Universiti Teknologi MARA

# Integrated Vegetable Crop Type Recommendation Using Rule-Based and Random Forest

Ilham Mustaqeem Bin Musa

**Thesis Submitted in Fulfillment of the Requirements for Bachelor of Information Systems (Hons.) Intelligent Systems Engineering**

**January 2026**

# SUPERVISOR APPROVAL

**INTEGRATED VEGETABLE CROP TYPE RECOMMENDATION USING RULE-BASED AND RANDOM FOREST**

**By**

**ILHAM MUSTAQEEM BIN MUSA**

**2024544671**

This thesis was conducted within the guidelines of the project supervisor, Dr. Mohd Zaki Bin Zakaria. It was presented to the Faculty of Computer and Mathematical Sciences and was accepted in partial fulfilment of the requirements for the degree of Bachelor of Information Systems (Hons.) Intelligent Systems Engineering.

Approved by

………………………………

Dr. Mohd Zaki Bin Zakaria

Project Supervisor

JANUARY 25, 2026

# STUDENT DECLARATION

I hereby declare that I own the copyright of this thesis and of the project to which it is related, that all the sources and references that I have used have been properly identified and acknowledged, and that the ideas or quotations taken from other people's work, whether published or in manuscript, are readily cited in the bibliography with what is strictly accepted practice in the discipline regarding the citing of the sources.

…………………………………

ILHAM MUSTAQEEM BIN MUSA

2024544671

JANUARY 25, 2026

# ACKNOWLEDGEMENT

Alhamdulillah, all praise to Allah SWT for giving me the strength, patience, and commitment to completing this Final Year Project. I would like to express my sincere gratitude to my supervisor, Dr. Mohd Zaki Bin Zakaria, for his direction, support, and constructive advice throughout this project. His expertise and ideas helped shape my research, and he mentored me as an individual. Additionally, I would also like to thank my CSP650 lecturer, in particular, for the understanding they gave me and encouragement to find the passion to learn, which gave me the background knowledge and confidence to approach this project with clarity and purpose. Most importantly, I would like to thank my beloved family for their love, support, and prayers. Their encouragement, patience, and belief in me have been so comforting, and my journey has been motivating because of their influence.

# ABSTRACT

Farming activity is highly influenced by environmental conditions and market dynamics, which can affect crop productivity and profitability. With these dynamics condition, it bring opportunity for farmers to leverage data-driven solution that can support them in selecting suitable vegetable crops based on current conditions. Many farmers rely on traditional knowledge when choosing crops, which may not reflect changing of environmental factors and fluctuating market prices. Vegetable market prices fluctuate weekly, making it difficult for farmers to choose profitable crop choices without systematic analysis of market data. This project proposes a hybrid crop recommendation application that combines a rule-based Machine Learning model that utilizes environmental and weekly vegetable market prices data collected from FAMA and Kaggle that has been pre-processed as input features. A predictive model is trained and evaluated using appropriate Random Forest algorithms to predict and suggest the highest crop price. The system is implemented within a mobile application framework to support user interaction and decision-making. A Random Forest regression model was trained and evaluated due to its robustness and suitability for crop price prediction. Grid Search hyperparameter tuning was applied to optimize the model performance. The developed model showed reliable prediction results with a strong $R^2$ score around 0.9655%, indicating that strong capability in learning market price patterns. This project successfully developed a hybrid vegetable crop recommendation application that combines rule-based environmental suitability analysis with market price prediction to support data-driven crop selection. The prototype help improves decision-making efficiency while reducing reliance on traditional trial-and-error methods. Future enhancements may include larger datasets, real-time weather data, and additional crop categories.

# TABLE OF CONTENTS

**CONTENT** **PAGE**

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| API | Application Programming Interface |
| CGC | Crop Growth Cycle |
| CNN | Convolutional Neural Network |
| CSV | Comma-Separated Values |
| DL | Deep Learning |
| GA | Genetic Algorithm |
| GDP | Gross Domestic Product |
| IoT | Internet of Things |
| KNN | K-Nearest Neighbour |
| LSTM | Long Short-Term Memory |
| MAE | Mean Absolute Error |
| ML | Machine Learning |
| MSE | Mean Squared Error |
| RMSE | Root Mean Squared Error |
| RF | Random Forest |
| SVR | Support Vector Regression |
| UI | User Interface |
| UX | User Experience |
| XGBoost | Extreme Gradient Boosting |

# CHAPTER 1

# INTRODUCTION

This section provides a brief overview of this project, which consists of the background of the study, the objectives, the problem statement, and the scope of the project. Agriculture is crucial to food security and socio-economic development. It plays an important role in many countries, promoting food security as well as economic and GDP contributions, especially in Malaysia. However, farmers struggle to make decisions about what crops to plant. It can be a result of variable or uncertain market prices, seasonal demand, or environmental factors. Therefore, this project proposes the development of recommendation applications that take into account certain factors to help farmers decide better what crops to plant.

## 1.1 Background of Study

Nowadays, the agricultural sector is important, as it produces food for humans in this world. It has been an important sector as it contributes to the economy in our country. In Malaysia, based on statistical data, this sector has increased 10.9 per cent of Gross Domestic Product (GDP), with cash crop industries such as palm oil and rubber as the main contributors, and this shows how important this industry is, especially in the farming sector. According to Sabirin et al. (2022), the agriculture industry plays an important role in directly enhancing availability and food security.

Farm planning is one of the important things that needs attention because this is the first stage before starting planting. The main component that needs to be decided is what the suitable vegetable crop type is for a specific location in order to have a good outcome from farming activity. Many farmers in Malaysia rely on traditional technology without any data-driven tools, which will lead to poor decision-making. Climate change has been the most challenging factor to overcome. According to Yash

Gupta & Garima Srivastava (2024), climate change has introduced more extreme and unpredictable weather patterns, making farming riskier. On the other hand, market price instability is a factor that needs to be considered by farmers to avoid investing in high-cost crops but with low returns due to the fluctuating market price of the yield product. This fluctuation is caused by a few factors, such as supply-demand imbalance, import competition, and seasonal variations such as normal days and the "Raya" festival. Market trends play a big role in farming planning in order to have an agricultural product that has high market value and is easy to sell.

In this project, the technique that was used is a supervised machine learning technique to develop a predictive model that is able to learn by itself from data and is expected to be implemented in a mobile application prototype. According to Dahiphale et al. (2023), Machine learning gives computers the ability to learn without being explicitly programmed. In other words, machine learning is turning things or data into numbers and finding patterns in those numbers. The identified patterns help in predicting output for new data points. Machine learning is different from traditional programming, where it involves training the model to learn from data given, to given while traditional programming requires programming to do the task and define the step should be taken to solve problems. There are three types of machine learning: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is usually trained with labelled data, such as random forests, decision trees, support vector machines, etc. For unsupervised learning, it opposite of supervised learning, where the model will be trained with unlabelled data. The model needs to learn to find patterns in the dataset and group similar data point such as clustering, apriori algorithm, etc. Reinforcement learning consists of a component known as an agent. According to Dahiphale et al. (2023), Reinforcement learning (RL) is a type of machine learning that allows an agent to learn how to behave in an environment by trial and error.

To address this, a vegetable crop type recommendation model is needed to help farmers on selecting suitable crop type based on a few conditions, such as environmental and market factors. This method involves machine learning to analyse data and provide the best vegetable crop type using real-time data-driven methods. It will help agricultural productivity and efficiency and provide sustainable agricultural practice by utilising and implementing technology in this sector. According to Wee Wei En & Shou Hui (2023), Smart farming technologies have an important purpose in agriculture and include improving status and quality, increasing quality, using better resources, and increasing sustainable farming practices. This paper will focus on identify vegetables crop type selection factor, developing a model that can give recommendations using machine learning and is expected to be implemented in a mobile application prototype.

## 1.2 Problem Statement

One of the most important challenges for farmers to contend with is awareness of crop suitability, both by environmental features and market demand. There is oftentimes insufficient reliable information about which crops are best suited for the soil type, weather conditions, and market demand. Ullah et al., (2022) explain how the lack of awareness and appropriate knowledge from credible sources results in farmers relying on input providers for crop information, thus leading to questionable decisions, which may result in profit depletion.

Another challenge is the lack of data-driven or machine learning solutions to provide suitable vegetable crop type recommendations. Most crop selection methods are based on the prior experiences of farmers or recommendations from some agricultural specialists. These processes of crop selection rely on subjective opinions and are often prolonged in time. Singhal et al., (2025) showed that relying solely on professional expertise and personal farmer experience of crop selection can lead to significant variability in outcomes. The above statement underlines the need for machine learning-based systems that can provide better, more timely and precise recommendations from data gathered from identified datasets.

Moreover, current vegetable crop type recommendation systems do not adequately support smallholder farmers. Most recommended technological solutions available do not consider the constraints required by smallholders, such as accessing real-time data, limited resources, and applications that suit land conditions. Shams et al., (2024) reported that smallholder farmers face many obstacles relating to technology accessibility, which affect how informed their decisions are regarding crop choice. For these reasons, it is essential to develop a vegetable crop type recommendation system that supports the needs and constraints of smallholder farmers.

## 1.3 Research Questions

1. What is the factor that influence vegetables crop selection?
2. How to develop a machine learning model that is able to predict the price of vegetable crop types at their harvest time?
3. How to develop a mobile application prototype that implement hybrid approach that combines rule-based filtering and a machine learning model?

## 1.4 Objectives

1. To identify the factors that influence vegetable crop type selection.
2. To develop a predictive machine learning model that can give vegetable crop type recommendations.
3. To develop a mobile application prototype that implements a hybrid approach by combining rule-based filtering and a machine learning model for price prediction for vegetable crop type recommendation.

## 1.5 Scope

The scope of this project is outlined below: _

1. **Study on influencing crop selection factor**
   Study, identify, collect, and process environmental and market trend factors, including soil conditions, weather patterns, and historical weekly market prices, that influence vegetable crop selection.
2. **Machine learning model development**
   Develop, train, and tune a Random Forest Regression model using historical weekly market price data, incorporating model training, testing, and hyperparameter tuning to predict vegetable crop prices and give the highest prediction price as the recommended crop.
3. **Mobile application prototype development**
   A Flutter-based mobile application prototype is developed that implements a hybrid approach by combining rule-based filtering with machine learning and enables users to input soil data, automatically retrieve environmental data via

API, and receive recommended vegetable crop types with predicted market prices.

## 1.6 Significance

A vegetable crop type recommendation system is indeed available nowadays. It helps farmers to identify suitable vegetable crop types to plant and helps them to maximize farming yield. This project would have significant value as follows: _

Support Sustainable Development Goal Component:

1) Sustainable Development Goal 2 (SDG2 - Zero Hunger)

   Knowledge and understanding of key factors that improve farming practices ensure better yield and food security.

2) Sustainable Development Goal 9 (SDG9 - Industry, Innovation, and Infrastructure)

   The system generates output-based technology advancement with data-driven innovations in agriculture.

3) Sustainable Development Goal 13 (SDG13 - Climate Action)

   Farmers can adapt to weather patterns through data-driven crop choice because the application developed is robust against environmental fluctuations.

This project offers a lot of benefits, as it helps to understand the components that have a big influence on vegetable crop selection to provide data insight and help to train a model to learn data patterns. Using area-based agricultural data, a predictive machine learning model is developed to recommend vegetable crop choices. This encourages innovation and facilitates technological advancement within the agriculture industry via data-informed interventions. Furthermore, by suggesting crop varieties that fit the current environmental conditions, this application allows the farmer to adjust to the changing environmental conditions and unpredictable market trends. This will support resilience and sustainability in agriculture.

## 1.7 Summary

In conclusion, the agriculture industry is a fast-growing industry that plays an important role in Malaysia's economy nowadays. The farming industry has been affected by many challenges, especially in the technological aspect, compared to other industries. This project aims to address this gap by doing some work as mentioned in the objective and scope. Having an application system for vegetable crop type recommendation help farmer for planning their activity. By integrating a machine learning model and a mobile application prototype, it is expected to be able to give vegetable crop type recommendations to support sustainable farming practices in Malaysia.

# CHAPTER 2

# LITERATURE REVIEW

This section presents the background of research on vegetable crop type recommendations. It intends to provide the rationale for starting the project and justification for the title of the project. Literature reviews are essential for not only showing what has been studied, but also what methods have been used and what has not been addressed. In agriculture research, crop recommendation has been widely researched to facilitate the farmer with an informed decision. The type of crop is generally influenced by environmental factors, including soil type, water recourses and climate. The objective of reviewing these works is to identify the limitations of traditional methods and practices, as well as to identify the key factors that influence vegetable crop type selection in agriculture.

## 2.1 Overview of Vegetables Crop Type Recommendation Application

Vegetables Crop type Recommendation application help on precision farming practice through data-driven insights. Pudumalar et al., (2017) emphasise that a crop recommendation system is a system that relies on ensemble learning methods like K-Nearest Neighbour to make predictions with high accuracy. Ahmed et al., (2024) highlight that transformative framing practice can be done by incorporating IoT devices, sensors, and data analytics to assist in terms of information, such as suitable crop selection, irrigation management, and farm operations.

A crop type recommendation system is able to assist farmers in selecting the best crop at the right time in order to have smooth farming operations. Pudumalar et al., (2017), the predictive model developed by them achieved prediction accuracy of 88%, this show that the ability of the predictive model to help farmers make an informed planting decision. As a specific application for precisions agriculture, vegetables crop type recommendation system works on the analysis of data based on variety factors such as climate, environmental, and market trend dynamic data analysis. Moreover, a statement from Akkem et al., (2023) said that the consideration of a few factors, such as production rate, market price, and government policies, is needed and relevant to consider for crop selection methods.

From a global perspective, vegetables crop recommendation system is growing fast as a technology for sustainable agriculture practices. According to Ahmed et al., (2024), the integration of data analytics, machine learning, and Iot represent a major step toward addressing global food security and environmental sustainability. It demonstrates how predictive models can be deployed to improve crop selection for farmers to make decisions and yield production, and not just for a specific region, but also can be localised to specific needs to support all types of farmers. In Malaysia context, the idea from this study is relevant, as Malaysia is going to modernise the agriculture sector to address main issues such as climate change and market inflation. However, most of the study mentioned above focuses on India and Bangladesh. Still, the principles can be applied in Malaysia context to support farmers in making informed decisions and make profit from these technologies.

### 2.1.1 Factors Influencing Vegetable Crop Type Selection

The choice of vegetable crops by farmers is a dynamic decision influenced by a variety of interconnected factors which environmental and market trends. Crop type selection involves a fundamental basis, such as environmental conditions, including climate, soil characteristics, market trends, and water availability. For example, Rizzo et al., (2023) highlight that factors like soil characteristic temperature significantly impact crop selection decisions. According to Yash Gupta & Garima Srivastava (2024), agricultural productivity, crop growth, and yield potential can be affected by weather conditions. Many countries have adjusted their crop choices to local climate conditions. Bhat et al. (2023) highlight that farmers nowadays must have awareness or knowledge of the crop suitability for different soil types based on their characteristics. This adjustment is crucial to ensure that the farmer chooses the crop type that suits the current environmental conditions of specific areas. Environmental conditions are mostly effected from unpredictable weather patterns, as climate change has become pronounced, considering environmental conditions are important to ensure sustainability.

The effect of commodity prices and consumer demand on market conditions and influencing factors for crop type selection is very important. Begho et al., (2022) point out that, related to profitability, one of the driving factors for a farmer's decisions is based on economic factors such as market access and expected returns. This further emphasises the importance of knowing markets and consumer demand before production happens.

In Malaysia, seasonal price fluctuations around festivals such as “Hari Raya”, Chinese New Year, Deepavali, etc., have a significant impact on crop profitability. Aligning planting patterns and seasons with the market enables farmers to time production in high-demand periods, which delivers maximum return. The weekly price variations from seasonal or weekly supply cycles and congestion can assist a farmer when making short-term planting decisions for perishable crops or any crop with weekly fluctuations. These are also common in commodity cycles as well. A study by Kephe et al., (2022) in South Africa also supports this perspective, emphasising that cost-benefit analysis helps farmers identify more profitable crops. Providing and using market demand data analysis builds a solid foundation for farmers to manage risk better and absorb price unpredictability in a developing and expanding market. Using market demand data analysis builds a solid foundation for farmers to manage risk better and absorb price unpredictability in a developing and expanding market. In turn, this also helps farmers make more informed, profit-driven decisions. Figure 2.1 summarises a comparison of features that are actual inputs for different crop type recommendation systems in different research papers.

Crop Recommendation Systems - Feature Comparison

| Characteristic | Pudumalar et al. (2017) & Begho et al. (2022) | Yash Gupta & Garima Srivastava (2024) & Ahmed et al. (2024) | Gunawan et al. (2024) & Rizzo et al. (2023) | Hamadani et al. (2021) & Elbasi et al. (2023) & Zheng et al. (n.d.) |
|---|---|---|---|---|
| Soil Properties | pH, depth, texture, water holding capacity | pH | Soil fertility | Soil conditions |
| Climate Factors | None | Temperature, humidity, rainfall | Climate suitability | Climate conditions |
| Agronomic Data | None | NPK | Agronomic factors | Crop production trends |
| Economic Factors | None | Market demand | Market demand | Import/export data |
| Sustainability | None | None | Sustainability score, water availability | None |
| Other Factors | None | Area, pests/diseases | Crop type, government policies | Data quality/type |

Made with Napkin

**Figure 2.1** Comparison of features used in crop recommendation across selected research studies

Based on the Figure 2.1 shows that the majority of the recommendation systems focus on the soil properties (pH, fertility and texture), and these are considered basic and required inputs for crop recommendation systems. More recent models have added climate factors (temperature, humidity and rainfall) to improve the environmental adaptability of the model. Some research papers also included agronomic variables like NPK levels and crop production trends, but these inputs were the least common. Most interestingly, only a small subset of recommendation systems incorporated economic factors, including but not limited to market demand or import/export data, to maximise profitability. Sustainability and other variables such as crop type, area, and data quality were considered less formally, but still all inputs contributed to the model. This analysis indicates there are opportunities to improve vegetable crop type recommendation systems by adding more environmental data and augmenting the economic inputs to set market trends, which is the focus of this project.

