estimators | Mean Test Score |
| 1 | squared\_error | None | log2 | 2 | 10 | 100 | -996.619 |
| 2 | squared\_error | None | log2 | 2 | 10 | 150 | -1001.33 |
| 3 | squared\_error | None | log2 | 1 | 10 | 100 | -1017.63 |
| 4 | squared\_error | None | log2 | 2 | 2 | 150 | -1023.22 |
| 5 | squared\_error | None | log2 | 2 | 5 | 150 | -1024.43 |

The best parameters are log2 for the maximum features and None for the maximum depth so trees grow to the maximum depth, which probably does not overfit complex patterns well. These setups differ, for the most part, in terms of the number of estimators (100 to 150) and the minimum samples that are required at the leaves and for splitting; this implies an optimal trade-off to avoid overfitting while at the same time, allow for model complexity. Notably, the worst parameter combinations resemble the best ones, but they have worse performance probably because of a different training subset or noise sensitivity, This means that small changes in hyperparameters such as the number of trees (n\_estimators) can have a huge impact on performance in predictive modeling.

Table 11   
*Top Five Best Parameters on Palay Data*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rank | Criterion | Max Depth | Max Features | Min Samples Leaf | Min Samples Split | N Estimators | Mean Test Score |
| 1 | squared\_error | None | log2 | 1 | 2 | 300 | -10717.4 |
| 2 | squared\_error | None | log2 | 1 | 2 | 200 | -10813.4 |
| 3 | squared\_error | None | log2 | 1 | 2 | 150 | -11172.7 |
| 4 | squared\_error | None | sqrt | 1 | 2 | 200 | -11390.5 |
| 5 | squared\_error | None | log2 | 1 | 2 | 100 | -11421 |

Table 12   
*Top Five Worse Parameters on Palay Data*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rank | Criterion | Max Depth | Max Features | Min Samples Leaf | Min Samples Split | N Estimators | Mean Test Score |
| 1 | squared\_error | None | log2 | 1 | 2 | 300 | -10717.4 |
| 2 | squared\_error | None | log2 | 1 | 2 | 200 | -10813.4 |
| 3 | squared\_error | None | log2 | 1 | 2 | 150 | -11172.7 |
| 4 | squared\_error | None | sqrt | 1 | 2 | 200 | -11390.5 |
| 5 | squared\_error | None | log2 | 1 | 2 | 100 | -11421 |

The optimal parameters usually involve squared\_error criterion, None for the max depth, log 2 for max features which means fewer features will be taken at each split, and 1 for min samples leaf and 2 for min samples split to avoid splitting into detail. Most of them differ by the number of estimators, which ranges from 100 to 300 where higher numbers tend to be better, but again, there is a limitation to the model complexity to avoid overfitting. The same setting in both best and worst tables mean that even a small change in data split or other conditions during cross-validation can greatly influence the performance, which is typical for agriculture where external conditions heavily impact the results of model tuning.

## Model Evaluation

Table 13   
*Corn Metrics Results*

|  |  |
| --- | --- |
| Metric | Value |
| Mean Absolute Error | 579.03 |
| Mean Squared Error | 13.09 |
| R-square | 0.9723 |

The parameters of the Random Forest Regression model show good results on the test data set. It is depicted from the graph that an MSE of 579.037 stands for the squared average difference between the actual and predicted values. The MSE is generally low as it is, however, it is sometimes highly sensitive to outliers because of the squaring of errors. The mean absolute error of 13.095 offers a more familiar measure of average deviation from the target variable that states that predictions are off, on average, by about 13 units in the same scale as the target variable. In addition, the R-squared (R²) value of 0.972 strongly supports the idea that the chosen model offers a good fit explaining approximately 97.2 % variation of a target variable. Altogether, these findings imply that the Random Forest model holds a small amount of error and a strong amount of predictability.

Figure 14   
*Actual vs Predicted Yield per Hectare on Corn*

A graph with red and orange dots

Description automatically generated

Random Forest model seems to capture majority of the patterns in the data well since most of the predicted values are very close to the actual values. This situation shows that the model is a good performer in terms of generalizing the outcomes on unseen data in the test set. There are cases where actual and predicted values vary with a small margin and can be observed in some cases. Such variations could be because of ‘noise’ in the data collection processes or inherent flaws of this approach in dealing with some classes of problems. However, these differences are minor and do not have a large effect on the execution of the model. This close relationship is evidenced by quantitative parameters such as R-squared of 0.972 and the MAE of 13.10 kg/ha that confirm model fitness. The model is well capable of using the data to predict the yields of corn and does so relatively accurately, with the estimated values closely resembling the true values. The few deviations made are insignificant, thereby making the model appropriate for future projection of corn yield per hectare.

Table 14   
*Palay Metric Results*

|  |  |
| --- | --- |
| Metric | Value |
| Mean Absolute Error | 8893.94 |
| Mean Squared Error | 70.83 |
| R-square | 0.8961 |

The results of the Random Forest Regression model show satisfactory performance on the test data set. The Mean Squared Error (MSE) is 8,893.94 which calculates the mean of the square of the difference between the estimated and observed values. Though this error is higher than ideal the extent of this error largely depends on the scale of the target variable. MAE equals 70.83, which means that on average the model is off the true values by approximately 70 in the same units as the target variable. The coefficient of determination (R-sq) is calculated to be 0.896 which tells about 89.6 percent variation in the target variable. This translates to a relatively strong correlation between the features and the target variable, which however is not as robust as the previous evaluation. These results suggest that, save for specific instances, the model is quite accurate and, if necessary, can be refined, and supplemented with more data or new variables.

Figure 15   
*Actual vs Predicted Yield per Hectare on Palay*

A graph with red and orange lines

Description automatically generated

That most of the values from the red and orange line are almost overlapping indicates that the model is useful in estimating the yields for palay. It can be noted that the predictions observe the fluctuations in the actual data points in the same way as an overall movement is observed. Nevertheless, there is some scatter in certain segments where the predicted values are slightly higher or lower than the actual values at some points of peaks or nadirs. Some of these differences may be due to random fluctuations in the information or slight imperfections in the potential to apply one or another pattern. In general, the model can provide a good forecast of palay yield since most of the figures forecasted by the model correlate well with the actual yield. This makes the model useful for the purpose of making forecasts, the small discrepancies noted in a few cases notwithstanding.

## Detailed Forecast Analysis

Figure 16   
*Corn Yield Predictions for the Next Ten Years*

A graph with a red line

Description automatically generated

Figure 12 shows the forecast of corn production yields in the next decade revealing a slight decline in 2023 and a small volatile fluctuation in the remained years till 2033. The initial drop in the model can indicate that the model is detecting upcoming difficulties or changes shortly and may be attributable to climate conditions, changes in policies on agriculture, or market changes. This is followed by more stable values suggesting that the model predicts conditions to become more settled and deliver a reasonably consistent level in the subsequent years. Such a type of forecasting is helpful for farmers and policymakers who anticipate the change and regulate production depending on the forecasted trends.