### 2.1.2 Challenges in Vegetable Crop Type Recommendation

A few years ago, vegetable crop selection relied on local knowledge, farmer intuition, and historical experience rather than systematic data analysis through AI. While this practice has contributed to the sustainable farming system from back then to now in certain contexts, it often lack flexibility needed in today's unpredictable environmental conditions. According to Wee & Lim (2022), the agricultural sector is also challenged by many issues, including crop and land quality, climate, poor economic for farmers and access to technology. Research from Hamadani et al. (2021) said that traditional farming practice provides a valuable means in preserving sustainability, but it still has a limitation in adapting to rapidly changing ecological and economic conditions. In the era where climate change is hard to predict across different region seasons, over relying on traditional methods may not ensure optimal farming productivity and resource efficiency.

The existing recommendation system still has significant limitations. The existing system relies on generalised recommendations that do not reflect the localised environment and economic conditions. As a result, irrelevant information may be provided to the user as recommendations that probably do not work well. Besides, Gunawan et al. (2024) highlight that the current existing systems are often constrained by technological gaps, especially in rural and poor resource regions. The system lacks real-time data analytics and does not integrate with other criteria needed for accurate recommendations. This will lead to poor decision-making and low farming yields.

The biggest challenge in vegetable crop type recommendation is to incorporate dynamic factors that need to be considered that will influence agricultural outcomes. For example, weather patterns, soil characteristics, and market prices of the product. These real time analysis on all of this factor makes it really difficult for the current existing system to keep updated for modern farmers. Gunawan et al. (2024), the process of selecting suitable vegetable crops is a problem that need of integrated, advanced, and data-driven approach. Manual interpretation of data by farmers is time-consuming and not practical. Without any AI application tool that is able to do analysis, it will affect the farming productivity and farmer economic.

## 2.2 Random Forest-Based Techniques

The discussion in this section covers a number of Random Forest methods that have been used across domains. It is discussed in detail with respect to the proposed methods rationale, comparative performance characteristics, strengths, and limitations of the techniques from the past research.

### 2.2.1 Random Forest for Vegetables Crop Type Prediction

Random Forest algorithm utilises the benefits of ensemble learning to produce accurate predictions. RF is a technique that uses combinations of multiple decision trees to produce accurate predictions and reduce the overfitting issue. A study from Paithane (2023) mentioned that Random Forest works by picking a random sample from the training set and make new decision tree for each sample. Every tree acts as an independent classifier; the output is produced through majority voting. Based research paper by Paithane (2023), the Random Forest work through choosing the prediction result with the most votes as the final result. This decision-making process structure makes Random Forest able to adapt to handling noisy and complex datasets.

The combination of multiple decision trees to make a more accurate and robust model is the foundation of Random Forest. According to Paithane (2023), in order to get a high-accuracy prediction, the Random Forest classifier using number of decision trees to various subset input and averages the results. From that, the technique used in this method comes from reducing the variety and bias in single decision trees to improve model performance. Apart from that, one of the important features that Random Forest has is the ability to calculate the feature importance. For example, variables such as soil pH, nitrogen, phosphorus, and climatic factors like rainfall and temperature are used by this method to make a score of how much each of these inputs will contribute to the final prediction. According to Behera & Mishra (2025), they highlight that their model is effective in determining parameters such as pH, nitrogen, phosphorus, and rainfall as strongly correlated to crop recommendation.

This technique provides several advantages in agriculture, especially in crop selection. The ability to handle both numerical data and categorical data makes it suitable for crop prediction for real-world farming scenario operations. The ensemble approach, which uses multiple decision trees, helps the model remain robust when having incomplete data. Random Forest also has high scalability, where it can adapt to new data sources, since in this project use real-time data analysis to ensure the generated recommendations from this model are relevant and align with current conditions.

### 2.2.2 Comparative Analysis on Random Forest Across Domain

Random Forest is one of the most widely used machine learning methods due to its versatility, robustness, and accuracy, regardless of problem type and data. Table 2.1 presents a comparison of Random Forest implementations in various fields, including a description of key applications, performance results, and challenges of deployment.

**Table 2.1** Comparative Analysis on Random Forest Across Domain.

| Domain | Key Application | Performances | Challenges |
| --- | --- | --- | --- |
| Healthcare (Özen, 2024) | Predicting COVID-19 daily cases and deaths in Turkey. | **Daily cases:** Accuracy: 92.30%. R2: 0.9893. **Death cases:** Accuracy: 91.39%. R2: 0.9834. | Model interpretability is low for healthcare decision-making. Training time and computational load increase with long time-series data. |
| Finance (Wong & Yeh, 2019) | Evaluating financial credit risk in imbalanced datasets | Improved Recall (up to 0.48 vs 0.43 in standard RF). | Recall remained below 0.2 for extremely imbalanced data. AUC improvement is minimal, showing limited gains in severe imbalance |

**Table 2.1** (Continued)

| Domain | Key Application | Performances | Challenges |
|---|---|---|---|
| Agriculture (Paithane, 2023) | Development of an intelligent Crop Recommender System using the Random Forest algorithm | Accuracy achieved 99.09%. It is scalable to large datasets and adaptable to real-time inputs. This technique can handle missing data, nonlinearity, and multiple features effectively. | Execution time for Random Forest is longer compared to simpler models. Limited generalisation if data coverage is low or regionally biased |
| Forecasting (Mallala et al., 2025) | Applied Random Forest Regression to forecast renewable energy production globally. Dataset from Kaggle, covering 175 countries (2000–2020) with 20+ socio-economic and energy features | R2 score: 0.998. MAPE: 0.21. | High feature correlation and model performance may vary across regions with unequal data quality |

Based on Table 2.1, Random Forest has been chosen because this algorithm consistently has strong predictive capability in multiple domains. It achieved good results with high accuracy in agriculture field, which outperforms some advance model. In Finance and Healthcare, it maintains consistent results within complex or imbalanced datasets. Despite the minor limitations such as execution time and low interpretability, the finding shows that Random Forest remain robust and versatile algorithm suitable for diverse fields for real world application.

### 2.2.3 Machine Learning Techniques for Vegetables Crop Type Recommendation

The implementation of machine learning has improved the crop recommendation system recently. According to Bahera & Mishra (2022), among the machine learning techniques, Random Forest show reliable and robust solution due to its ability to handle a variety of data sources and its ensemble learning approach. The traditional approach relies on historical data and expert judgement, where it often struggles to adapt to rapidly changing environment conditions and market demand. However, ML approach particularly Random Forest, work with vast datasets such as soil conditions, climatic factors, and crop growth parameters. A study from Kacprzyk et al. (2022) highlight that ability of ML to process real-time data, helping to transform farming management and predictive systems. This technique will enhance the recommendation system to work with dynamic data compared to the traditional method.

A supervised learning method like Random Forest is suitable for vegetable crop recommendation because of their ability to make predictions based on historical data while maintaining interpretability. Kacprzyk et al. (2022). This technique make prediction by splitting the decision-making process into multiple trees and capturing complex interactions between variables without having an overfitting issue. Random Forest is an ensemble of decisions tree aggregate multiple predictions in order to provide robust results even when dealing with noisy and incomplete data. The benefits that can be utilised from this technique will enable experts to prioritise critical variables for crop success. The integration of an ML model into a crop recommendation system can enhance current farming operations through data-driven analysis.

### 2.2.4 Challenges in Machine Learning-Based Vegetables Crop Type Recommendation

One of the main challenges is the availability of localised agriculture datasets. A high-quality dataset is important because it acts as a fuel to train a machine learning model. Crop suitability is highly dependent on local composition in a specific region. As said by Behera & Mishra (2025), a recommendation system strongly relies on the availability of data and the accuracy of data. Without high-quality data, the model predictions will be unreliable or misleading for farmers in underrepresented areas. Besides, the integration of markets is also a major issue with dynamic market data such as fluctuating crop prices, demand trends, and seasonal festivals. It is not just about crop selection based on environmental data only, but it is also an economic decision to make sure the farmer will maximise profit from farming activity.

Another challenge is the class imbalance issue, such as rare food crops and common food crops. According to Behera & Mishra (2025), the model that they develop perform with common crop types like wheat, maize, and barley, but the model has slightly lower accuracy when it comes to crops with fewer samples. This means a rare vegetable crop may affect the predictions, even if the suggested vegetable crop is suitable environmentally and economically profitable for a specific location. Many farmers lack access to compatible devices that can capture real-time information such as temperature, humidity, and soil. But a suggestion from Behera & Mishra (2025) Real-time weather data can be retrieved from OpenWeatherApp API, and soil data can be retrieved from SoilGrid API based on user location. It will help the system to retrieve precise environmental data from external sources. In reality, a rural user may have issues with the internet, a lack of technical skills, which leads to another issue. However, real-time-data integration is a major strength in crop recommendation systems, but it is also a critical issue need to be solved.

## 2.3 Related Works & Research Gaps

Through critical reviews of previously published studies on agriculture related field, this section identifies weaknesses and opportunities for future research in these areas, thereby providing justification for the proposed adaptive approach.

### 2.3.1 Role of Machine Learning in Vegetables Crop Type Recommendation

The application of machine learning in vegetables crop type recommendation system represents a transformative modern approach in the farming industry. Machine learning provides a robust framework where it very useful in the agriculture industry. Machine learning algorithm learns from real-time and historical data, which allows the model to adapt to changing environmental and economic conditions. In the farming context, machine learning for predictive models is used to reduce human error and minimize resources waste. This model will contribute to increasing farming productivity and reducing financial risk. A study from Manish Lad et al. (2022) said that the combination of AI and ML technology into farming activity has enabled crop prediction of appropriate crops by analysing a few variables such as nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, pH, and rainfall. These variables are always changing and affected by seasonal fluctuations, making it hard to handle by conventional system.

In machine learning paradigms, supervised learning has become the most significant technique for the crop recommendation task. In supervised learning, the model is trained using a labelled dataset. In the farming context, it will learn historical crop yield that is linked to soil and climate data in order to make new predictions. A work that has been done by Manish Lad et al. (2022) few supervised learning models, including Random Forest, Decision Trees (DT), Gaussian Naive Bayes (GNB), Support Vector Machines (SVM), Logistic Regression, and XGBoost has been tested for doing analysis. Their result showing that Random Forest, XGBoost, and Gaussian Naive Bayes (GNB) are among algorithm with higher accuracy. These models effectively handled complex interactions among the predictor variables and were able to generalise well across different environmental scenarios.

While supervised learning is able to give a strong prediction, it requires well-labelled data for training. Poor data collection, incomplete data, and missing values could affect the practical deployment of this system. However, continuous evaluation of the model and re-training are needed to ensure its performance and robustness of the model can evolve from time to time for farming conditions. But this can be limitations to the model ion term region factor where digital record-keeping and sensor infrastructure may still be underdeveloped. Nonetheless, as digital agriculture initiatives grow and open agricultural datasets become easier to access, the scalability of machine learning based crop recommendation systems will improve significantly.

### 2.3.2 Comparative Analysis on Random Forest Model vs Other ML Techniques

Machine learning methods are increasingly important in the design of smart agricultural systems, as they can improve efficiency and decision support by assisting systems in learning from new data. Machines can use different algorithms, such as Random Forest, Support Vector Machines (SVM), Gradient Boosting, and deep learning (Deep Learning, LSTM, and CNN) approaches, all of which have been commonly studied for many different applications within smart agricultural systems, such as crop type classification, yield prediction, and soil property assessment. Every algorithm comes with different benefits and costs, in terms of interpretability, computational efficiency, and resilience to noisy data. Table 2.2 shows a comparison of several machine learning approaches that have been mentioned with respect to crop type prediction, including the strengths and limitations of the techniques.

**Table 2.2** Comparative Analysis on Random Forest Model vs Other Machine Learning Techniques

| Technique | Strength | Limitation |
|---|---|---|
| **Random Forest (RF)** | "In the case of the Random Forest algorithm, the R2 score is the highest. Thus, it has higher accuracy when compared to other models for both cases, and that is about 97% in the case of selected parameters and about 96.3% in the case of all parameters" Jhajharia et al. (2022) | "Require a larger quantum of data for a better predictive analysis." Jhajharia et al. (2022) |
| **XGBoost** | High predictive power and robust with regularisation | Sensitive to hyperparameter tuning and encoding choices |
| **LightGBM** | "LightGBM outperforms conventional models with a 94.7% accuracy rate in yield prediction and reduces maintenance costs by up to 20%." Kumar et al. (2024) | "The primary issues involve handling the vast diversity and variability in data arising from different geographical regions, crop types, and growing conditions [4]-[5]." Kumar et al. (2024) |

**Table 2.2** (Continued)

| Technique | Strength | Limitation |
|---|---|---|
| **LSTM (Long Short-Term Memory)** | "The employed LSTM model has been proving capable of maintaining its predictive accuracy, highlighting the potential for real-time, season-specific irrigation management." Dolaptsis et al. (2024) | Requires large datasets to train a complex black-box model |
| **CNN (Convolutional Neural Network)** | "CNNs are effective for spatial data analysis, making them suitable for tasks like crop classification and disease detection." Kumar et al. (2024) | "However, deep networks typically require extensive labelled data and significant computational resources, which can limit their practical application in real-time prediction settings for agriculture [12]." Kumar et al. (2024) |

Based on Table 2.2, Random Forest frequently demonstrates excellent accuracy, reliability, and flexibility across numerous datasets and conditions in the context of agricultural data modelling, particularly for crop yield prediction, fertiliser recommendation, and real-time decision making. It has been shown by (Jhajharia et al., 2022) that RF is more effective than any other models (Gradient Descent and Long Short-Term Memory (LSTM)) for complete and selected parameter sets, with an R2 of 0.963. Random Forest can account for high-dimensional, noisy, and non-conventional data conditions that can be common in agricultural datasets. This is due to RF being an ensemble of decision trees. In addition to this, Random Forest can perform really well with datasets that are not tabular and are moderately sized, which is common with soil, weather, and crop features, but models such as CNN or LSTM are not appropriate due to their input of large volumes of time series or image data.

Because of this, RF is well-suited to agricultural projects, as access to data may be scarce or inconsistent. Assuming interpretability, computational efficiency, and predictive ability provide a good balance with respect to the comparison to other methods, Random Forest is the best all-around method. Even though methods such as LightGBM and XGBoost provide high predictive capabilities, typically they require a higher degree of model tuning skills and are sensitive to hyperparameter tuning.

### 2.3.3 Applications of Machine Learning in Vegetables Crop Type Recommendation

Many cases study has highlighted the successful of implementation machine learning techniques in crop recommendation systems. These successful verified how data driven methods decision making can improve farming outcomes. For example, according to Elbasi et al. (2023), the model that has been developed reaches high accuracy around 99.59% for the BayesNet Classifier, Naïve Bayes Classifier and Hoeffding Tree, where all of them have the same accuracy, which 99.46%. This result shows the significant of the ensembled method for crop prediction. Moreover, many studies highlight about random forest for dynamic features. This can be seen in a study by Jain et al. (2017), where the proposed model has achieved 96.5% for crop recommendation using a Random Forest classifier through environmental data. This shows that data-driven models simulate the significant of that technique for the model performance over the traditional method.

In addition, the implementation of Random Forest for crop type recommendation from Behera & Mishra (2025) in their research shows that incorporating real-time weather data from external sources, such as OpenWeatherMap and SoilGrids APIs, can yield solid results with high accuracy. From that, it enables the application to retrieve data directly from the user's location to make suitable recommendations. In comparison with traditional method, machine learning based method offer real-time adaptability, and automation (the ability of the model to learn by itself) and they can process large volume of data and adapt evolving conditions, which will enhance the decision making process for the farmers, making huge advancement over traditional method of crop selection where it based on solely on experience or static guides.

### 2.3.4 Limitations of Existing Solutions

Machine learning-based vegetable crop type recommendation systems offer a lot of benefits in this sector, but they still have some serious drawbacks. In technical terms, adapting in real time and being unable to dynamically retrain in response to abrupt environmental changes like pest outbreaks or droughts has been challenging for the ML model. These align with the statement by Behera & Mishra (2024), who said there is a limitation on the imbalance data and environmental unpredictability. Additionally, they frequently exhibit class imbalance, doing well on common crops but poorly on rare but locally appropriate ones. Furthermore, economic factors such as market prices and crop demand are typically excluded, limiting practical value as mentioned by Paithane (2023) that the need for crop selection based on economic and environmental aspects. Most systems rely on datasets from generic sources, reducing the applicability in different regions like Malaysia, which have distinct conditions. System usability is another barrier, as the application is integrated with external sources to get real-time data, as many platforms require internet, a smartphone, and digital literacy, which are not always present in rural areas. Lastly, black-box models like Random Forest and XGBoost, while accurate, lack interpretability, reducing user trust and hindering wider adoption among farmers and agricultural advisors.

## 2.4 Summary

To conclude, it discussed some of the research on vegetable crop type recommendations, primarily identifying them as supporting precision agriculture by making decisions that are ultimately data-driven. Vegetable crop type recommendation systems have typically used environmental factors, including climate, soil type, and water availability, since these characteristics pertain to the suitability of a crop. Newer literature identifies the addition of economic factors like market demand, seasonal price variations, and profitability. In Malaysia, the possibility of cropping to market cycles (including festivals) and weekly price variations represents an opportunity to improve farmers' income through planning and planting on time. This chapter also addressed existing system issues such as the ability to combine real-time data, low flexibility to localised conditions, and lack of access to rural areas. Machine learning methodologies, particularly Random Forest, were mentioned as a promising option to better fit into the challenges of addressing these gaps offered their sturdiness, interpretability, and ability to combine many kinds of inputs. Also, even compared to the other machine learning methods, Random Forest offers a balanced approach to combining environmental and market data. Overall, this literature review provides a great base for the system proposed for this project.

# CHAPTER 3

# RESEARCH METHODOLOGY

This chapter outlines the full methodology used to create a vegetable crop recommendation application using machine learning, particularly the Random Forest algorithm. This methodology is designed to ensure this project has a application that can predict suitable vegetables crops. The methodology undergoes several phases, which include preliminary study, data collection and pre-processing, model creation, model implementation, hyperparameter tuning, application design, development of the application, and final documents. Each phase in the methodology supports the project objectives.

## 3.1 Overview of Research Framework

The research framework of this project is organized around three main goals. Each objective provides an overall guide to the methods used from initial data collection to deliver a working vegetable crop type recommendation application.