Figure 17   
*Palay Yield Predictions for the Next Ten Years*

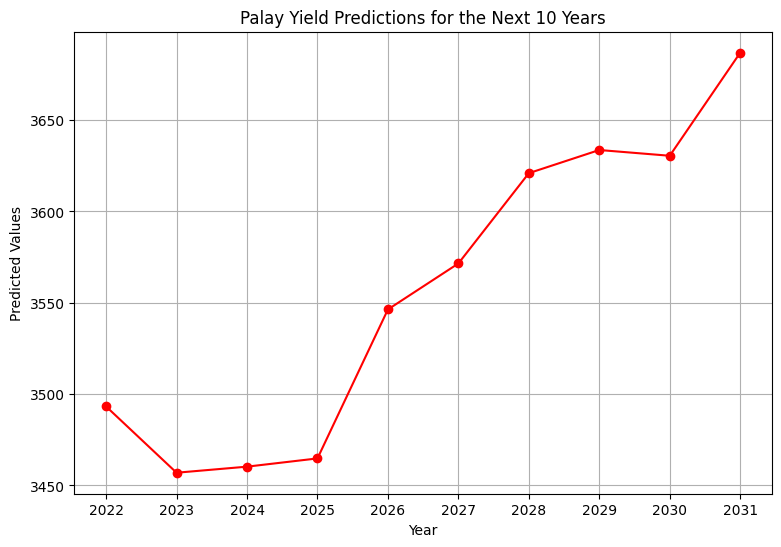
**

Table 13 shows the forecasted yield for palay up to the coming ten years whereby the yield will slightly decrease in the first few years then gradually rise. After that, there is a slight increase from 2023 to 2025, a further increase from 2029 and a sharp rise in 2031. This upward trend in the later years may be due to better practices in agriculture or technology or better environment conditions expected by the model. Such forecasts are useful for strategic purposes in the agricultural industry and offer information that can be used in decision making regarding crop productivity, investment in technologies and resources to obtain optimum yield results in the subsequent ten years.

# Chapter 5 **CONCLUSION AND RECOMMENDATIONS**

## Conclusion

This project was able to establish a machine learning-driven system for projecting long-term crop yields of palay and corn in the Philippines. The system is based on the random forest regression, the decision tree was chosen due to its capacity to work with big and complex datasets. This model performs well in modeling non-linear patterns and is less volatile across different data settings; thus, suitable for the agricultural data of this research. The system development process included extensive stages such as data gathering from the Philippine Statistics Authority and data preprocessing due to data errors and missing values. Features were drawn out using feature extraction where much emphasis was put on the features that would give meaningful predictors of the outcome variable and the subsequent model tuning to enhance the predictive performance of the model.

The research also benefits the agricultural field by improving the ability to predict crop yield more efficiently. This advancement is important to enhance agricultural yield, efficient utilization of resources, and market regulation in the Philippines and has made a significant contribution to food security at the community and national levels. The effectiveness of the model was measured based on Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Coefficient of determination (R-squared). All these metrics together confirmed the model proving that it can generate accurate and reliable forecasts within acceptable error bounds.

The results showed that the random forest model provided the values of exceptional predictive accuracy; the value of R² explained that a large amount of variance in the crop yields was explained well. The lower MAE and RMSE values also similarly supported the conclusion that the proposed model can, with reasonable accuracy estimate crop yields, thereby enriching the possibilities of precise planning and management of plant production. The forecasting system brings a lot of practical values to the farmers, the agricultural businesses, and the related policy makers. Through its realistic yield estimates it underpins effective and informed agricultural planning, efficient utilization of resources, and managing the risks arising from crop yield volatilities and instability hence enhancing food production stability and economic stability.

## Recommendations

For future research to improve on the findings of this thesis on long-term crop yield forecasting using machine learning in the Philippines, the following enhancements should be considered. Adding real time data would add a lot of value to the model; climatic and soil data obtained through *Internet of Thing*s sensors and satellite would enable the model to respond to current climate conditions. Further, the expansion of other machine learning methods, like Deep Learning Neural Networks, especially Long Short-Term Memory (LSTM) network, might serve temporal dependencies and non-linear relationship, thereby enhancing yield forecasts. Moreover, applying the described model to the other agricultural areas with different climates could be useful to verify its efficiency in different climate zones to get deeper understanding of its versatility.

# **Bibliography**

Alam, M. (2024, April 18). *What is a Decision Tree? Definition, Examples, Model, Advantages, Analysis, and Samples*. IdeaScale.com. <https://ideascale.com/blog/what-is-a-decision-tree/>

Bajaj, A. (2022, July 21). *Performance Metrics in Machine Learning [Complete Guide] - neptune.ai*. Neptune.ai. <https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide>

Chauhan, A. (2021, February 23). *Random Forest Classifier and its Hyperparameters - Analytics Vidhya*. Medium; Analytics Vidhya. <https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6>

Codecademy. (2024). *What is Scikit-Learn?* Codecademy; Codecademy. <https://www.codecademy.com/article/scikit-learn>

Fetzer, J. H. (1990). What is Artificial Intelligence? *Studies in Cognitive Systems, 3*–27. <https://doi.org/10.1007/978-94-009-1900-6_1>

Frost, J. (2023, May 6). *Root Mean Square Error (RMSE)*. Statistics by Jim. <https://statisticsbyjim.com/regression/root-mean-square-error-rmse/>

Genuer, R., & Poggi, J.-M. (2020). Random Forests. *Use R!, 33*–55. <https://doi.org/10.1007/978-3-030-56485-8_3>

Harris, C. R., K Jarrod Millman, van, Ralf Gommers, Pauli Virtanen, Cournapeau, D., Wieser, E. S., Taylor, J., Berg, S., Smith, N. J., Kern, R., Matti Picus, Hoyer, S., Kerkwijk, van, Brett, M., Haldane, A., Fernández, J., Wiebe, M. W., Peterson, P., & Gérard-Marchant, P. (2020*).* Array programming with NumPy. *Nature, 585*(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>

Hassan, M. (2024, March 25). *Regression Analysis - Methods, Types, and Examples*. Research Method. <https://researchmethod.net/regression-analysis/>

Ihechikara Vincent Abba. (2023, March 28). *What is R-squared? R2 Value Meaning and Definition*. FreeCodeCamp.org. <https://www.freecodecamp.org/news/what-is-r-squared-r2-value-meaning-and-definition/>

Issam El Naqa, & Murphy, M. J. (2015). What Is Machine Learning*? Springer EBooks, 3*–11. <https://doi.org/10.1007/978-3-319-18305-3_1>

Joshi, S. (2019). *Time Series Analysis and Forecasting of the US Housing Starts Using Econometric and Machine Learning Model*. ArXiv.org. <https://arxiv.org/abs/1905.07848>

Khaki, S., & Wang, L. (2019). Crop Yield Prediction Using Deep Neural Networks*.* *Frontiers in Plant Science, 10*. <https://doi.org/10.3389/fpls.2019.00621>

Maazallahi, A., Thota, S., Kondaboina, Naga Prasad, Muktineni, V., Annem, D., Rokkam, Abhi Stephen, Amini, M. H., Salari, M. A., Norouzzadeh, P., Snir, E., & Rahmani, B. (2024). *Naive Bayes and Random Forest for Crop Yield Prediction*. ArXiv.org. <https://arxiv.org/abs/2404.15392>

Rana, H., Muhammad Umer Farooq, Abdul Karim Kazi, Mirza Adnan Baig, & Muhammad Ali Akhtar. (2024). Prediction of Agricultural Commodity Prices using Big Data Framework*.* *Engineering, Technology and Applied Science Research/Engineering, Technology and Applied Science Research, 14*(1), 12652–12658. <https://doi.org/10.48084/etasr.6468>

Rong, S., & Zhang Bao-wen. (2018). The research of regression model in the machine learning field. *MATEC Web of Conferences, 176*, 01033–01033. <https://doi.org/10.1051/matecconf/201817601033>

Sari, M., Duran, S., Huseyin Kutlu, Bulent Guloglu, & Atik, Z. (2024). Various optimized machine learning techniques to predict agricultural commodity prices*. Neural Computing & Applications, 36*(19), 11439–11459. <https://doi.org/10.1007/s00521-024-09679-x>

Singh, K., Kumar, R., Thakur, P., Singh, H., & Singh, S. (2024). Dengue Fever Outbreak Prediction Using Machine Learning Models: A Comparative Study*.* *Lecture Notes in Networks and Systems, 443*–455. <https://doi.org/10.1007/978-981-99-7820-5_36>

Spyros Makridakis, Spiliotis, E., & Vassilios Assimakopoulos. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PloS One, 13*(3), e0194889–e0194889. <https://doi.org/10.1371/journal.pone.0194889>

Staff, C. (2024). *Data Analytics: Definition, Uses, Examples, and More*. Coursera. <https://www.coursera.org/articles/data-analytics>

Suleymanov, A., Tuktarova, I., Belan, L., Suleymanov, R., Ilyusya Gabbasova, & Lyasan Araslanova. (2023*).* Spatial prediction of soil properties using random forest, k-nearest neighbors, and cubist approaches in the foothills of the Ural Mountains, Russia*. Modeling Earth Systems and Environment, 9*(3), 3461–3471. <https://doi.org/10.1007/s40808-023-01723-4>

Sumido, E. C., Larmie Feliscuzo, & Chris Jordan Aliac. (2023, December). *Pest Classification And Prediction: Analyzing The Impact Of Weather To Pest Occurence Through Machine Learning.* ResearchGate; unknown. <https://www.researchgate.net/publication/377331491_PEST_CLASSIFICATION_AND_PREDICTION_ANALYZING_THE_IMPACT_OF_WEATHER_TO_PEST_OCCURRENCE_THROUGH_MACHINE_LEARNING>

Umberger, W. (2024). *Demographic Trends: Implications for Future Food Demand*. Retrieved July 10, 2024, from [https://www.kansascityfed.org/documents/7023/umberger-paper.pdf](https://www.kansascityfed.org/documents/7023/umberger-paper.pdf#:~:text=URL%3A%20https%3A%2F%2Fwww.kansascityfed.org%2Fdocuments%2F7023%2Fumberger)

Usharani Bhimavarapu, Gopi Battineni, & Nalini Chintalapudi. (2023). Improved Optimization Algorithm in LSTM to Predict Crop Yield*. Computers, 12*(1), 10–10. <https://doi.org/10.3390/computers12010010>

Valera, H.G., Mayorga, J., Pede, V. O., & Mishra, A. K. (2022*).* Estimating food demand and the impact of market shocks on food expenditures: The case for the Philippines and missing price data. *Q Open, 2*(2). <https://doi.org/10.1093/qopen/qoac030>

Venkatesh, K., & K. Jairam Naik. (2024). *An ensemble transfer learning for nutrient deficiency identification and yield-loss prediction in crops*. Multimedia Tools and Applications. <https://doi.org/10.1007/s11042-024-18592-3>

Vijayatai Hukare, Vidya Kumbhar, & Shah, S. K. (2023*).* Machine Learning Methods for Crop Yield Prediction.*Communications in Computer and Information Science, 195*–209. <https://doi.org/10.1007/978-3-031-43605-5_15>

Vovsha, P. (2021). Comparison of Traditional Econometric Models and Machine Learning Methods in the Context of Travel Decision Making and Perspectives for Synergy*.* *Studies in Computational Intelligence, 177*–186. <https://doi.org/10.1007/978-3-030-75583-6_18>

Yang, X., Hua, Z., Li, L., Huo, X., & Zhao, Z. (2024). Multi-source information fusion-driven corn yield prediction using the Random Forest from the perspective of Agricultural and Forestry Economic Management. *Scientific Reports, 14*(1), 4052. <https://doi.org/10.1038/s41598-024-54354-9>

Zhou, Z.-H. (2021). Ensemble Learning*. Springer EBooks, 181*–210. <https://doi.org/10.1007/978-981-15-1967-3_8>

Ibañez, S. C., & Monterola, C. P. (2023). A Global Forecasting Approach to Large-Scale Crop Production Prediction with Time Series Transformers. *Agriculture*, *13*(9), 1855–1855. <https://doi.org/10.3390/agriculture13091855>

Philippine Statistics Authority (2024). *Rice and Corn Situation and Outlook Reports.* PSA.gov.ph. <https://www.psa.gov.ph/content/rice-and-corn-situation-and-outlook-reports>

Shin, H. (2021, February 10). *Council Post: Preparing For The Unpredictable In Food And Agriculture*. Forbes. <https://www.forbes.com/sites/forbesbusinessdevelopmentcouncil/2021/02/11/preparing-for-the-unpredictable-in-food-and-agriculture/>

Morales, A., & Villalobos, F. J. (2023). Using machine learning for crop yield prediction in the past or the future. Frontiers in Plant Science, 14. <https://doi.org/10.3389/fpls.2023.1128388>

‌D. M. P. W. Dissanayake, R. M. K. T. Rathnayake, & L. L. Gihan Chathuranga. (2023). Crop Yield Forecasting using Machine Learning Techniques - A Systematic Literature Review. KDU Journal of Multidisciplinary Studies, 5(1), 54–65. <https://doi.org/10.4038/kjms.v5i1.62>

‌Gera, R., & Jain, A. (2023). Predicting Crop Yield in Smart Agriculture Using IoT and Machine Learning for Sustainable Development. Communications in Computer and Information Science, 64–76. <https://doi.org/10.1007/978-3-031-47055-4_6>