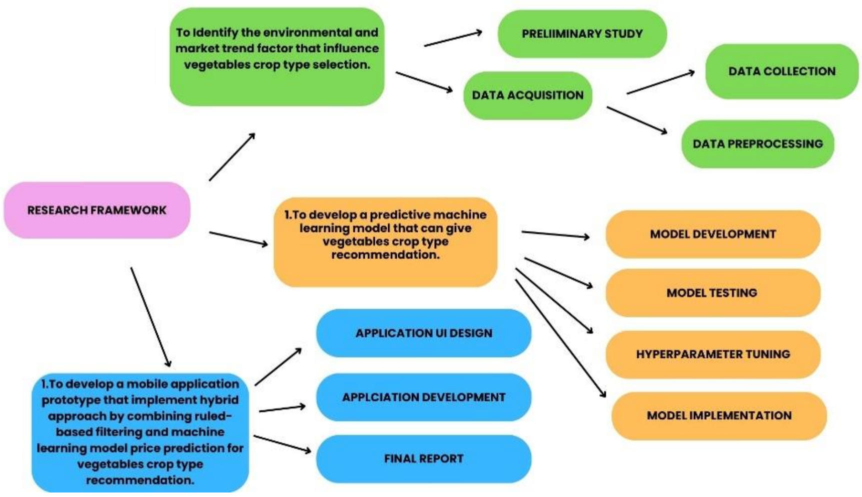



**Figure 3.1** Research framework methodology structure

Figure 3.1 shows the research framework breakdown for this project. The first objective is to identify key factors that influence vegetable crop type selection. This is done in the preliminary study and data acquisition stages by reviewing articles on important environmental factors which affect agriculture and the effects of soil nutrients (nitrogen, phosphorus, potassium), soil pH, temperature, humidity, and rainfall. Also considering market information, especially crop selling prices (weekly market price), etc to ensure recommendations can provide a return on investment.

The second objective is to develop and train a machine learning model that predicts the most suitable and profitable vegetable crop based on the collected parameters. The Random Forest algorithm is selected for this model, as it is more accurate than typical machine learning approaches, can accommodate both numerical and categorical variables, and is resistant to overfitting. Random Forest is an ensemble of decision trees, in which each tree is trained using a random subset of data and features, and the final classification is taken as the majority vote from the decisions made by each tree. There are a number of inherent benefits to using this model, such as that prediction is more stable and the predictions are readily interpretable. This specific algorithm is trained using weekly market vegetable crop price data, which will capture many of the complicated relationships between the ups and downs of the crop price.

The third objective is to design and develop a mobile-based prototype vegetables crop type recommendation application. This can be achieved through the application design, development and documentation stages. The mobile application is created through both for front end and back-end of the application. The application provides real-time data through APIs such as OpenWeatherMap in order to get environmental parameters, based on the location of the user. The mobile application takes environmental parameters and soil data (manually input), processes them through ruled based for filtering and send it into Random Forest model, and the application presents crop recommendations to the user in a friendly manner.

Table 3.1 shows how the preliminary study and overall project flow together. Each goal is broken down into stages that follow the progression from identifying the factors, collecting the necessary data, creating the model, and finally implementing the mobile application so that the project systematically follows the path to developing a vegetable crop type recommendation application.

**Table 3.1** Research Methodology Table

| Phases | | Activities | Deliverables |
|---|---|---|---|
| Preliminary study | | Reviewing the articles, domain understanding and analyzing the past related studies. | The key factors that determine which vegetables to grow will be identified from the information gained through the preliminary study and literature review. The key factors will be put together to enable farmers to make good decisions regarding what vegetables to grow. |
| Data acquisition | Data collection | Environmental and weekly market price data collected from credible sources such as government agencies | Weather and Soil data:<br>collected from Kaggle<br>https://www.kaggle.com/datasets/emrahaydemr/realfake-signature-datasets<br><br>Weekly Market Price Data:<br>Collected data from Federal Agricultural Marketing Authority (FAMA)<br>https://docs.google.com/spreadsheets/d/1ZOdFN9xhtvYv0u9_31JaF3N5Y4xRgMct/edit?usp=drive_link&ouid=104246425162379962893&rtpof=true&sd=true |
| | Data preprocessing | The data will be cleaned, missing values will be imputed, and categorical values will be encoded appropriately. | A structured and cleaned dataset will be prepared for model training. |
| Model Development | | Developed Random Forest Model that able to predict the price of the vegetables crop type | RF models will be developed and trained using historical weekly market price data between 2015 until 2025 (July). |
| Model training and testing | | Training and testing the Random Forest model using an 80:20 train–test split. | The final product is a Random Forest model that has been trained and tested. |
| Hyperparameter Tuning | | Tuning model parameters using Grid Search | Key hyperparameters, including number of trees (estimators), maximum tree depth, feature subset size, minimum samples to split a node (min_samples_split), and minimum samples per leaf node (min_samples_leaf) will be tuned. |
| Model Implementation | | Implement the trained model with rule-based filtering and deploy it through a Flask API for real-time crop recommendation. | A functional deployed model with an API endpoint that returns the final crop recommendation and predicted price. |
| Application Design | | Design the interface for mobile applications to be user friendly for farmers. | Interface of mobile vegetables crop type recommendation application. |
| Application Development | | The trained model integrated into the app with APIs for real-time data, using Flutter for development and VS Code as the coding environment. | Developed mobile crop type recommendation application and integrating it with developed model. |
| Reporting | | Write full documentation | Completed FYP report |

## 3.2 Preliminary Study

The exploratory study stage aimed to establish a good understanding of the problem space and to set the direction for the project. This was done through the review of relevant articles, reviewing previous work, and exploring the domain of agricultural decision-making, smart farming and predictive analytics for vegetable crop type selection. Table 3.2 shows the different types of factors selected for this project. These factors were derived from the information in Chapter 2 and represent a well-balanced view between the soil's ability to grow crops, climatic characteristics, and the economic success of vegetable crops.

**Table 3.2** Selected Factor Used for This Project

| Factor Category | Variables Included | Justification |
|---|---|---|
| **Soil Properties** | Soil pH, Nitrogen, Phosphorus, Potassium | Commonly used in previous studies and directly influence crop suitability and growth |
| **Climate/Environmental Factors** | Temperature, Humidity, Rainfall | Frequently identified as key environmental factors affecting crop development and yield. |
| **Economic Factors** | Month, week, year, crop type, price | Included to capture market trends and support economically for vegetables crop type recommendations |

Based on Table 3.2, the review of literature performed in Chapter 2 found that the majority of past research combined numerous environmental, economic and soil factors to determine what will affect yield maximization as well as crop recommendations. A group of key variables and indicators was discovered as a result of comparing numerous research papers, all of which would be of both usefulness and applicable to current agricultural practices selected for this project.

The selected variables for this project are the key indicators for soil nutrient status, such as N (nitrogen), P (phosphorus), K (potassium), and soil pH, all of which have been clearly documented in numerous publications in the literature on how important these indicators are to successful crop selection and plant growth. Each of the above-mentioned variables has a direct impact on how much crop nutrients can be taken from the soil and develop into healthy plants. Therefore, they were included as core input values in the rule-based part of the assessment system.

Additionally, climate and environmental conditions such as temperature, humidity and rainfall were also included because of their significant impact on growing seasons, crop life cycles and seasonal crop performance. These conditions were prominently mentioned through prior research articles, as well as cited as necessary conditions to be satisfied prior to making a crop recommendation to the farmer.

Economic factors related to the historical price data of the market have been chosen to represent the quick crop selection financial element. Market price trends are included to enable crop recommendations that are not only agronomically sound but also economically feasible for farmers to utilize the machine learning price prediction model while making informed decisions regarding crop selection.

In summary, the preliminary research has shown that the factors selected are based on reliable evidence and are pertinent to the Malaysian agricultural environment and correspond with previously published studies. Thus, the development of a balanced collection of factors that support both agronomic suitability and economic sustainability for crop recommendations was developed.

## 3.3 Data Acquisition

This project used two main datasets, the first one is an environmental sector dataset taken from Kaggle. The data extracts and includes temperature, humidity, and rain level along with other variables. These variables are necessary to support the rule-based filtering of crop suitability. The second dataset consists of historical weekly vegetable prices as sourced from the Federal Agricultural Marketing Authority (FAMA). These datasets combine to provide the needed environmental context and pattern of price for training the vegetable crop type recommendations.

### 3.3.1 Data Collection

The crop price dataset provided by FAMA captures Malaysian vegetable market prices weekly between 2015 and 2025. The dataset contains 24 vegetable types frequently sold in local markets, with weekly entries indicating the average national market price calculated by FAMA. The dataset tracks weekly prices by crop and indicates the historical trend and fluctuation of market conditions. The dataset is useful because it reflects the workings of the market over a long period across numerous crop types, which makes it an effective tool to evaluate seasonal trends and price trend behaviours for crops. In contrast to how the crop price data is formatted, the environmental data from Kaggle was not modified and has been used for the final product because it contained complete features necessary for conducting rule-based evaluation on the data. By combining these two datasets, a more effective model will be able to leverage both the environmental variables and price trends in creating accurate recommendations regarding crop production.

Table 3.3 includes a sample that shows how vegetable prices are reported by FAMA, that include commodity, grade, measurement unit, and weekly market price.

**Table 3.3** Example of Weekly Market Price Raw Data from FAMA

| Commodity | Grade | Unit | Week 1 | Week 2 | Week 3 | Week 4 |
|---|---|---|---|---|---|---|
| Spinach | F.A.Q | KG | 2.43 | 2.71 | 2.91 | … |
| Green Chili | F.A.Q | KG | 7.63 | 7.66 | 7.55 | … |
| Kulai Red Chili / Hybrid Kulai Chili | F.A.Q | KG | 13.15 | 12.61 | 12.45 | … |
| Oily Red Chili | F.A.Q | KG | 11.81 | 11.73 | 11.31 | … |
| Bird's Eye Chili | F.A.Q | KG | 18.00 | 18.40 | 17.10 | … |

Table 3.4 shows that all features of the weekly market price dataset used for this study are outlined. Features in the dataset include critical information enabling the predictive capabilities for the vegetable market price analysis, with data type and a description of the features.

**Table 3.4** Description of Features in Weekly Market Price Data

| Feature | Data Type | Description |
|---|---|---|
| **Commodity** | Categorical | Type of vegetable commodity contained in the dataset (e.g., spinach, bird's eye Chili). |
| **Grade** | Categorical | The corresponding month (can be used for seasonal trend analysis) |
| **Unit** | Float | Unit of measurement for the commodity pricing (in kilograms) |
| **Week** | Float | The average price (RM) of the commodity recorded in a specific week. |

Table 3.5 shows data included in this study for environmental information. The information collected includes soil nutrient and climate parameters affecting crop growth or potential yield, and a list of crop labels for evaluating environmental suitability for vegetable crop recommendations.

**Table 3.5** Example of Environmental Data

| N | P | K | Temperature | Humidity | ph | Rainfall | Label |
|---|---|---|---|---|---|---|---|
| 90 | 42 | 43 | 20.87974 | 82.00274 | 6.502985 | 202.9355 | SPINACH |
| 30 | 65 | 82 | 20.71424 | 15.27824 | 7.103798 | 76.77889 | TOMATO |
| 14 | 67 | 22 | 23.82577 | 24.75485 | 5.62469 | 84.64144 | CHILI |
| .. | … | … | .. | .. | …. | … | … |

Table 3.6 provides an overview of the environmental features used in this study for crop recommendation. Each feature also includes information about the type of data associated with that feature and a description of the feature.

**Table 3.6** Description of Features Environmental Data

| Feature | Data Type | Description |
|---|---|---|
| **N** | Integer | Nitrogen content in the soil (essential for leaf growth) |
| **P** | Integer | Phosphorus content (important for root and flower development) |
| **K** | Integer | Potassium content (affects fruit quality and overall plant health) |
| **Temperature** | Float | Average temperature of the environment (°c) |
| **Humidity** | Float | Relative humidity (%) |
| **ph** | Float | Soil ph level |
| **Rainfall** | Float | Total rainfall (mm) over a given period |
| **Label** | Categorical | The target crop type (vegetables such as tomato, etc. |

### 3.3.2 Data Pre-Processing

To prepare the crop price data from FAMA for model training, FAMA crop prices must first be cleaned, as the original data contains raw data or corrupted data as shown in Table 4, which cannot be used directly for model training. Feature engineering is done because the original data format is not suitable for directly using it to train machine-learning models. Feature Engineering involves transforming the data, thereby improving the dataset structure and providing a more interpretable way of presenting the data, as well as making it clearer for use in model training. The cleaned crop price data has been transformed, where it has been reorganized and reformatted into a better format that is also easier to train with. In addition, the new structure makes it easier for the algorithm to interpret and learn from the data that has been formatted into new way. Table 3.7 shows the feature engineering steps for the dataset, including the code and its explanation.

**Table 3.7** Feature Engineering Code Block

| Step | Code Script | Explanation |
|---|---|---|
| 1 | xls = pd.ExcelFile(file_path)<br>sheets = xls.sheet_names | Load the Excel file and read all the sheets to capture all weekly price records from multiple years |
| 2 | df = pd.read_excel(file_path, sheet_name=sheet, skiprows=7)<br>df.columns = df.columns.str.strip().str.upper() | Delete any rows containing metadata, as well as standardize the headings on each of your columns. Clean up the column names to make sure those names are consistent and remove any rows not containing data or containing text used only for formatting purposes. |
| 3 | week_cols = [col for col in df.columns if "MINGGU" in col]<br>keep_cols = ["KOMODITI"] + week_cols<br>df = df[keep_cols] | Select chosen columns (KOMODITI and weekly prices). Only crop names and weekly market price columns are retained. |
| 4 | df = df[df["KOMODITI"].notna()]<br>df = df[df["KOMODITI"].str.strip().str.upper() != "SUMBER : BAHAGIAN MAKLUMAT PASARAN"] | Remove invalid and source rows. Rows with empty crop names or metadata are filtered out. |
| 5 | df_melted = df.melt(id_vars=["KOMODITI"], var_name="Week_Info", value_name="Price") | Convert wide format to long format (melt operation). Weekly price columns are transformed into individual rows for each crop per week. |

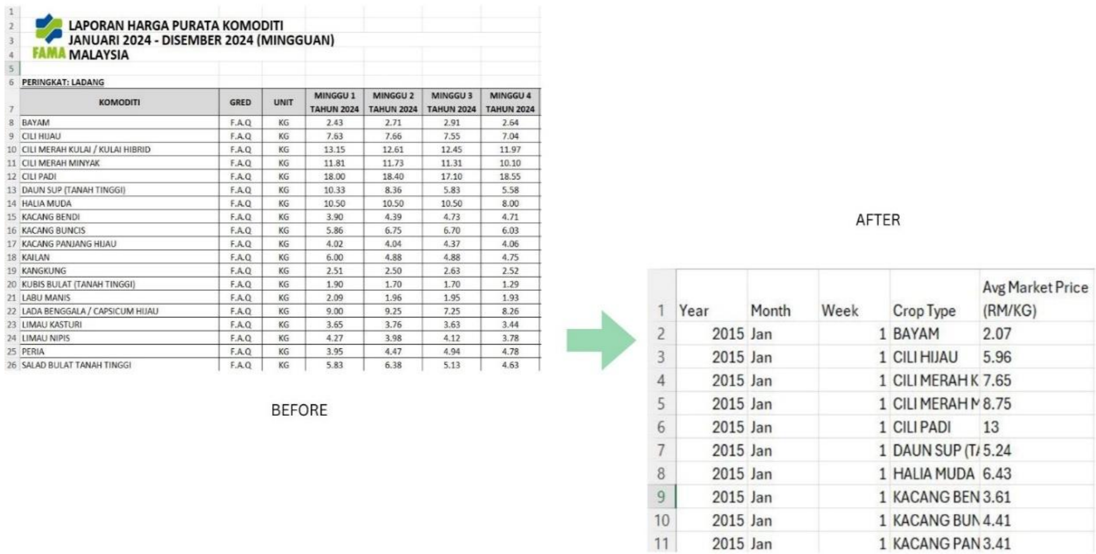
**Table 3.7** (Continued)

| Step | Code Script | Explanation |
|---|---|---|
| 6 | df_melted["Week"] = df_melted["Week_Info"].str.extract(r"MINGGU\s+(\d+)") df_melted["Year"] = df_melted["Week_Info"].str.extract(r"TAHUN\s+(\d+)") | Extract week and year from column labels to create explicit temporal features. |
| 7 | df_melted["Date"] = pd.to_datetime(df_melted["Year"].astype(str) + df_melted["Week"].astype(str) + "1", format="%Y%W%w", errors="coerce") df_melted["Month"] = df_melted["Date"].dt.strftime("%b") | Generate a month feature from the year and week. A date is constructed to derive the corresponding month for seasonal analysis. |
| 8 | df_melted = df_melted.rename(columns={"KOMODITI": "Crop Type"}) df_final = df_melted[["Year", "Week", "Crop Type", "Price", "Month"]] | Rename columns and select final attributes for clarity and consistency. |
| 9 | final_df = pd.concat(all_data, ignore_index=True) final_df = final_df[(final_df["Year"] >= 2015) & (final_df["Year"] <= 2025)] | Filter data between 2015–2025 to include only relevant years. |
| 10 | final_df = final_df.sort_values(by=["Year", "Week", "Crop Type"]).reset_index(drop=True) final_df.to_csv(output_file, index=False, encoding="utf-8-sig") | Sort and export the cleaned dataset chronologically as a CSV for further analysis or model training. |

Table 3.8 shows weekly market price information for several crops, after feature engineering. Week, month, and year, crop type, and average market price (RM/kg) contained in the table demonstrate cleaned and structured data in preparation for model development.