‌Hatfield, J. L., & Prueger, J. H. (2015). Temperature extremes: Effect on plant growth and development. Weather and Climate Extremes, 10, 4–10. <https://doi.org/10.1016/j.wace.2015.08.001>

‌I. Supit, C.A. van Diepen, Wit, A. J. W. de, Wolf, J., Kabat, P., Baruth, B., & Ludwig, F. (2012). Assessing climate change effects on European crop yields using the Crop Growth Monitoring System and a weather generator. Agricultural and Forest Meteorology, 164, 96–111. <https://doi.org/10.1016/j.agrformet.2012.05.005>

‌Wankhede, D. S. (2020). Analysis and Prediction of Soil Nutrients pH,N,P,K for Crop Using Machine Learning Classifier: A Review. EAI/Springer Innovations in Communication and Computing, 111–121. <https://doi.org/10.1007/978-3-030-49795-8_10>

‌

# Appendix A **SOURCE CODE**

## Corn

import pandas as pd

data = pd.read\_csv('corn\_data.csv')

data.head()

### Data Preprocessing

# Check for missing values in the merged dataset

data.isnull().sum()

data.dtypes

# Convert object columns to numeric, forcing errors to NaN (in case of any non-numeric values)

columns\_to\_convert = ['Year', 'CASH COSTS', 'NON-CASH COSTS', 'IMPUTED COSTS', 'TOTAL COSTS', 'GROSS RETURNS',

                      'RETURNS ABOVE CASH COSTS', 'RETURNS ABOVE CASH AND NON-CASH COSTS', 'NET RETURNS',

                      'NET PROFIT-COST RATIO', 'Cost per kilogram (pesos)', 'Yield per hectare (kg)',

                      'Farmgate price (pesos/kg)']

# Convert the columns to numeric, forcing errors='coerce' to handle any invalid parsing

data[columns\_to\_convert] = data[columns\_to\_convert].apply(pd.to\_numeric, errors='coerce'

data.isnull().sum()

data.head()

### Data Cleaning

import missingno as msno

import matplotlib.pyplot as plt

# Visualize missing data using missingno

msno.matrix(data)

plt.title('Missing Data Visualization with missingno')

plt.show()

# Impute the specified columns using median imputation first

columns\_to\_impute\_median = ['Year', 'CASH COSTS', 'NON-CASH COSTS', 'IMPUTED COSTS', 'TOTAL COSTS', 'GROSS RETURNS',

                            'RETURNS ABOVE CASH COSTS', 'RETURNS ABOVE CASH AND NON-CASH COSTS', 'NET RETURNS',

                            'NET PROFIT-COST RATIO', 'Cost per kilogram (pesos)', 'Yield per hectare (kg)',

                            'Farmgate price (pesos/kg)']

# Apply median imputation

data[columns\_to\_impute\_median] = data[columns\_to\_impute\_median].fillna(data[columns\_to\_impute\_median].median())

# Now check how many missing values remain

data.isnull().sum()

import seaborn as sns

# Plot distribution of key numerical variables

plt.figure(figsize=(12, 8))

sns.histplot(data['CASH COSTS'], kde=True)

plt.title('Distribution of CASH COSTS')

plt.xlabel('CASH COSTS')

plt.ylabel('Frequency')

plt.show()

plt.figure(figsize=(12, 8))

sns.histplot(data['TOTAL COSTS'], kde=True)

plt.title('Distribution of TOTAL COSTS')

plt.xlabel('TOTAL COSTS')

plt.ylabel('Frequency')

plt.show()

plt.figure(figsize=(12, 8))

sns.histplot(data['Yield per hectare (kg)'], kde=True)

plt.title('Distribution of Yield per hectare (kg)')

plt.xlabel('Yield per hectare (kg)')

plt.ylabel('Frequency')

plt.show()

plt.figure(figsize=(12, 8))

sns.histplot(data['Farmgate price (pesos/kg)'], kde=True)

plt.title('Distribution of Farmgate price (pesos/kg)')

plt.xlabel('Farmgate price (pesos/kg)')

plt.ylabel('Frequency')

plt.show()

# Plot categorical variable 'Type'

plt.figure(figsize=(12, 8))

sns.countplot(data=data, x='Type')

plt.title('Distribution of Crop Types')

plt.xlabel('Crop Type')

plt.ylabel('Count')

plt.show()

# Plot categorical variable 'Season'

plt.figure(figsize=(12, 8))

sns.countplot(data=data, x='Season')

plt.title('Distribution of Seasons')

plt.xlabel('Season')

plt.ylabel('Count')

plt.show()

# Scatterplot: CASH COSTS vs Yield per hectare (kg)

plt.figure(figsize=(12, 8))

sns.scatterplot(data=data, x='CASH COSTS', y='Yield per hectare (kg)')

plt.title('CASH COSTS vs Yield per hectare (kg)')

plt.xlabel('CASH COSTS')

plt.ylabel('Yield per hectare (kg)')

plt.show()

# Scatterplot: Farmgate price vs Yield per hectare (kg)

plt.figure(figsize=(12, 8))

sns.scatterplot(data=data, x='Farmgate price (pesos/kg)', y='Yield per hectare (kg)')

plt.title('Farmgate price vs Yield per hectare (kg)')

plt.xlabel('Farmgate price (pesos/kg)')

plt.ylabel('Yield per hectare (kg)')

plt.show()

# Select only the numeric columns for correlation analysis

numeric\_columns = data.select\_dtypes(include=['float64', 'int64'])

# Generate a correlation matrix for the numerical features

correlation\_matrix = numeric\_columns.corr()

# Set up the plot for a heatmap

plt.figure(figsize=(14, 10))

# Create a heatmap of the correlation matrix

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)

plt.title('Correlation Matrix of Numerical Features')

plt.show()

# Set up the figure for outlier detection using boxplots

plt.figure(figsize=(14, 10))

# Create boxplots for key numerical features to visualize outliers

sns.boxplot(data=data[['CASH COSTS', 'TOTAL COSTS', 'Farmgate price (pesos/kg)', 'Yield per hectare (kg)']])

plt.title('Boxplots for Outlier Detection')

plt.ylabel('Values')

plt.xticks(rotation=45)

plt.show()

# Generate summary statistics for the numerical features

summary\_statistics = data.describe()

# Display the summary statistics

summary\_statistics

# Function to cap outliers based on IQR

def cap\_outliers(column):

    Q1 = column.quantile(0.25)

    Q3 = column.quantile(0.75)