**Table 3.8** Weekly Market Price New Formatted Data

| Week | Month | Year | Crop Type | Avg Market Price (RM/KG) |
|---|---|---|---|---|
| 1 | January | 2024 | Chili | 4.50 |
| 2 | January | 2024 | Tomato | 2.50 |
| …. | …. | …. | .. | …. |



LAPORAN HARGA PURATA KOMODITI
JANUARI 2024 - DISEMBER 2024 (MINGGUAN)
FAMA MALAYSIA

PERINGKAT: LADANG

| KOMODITI | GRED | UNIT | MINGGU 1 TAHUN 2024 | MINGGU 2 TAHUN 2024 | MINGGU 3 TAHUN 2024 | MINGGU 4 TAHUN 2024 |
|---|---|---|---|---|---|---|
| BAYAM | F.A.Q | KG | 2.43 | 2.71 | 2.91 | 2.64 |
| CILI HIJAU | F.A.Q | KG | 7.63 | 7.66 | 7.55 | 7.04 |
| CILI MERAH KULAI / KULAI HIBRID | F.A.Q | KG | 13.15 | 12.61 | 12.45 | 11.97 |
| CILI MERAH MINYAK | F.A.Q | KG | 11.81 | 11.73 | 11.31 | 10.10 |
| CILI PADI | F.A.Q | KG | 18.00 | 18.40 | 17.10 | 18.55 |
| DAUN SUP (TANAH TINGGI) | F.A.Q | KG | 10.33 | 8.36 | 5.83 | 5.58 |
| HALIA MUDA | F.A.Q | KG | 10.50 | 10.50 | 10.50 | 8.00 |
| KACANG BENDI | F.A.Q | KG | 3.90 | 4.39 | 4.73 | 4.71 |
| KACANG BUNCIS | F.A.Q | KG | 5.86 | 6.75 | 6.70 | 6.03 |
| KACANG PANJANG HIJAU | F.A.Q | KG | 4.02 | 4.04 | 4.37 | 4.06 |
| KAILAN | F.A.Q | KG | 6.00 | 4.88 | 4.88 | 4.75 |
| KANGKUNG | F.A.Q | KG | 2.51 | 2.50 | 2.63 | 2.52 |
| KUBIS BULAT (TANAH TINGGI) | F.A.Q | KG | 1.90 | 1.70 | 1.70 | 1.29 |
| LABU MANIS | F.A.Q | KG | 2.09 | 1.96 | 1.95 | 1.93 |
| LADA BENGGALA / CAPSICUM HIJAU | F.A.Q | KG | 9.00 | 9.25 | 7.25 | 8.26 |
| LIMAU KASTURI | F.A.Q | KG | 3.65 | 3.76 | 3.63 | 3.44 |
| LIMAU NIPIS | F.A.Q | KG | 4.27 | 3.98 | 4.12 | 3.78 |
| PERIA | F.A.Q | KG | 3.95 | 4.47 | 4.94 | 4.78 |
| SALAD BULAT TANAH TINGGI | F.A.Q | KG | 5.83 | 6.38 | 5.13 | 4.63 |

BEFORE

AFTER

| Year | Month | Week | Crop Type | Avg Market Price (RM/KG) |
|---|---|---|---|---|
| 2015 | Jan | 1 | BAYAM | 2.07 |
| 2015 | Jan | 1 | CILI HIJAU | 5.96 |
| 2015 | Jan | 1 | CILI MERAH K | 7.65 |
| 2015 | Jan | 1 | CILI MERAH M | 8.75 |
| 2015 | Jan | 1 | CILI PADI | 13 |
| 2015 | Jan | 1 | DAUN SUP (T/ | 5.24 |
| 2015 | Jan | 1 | HALIA MUDA | 6.43 |
| 2015 | Jan | 1 | KACANG BEN | 3.61 |
| 2015 | Jan | 1 | KACANG BUN | 4.41 |
| 2015 | Jan | 1 | KACANG PAN | 3.41 |

**Figure 3.2** Before and after feature engineering on the weekly market price dataset

Figure 3.2 shows the transformation of the raw, weekly market price data as represented. The original dataset recorded the weekly prices for commodities in a tabular format that is very wide with multiple columns for the weekly prices for each commodity. The data was then transformed into a normalized (time-series-friendly) format through feature engineering. This resulted in a new format consisting of an explicit temporal attribute (year, month, week), crop type, and average price for market pricing.

Before starting training the model, the collected data will go through several preprocessing steps to ensure a smooth training process.

- Handling Missing Values:
  Missing Data were handled according to the data types of each feature within the dataset. For numerical features, missing values were filled with the column median. For categorical features, the mode has been used as the fill-in value since it is defined as the category found most frequently within the column. This results in filling those records so that not only does the original data distribution remain intact, but doesn't pass through the model when it's training on a given dataset.

- Label Encoding:
  Categorical variables in the dataset needed to be converted into numerical form. Categorical variables of crop type and month were converted to label encoding, where each category received a numeric representation. By encoding categorical variables this way, Random Forest will effectively utilize the information from categorical variables and not disrupt the structure of the dataset. Since the Random Forest method is tree-based, label encoding does not create an implied order among the categories, and thus it is appropriate to use it for this purpose.

## 3.4 Model Development

The Random Forest Regression model used in this project is able to predict crop prices in the future based on trends in weekly markets as well as growth cycles for the crop.

### 3.4.1 Algorithm Used

The model development was based on historical market prices of vegetable crop commodities. To support the development of predictive models for future vegetable crop prices, the Random Forest Regressor from the Python library was chosen because of its ability to handle non-linear data as well as mixed or noisy datasets common to agriculture. By building many different trees and averaging them together, the Random Forest captures all of the complex interactions between crop varieties and timing within a given agricultural market. Because of this, Random Forest provides an effective framework for building predictive models based on the relationships between crop prices and seasonality. The primary goal of this predictive model is to provide a price estimate (harvest) at the time of planting, allowing farmers to have better planning capabilities regarding their crop choices.

### 3.4.2 Growth Cycle Mapping (Target Shifting) - Function

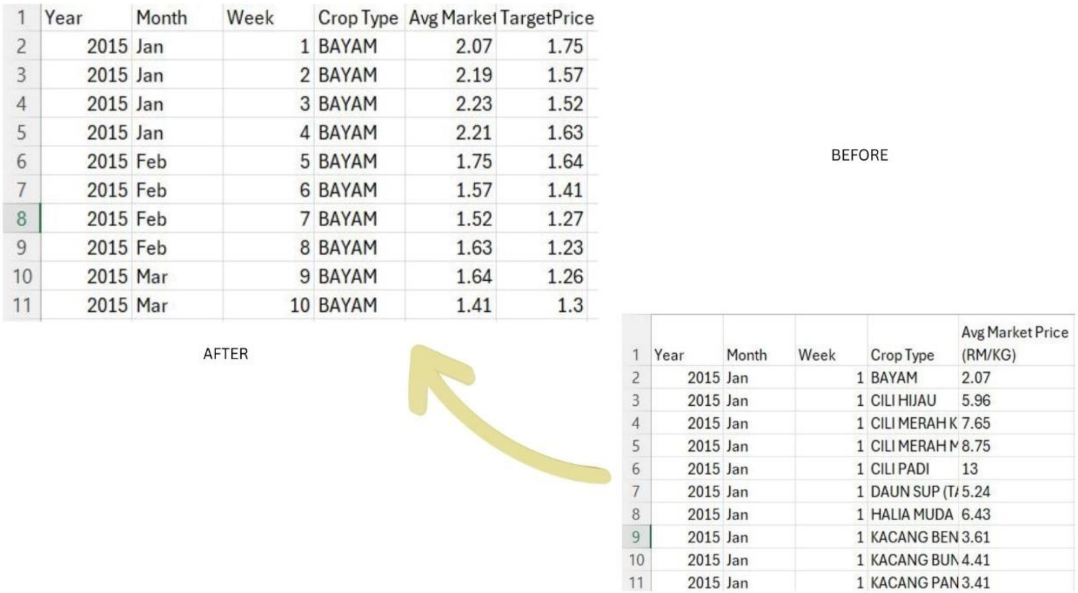
The most important part of this model was the creation of a target shifting approach that considers the lifecycle of crops during their growth. Vegetables have different lengths of time until they reach maturity and their respective harvest times. Therefore, it will be the price predicted by the model for the week the crop will be harvested. A specific function called “prepare_growth_cycle_dataset_crossyear,” (CGC) has been created to allow the model to create a mapping for each crop to show the price it will be sold for based on the expected growth cycle of that crop such as “bayam” equal to 4 weeks, “cili merah” equal to 16 weeks. In the training code, the function is applied after data loading and cleaning, and before feature encoding and model training. Also included in this function is logic on how to handle years where crops are planted close to the end of a year and will not be harvested until after January 1st of the next year. Once all data has been processed through this function, the TargetPrice variable has been created for crops, and all rows that do not have valid future price information have been removed. The shifting of target prices allows the model to accurately represent the market price that will be available in time for the appropriate time frame for the growth duration of each crop. Table 3.9 shows the procedure outlined to determine growth cycle-based target prices for all vegetable crops during the agricultural data preparation phase. This procedure matches each crop's weekly market price records to its estimated harvest period, based on the growth cycle duration specific to each crop.

**Table 3.9** Crop Growth Cycle Mapping Code Block

| Step | Code Logic | Explanation |
|---|---|---|
| 1 | df_prepared = df.copy() | Another copy of the original dataset has been produced to protect the original data from alteration during the data prep process. |
| 2 | sort_values(["Crop Type", "Year", "Week"]) | The datasets will be organized and arranged first by crop type, then by year and week number, to maintain the correct growth cycle and timeline of each crop |

**Table 3.9** (Continued)

| Step | Code Logic | Explanation |
|---|---|---|
| 3 | df_prepared["TargetPrice"] = pd.NA | A new field called TargetPrice was added to store the predicted price of the crop at the end of the growth cycle. |
| 4 | for crop, cycle in growth_cycles.items() | Each crop type will be individually processed based off their established growth cycle length (in weeks). |
| 5 | crop_mask = df_prepared["Crop Type"] == crop | Records will be filtered by the current crop type in order to work on them specifically. |
| 6 | for idx, row in crop_rows.iterrows() | Each weekly data point for the selected crop is iterated to calculate the future target price. |
| 7 | target_week = start_week + cycle | The target week is determined by taking the current week number and adding the crop's growth cycle length to it. |
| 8 | while target_week > 52<br>target_week -= 52<br>target_year += 1 | When a crop grows across years, the week number that exceeds 52 must be changed back to 1 for the new year, and the corresponding year number increased. |
| 9 | target_row = crop_rows[<br>(crop_rows["Year"] == target_year) & (crop_rows["Week"] == target_week)<br>] | Once the future year and week have been determined, the dataset will be searched for the corresponding Market Price. |
| 10 | if not target_row.empty:<br><br>target_prices.append(target_row.iloc[0]["Avg Market Price (RM/KG)"])<br>else:<br>target_prices.append(np.nan) | If a matching future price exists, it is stored as the target price; otherwise, a missing value is assigned. |
| 11 | df_prepared.loc[...] = target_prices | Target Prices are entered back into the Dataset where the crop records for the Scores have been recorded. |
| 12 | dropna(subset=["TargetPrice"]) | Existing rows without a valid future price are removed. |
| 13 | return df_prepared | The final dataset containing growth cycle–adjusted target prices is returned for model training. |




AFTER:

| 1 | Year | Month | Week | Crop Type | Avg Market | TargetPrice |
|---|---|---|---|---|---|---|
| 2 | 2015 | Jan | 1 | BAYAM | 2.07 | 1.75 |
| 3 | 2015 | Jan | 2 | BAYAM | 2.19 | 1.57 |
| 4 | 2015 | Jan | 3 | BAYAM | 2.23 | 1.52 |
| 5 | 2015 | Jan | 4 | BAYAM | 2.21 | 1.63 |
| 6 | 2015 | Feb | 5 | BAYAM | 1.75 | 1.64 |
| 7 | 2015 | Feb | 6 | BAYAM | 1.57 | 1.41 |
| 8 | 2015 | Feb | 7 | BAYAM | 1.52 | 1.27 |
| 9 | 2015 | Feb | 8 | BAYAM | 1.63 | 1.23 |
| 10 | 2015 | Mar | 9 | BAYAM | 1.64 | 1.26 |
| 11 | 2015 | Mar | 10 | BAYAM | 1.41 | 1.3 |

BEFORE:

| 1 | Year | Month | Week | Crop Type | Avg Market Price (RM/KG) |
|---|---|---|---|---|---|
| 2 | 2015 | Jan | 1 | BAYAM | 2.07 |
| 3 | 2015 | Jan | 1 | CILI HIJAU | 5.96 |
| 4 | 2015 | Jan | 1 | CILI MERAH K | 7.65 |
| 5 | 2015 | Jan | 1 | CILI MERAH M | 8.75 |
| 6 | 2015 | Jan | 1 | CILI PADI | 13 |
| 7 | 2015 | Jan | 1 | DAUN SUP (T/ | 5.24 |
| 8 | 2015 | Jan | 1 | HALIA MUDA | 6.43 |
| 9 | 2015 | Jan | 1 | KACANG BEN | 3.61 |
| 10 | 2015 | Jan | 1 | KACANG BUN | 4.41 |
| 11 | 2015 | Jan | 1 | KACANG PAN | 3.41 |

**Figure 3.3** Dataset structure before and after applying the CGC method

Figure 3.3 show before applying the CGC method, the weekly market price dataset's structure indicated that all records contained only an average price per crop over time (week). This data had no direct connection to the expected harvest time price. However, after applying the CGC method, an additional attribute for target price is created for each crop, which indicates the estimated price at the completion of the crop growing cycle.

### 3.4.3 Data Preprocessing and Feature Engineering

Preprocessing of the dataset was performed before training the model improve consistency in the way the model learns. Missing Values in the numerical columns were filled in by taking an average value and then using the median. In contrast, Missing Values in the categorical columns, such as 'Crop Type' and 'Month', were replaced by the mode of each column, as it made sense from a logical standpoint. Once all the categorical variables were encoded, both the mapping from the crop types and months (to be used by the user of the model after deployment) were retained. Therefore, the TargetPrice (shifted TargetPrice) is the only price-related variable remaining in the modelling process. The last step is to create the input Feature Matrix

(X) and target vector (y) for training purposes, where the "TargetPrice” and “AvgMarketPrice" from the dataset were removed from the input Feature Matrix (X) and "TargetPrice" was assigned as the target vector (y).

| 1 | Year | Week | Crop Type | Month |
|---|---|---|---|---|
| 2 | 2015 | 1 | 0 | 4 |
| 3 | 2015 | 2 | 0 | 4 |
| 4 | 2015 | 3 | 0 | 4 |
| 5 | 2015 | 4 | 0 | 4 |
| 6 | 2015 | 5 | 0 | 3 |
| 7 | 2015 | 6 | 0 | 3 |
| 8 | 2015 | 7 | 0 | 3 |
| 9 | 2015 | 8 | 0 | 3 |
| 10 | 2015 | 9 | 0 | 7 |

**Figure 3.4** New formatted data for training process (pre-process + target variable drop)

Figure 3.4 represents the dataset which has been produced after feature engineering, pre-processing and target variable dropout for the training process. All features have a numeric value that will enable them to be incorporated into a Random Forest Regression model equation for the purposes of creating an accurate predictive model or set of models. The temporal features of Year, Week and Month are maintained, whereas the categorical description of crop variety is converted into a numeric representation.

### 3.4.4 Train Test Split and Parameter Setting

To determine how well the model will do outside the training/data set, first I shuffled my dataset with a random seed, then divided the dataset into a separate 80% for the training set and 20% for the testing set. Hyperparameter tuning was done using a manual grid-search method with a 5-fold method of measuring 'fit' using different combinations of various hyperparameters such as the number of trees, maximum depth of tree, minimum sample required at the end of tree node, and feature selection techniques used. All possible combinations of these parameter settings were tested with the cross-validation function from the library of sklearn. The performance of each tree was evaluated using four different error metrics, which is R2 coefficient of determination, Mean Square Error (MSE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). Each fold mean performance score was used to create a mean score across all 5 folds. The model with the highest average R2 coefficient of determination was selected.

### 3.4.5 Final Model Training, Saving, and Performance Evaluation

The model is finalized after determining the optimal hyperparameters through the retraining of the Random Forest model on a full training sample. An additional performance metric was calculated using Out-of-Bag (OOB) error estimates calculated using the built-in "validation method" of Random Forests. The performance of the final Random Forest model was then evaluated using several metrics such as Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE), and Coefficient of Determination (R2). Predictions made by the model will be compared against actual values to visualize how well the model works. The highest predicted price of any crop in the training sample is the basis for creating a simple crop recommendation in a way to demonstrate how this model could help decision-making.

### 3.4.6 Model Serialisation and Deployment Integration

The trained Random Forest Model and relevant components have been saved into a .pkl file using the Pickle Library provided by Python as part of deployment preparation. The filename of this file is rf_growth_cycle_weekly.pkl and contains everything necessary, including the Encoded Crops/Months mappings as well as the test dataset to be used for Validating API Development. When deploying the Model to an API, the API will utilise the saved pickle file when accepting client input to create a transformed input for predicting the expected Harvest Price based on the Model's response to that input.

### 3.5 Model Training and Testing

To create the final datasets, the prepared dataset was split into two parts 80% of the data for training and 20% of the data for testing the model through a random split. A search for optimal hyperparameters was conducted through five-fold cross-validation using the training set to determine which hyperparameters would produce the strongest model. Combinations of hyperparameters were tested and compared based on multiple regression model metrics such as $R^2$, MSE, MAE, and RMSE. From those results, the configuration with the highest average $R^2$ score was selected as the best configuration and retrained using all of the training data.

The testing of the model used the saved model file in conjunction with the same metrics that were used to assess performance on the training dataset. The script used for testing also selected from the test samples the highest predicted price of crops in order to illustrate how well the model supports recommendation scenarios. Finally, scatterplots and regression line graphic visualisations were created in order to compare the actual prices and the predicted prices so that there was an easy visual way to understand how accurately the model predicted prices and its overall predictability.

### 3.6 Hyperparameter Tuning

In Random Forest, the 5 main parameters involve is tree depth, number of trees, and feature subset size.

Tree depth (maximum depth):

Tree depth is the length of the longest path from the root node to the leaf node. It controls how deep each decision tree can grow, but a deeper tree captures complex patterns, which may be overfit, and shallower trees are faster but may be underfit. In a crop prediction scenario, deeper trees can capture complex interactions between variables such as rainfall, temperature, soil, etc but if too deep, they may be overfit and give poor predictions.

Number of trees (number estimator):

It determines how many trees the forest requires. More trees usually require more time to execute, but improve accuracy and stability. In this case, where it involves soil, weather, and market data, having more trees ensures the model is less sensitive to noise and anomalies that are contained in the data. A higher number of trees provides more stability when recommending a crop under various factors.

Features subset size:

Determine how many features/columns such as temperature, rainfall, nitrogen, etc at each split. The randomising in the subset helps the model avoid relying on 1 factor, avoid the overfitting issue and prevents any single feature from dominating the model, which is particularly important in agriculture where multiple factors influence crop suitability.

Minimum samples to split a node (min_samples_split):

This controls how many samples are required to split an internal node. Setting this value to a higher number may help avoid splits based on small potential misleading subsets of data. In crop prediction, this reduces the chance that the model assigns crop decisions (e.g., Chili or tomato) based on rare or uncommon conditions, so that outcomes are more reliable and make sense.

Minimum samples per leaf node (min_samples_leaf):

This parameter establishes the minimum number of samples needed at a leaf node. If it is set into a larger minimum number of samples, it enhances the final decision by making sure that at least a minimum number of examples were taken into consideration.

## 3.7 Model Implementation

The implementation section will consist of input and output descriptions, integration of rule-based. The integration of the model into the overall architecture of the system will be described to enable conducting a crop recommendation.

### 3.7.1 Input and Output Description

The model implementation starts by defining the inputs and outputs for the system. The crop recommendation process will be based on these two types of inputs, environmental factors and weekly market price of vegetable crops, where the environmental data are sourced through API integration, such as into the rule-based filter module. Meanwhile, weekly crop market prices are represented by year, month and week of the month, crop type and the price of the crop as they are necessary for the regression model to estimate pricing for recommended crops. The two types of inputs are presented numerically and encoded via previously saved crop_mapping and month_mapping tables.

For the output, the model only gives one single "final" recommendation when using the REST API to access it. Each final recommendation provides three pieces of information which is the name of the "best" crop recommendation, the predicted market price for that recommended crop, and a brief description of the recommended crop based on the pre-defined dictionary. If the rule filtering and/or prediction process does not identify any appropriate crop for the input provided, then the system produces an easily parsable, structured JSON message that communicates that no recommendation can be given. By developing a simple, clear input-output structure, it allows easy integration with other external applications such as dashboards and tools that provide automated decision making.

### 3.7.2 Integration of Rule-Based Filtering

Before the model makes any predictions, rules will be applied to filter the crops that meet the environmental criteria so that nothing will be recommended that is either impractical or potentially damaging from an environmental perspective due to the comparison of the environmental data at the farm with predetermined limits (in a JSON file named crop_rules_2.json) set for each crop. The limits for recommended crops include the lowest and highest/maximum Recommended Limits for environmental parameters such as Soil Nutrients, Temperature, Humidity, pH, and Rainfall. The function called rule_based_filtering() goes through every crop while checking to see whether any user's environmental data is within an acceptable range for each of the crops. The only crops that meet all of the required criteria for the user's environmental data will be selected as final predictions.

### 3.7.3 Model Deployment Workflow

The Flask framework is used to deploy the model as a good API facility that will be accessible to other applications via a remote call from the API service. The workflow consists of a series of steps that take place once a POST request containing data for environmental parameters and time-related information has been sent to the /recommend endpoint. Since Flutter cannot access a model built from Python directly, Ngrok is used to expose the Flask API via a secure public URL so that the mobile application can interact with the locally hosted model while in development. Once the POST request has been received, the API service loads the saved Model along with its required Artefacts that consist of the trained model, etc. The service then extracts from the POST request the values that were submitted and then utilizes the rule-based filter module to find all of the crops that are suitable for the environmental conditions specified in the POST request. The Final Crop Selection process includes building a Feature Vector that incorporates the information passed on to a Random Forest Regression Model to predict the future price of each shortlisted crop. Once all predicted prices have been calculated, the Model will evaluate which Shortlisted Crop has the highest predicted price and return this crop name as a recommendation to the user along with the predicted price and a Supporting Explanation to facilitate user understanding.

## 3.8 Mobile Application Design

This section shows the application design for the mobile application prototype.

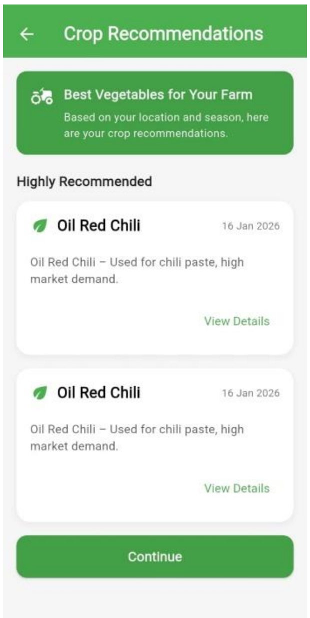



**Figure 3.5** Recommendation Screen

Figure 3.5 show recommendation screen. This page contains the recommended crop type based on the user's current condition and market condition. The system provides a list of the most highly recommended vegetable and crop. For each recommended crop there is a brief explanation of what makes it suitable to grow.

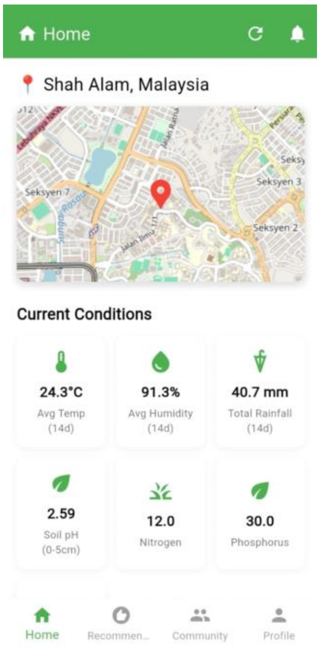



**Figure 3.6** Home Screen

Figure 3.6 show home screen, where this page providing users with an overview of the environmental and soil conditions at their farm location. At the top of the page is a map showing where the user is located, and the map enables the application to return data to the user that is relevant to their specific location. The Current Condition section of the page provides the user with information regarding the average monthly temperatures, humidity, total rainfall, and soil properties from the past, in addition to the soil characteristics. Moreover, clicking on the Get Recommendation button will allow users to proceed directly to the crop recommendation page. By combining both environmental and soil characteristics in one place, farmers are able to understand more about the current state of the farming environment for their farm.

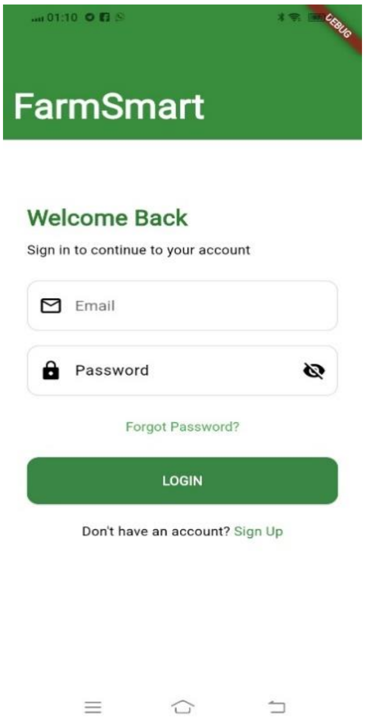



**Figure 3.7** Sign-in Screen

Figure 3.7 is a login interface for the application. Existing users can use their email and password to login. Also available are forgotten password help and new account registration.

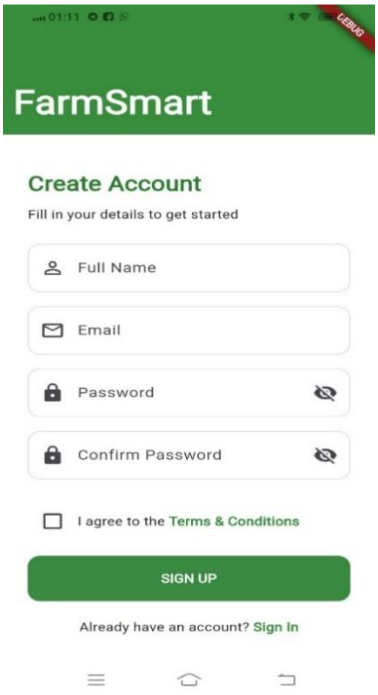



**Figure 3.8** Sign Up Screen

Figure 3.8 shows the sign up screen. This page is used to register for a FarmSmart Account. Here, it captures some basic user information, such as the user's first and last name, email address, and password, to create a secure profile that identifies the user. Additionally, users are required to accept the terms and conditions of use as part of the sign-up process before being granted access to the features of the application.

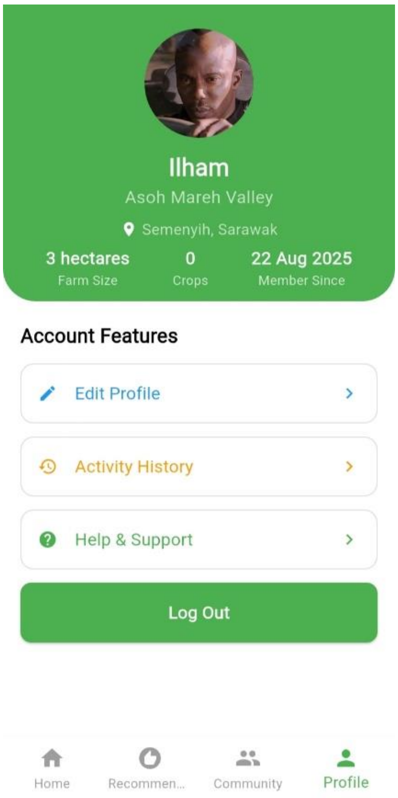



**Figure 3.9** Profile Screen

Figure 3.9 show the profile screen where this page functions as a settings or profile management section. It displays user and farm details, including name, farm location, farm size, number of crops. It also provides access to account features such as editing the profile, viewing activity history, getting help and support, and logging out.

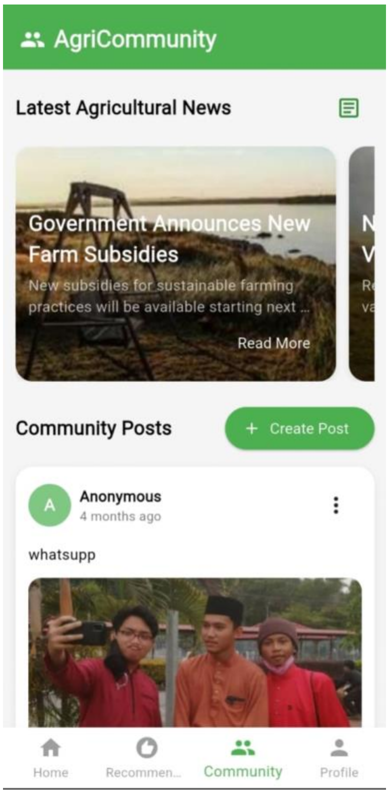



**Figure 3.10** Community Screen

Figure 3.10 shows the community screen. The purpose of this page is to allow users to connect with other users. Users can use the page to share their experiences, get answers to their questions, and give and get information about the agricultural industry. On the Community page, this page also has a "Latest Agricultural News" section. The Latest Agricultural News section contains all of the newest information, such as announcements, upcoming grants and loan programs that farmers may qualify for and what sustainable agricultural practices are.

The Community Posts area, located beneath the news section, allows users to see what other community members have written about. In addition to being able to see posts by other users, users can create their own new post from the "Create Post" button and upload any text or images related to farming activity/issue/general discussion.

## 3.9 Application Development

This stage involves of developing the system based on the previous design. The system is built using Flutter for both the front and back ends of the mobile application and Python to develop a machine learning model. Firebase is used for application data storage and for basic application services. It used to store user data, application input, recommendation results, etc. The purpose of this stage is to create a working prototype that can retrieve the environmental inputs, make the predictions, and display the vegetable crop recommendations through a user-friendly interface. At this stage, the system is still in its early prototype phase, with a simple layout and core features in place.

### 3.9.1 Overview of the Integrated Workflow

This application combines three elements, which are data collection, filtering data according to rules, and predicting crop performance using Random Forest, into an integrated process. The entire process begins when the user enters the primary inputs, which are soil data such as soil type, pH, etc since there is no API that provides the parameter values for each location. Next, the system will retrieve the current environmental conditions, such as temperature, humidity, and weather status using an external weather API. At this point in time, the application utilises a rules-based filtering engine (RBFE), used to determine which types of crops would be ideal for growing based on both the current environmental conditions and requirements. After getting the valid set of possible crops, the application activates the trained model to predict the expected harvest-week price for each of those possible crops. Once the price has been predicted for each crop, the application ranks all of the crops according to the price predictions and provides the end user with the highest price of vegetable crop type. The combination of these processes ensures that this application allows farmers to consider potential revenues in addition to the environmental suitability of a crop.

### 3.9.2 API Integration (OpenWeatherMap)

The app gives real-time weather info by using the OpenWeatherMap API. The API provides environmental information such as temperature, humidity, rainfall, etc based on the user's location. The values provided are then used by the rule-based filtration module to determine whether the environment conditions are suitable for growing crops. API requests are sent through a secure HTTP call, and the JSON responses received are parsed to weather attributes. By using the API to deliver real-time weather data, the recommendations made by the app can be updated in real-time based on the weather conditions. This provides users with more trust in their decision-making process since it is based on live environmental data.

### 3.9.3 Rule-Based and Machine Learning Combination in the Application

The recommendation application uses a hybrid for its filtering capabilities with a Rule-Based filtering methodology version and Machine Learning Prediction approach, where Rule-Based serves as a first decision layer by evaluating weather parameters against a set of defined crop requirements stored in a JSON rule file. Crop requirements for this system consist of the scope of acceptable temperatures, humidities, etc., for determining crop compatibility. Only those crops that meet the environmental condition requirements will continue on to the second decision stage and will be further evaluated by the second decision layer. After creating the list of crop types that are environmentally suitable, the model was utilized to generate predictions of the expected harvest price for each crop type, based on the price information received weekly from FAMA and encoded crop types and time features. This hybrid design gives farmers better decision support because it combines the predictive accuracy of machine learning models with the practical application of rule-based filtering to create more valuable recommendations.

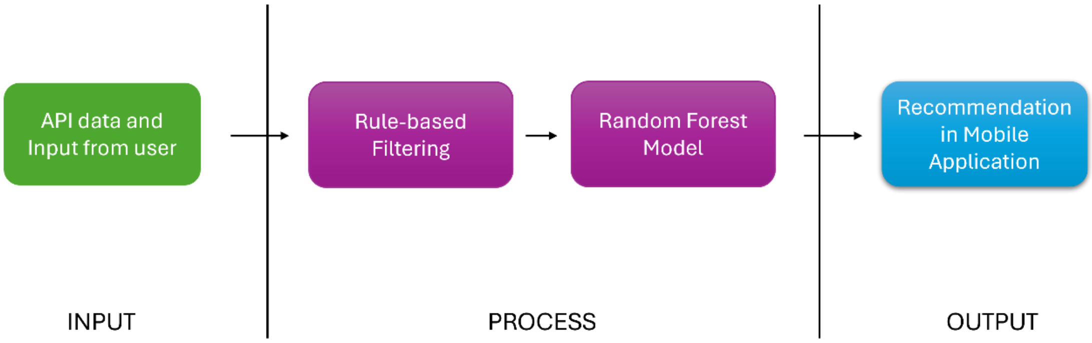



**Figure 3.11** Recommendation process workflow

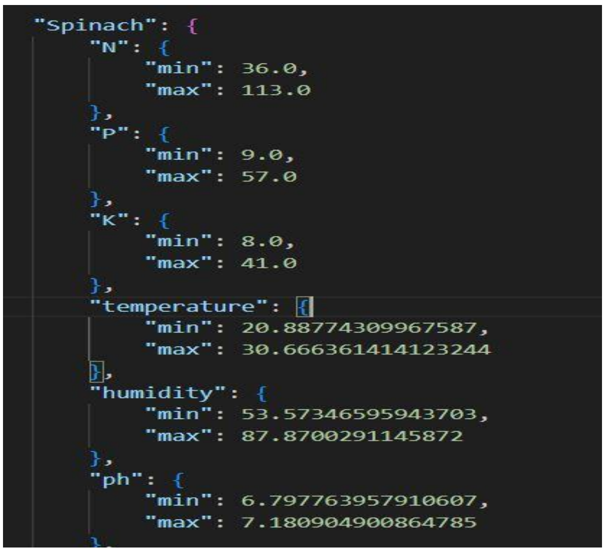
Figure 3.11 shows the flow of how the recommendation is made. To understand the recommendation workflow. Below are the steps for producing the recommendation:

### Input

The system gets the input information from two major sources. The first type of data collected is environmental data that is collected in real-time from outside using the public API in Open WeatherMap. In addition, users will manually enter soil condition values such as Nitrogen, pH soil, Phosphorus, Potassium into the application for the correct evaluation of crop suitability based on both the current environmental conditions as well as the current soil conditions.

### Process

Rule-Based Filtering (Environmental Suitability):



```
"Spinach": {
    "N": {
        "min": 36.0,
        "max": 113.0
    },
    "P": {
        "min": 9.0,
        "max": 57.0
    },
    "K": {
        "min": 8.0,
        "max": 41.0
    },
    "temperature": {
        "min": 20.88774309967587,
        "max": 30.666361414123244
    },
    "humidity": {
        "min": 53.57346595943703,
        "max": 87.8700291145872
    },
    "ph": {
        "min": 6.797763957910607,
        "max": 7.180904900864785
    },
```

**Figure 3.12** Example of a rule for the Spinach crop

- Figure 3.12 shows the rule-based approach used for identifying the suitable crop. This filtering system uses rule-based logic to apply rules associated with certain crops based on the conditions of environmental and soil inputs. These environmental and soil input data are considered facts from which the rules about the crops will be activated from a defined knowledge base. For each crop, the system checks the input parameters (N, P, K, temp, humidity, pH, and rainfall) to see if they are between the proper minimums and maximums in order to activate the crop specific suitability rules. Unlike other automated filtering systems, the rules are fired through direct conditional comparisons (forward chaining), and a suitability score is calculated based on how closely the inputs match the ideal ranges. Once cropped achieve a suitability score above predefined thresholds, and are included in a shortlist that is transferred to the Random Forest Model for continued pricing predictions.

Machine learning model (Price Prediction):

- The shortlisted crop types produced in the rule-based stage are then passed into the machine learning model that is trained on historical market price data. This model predicts the market prices of the crops at their expected harvest periods for the crops in the shortlist and determines the most profitable crop based on future predicted prices.