    IQR = Q3 - Q1

    # Define lower and upper bounds for capping

    lower\_bound = Q1 - 1.5 \* IQR

    upper\_bound = Q3 + 1.5 \* IQR

    # Cap the outliers

    column\_capped = column.clip(lower=lower\_bound, upper=upper\_bound)

    return column\_capped

# Capping outliers for the selected numerical features

data['CASH COSTS'] = cap\_outliers(data['CASH COSTS'])

data['TOTAL COSTS'] = cap\_outliers(data['TOTAL COSTS'])

data['Farmgate price (pesos/kg)'] = cap\_outliers(data['Farmgate price (pesos/kg)'])

data['Yield per hectare (kg)'] = cap\_outliers(data['Yield per hectare (kg)'])

# Check the summary statistics after capping outliers to see the effect

summary\_statistics\_after\_capping = data.describe()

# Display the summary statistics after capping

summary\_statistics\_after\_capping

from sklearn.preprocessing import LabelEncoder

# Re-encode the categorical columns 'Season' and 'Geolocation' to ensure they are present

label\_encoder = LabelEncoder()

# Create encoded versions of 'Season' and 'Geolocation'

data['Season\_encoded'] = label\_encoder.fit\_transform(data['Season'])

data['Geolocation\_encoded'] = label\_encoder.fit\_transform(data['Geolocation'])

# Now retry the correlation analysis on the proposed features

proposed\_features = [

    'Year', 'CASH COSTS', 'NON-CASH COSTS', 'IMPUTED COSTS', 'TOTAL COSTS',

    'GROSS RETURNS', 'RETURNS ABOVE CASH COSTS', 'Population',

    'Season\_encoded', 'Geolocation\_encoded', 'Yield per hectare (kg)'

]

# Generate a correlation matrix for the numerical features

correlation\_matrix\_proposed = data[proposed\_features].corr()

# Set up the plot for a heatmap of the proposed features' correlation matrix

plt.figure(figsize=(14, 10))

# Create a heatmap of the correlation matrix for proposed features

sns.heatmap(correlation\_matrix\_proposed, annot=True, cmap='coolwarm', linewidths=0.5)

plt.title('Correlation Matrix of Proposed Features')

plt.show()

### Model Development and Training

from sklearn.model\_selection import train\_test\_split

# Define the target variable and features

target\_variable = 'Yield per hectare (kg)'  # or 'Farmgate price (pesos/kg)' if predicting prices

features = [

    'CASH COSTS', 'NON-CASH COSTS', 'IMPUTED COSTS', 'Year',

    'TOTAL COSTS', 'GROSS RETURNS', 'RETURNS ABOVE CASH COSTS',

    'Population', 'Season\_encoded'

]

# Split the data into training and testing sets (80% training, 20% testing)

X = data[features]

y = data[target\_variable]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Check the shapes of the resulting datasets

X\_train.shape, X\_test.shape, y\_train.shape, y\_test.shape

#### **Model Evaluation Random Forest Regressor**

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score, mean\_absolute\_error

# Initialize the Random Forest regressor

rf\_model = RandomForestRegressor(random\_state=42)

# Fit the model on the training data

rf\_model.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred\_rf = rf\_model.predict(X\_test)

# Evaluate the model's performance

mse\_rf = mean\_squared\_error(y\_test, y\_pred\_rf)

mae\_rf = mean\_absolute\_error(y\_test, y\_pred\_rf)

r2\_rf = r2\_score(y\_test, y\_pred\_rf)

print(f'Mean Squared Error: {mse\_rf}')

print(f'Mean Absolute Error: {mae\_rf}')

print(f'R-squared: {r2\_rf}')

### Model Analysis and Visualization

# Reset the indices of train and test sets

X\_train = X\_train.reset\_index(drop=True)

X\_test = X\_test.reset\_index(drop=True)

y\_train = y\_train.reset\_index(drop=True)

y\_test = y\_test.reset\_index(drop=True)

# Plot the training and test data with predictions

plt.figure(figsize=(12, 6))

# Plot training data

plt.plot(range(len(y\_train)), y\_train, label='Train', color='blue', marker='o')

# Plot testing data

plt.plot(range(len(y\_train), len(y\_train) + len(y\_test)), y\_test, label='Test', color='red', marker='o')

# Plot predictions

plt.plot(range(len(y\_train), len(y\_train) + len(y\_test)), y\_pred\_rf, label='Predictions', color='orange', linestyle='--', marker='o')

# Add labels, title, and legend

plt.xlabel('Index (Reset)')

plt.ylabel('Yield per hectare (kg)')

plt.title('Train-Test Split with Predictions on Corn')

plt.legend()

plt.tight\_layout()

plt.show()

### Final Model Detailed Forecast Analysis

from sklearn.model\_selection import GridSearchCV

from skforecast.ForecasterAutoreg import ForecasterAutoreg

from sklearn.ensemble import RandomForestRegressor

from skforecast.model\_selection import backtesting\_forecaster

# Initialize and configure the RandomForestRegressor

rf = RandomForestRegressor(random\_state=42)

# Define the hyperparameter grid for the RandomForestRegressor

param\_grid = {

    'n\_estimators': [100, 150, 200, 300],  # Number of trees

    'max\_depth': [None],  # Maximum depth of each tree

    'min\_samples\_split': [2, 5, 10],  # Minimum samples required to split a node

    'max\_features' : ['sqrt', 'log2'],

    'criterion' : ['squared\_error'],

    'min\_samples\_leaf' : [1, 2, 4]

    }

# Use GridSearchCV to find the best hyperparameters

grid\_search = GridSearchCV(estimator=rf, param\_grid=param\_grid, cv=30, scoring='neg\_mean\_squared\_error', verbose=2)

grid\_search.fit(data[features], data[target\_variable])

#Get cross-validation results

cv\_results = pd.DataFrame(grid\_search.cv\_results\_)

# Sort by mean test score in descending order (since higher scores are better for negative MSE)

cv\_results\_sorted = cv\_results.sort\_values(by='mean\_test\_score', ascending=False)

# Display the 5 best hyperparameter combinations with improved visualization

print("Top 5 Best Hyperparameter Combinations with Scores:")

display(cv\_results\_sorted[['params', 'mean\_test\_score']].head(5).style.background\_gradient(cmap='Greens'))

# Display the 5 worst hyperparameter combinations with improved visualization

print("\nTop 5 Worst Hyperparameter Combinations with Scores:")

display(cv\_results\_sorted[['params', 'mean\_test\_score']].tail(5).style.background\_gradient(cmap='Reds'))

# Initialize the forecaster with the best RandomForestRegressor and configure the lags

forecaster = ForecasterAutoreg(

    regressor=rf,

    lags=12  # Adjust this based on your specific needs

)

# Fit the forecaster to the entire dataset

forecaster.fit(y=data[target\_variable], exog=data[features])

# backtesting to evaluate the model on the most recent part of the data

initial\_train\_size = len(data) - forecast\_horizon

# Backtesting the model: predictions for the most recent part of the data

backtest\_predictions = backtesting\_forecaster(

    forecaster=forecaster,

    y=data[target\_variable],

    exog=data[features],

    initial\_train\_size=initial\_train\_size,

    steps=forecast\_horizon,

    metric='mean\_squared\_error',

    verbose=True

)