**Output**

The mobile application presents the best vegetable crop recommendations that can be planted based on the environment and economics conditions. The rule-based filtration process uses soil data and weather patterns combined with profitability from the machine learning model to recommend the most suitable crop for the current environmental conditions and expected economic return.

## 3.10 Reporting Documentation

In this stage, all key elements of the project will be recorded and assembled into a finalized report. This phase encompasses everything that has taken place, from the very beginning until the prototype of the system was developed. The documentation will set forth the background of the project, stated goals, research completed, data collected, model produced, design produced, and tests completed. Documentation will record all technical aspects, such as how the data was collected, how the Random Forest model was created, and how the mobile application was developed. The explained process will be documented by diagram, flowchart, table, and screenshot.

## 3.11 Summary

This chapter has described the methodology used to develop a vegetable crop recommendation application through the integration of a mobile application and the Random Forest algorithm. The methodology begins with the preliminary analysis of the most important environmental and market variables that will influence decisions about crops. Data acquisition was done using two datasets, the first containing weekly vegetable prices from FAMA and the second providing environmental information from Kaggle. Both datasets were cleaned, formatted, encoded and preprocessed to feed into the model to ensure that they are complete and consistent. The process for developing a Random Forest regression model includes the use of growth cycle shifting to predict the harvest time price. The process of feature engineering, Train and Test Split, and hyperparameter tuning is done to improve model quality. After the Model is completed, it will be saved, evaluated using R2, MSE, MAE, and RMSE, and then deployed using a Flask API. Before making predictions, the rule-based filter will limit the number of crop types to only those that are suited for the environment. The model implementation also provides a clear input and output structure, as well as utilising the predicted results of the crops being considered to create a crop recommendation list. Lastly, the interface of the mobile application was created using Flutter. This involves both real-time environmental data pulled from the API and manually entered data about soils, combined with the rule-based filter and trained model. The result of this combination was a very easy-to-use recommendation application that is capable of providing end-users with an efficient interaction.

# CHAPTER 4

# RESULTS AND FINDINGS

This chapter highlights the results of evaluating the proposed model and application. In this chapter, the performance of the Machine Learning model created during the previous chapters will be analysed, and how accurate and reliable the predictions produced by the Random Forest Regression model were. The results of this analysis will detail how well the Random Forest Regression predictions using Cross Validation, Hyperparameter Tuning through the use of Grid Search, and the performance of the model on both training and testing datasets. Multiple Evaluation Metrics, such as MAE, MSE, RMSE, and $R^2$, were used to measure how well the Model performed. Finally, it has the discussion that summarise the findings, identify the strengths, and explain any deficiencies that may limit its success in the future.

## 4.1 Before and After Tuning Model Performance Results

The listed results in this section are those from the Random Forest Regression model used in this study. The performance was evaluated based on various statistical measures to determine how accurate, reliable, and effective Random Forest is for predicting Vegetable Crop prices.

### 4.1.1 Baseline Results

The following subsection of this section describes what the Random Forest regression model would have been like before Hyperparameter Tuning. The baseline Random Forest regression model will be a basis for evaluating how effective the Hyperparameter tuning techniques were at improving model performance in the future. Table 4.1 shows the results from the 5-Fold Cross Validation of the baseline Random Forest regression model. Both selected parameter settings and associated evaluation metrics are reported within the table as a preliminary evaluation of the modelling prediction capabilities.

**Table 4.1** Baseline Results Before Hyperparameter Tuning

| **Parameter Setting** | **Value** |
|---|---|
| Number of Estimators (n_estimators) | 100 |
| Maximum Depth (max_depth) | 10 |
| Minimum Samples Split (min_samples_split) | 2 |
| Minimum Samples Leaf (min_samples_leaf) | 1 |
| Maximum Features (max_features) | None |
| **Evaluation Metric** | **Result** |
| $R^2$ Score (Mean ± Std) | 0.9362 ± 0.0058 |
| Mean Squared Error (MSE) | 0.8301 |
| Mean Absolute Error (MAE) | 0.5640 |
| Root Mean Squared Error (RMSE) | 0.9103 |

Based on Table 4.1, a Random Forest Regression baseline model was evaluated by applying 5-fold cross-validation to its training dataset to verify that the model results would hold true for other splits of the data and not just be dependent on one particular partition of the data. From the results of 5-fold cross-validation, the mean $R^2$ value for the proposed Random Forest Regression model was 0.9362, with a standard deviation of ± 0.0058. This indicates that the Random Forest Regression model was able to explain approximately 93.62% of the variance associated with crop pricing data across all 5 folds, with minimal variability in the results across folds. Due to the small standard deviation across the 5 folds, it indicates that the Random Forest Regression model's performance was stable and consistent across all 5 folds used for the five-fold cross-validation process.

With respect to the error metrics, the Random Forest Regression model produced a mean MSE of 0.8301, mean MAE of 0.5640 and mean RMSE of 0.9103. The above values show that the model predicted crop pricing values on average were less than

RM 1.00 per kilogram different from the actual target crop pricing values throughout the entire training sample. Given the natural fluctuations and volatility associated with crop pricing in agricultural markets, this accuracy level can be considered adequate for a baseline Random Forest Regression model.

To establish the baseline hyperparameter configuration for cross-validation, this model uses a total of 100 trees with a maximum depth of 10, a minimum sample size of 2 required for any given internal node to be split, 1 required at every leaf, and None for max features, where all features are considered when splitting. Cross-validation results clearly show that this baseline Random Forest model has the potential to learn useful patterns from this data set, pointing to the fact that this model serves as a good basis on which to improve performance via hyperparameter tuning and that the $R^2$ is high while the error values are low enough to indicate suitability for crop price prediction.

### 4.1.2 Hyperparameter Tuning Results (Grid Search Combination Analysis)

This section describes the outcomes of the hyperparameter tuning for the Random Forest Regression model, which was used to optimise the Random Forest Regression Model's performance. The training dataset was evaluated using a grid search technique with 5-fold cross-validation in order to explore multiple combinations of hyperparameters. The goal of this tuning process was to find a set of parameters that provides the best predictive accuracy and the least amount of prediction error.

#### a) Hyperparameter Search Space

Table 4.2 shows the hyperparameter search space used for tuning the Random Forest regression model. The listed hyperparameters and their corresponding values were selected to explore different model configurations and identify the combination that produces the best predictive performance.

**Table 4.2** Parameter and Value Combinations

| Hyperparameter | Values | Number of Options |
|---|---|---|
| n_estimators | [100, 150, 200, 300, 400, 500] | 6 |
| max_depth | [6, 10, 15, 20, None] | 5 |
| min_samples_split | [2, 5, 10] | 3 |
| min_samples_leaf | [1, 2, 4] | 3 |
| max_features | ['sqrt', 'log2', None] | 3 |

**Total combinations = 6×5×3×3×3**

**= 810 combinations**

Based on Table 4.2, a total of 810 unique hyperparameter combinations were created based on the hyperparameter grid. Each of these combinations is an individual Random Forest configuration and was tested using K-Fold Cross Validation against the training dataset. For each configuration, the model performance been tracked using four different metrics, which are $R^2$, MSE, MAE, and RMSE, in order to make a thorough comparison among all of the different parameter configurations.

An initial exploratory hyperparameter search space was defined for the Random Forest Model by testing a range for each parameter such as n_estimators between 50 and 1,000, etc. The results suggested that the $R^2$ score and error metric began to reach a stable level of performance after about 700-1000 trees and showed diminishing returns at amounts greater than that. There would be no point in extending the value for each parameter as it did not produce a better result. This strategy of starting with a broad exploratory space and then refining it based on performance trends has been well established as best practice in Random Forest hyperparameter tuning. According to Probst et al. (2019), tuning hyperparameters like n_estimators, max_depth, min_samples_split, min_samples_leaf, and max_features over a wide variety of values will lead to a more effective model than is possible with the default parameter values.

**b) Result of Combination Tested**

Because of the huge volume of evaluated configurations, it is impractical to show all results in this main chapter. In this section, only the 10 best and worst hyperparameter combination results are shown.

Here are the best and worst 10 hyperparameter combinations based on the performance of those combinations as measured by a 5 folds cross-validation-based $R^2$ score. The purpose of this method is to report configurations that provide both high accuracy in predicting test data, as well as a stable result when evaluated with different training data splits.

Here also have information regarding error performance metrics (MAE, MSE, and RMSE) for all the best and worst hyperparameter combinations, however, these additional metrics are not considered when ranking the hyperparameter combinations within tables 4.3 and 4.4.

Table 4.3 presents the top 10 best-performing hyperparameter combinations obtained from the grid search process. These combinations are ranked based on their average $R^2$ score, providing an overview of the most accurate and stable Random Forest model configurations.

**Table 4.3** Top 10 best Hyperparameter Overall Combination Performance

| Rank | Hyperparameter Configuration | $R^2$ mean + Std | MAE | MSE | RMSE |
|---|---|---|---|---|---|
| 1 | n=500, depth=None, split=2, leaf=1, feat=None | 0.9655 ± 0.0035 | 0.3597 | 0.4499 | 0.6697 |

**Table 4.3** (Continued)

| **Rank** | **Hyperparameter Configuration** | **R² mean + Std** | **MAE** | **MSE** | **RMSE** |
|---|---|---|---|---|---|
| 2 | n=400, depth=None, split=2, leaf=1, feat=None | 0.9655 ± 0.0034 | 0.3596 | 0.4500 | 0.6699 |
| 3 | n=500, depth=20, split=2, leaf=1, feat=None | 0.9654 ± 0.0035 | 0.3604 | 0.4505 | 0.6702 |
| 4 | n=300, depth=None, split=2, leaf=1, feat=None | 0.9654 ± 0.0034 | 0.4514 | 0.3599 | 0.6710 |
| 5 | n=300, depth=20, split=2, leaf=1, feat=None | 0.9653 ± 0.0034 | 0.4520 | 0.3605 | 0.6714 |
| 6 | n=200, depth=None, split=2, leaf=1, feat=None | 0.9653 ± 0.0032 | 0.3597 | 0.4522 | 0.6716 |
| 7 | n=200, depth=20, split=2, leaf=1, feat=None | 0.9653 ± 0.0032 | 0.3602 | 0.4525 | 0.6718 |
| 8 | n=150, depth=None, split=2, leaf=1, feat=None | 0.9652 ± 0.0032 | 0.3597 | 0.4538 | 0.6728 |
| 9 | n=150, depth=20, split=2, leaf=1, feat=None | 0.9652 ± 0.0032 | 0.3603 | 0.4542 | 0.6731 |
| 10 | n=100, depth=20, split=2, leaf=1, feat=None | 0.9652 ± 0.0034 | 0.3606 | 0.4535 | 0.6725 |

Table 4.4 displays the 10 most poorly performing hyperparameters from the grid search. The average $R^2$ scores for five-fold cross-validation were used to sort these hyperparameters from worst to best in terms of predicted accuracy and usage as predictors.

**Table 4.4** Top 10 Worst Hyperparameter Overall Combinations Performance

| Rank | Hyperparameter Configuration | $R^2$ mean + Std | MAE | MSE | RMSE |
|---|---|---|---|---|---|
| 1 | n=100, depth=6, split=2, leaf=1, feat=Sqrt | 0.7031 ± 0.0089 | 1.5107 | 3.8674 | 1.9663 |
| 2 | n=100, depth=6, split=2, leaf=1, feat=Log2 | 0.7031 ± 0.0089 | 1.5107 | 3.8674 | 1.9663 |
| 3 | n=100, depth=6, split=5, leaf=2, feat=Sqrt | 0.7037 ± 0.0093 | 1.5109 | 3.8590 | 1.9641 |
| 4 | n=100, depth=6, split=5, leaf=2, feat=log2 | 0.7037 ± 0.0093 | 1.5109 | 3.8590 | 1.9641 |
| 5 | n=100, depth=6, split=6, leaf=2, feat=Sqrt | 0.7049 ± 0.0118 | 1.5055 | 3.8414 | 1.9598 |
| 6 | n=100, depth=6, split=6, leaf=2, feat=Log2 | 0.7049 ± 0.0118 | 1.5055 | 3.8414 | 1.9598 |
| 7 | n=100, depth=6, split=5, leaf=1, feat=Sqrt | 0.7052 ± 0.0090 | 1.5071 | 3.8398 | 1.9592 |
| 8 | n=100, depth=6, split=5, leaf=1, feat=Log2 | 0.7052 ± 0.0090 | 1.5071 | 3.8398 | 1.9592 |
| 9 | n=100, depth=6, split=10, leaf=1, feat=Sqrt | 0.7059 ± 0.0133 | 1.5031 | 3.8295 | 1.9564 |
| 10 | n=100, depth=6, split=10, leaf=1, feat=log2 | 0.7059 ± 0.0133 | 3.8295 | 1.5031 | 1.9564 |

Table 4.5 show best and worst hyperparameters based on MAE. In Table 4.5 it is possible to see how different hyperparameter configurations will produce lower prediction errors than others, where lower MAE values indicate that the predicted price was more accurate.

**Table 4.5** Top 10 best and worst performance for MAE

| Rank | Best Combinations | MAE |
|---|---|---|
| 1 | n=400, depth=None, split=2, leaf=1, feat=None | 0.3596 |
| 2 | n=150, depth=None, split=2, leaf=1, feat=None | 0.3597 |
| 3 | n=200, depth=None, split=2, leaf=1, feat=None | 0.3597 |
| 4 | n=500, depth=None, split=2, leaf=1, feat=None | 0.3597 |
| 5 | n=300, depth=None, split=2, leaf=1, feat=None | 0.3599 |
| 6 | n=200, depth=20, split=2, leaf=1, feat=None | 0.3602 |
| 7 | n=400, depth=20, split=2, leaf=1, feat=None | 0.3602 |
| 8 | n=150, depth=20, split=2, leaf=1, feat=None | 0.3603 |
| 9 | n=500, depth=20, split=2, leaf=1, feat=None | 0.3604 |
| 10 | n=300, depth=20, split=2, leaf=1, feat=None | 0.3605 |
| **Rank** | **Worst Combinations** | **MAE** |
| 1 | n=100, depth=6, split=5, leaf=2, feat=Sqrt | 1.5109 |
| 2 | n=100, depth=6, split=5, leaf=2, feat=Log2 | 1.5109 |
| 3 | n=100, depth=6, split=2, leaf=1, feat=Sqrt | 1.5107 |
| 4 | n=100, depth=6, split=2, leaf=1, feat=Log2 | 1.5107 |
| 5 | n=100, depth=6, split=5, leaf=1, feat=Sqrt | 1.5071 |
| 6 | n=100, depth=6, split=5, leaf=1, feat=Log2 | 1.5071 |
| 7 | n=100, depth=6, split=2, leaf=2, feat=Sqrt | 1.5055 |
| 8 | n=100, depth=6, split=2, leaf=2, feat=Log2 | 1.5055 |
| 9 | n=100, depth=6, split=10, leaf=1, feat=Sqrt | 1.5031 |
| 10 | n=100, depth=6, split=10, leaf=1, feat=Log2 | 1.5031 |

Table 4.6 below outlines the 10 best and 10 worst hyperparameter combinations according to the MSE metric. This table highlights how different model configurations affect prediction error, where lower MAE values indicate more accurate price predictions.

**Table 4.6** Top 10 best and worst performance for MSE

| Rank | Best Combinations | MSE |
|---|---|---|
| 1 | n=500, depth=None, split=2, leaf=1, feat=None | **0.4499** |
| 2 | n=400, depth=None, split=2, leaf=1, feat=None | 0.4500 |
| 3 | n=400, depth=20, split=2, leaf=1, feat=None | 0.4505 |
| 4 | n=500, depth=20, split=2, leaf=1, feat=None | 0.4505 |
| 5 | n=300, depth=None, split=2, leaf=1, feat=None | 0.4514 |
| 6 | n=300, depth=20, split=2, leaf=1, feat=None | 0.4520 |
| 7 | n=200, depth=None, split=2, leaf=1, feat=None | 0.4522 |
| 8 | n=200, depth=20, split=2, leaf=1, feat=None | 0.4525 |
| 9 | n=100, depth=None, split=2, leaf=1, feat=None | 0.4535 |
| 10 | n=100, depth=20, split=2, leaf=1, feat=None | 0.4537 |
| **Rank** | **Worst Combinations** | **MSE** |
| 1 | n=100, depth=6, split=2, leaf=1, feat=Sqrt | **3.8674** |
| 2 | n=100, depth=6, split=2, leaf=1, feat=Log2 | **3.8674** |
| 3 | n=100, depth=6, split=5, leaf=2, feat=Sqrt | 3.8590 |
| 4 | n=100, depth=6, split=5, leaf=2, feat=Log2 | 3.8590 |
| 5 | n=100, depth=6, split=2, leaf=2, feat=Sqrt | 3.8414 |
| 6 | n=100, depth=6, split=2, leaf=2, feat=Log2 | 3.8414 |
| 7 | n=100, depth=6, split=5, leaf=1, feat=Sqrt | 3.8398 |
| 8 | n=100, depth=6, split=5, leaf=1, feat=Log2 | 3.8398 |
| 9 | n=100, depth=6, split=10, leaf=1, feat=Sqrt | 3.8295 |
| 10 | n=100, depth=6, split=10, leaf=1, feat=Log2 | 3.8295 |

Table 4.7 below outlines the 10 best and 10 worst hyperparameter combinations according to the RMSE metric. This table provides insight into the overall prediction error of the model, where lower RMSE values indicate better performance and more accurate price predictions.

**Table 4.7** Top 10 best and worst performance for RMSE

| Rank | Best Combinations | RMSE |
|---|---|---|
| 1 | n=500, depth=None, split=2, leaf=1, feat=None | **0.6697** |
| 2 | n=400, depth=None, split=2, leaf=1, feat=None | 0.6699 |
| 3 | n=500, depth=20, split=2, leaf=1, feat=None | 0.6702 |
| 4 | n=400, depth=20, split=2, leaf=1, feat=None | 0.6703 |
| 5 | n=300, depth=None, split=2, leaf=1, feat=None | 0.6710 |
| 6 | n=300, depth=20, split=2, leaf=1, feat=None | 0.6714 |
| 7 | n=200, depth=None, split=2, leaf=1, feat=None | 0.6716 |
| 8 | n=200, depth=20, split=2, leaf=1, feat=None | 0.6718 |
| 9 | n=100, depth=None, split=2, leaf=1, feat=None | 0.6725 |
| 10 | n=100, depth=20, split=2, leaf=1, feat=None | 0.6726 |
| **Rank** | **Worst Combinations** | **RMSE** |
| 1 | n=100, depth=6, split=2, leaf=1, feat=Sqrt | 1.9663 |
| 2 | n=100, depth=6, split=2, leaf=1, feat=Log2 | 1.9663 |
| 3 | n=100, depth=6, split=5, leaf=2, feat=Sqrt | 1.9641 |
| 4 | n=100, depth=6, split=5, leaf=2, feat=Log2 | 1.9641 |
| 5 | n=100, depth=6, split=2, leaf=2, feat=Sqrt | 1.9598 |
| 6 | n=100, depth=6, split=2, leaf=2, feat=Log2 | 1.9598 |
| 7 | n=100, depth=6, split=5, leaf=1, feat=Sqrt | 1.9592 |
| 8 | n=100, depth=6, split=5, leaf=1, feat=Log2 | 1.9592 |
| 9 | n=100, depth=6, split=10, leaf=1, feat=Sqrt | 1.9564 |
| 10 | n=100, depth=6, split=10, leaf=1, feat=Log2 | 1.9564 |

Table 4.3 and 4.4 summarizes the hyperparameter configurations that were selected from the best cross-validated $R^2$ scores. By including the mean $R^2$ value along with the standard deviation (mean ± std), it is possible to get an idea of how consistently each model performed across its validation folds, which is useful when comparing models trained on limited or variable datasets. In addition, all hyperparameter combinations that produced the lowest performance are included to provide a more comprehensive view of the hyperparameter space being searched and are thus helpful in illustrating how poorly a model performance could potentially be within the bounds of this research. Additional evaluation metrics table, including MAE, MSE, and RMSE, are included to maintain a uniform comparison framework with the best and worst performing configurations.

**c) Analysis of Results**

In this section, the results of hyperparameter tuning are presented using multiple tables to allow a clearer and more systematic evaluation of model performance. The hyperparameter search space is presented as above, where the performance for each hyperparameter combination is shown in result table. Each metric used to evaluate numerical performance, such as $R^2$, MAE, MSE and RMSE is provided via the error metric tables.

In addition, each table has been developed in a manner so as to provide a clear difference between the configuration and evaluation of a model performance for the respective hyperparameter settings. The hyperparameter combination tables emphasise how a combination of hyperparameters will affect the structure of a model and its ability to learn, whereas the error metric tables provide the quantitative results for predicting accuracy and error size. Thus, separating the results provides both an easier way to read and an opportunity to assess each aspect of tuning, to making an overall judgement regarding the findings.

i. **Outcome description from Hyperparameter Combination Tables (10 and 11)**

   From the hyperparameter combination tables, it is clear that the Random Forest Hyperparameter Combination has a strong influence over the accuracy of the predictions. It is possible to see how each of these hyperparameters influences the behaviour of the Random Forest Model by comparing and contrasting the highest performing configurations to those with the lowest predicted accuracies.

The best performing hyperparameters were a higher number of decision trees created (n_estimators = 500), moderate depth of trees (max_depth = None), minimal tree splitting criteria (min_samples_split = 2; min_samples_leaf = 1), and used all features (max_features = None). This combination also had the highest cross-validated $R^2$ score around 0.9655 and had consistently low error rates. The fact that this hyperparameter combination produced such a high $R^2$ score means that this particular configuration could describe a large percentage of the variation in crop prices while also being able to show a consistent performance across different subsets of data (cross-validation folds).

The combinations with the lowest performance were the configurations that used a lower number of trees and limited tree growth. The combinations with fewer estimators and lower tree depths exhibited significantly lower performance, as evidenced by the $R^2$ score dropping to approximately 0.731, indicating that approximately +-30% of the variance of the target price could not be explained by the model. This indicates that these types of configurations do not have adequate model capacity for learning the complex relationships that exist in market price data.

The comparison between the best and worst combinations illustrates that increasing the number of estimators is a fundamental factor in stabilizing predictions because it reduces the amount of variability in the predictions made by individual trees. Also, the additional growth allowed to the trees enables the model to better represent seasonal effects, market trends and crop-specific price patterns compared to models with the lowest number of tree estimators and deepest tree growth.

ii. **Outcome Analysis from Error Metric Tables**

The hyperparameter combination tables show trends of general performance but do not contain data showing how accurate and consistent each hyperparameter combination can demonstrate its predictive ability.

The hyperparameter combination producing the highest amount of accuracy produced the least amount of error when predicting harvest price, as supported by the MAE containing values of less than RM 0.50 and the RMSE containing values of less than RM 0.90, indicating that on average, the harvest price predicted by the model was off less than RM 1.00 per kilogram from the actual market price, which would still be acceptable based on the volatility associated with agricultural markets but demonstrates a strong amount of predictive ability.

By contrast, the combinations that performed the worst had substantially more errors. The MAE value was greater than RM 1.50, and the RMSE value approached RM 2.00, which indicates a significant difference between the predicted price and the actual price. The high RMSE value indicates there are a number of large prediction errors present which are heavily weighted in this evaluation metric. Therefore, this demonstrates that these configurations are not suitable for practical application and are prone to extreme instability.

When reviewing the error metrics tables, one major takeaway is that $R^2$ by itself does not indicate the quality of a model. Some combinations will yield moderate $R^2$ values but may produce both moderate MAE and RMSE values. This demonstrates that multiple evaluation metrics should be considered and combined to provide a complete picture of how well a model will perform. It is easier to identify those combinations that may appear to be acceptable under $R^2$ but are not capable of providing accurate predictions when reviewing the error metrics in a separate table.

It important to emphasise that the MAE and RMSE results mentioned in the present research are derived from using raw crop price data that were expressed in Malaysian Ringgit (RM). The target variable was not normalised or scaled during training, as preserving the original price units allows the prediction errors to be directly interpreted in real-world terms. Therefore, MAE and RMSE results could be much higher than what is commonly seen in models based on normalised target prices, where the error results are frequently less than 1. In this setting, although the MAE or RMSE may appear higher than normal, it does not serve as an indication of an inefficient model. Rather, it provides a reflection of the inherent price magnitude of agricultural items. By utilising this approach, the resulting error values are still practical and easy to interpret.

**iii. Comparison Between Before and After Hyperparameter Tuning**

After analysing the hyperparameter combination tables and error metric tables individually. It is essential to compare models' comparisons of pre- and post-hyperparameter tuning performance to identify any improvements due to the hyperparameter tuning process through improved model performance in terms of increased prediction accuracy and reliability. Table 4.8 show the comparison result before and after hyperparameter tuning are applied.

**Table 4.8** Baseline and Tuned Model Result Comparison

| **Metric** | **Before Tuning (Baseline Model)** | **After Tuning (Best Model)** | **Performance Change** |
|---|---|---|---|
| $R^2$ (Mean) | 0.9362 | 0.9655 | R2 has increased to signify that the optimised model is fitting the data more accurately than the base model, with more variance explained in the crop price. |
| $R^2$ Standard Deviation | ± 0.0058 | ± 0.0035 | The standard deviation for the R2 decreased to indicate that the model has become more stable when using the cross-validation folds. |
| MAE | 0.5640 | 0.4499 | The average MAE has decreased, which reflects a lower average price error per kilogram. |

**Table 4.8** (Continued)

| Metric | Before Tuning (Baseline Model) | After Tuning (Best Model) | Performance Change |
|---|---|---|---|
| MSE | 0.8301 | 0.3597 | MSE has improved significantly due to having fewer large MSE values and overall prediction errors. |
| RMSE | 0.9103 | 0.6697 | RMSE has decreased after model tuning indicating that predictions made by the optimised model will be closer to market prices. |

Before hyperparameter tuning, the baseline Random Forest regression model already demonstrated strong predictive capability. The mean $R^2$ score hyperparameter tuning was 0.9362, meaning that this baseline model was able to explain more than 93% of the variance in crop price data. Furthermore, the RMSE of 0.9103 RM/kg indicates that the predicted price of a crop should error by an average of less than RM 1.00 per kg. This value is acceptable given the significant volatility of agricultural market prices.

However, after applying the hyperparameter tuning using the Grid Search technique. The tuned model achieved a mean $R^2$ score of 0.9655, indicating an improved ability to explain the variance of the dependent variable. Although the numerical increase between pre and post hyperparameter tuning in terms of the $R^2$ is small, it represents a significant increase in accuracy for a regression-type model that is dealing with noisy real-world data.

A significant improvement in the performance of the tuned model compared to the original model on error metrics. For example, when using MAE as an error metric, the difference between the two models was approximately 12%, from 0.5640 to 0.4499. Also, the difference in RMSE between the two models was 0.9103 to 0.6697, which is a drop of 0.2406 in RMSE. The drop in RMSE is important as this metric highlights the reduction of large errors.

The significant decrease in MSE from 0.8301 to 0.6697 also indicates that the tuned model generates more consistent predictions. The performance gains of these two models indicate that tuning the hyperparameters was successful in improving the overall accuracy and stability of the model.

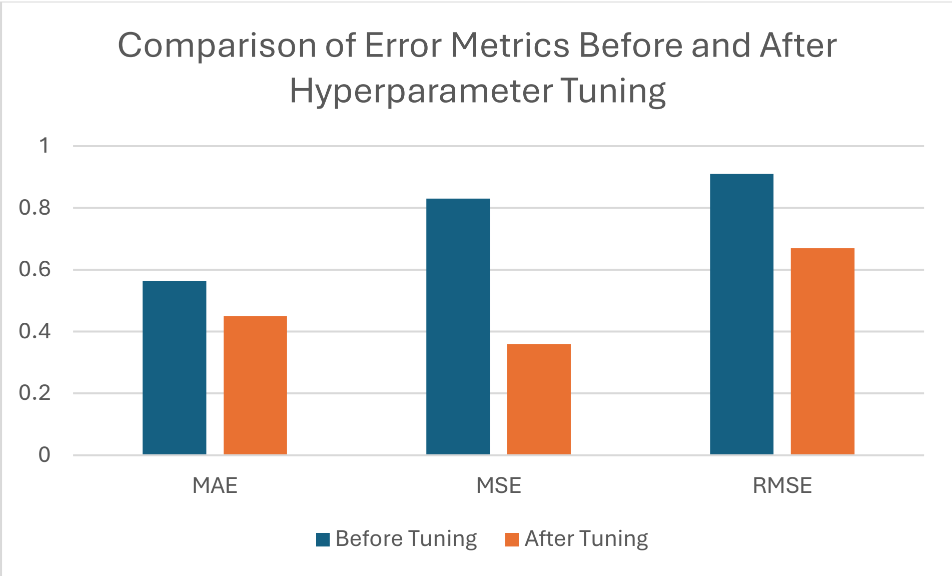



**Figure 4.1** Model error comparison

Figure 4.1 show error comparison before and after hyperparameter tuning. Hyperparameter tuning makes a significant difference in the prediction accuracy of the Random Forest regression model. A comparison of the Random Forest regression model performance with hyperparameter tuning versus baseline model, using three different error metrics (MAE, MSE, RMSE), show that the model produces higher errors before the tuning process. The results produced by the tuned model (MAE, MSE, RMSE) indicate improvements and highlight the increased stability of the predictions produced by the tuned model.

### iv. Hyperparameters That Increased Model Performance

Consistent patterns were noted in an analysis of the best-performing hyperparameter combinations. To begin with, nearly all of the combinations that resulted in the best model accuracies had at least 150 trees (n_estimators) as opposed to the baseline model, which had only 100 trees. The reason that larger numbers of trees were found in all of the highest performing combinations is that they provide a greater degree of stability, since the more trees that are included, the less variance will just be included in the average predictions from the ensemble, thus improving the model's ability to generalize.

The models that achieved the high accuracy consistently included either no maximum depth parameter (max_depth = None) or a very large value for maximum depth, such as 20. This enabled the decision trees to identify more complex and, therefore, more realistic relationships among the agricultural price data. Also, the highest-performing models all demonstrate very little splitting constraint on trees such as min_samples_split = 2 and min_samples_leaf = 1, allowing trees to create more detailed patterns from the data and to learn better. The ability of these models to be flexible does not introduce uncertainty or variability in the cross-validation folds since their $R^2$ values have low standard deviations. The models that used max_features = None performed consistently better than the other value such as sqrt and log2 for the various features used in the model. Thus, it can be concluded that a model with the full set of features is more able to learn significant relationships, while limiting the set of features limits the model ability to learn valuable relationships.

## d) Best Parameter Selection

Based on the hyperparameter tuning results and evaluation using multiple performance metrics, the following hyperparameter configuration was selected as the final optimal model for subsequent testing and application deployment:

**Selected Hyperparameter Configuration:**

- *n_estimators* = 500
- *max_depth* = None
- *min_samples_split* = 2
- *min_samples_leaf* = 1
- *max_features* = None

This configuration corresponds to the top ranked model in the hyperparameter combination analysis, achieving a cross-validated $R^2$ score of 0.9655 ± 0.0035, together with consistently low error values across MAE, MSE, and RMSE. Certain hyperparameter combinations produce individual error metric values as shown in the top 10 performance ranking that were slightly lower than above selected hyperparameter configurations. However, these were not the basis upon which the final parameter set were chosen. Instead, a balanced and comprehensive evaluation method was employed when selecting final parameter versus merely minimising one error metric. MAE, MSE and RMSE provide numeric values indicating the accuracy of a model predictions, while the $R^2$ Value quantifies the extent to which the model can explain for variability in crop prices, which is essential for ensuring reliable learning of crop pricing patterns.

The chosen parameter combination demonstrated a mean absolute error (MAE) of 0.3597, meaning that the average predicted price deviates from the average actual market price by less than 0.40 RM per kg, which is considered acceptable given the natural volatility typical of agricultural markets. Furthermore, the root mean square error (RMSE) of 0.6697 also demonstrates that any large prediction error is effectively regulated by the model using that parameter, and because RMSE gives greater weight to larger errors, the RMSE of this model provides an indication of greater confidence that extreme cases of misprediction are infrequent, and therefore, more reliable for practical use in agricultural decision-making.

The model $R^2$ score shows little to no variation between different cross validated splits. A low standard deviation indicates of the $R^2$ score (0.0035) the performance of the model is quite stable, hence has good generalisation ability, and not overly dependent upon how the data was partitioned. When compared to the other best performing combinations in terms of their high levels of low errors, this particular parameter combination has achieved a great balance between these three characteristics where it demonstrating significant explanatory power $R^2$ score while exhibiting low prediction error and high levels of deleted residuals. The final hyperparameter configuration was not simply chosen from the lowest error, but from an overall evaluation approach to measure prediction performance, error magnitudes and model stability combined as a selection factor for the next step of deployment and evaluate on the test dataset.

### 4.1.3 Final Model Performance (Train vs Test)

This part is to describe the model performance through comparing the results obtained from using the training dataset for 5-fold cross-validation vs the results obtained using the test dataset after the hyperparameter tuning process. The purpose of this comparison is to determine how well the model generalises to new data, as well as to identify potential areas of overfitting or underfitting with respect to the training data. Table 4.9 shows the best hyperparameter sets. The training set produced an average $R^2$ score of 0.9655 ± 0.0035, along with RMSE = 0.6697, MAE = 0.3597, and MSE = 0.4499. This means that the model was able to capture the variation in the crop prices but had low prediction error. In the test set, the model produced an $R^2$ score of 0.9607, with RMSE = 0.6952, MAE = 0.3248, and MSE = 0.4833. The close relationship between the training and test metrics indicates that the model can generalise well to data that was not available during training.

**Table 4.9** Result Training and Testing Tuned Model

| Dataset (80:20) | $R^2$ Score | MAE | MSE | RMSE |
|---|---|---|---|---|
| **Training (Cross-Validation)** | 0.9655 ± 0.0035 | 0.3597 | 0.4499 | 0.6697 |
| **Testing (Unseen Data)** | 0.9607 | 0.3248 | 0.4833 | 0.6952 |

Based on Table 4.9, the difference in $R^2$ score and RMSE results from comparing the training set and test set is minimal, as expected when evaluating a model using an unknown data set. Therefore, this behaviour does not indicate an extreme instance of overfitting. The very small difference between the training and test set results may be an indicator that the model does not memorise training data but instead identifies meaningful patterns in training data which will generalise well. Additionally, the testing data and training data contain consistently high R-squared values along with low error metric values, which indicates that the model is not underfitting because it can account for a significant amount of variation in price and produce accurate predictions.

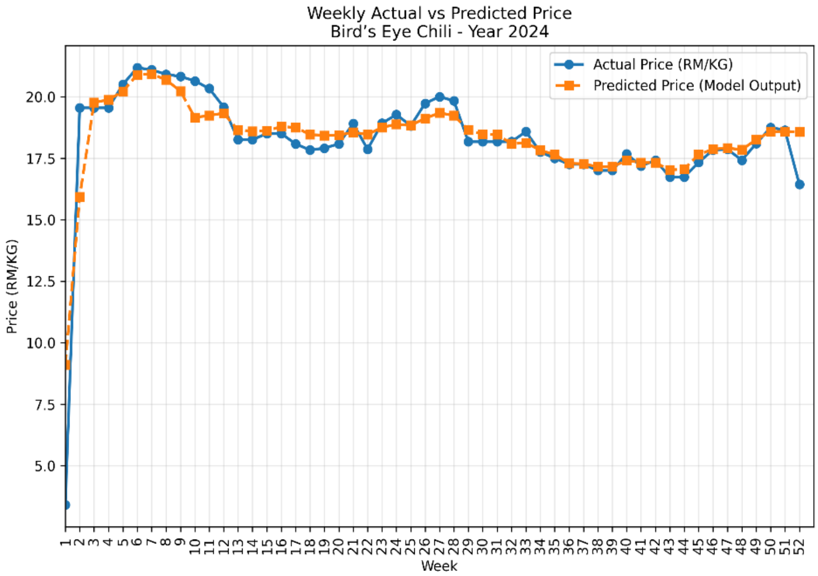
The best model to use was determined to be the one that had nice balanced performance and consistent results using both the training data and the test data sets. The model strong $R^2$ values and low mean absolute error MAE, MSE, and RMSE indicate that the model accurately predicts crop prices while explaining the reasons why those prices change.