# Make predictions for the next 10 years

# Exogenous variables for future periods must be provided if they are used in the model

future\_exog = pd.DataFrame({

    'CASH COSTS': np.random.rand(forecast\_horizon),

    'NON-CASH COSTS': np.random.rand(forecast\_horizon),

    'IMPUTED COSTS': np.random.rand(forecast\_horizon),

    'TOTAL COSTS': np.random.rand(forecast\_horizon),

    'RETURNS ABOVE CASH COSTS': np.random.rand(forecast\_horizon),

    'Population': np.linspace(start=data['Population'].iloc[-1], stop=data['Population'].iloc[-1] \* 1.1, num=forecast\_horizon),

    'Season\_encoded': np.random.randint(low=0, high=4, size=forecast\_horizon),  # Adjust as per actual categories

    'GROSS RETURNS': np.random.rand(forecast\_horizon)

})

# Define the years corresponding to the predictions

years = np.arange(2022, 2032)

last\_index = 59  # The last index of your training data

steps = 10  # Number of steps you want to predict

# Create a new index for the exog data starting from last\_index + 1

new\_index = np.arange(last\_index + 1, last\_index + 1 + steps)

exog\_future = future\_exog.copy()

exog\_future.index = new\_index

# Now use this exog to make predictions

predictions = forecaster.predict(steps=steps, exog=exog\_future)

# Print or plot your predictions

print(predictions)

plt.figure(figsize=(9, 6))

plt.plot(years, predictions, marker='o', color='r')

plt.title('Corn Yield Predictions for the Next 10 Years')

plt.xlabel('Year')

plt.ylabel('Predicted Values')

plt.grid(True)

plt.xticks(years)  # Set x-axis ticks to the defined years

plt.show()

## Palay

import pandas as pd

data = pd.read\_csv('palay\_data.csv')

data.head()

### Data Preprocessing

# Check for missing values in the merged dataset

data.isnull().sum()

data.dtypes

# Convert object columns to numeric, forcing errors to NaN (in case of any non-numeric values)

columns\_to\_convert = ['Year', 'CASH COSTS', 'NON-CASH COSTS', 'IMPUTED COSTS', 'TOTAL COSTS', 'GROSS RETURNS',

                      'RETURNS ABOVE CASH COSTS', 'RETURNS ABOVE CASH AND NON-CASH COSTS', 'NET RETURNS',

                      'NET PROFIT-COST RATIO', 'Cost per kilogram (pesos)', 'Yield per hectare (kg)',

                      'Farmgate price (pesos/kg)']

# Convert the columns to numeric, forcing errors='coerce' to handle any invalid parsing

data[columns\_to\_convert] = data[columns\_to\_convert].apply(pd.to\_numeric, errors='coerce')

data.isnull().sum()

data.head()

### Data Cleaning

import missingno as msno

import matplotlib.pyplot as plt

# Visualize missing data using missingno

msno.matrix(data)

plt.title('Missing Data Visualization with missingno')

plt.show()

# Impute the specified columns using median imputation first

columns\_to\_impute\_median = ['Year', 'CASH COSTS', 'NON-CASH COSTS', 'IMPUTED COSTS', 'TOTAL COSTS', 'GROSS RETURNS',

                            'RETURNS ABOVE CASH COSTS', 'RETURNS ABOVE CASH AND NON-CASH COSTS', 'NET RETURNS',

                            'NET PROFIT-COST RATIO', 'Cost per kilogram (pesos)', 'Yield per hectare (kg)',

                            'Farmgate price (pesos/kg)']

# Apply median imputation

data[columns\_to\_impute\_median] = data[columns\_to\_impute\_median].fillna(data[columns\_to\_impute\_median].median())

# Now check how many missing values remain

data.isnull().sum()

import seaborn as sns

# Plot distribution of key numerical variables

plt.figure(figsize=(12, 8))

sns.histplot(data['CASH COSTS'], kde=True)

plt.title('Distribution of CASH COSTS')

plt.xlabel('CASH COSTS')

plt.ylabel('Frequency')

plt.show()

plt.figure(figsize=(12, 8))

sns.histplot(data['TOTAL COSTS'], kde=True)

plt.title('Distribution of TOTAL COSTS')

plt.xlabel('TOTAL COSTS')

plt.ylabel('Frequency')

plt.show()

plt.figure(figsize=(12, 8))

sns.histplot(data['Yield per hectare (kg)'], kde=True)

plt.title('Distribution of Yield per hectare (kg)')

plt.xlabel('Yield per hectare (kg)')

plt.ylabel('Frequency')

plt.show()

plt.figure(figsize=(12, 8))

sns.histplot(data['Farmgate price (pesos/kg)'], kde=True)

plt.title('Distribution of Farmgate price (pesos/kg)')

plt.xlabel('Farmgate price (pesos/kg)')

plt.ylabel('Frequency')

plt.show()

# Plot categorical variable 'Type'

plt.figure(figsize=(12, 8))

sns.countplot(data=data, x='Type')

plt.title('Distribution of Crop Types')

plt.xlabel('Crop Type')

plt.ylabel('Count')

plt.show()

# Plot categorical variable 'Season'

plt.figure(figsize=(12, 8))

sns.countplot(data=data, x='Season')

plt.title('Distribution of Seasons')

plt.xlabel('Season')

plt.ylabel('Count')

plt.show()

# Scatterplot: CASH COSTS vs Yield per hectare (kg)

plt.figure(figsize=(12, 8))

sns.scatterplot(data=data, x='CASH COSTS', y='Yield per hectare (kg)')

plt.title('CASH COSTS vs Yield per hectare (kg)')

plt.xlabel('CASH COSTS')

plt.ylabel('Yield per hectare (kg)')

plt.show()

# Scatterplot: Farmgate price vs Yield per hectare (kg)

plt.figure(figsize=(12, 8))

sns.scatterplot(data=data, x='Farmgate price (pesos/kg)', y='Yield per hectare (kg)')

plt.title('Farmgate price vs Yield per hectare (kg)')

plt.xlabel('Farmgate price (pesos/kg)')

plt.ylabel('Yield per hectare (kg)')

plt.show()

# Select only the numeric columns for correlation analysis

numeric\_columns = data.select\_dtypes(include=['float64', 'int64'])

# Generate a correlation matrix for the numerical features

correlation\_matrix = numeric\_columns.corr()

# Set up the plot for a heatmap

plt.figure(figsize=(14, 10))

# Create a heatmap of the correlation matrix

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)

plt.title('Correlation Matrix of Numerical Features')

plt.show()

# Set up the figure for outlier detection using boxplots

plt.figure(figsize=(14, 10))

# Create boxplots for key numerical features to visualize outliers

sns.boxplot(data=data[['CASH COSTS', 'TOTAL COSTS', 'Farmgate price (pesos/kg)', 'Yield per hectare (kg)']])

plt.title('Boxplots for Outlier Detection')

plt.ylabel('Values')

plt.xticks(rotation=45)

plt.show()

# Generate summary statistics for the numerical features

summary\_statistics = data.describe()

# Display the summary statistics

summary\_statistics

# Function to cap outliers based on IQR

def cap\_outliers(column):

    Q1 = column.quantile(0.25)

    Q3 = column.quantile(0.75)

    IQR = Q3 - Q1

    # Define lower and upper bounds for capping

    lower\_bound = Q1 - 1.5 \* IQR

    upper\_bound = Q3 + 1.5 \* IQR

    # Cap the outliers

    column\_capped = column.clip(lower=lower\_bound, upper=upper\_bound)

    return column\_capped

# Capping outliers for the selected numerical features

data['CASH COSTS'] = cap\_outliers(data['CASH COSTS'])

data['TOTAL COSTS'] = cap\_outliers(data['TOTAL COSTS'])

data['Farmgate price (pesos/kg)'] = cap\_outliers(data['Farmgate price (pesos/kg)'])

data['Yield per hectare (kg)'] = cap\_outliers(data['Yield per hectare (kg)'])

# Check the summary statistics after capping outliers to see the effect

summary\_statistics\_after\_capping = data.describe()

# Display the summary statistics after capping

summary\_statistics\_after\_capping

from sklearn.preprocessing import LabelEncoder

# Re-encode the categorical columns 'Season' and 'Geolocation' to ensure they are present

label\_encoder = LabelEncoder()

# Create encoded versions of 'Season' and 'Geolocation'

data['Season\_encoded'] = label\_encoder.fit\_transform(data['Season'])

data['Geolocation\_encoded'] = label\_encoder.fit\_transform(data['Geolocation'])

# Now retry the correlation analysis on the proposed features

proposed\_features = [

    'Year', 'CASH COSTS', 'NON-CASH COSTS', 'IMPUTED COSTS', 'TOTAL COSTS',

    'GROSS RETURNS', 'RETURNS ABOVE CASH COSTS', 'Population',

    'Season\_encoded', 'Geolocation\_encoded', 'Yield per hectare (kg)'

]

# Generate a correlation matrix for the numerical features

correlation\_matrix\_proposed = data[proposed\_features].corr()

# Set up the plot for a heatmap of the proposed features' correlation matrix

plt.figure(figsize=(14, 10))

# Create a heatmap of the correlation matrix for proposed features

sns.heatmap(correlation\_matrix\_proposed, annot=True, cmap='coolwarm', linewidths=0.5)

plt.title('Correlation Matrix of Proposed Features')

plt.show()

### Model Development and Training

from sklearn.model\_selection import train\_test\_split

# Define the target variable and features

target\_variable = 'Yield per hectare (kg)'  # or 'Farmgate price (pesos/kg)' if predicting prices

features = [

    'CASH COSTS', 'NON-CASH COSTS', 'IMPUTED COSTS', 'Year',

    'TOTAL COSTS', 'GROSS RETURNS', 'RETURNS ABOVE CASH COSTS',

    'Population', 'Season\_encoded'

]

# Split the data into training and testing sets (80% training, 20% testing)

X = data[features]

y = data[target\_variable]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Check the shapes of the resulting datasets

X\_train.shape, X\_test.shape, y\_train.shape, y\_test.shape

#### **Model Evaluation Random Forest Regressor**

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score, mean\_absolute\_error

# Initialize the Random Forest regressor

rf\_model = RandomForestRegressor(random\_state=42)

# Fit the model on the training data

rf\_model.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred\_rf = rf\_model.predict(X\_test)

# Evaluate the model's performance

mse\_rf = mean\_squared\_error(y\_test, y\_pred\_rf)

mae\_rf = mean\_absolute\_error(y\_test, y\_pred\_rf)

r2\_rf = r2\_score(y\_test, y\_pred\_rf)

print(f'Mean Squared Error: {mse\_rf}')

print(f'Mean Absolute Error: {mae\_rf}')

print(f'R-squared: {r2\_rf}')

### Model Analysis and Visualization

# Reset the indices of train and test sets

X\_train = X\_train.reset\_index(drop=True)

X\_test = X\_test.reset\_index(drop=True)

y\_train = y\_train.reset\_index(drop=True)

y\_test = y\_test.reset\_index(drop=True)

# Plot the training and test data with predictions

plt.figure(figsize=(12, 6))

# Plot training data

plt.plot(range(len(y\_train)), y\_train, label='Train', color='blue', marker='o')

# Plot testing data

plt.plot(range(len(y\_train), len(y\_train) + len(y\_test)), y\_test, label='Test', color='red', marker='o')

# Plot predictions

plt.plot(range(len(y\_train), len(y\_train) + len(y\_test)), y\_pred\_rf, label='Predictions', color='orange', linestyle='--', marker='o')

# Add labels, title, and legend

plt.xlabel('Index (Reset)')

plt.ylabel('Yield per hectare (kg)')

plt.title('Train-Test Split with Predictions on Palay')

plt.legend()

plt.tight\_layout()

plt.show()

### Detailed Forecast Analysis Final Model

from sklearn.model\_selection import GridSearchCV

from skforecast.ForecasterAutoreg import ForecasterAutoreg

from sklearn.ensemble import RandomForestRegressor

from skforecast.model\_selection import backtesting\_forecaster

# Initialize and configure the RandomForestRegressor

rf = RandomForestRegressor(random\_state=42)

# Define the hyperparameter grid for the RandomForestRegressor

param\_grid = {

    'n\_estimators': [100, 150, 200, 300],  # Number of trees

    'max\_depth': [None],  # Maximum depth of each tree

    'min\_samples\_split': [2, 5, 10],  # Minimum samples required to split a node

    'max\_features' : ['sqrt', 'log2'],

    'criterion' : ['squared\_error'],

    'min\_samples\_leaf' : [1, 2, 4]

    }

# Use GridSearchCV to find the best hyperparameters

grid\_search = GridSearchCV(estimator=rf, param\_grid=param\_grid, cv=30, scoring='neg\_mean\_squared\_error', verbose=2)

grid\_search.fit(data[features], data[target\_variable])

#Get cross-validation results

cv\_results = pd.DataFrame(grid\_search.cv\_results\_)

# Sort by mean test score in descending order (since higher scores are better for negative MSE)

cv\_results\_sorted = cv\_results.sort\_values(by='mean\_test\_score', ascending=False)

# Display the 5 best hyperparameter combinations with improved visualization

print("Top 5 Best Hyperparameter Combinations with Scores:")

display(cv\_results\_sorted[['params', 'mean\_test\_score']].head(5).style.background\_gradient(cmap='Greens'))

# Display the 5 worst hyperparameter combinations with improved visualization

print("\nTop 5 Worst Hyperparameter Combinations with Scores:")

display(cv\_results\_sorted[['params', 'mean\_test\_score']].tail(5).style.background\_gradient(cmap='Reds'))

# Initialize the forecaster with the best RandomForestRegressor and configure the lags

forecaster = ForecasterAutoreg(

    regressor=rf,

    lags=12  # Adjust this based on your specific needs

)

# Fit the forecaster to the entire dataset

forecaster.fit(y=data[target\_variable], exog=data[features])

# backtesting to evaluate the model on the most recent part of the data

initial\_train\_size = len(data) - forecast\_horizon

# Backtesting the model: predictions for the most recent part of the data

backtest\_predictions = backtesting\_forecaster(

    forecaster=forecaster,

    y=data[target\_variable],

    exog=data[features],

    initial\_train\_size=initial\_train\_size,

    steps=forecast\_horizon,

    metric='mean\_squared\_error',

    verbose=True

)

# Make predictions for the next 10 years

# Exogenous variables for future periods must be provided if they are used in the model

future\_exog = pd.DataFrame({

    'CASH COSTS': np.random.rand(forecast\_horizon),

    'NON-CASH COSTS': np.random.rand(forecast\_horizon),

    'IMPUTED COSTS': np.random.rand(forecast\_horizon),

    'TOTAL COSTS': np.random.rand(forecast\_horizon),

    'RETURNS ABOVE CASH COSTS': np.random.rand(forecast\_horizon),

    'Population': np.linspace(start=data['Population'].iloc[-1], stop=data['Population'].iloc[-1] \* 1.1, num=forecast\_horizon),

    'Season\_encoded': np.random.randint(low=0, high=4, size=forecast\_horizon),  # Adjust as per actual categories

    'GROSS RETURNS': np.random.rand(forecast\_horizon)

})

# Define the years corresponding to the predictions

years = np.arange(2022, 2032)

last\_index = 59  # The last index of your training data

steps = 10  # Number of steps you want to predict

# Create a new index for the exog data starting from last\_index + 1

new\_index = np.arange(last\_index + 1, last\_index + 1 + steps)

exog\_future = future\_exog.copy()

exog\_future.index = new\_index

# Now use this exog to make predictions

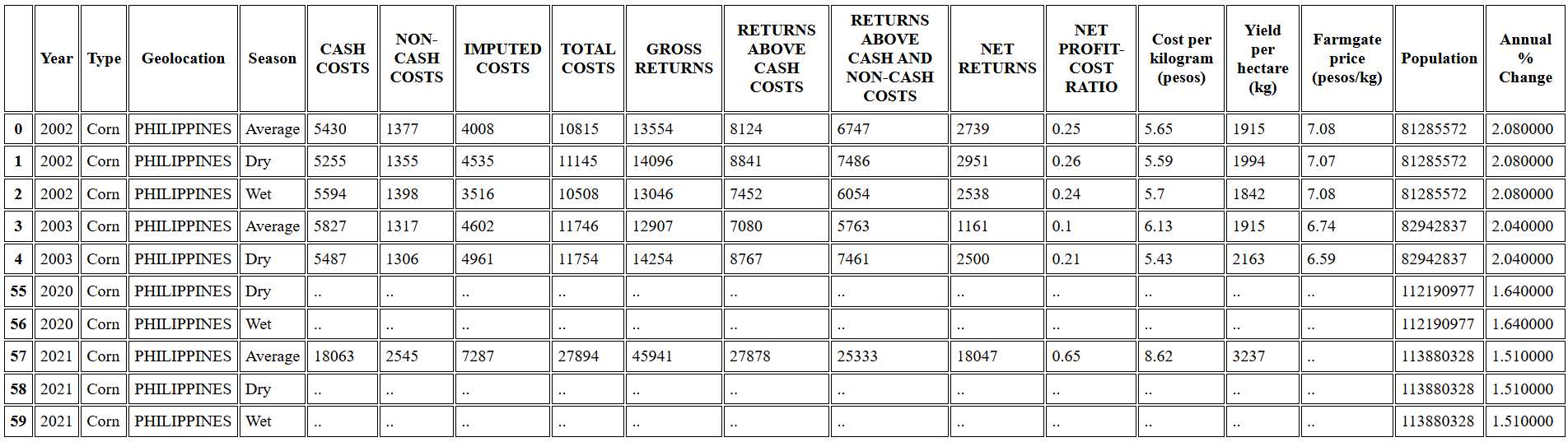
predictions = forecaster.predict(steps=steps, exog=exog\_future)

# Print or plot your predictions

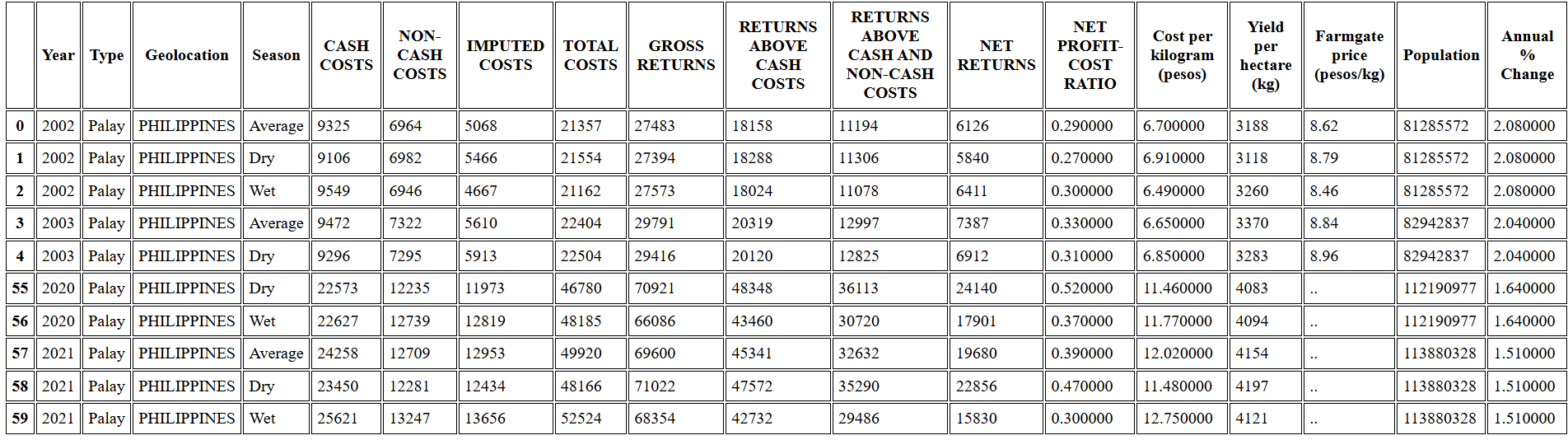
print(predictions)

# Appendix B DATASETS

*Corn Dataset*



*Palay Dataset*



# Appendix C CURRICULUM VITAE