Practically speaking, with a test MAE of 0.3248, there is an average difference in predicted crop prices versus actual market prices of around RM 0.32 per kilogram, which is reasonable for making real-world agricultural decisions. Additionally, this model has been applied in the crop recommendation module, which forecasted a harvest price of RM 23.17 per kilogram for the Bird's Eye Chili crop, proving that the forecasting model is appropriate in developing crop recommendations.

### 4.1.4 Predicted vs Actual Visualisation

This section will illustrate the prediction and actual market prices weekly over time of the best crop as determined by the model during both the training and test phases. The focus of this analysis is on the highest performing crop identified by the Random Forest model, and for this case, Bird's Eye Chili was produced as the best crop from the model prediction. This graph highlights how the model captures weekly crop price variation to make predictions from previous weekly market price data.

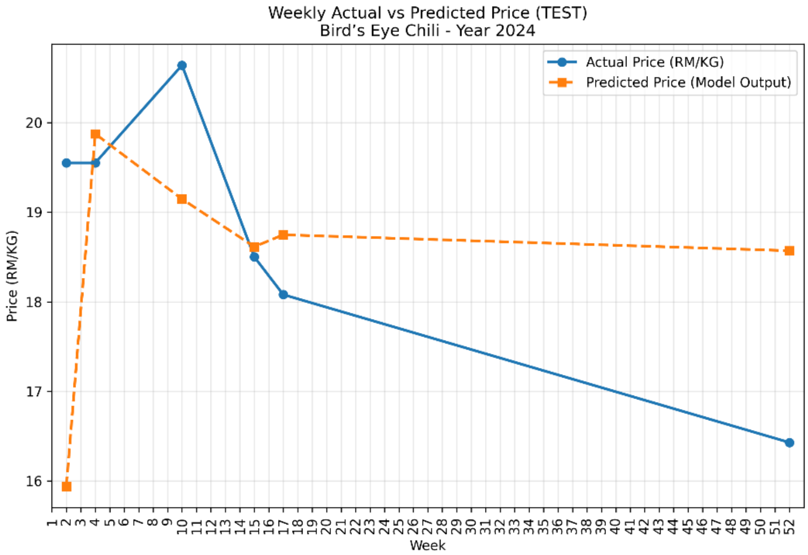
The visualisations illustrate the performance of both the baseline model and the optimised model in relation to hyperparameter tuning. The results of this comparison clearly demonstrate the improvement in predictive accuracy as well as the improved agreement of predicted trends with the historical data following hyperparameter tuning. The price trends are illustrated with a price vs time (week number) graph that enables a straightforward comparison between historical market prices and model-derived predicted prices for the specified time frame.




| Actual Price (RM) | Predicted Price (RM) |
|---|---|
| 3.40 (W1) | 9.10 |
| 19.55 (W2) | 15.94 |
| 19.55 (W3) | 19.76 |
| 19.55 (W4) | 19.87 |
| 20.50 (W5) | 20.22 |
| 21.18 (W6) | 20.90 |
| 21.09 (W7) | 20.92 |
| 20.91 (W8) | 20.68 |
| 20.82 (W9) | 20.23 |
| 20.64 (W10) | 19.15 |
| 20.33 (W11) | 19.24 |
| 19.58 (W12) | 19.33 |
| 18.25 (W13) | 18.64 |
| 18.25 (W14) | 18.60 |
| 18.50 (W15) | 18.61 |
| 18.50 (W16) | 18.78 |
| 18.08 (W17) | 18.75 |
| 17.83 (W18) | 18.46 |
| 17.91 (W19) | 18.42 |
| 18.08 (W20) | 18.44 |
| 18.91 (W21) | 18.55 |
| 17.86 (W22) | 18.46 |
| 18.92 (W23) | 18.75 |
| 19.27 (W24) | 18.88 |
| 18.82 (W25) | 18.84 |
| 19.73 (W26) | 19.12 |
| 20.00 (W27) | 19.34 |
| 19.83 (W28) | 19.23 |
| 18.17 (W29) | 18.64 |
| 18.17 (W30) | 18.47 |
| 18.17 (W31) | 18.45 |
| 18.17 (W32) | 18.09 |
| 18.58 (W33) | 18.13 |
| 17.75 (W34) | 17.82 |
| 17.50 (W35) | 17.65 |
| 17.25 (W36) | 17.30 |
| 17.25 (W37) | 17.27 |
| 17.00 (W38) | 17.15 |
| 17.00 (W39) | 17.15 |
| 17.67 (W40) | 17.41 |
| 17.18 (W41) | 17.32 |
| 17.42 (W42) | 17.32 |
| 16.73 (W43) | 17.01 |
| 16.73 (W44) | 17.06 |
| 17.33 (W45) | 17.65 |
| 17.83 (W46) | 17.87 |
| 17.86 (W47) | 17.91 |
| 17.42 (W48) | 17.84 |
| 18.09 (W49) | 18.26 |
| 18.75 (W50) | 18.58 |
| 18.64 (W51) | 18.59 |
| 16.43 (W52) | 18.57 |

**Figure 4.2** Training graph baseline model (Bird's Eye Chili)

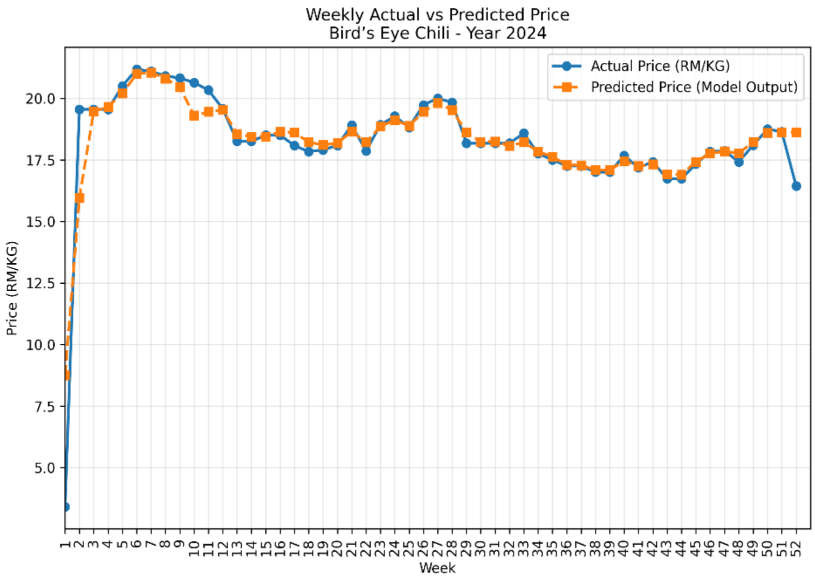
Bird's Eye Chili weekly actual vs predicted prices during 2024, shown in Figure 4.2, used baseline models before hyperparameter tuning. Overall, in comparison to training on all the years from 2015-2025, the baseline model can be considered as a well accomplished model because the predicted price is very close to the actual overall price trend of the market. Even during occasional large differences in price on a weekly basis, the model has been able to accurately show the price movement and seasonal patterns. Therefore, prior to any hyperparameter optimisation, baseline models have shown great success with Bird's Eye Chili price prediction.




| Actual Price (RM) | Predicted Price (RM) |
|---|---|
| 19.55 (W2) | 15.94 |
| 19.55 (W4) | 19.87 |
| 20.64 (W10) | 19.15 |
| 18.50 (W15) | 18.61 |
| 18.08 (W17) | 18.75 |
| 16.43 (W52) | 18.57 |

**Figure 4.3** Testing graph baseline model (Bird's Eye Chili)

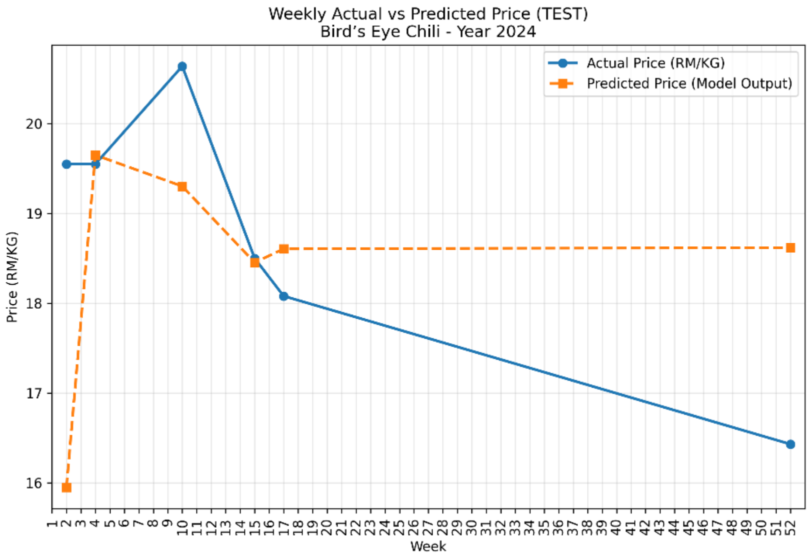
Figure 4.3 shows the weekly market price of Bird's Eye Chili and the predicted price in 2024 generated by the Random Forest model from the baseline model. The model was built on data from 80% of the sample, the test set only reflect 20% of the overall data. So while the figure cannot show predictions for all weeks in the sample, it does illustrate the ability of the random forest model to produce reasonable estimates of week-to-week price changes, which correlate with the actual prices.




| Actual Price (RM) | Predicted Price (RM) |
|---|---|
| 3.40 (W1) | 8.74 |
| 19.55 (W2) | 15.95 |
| 19.55 (W3) | 19.47 |
| 19.55 (W4) | 19.65 |
| 20.50 (W5) | 20.19 |
| 21.18 (W6) | 20.99 |
| 21.09 (W7) | 21.05 |
| 20.91 (W8) | 20.80 |
| 20.82 (W9) | 20.46 |
| 20.64 (W10) | 19.30 |
| 20.33 (W11) | 19.45 |
| 19.58 (W12) | 19.53 |
| 18.25 (W13) | 18.56 |
| 18.25 (W14) | 18.44 |
| 18.50 (W15) | 18.45 |
| 18.50 (W16) | 18.63 |
| 18.08 (W17) | 18.61 |
| 17.83 (W18) | 18.22 |
| 17.91 (W19) | 18.10 |
| 18.08 (W20) | 18.18 |
| 18.91 (W21) | 18.65 |
| 17.86 (W22) | 18.22 |
| 18.92 (W23) | 18.85 |
| 19.27 (W24) | 19.11 |
| 18.82 (W25) | 18.88 |
| 19.73 (W26) | 19.45 |
| 20.00 (W27) | 19.81 |
| 19.83 (W28) | 19.53 |
| 18.17 (W29) | 18.62 |
| 18.17 (W30) | 18.23 |
| 18.17 (W31) | 18.24 |
| 18.17 (W32) | 18.06 |
| 18.58 (W33) | 18.21 |
| 17.75 (W34) | 17.84 |
| 17.50 (W35) | 17.62 |
| 17.25 (W36) | 17.30 |
| 17.25 (W37) | 17.28 |
| 17.00 (W38) | 17.08 |
| 17.00 (W39) | 17.09 |
| 17.67 (W40) | 17.44 |
| 17.18 (W41) | 17.25 |
| 17.42 (W42) | 17.32 |
| 16.73 (W43) | 16.91 |
| 16.73 (W44) | 16.90 |
| 17.33 (W45) | 17.42 |
| 17.83 (W46) | 17.76 |
| 17.86 (W47) | 17.85 |
| 17.42 (W48) | 17.77 |
| 18.09 (W49) | 18.23 |
| 18.75 (W50) | 18.60 |
| 18.64 (W51) | 18.62 |
| 16.43 (W52) | 18.62 |

**Figure 4.4** Training graph tuned model (Bird's Eye Chili)

Based on Figure 4.4, the Random Forest model fitted with hyperparameters accurately reflects the actual weekly prices for Bird's Eye Chili in 2024 in comparison to the expected weekly prices. Because the graph shows that the expected weekly prices follow the actual weekly prices from week to week throughout the year 2024, it demonstrates an outstanding ability of the Random Forest model to fulfil the general price increase at the beginning of the year and decline in price through the third quarter of 2024 while stabilising back to near the average set price late in the year. This confirms that the Random Forest model trained with hyperparameters is successfully predicting the seasonal prices for Bird's Eye Chili. Furthermore, the model reacts positively to week-to-week fluctuations in price, with only slight discrepancies detected in weeks where prices experienced larger fluctuations. When comparing the tuned model with the baseline model, the tuned model has a more consistent approach to prices. The peaks and bottoms of pricing were more closely aligned than the baseline model. Therefore, it follows that the hyperparameter tuning of the tuned model reduced the overall prediction error and created more stability within the model.




| Actual Price (RM) | Predicted Price (RM) |
|---|---|
| 19.55 (W2) | 15.95 |
| 19.55 (W4) | 19.65 |
| 20.64 (W10) | 19.30 |
| 18.50 (W15) | 18.45 |
| 18.08 (W17) | 18.61 |
| 16.43 (W52) | 18.62 |

**Figure 4.5** Testing graph tuned model (Bird's Eye Chili)

The 2024 weekly actual vs predicted prices for Bird's Eye Chili shown in Figure 4.5, using 20% of the dataset still unseen. This graph shows how well the tuned Random Forest model generalise when forecasting future market prices. As shown, the predicted prices are very similar to the actual market prices in most of the weeks. The fitted model was able to correctly feature the general trends of price changes. For example, the initial decline in price during the first weeks of the year, then a gradual rise and finally some fluctuations towards the end. This demonstrates that the fitted model is able to learn and generalise from the training data effectively to be able to apply it to new, unseen data.

### 4.1.5 Sample Crop Price Predictions (Predicted Model Only)

In this section, the sample inputs and outputs of the crop price prediction from the model are presented. The objective is to show how the crop price prediction model works by demonstrating what types of data inputs are needed by the model and what kinds of results are produced from those inputs. The purpose of this section is not to evaluate the model or assess its performance because the detailed evaluation of the model was discussed in the previous sections.

**Sample Model Input**

The developed model requires a small set of structured inputs for utilizing the model just described. These inputs will allow estimation of future prices based on when the user anticipates harvesting their products. Below is the inputs to the model:

1) Year
2) Month
3) Week
4) Crop Type

The user is not the one who determines the crop type, but instead determines it through a rule-based filtering process based on environmental conditions to identify crops appropriate for use related to predicting price. At this stage, the model will consider all eligible crops prior to estimating their prices at harvest.

**Sample Model Output**

The model creates an estimate of the crop price using the optimum hyperparameter configuration determined from cross-validation. An example of the output created during the model development process is shown below.

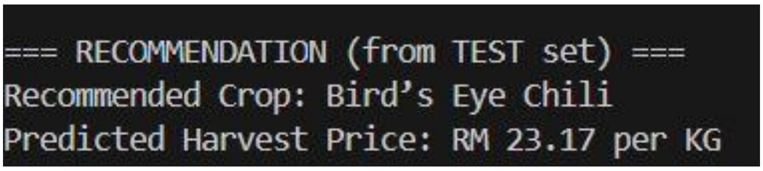



**Figure 4.6** Output from model

Figure 4.6 is an example of output produced directly from the model. This represents the clarity and interpretability of the crop recommendation with price prediction output from the model with the appropriate data input. Thus, this creates a conclusion that the Model can effectively provide valid and usable predictions when given structured input data for use in crop recommendations.

## 4.2 Application Development Results

This section only shows the generated input and output of the main process from the application that has been developed. It does not aim to showcase all user interface components, as the full system design and interface layout are in the appendices section and have already been discussed in the earlier chapter. It describes the input data used, and the output from the crop price recommendation application. The results provided showed that the crop price recommendation model developed has practical use in the real world and will be useful for supporting decision-making using machine learning models.

### 4.2.1 User Input Interface Results

The User Input Interface is the main way users interact with the application. The interface has been designed easy to use so that the user can easily data needed to make predictions about crop prices.

Soil related data is manually entered by the user in numeric format. The Primary reason is that currently no API exists to provide accurate, locally relevant soil data that could be automatically integrated into the application. Therefore, users will be prompted to manually enter their own soil parameter numbers. They may include some combination of soil type classifications and/or measurements taken from soil field studies or standard references. Manually entering numeric soil parameters also makes them compatible with the machine learning model, which requires the use of numeric features to predict crop prices.

Figure 4.7 show the screen for input need to be fill by the user in order to receive the recommendation. It has 4 different values of soil properties, and all the value is in numeric. Below is an example of soil data needed as shown in Figure 4.7:

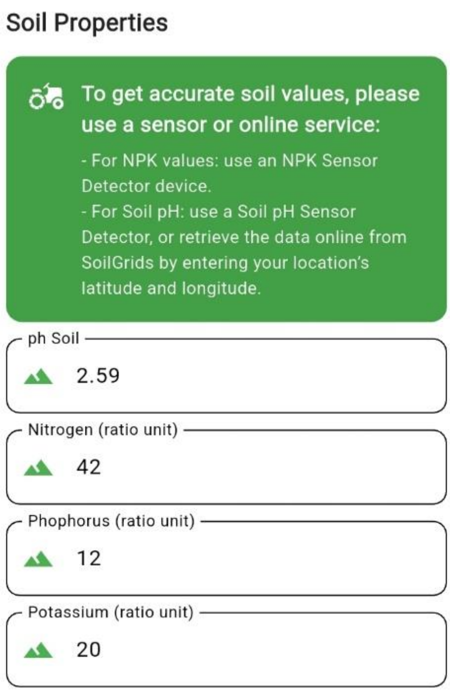



**Figure 4.7** Generated interface input for soil data

Figure 4.7 shows the screen for input need to be fill by the user in order to receive the recommendation. It has 4 different values of soil properties, and all the value is in numeric. Below is an example of soil data needed as shown in Figure 4.7:

1) Nitrogen
2) Soil pH
3) Phosphorus
4) Potassium

In addition to manual soil inputs, the environmental information is received automatically through an external API that allows the system to acquire this information. The environmental data used by the system will be in terms of monthly averages instead of real-time data, as the monthly average data reflects the data in the environmental dataset as it represents the data in monthly average and its more useful for agricultural planning purposes as it provides a better representation of environmental conditions throughout the crop cycle and reduces the impact of short term weather fluctuations.

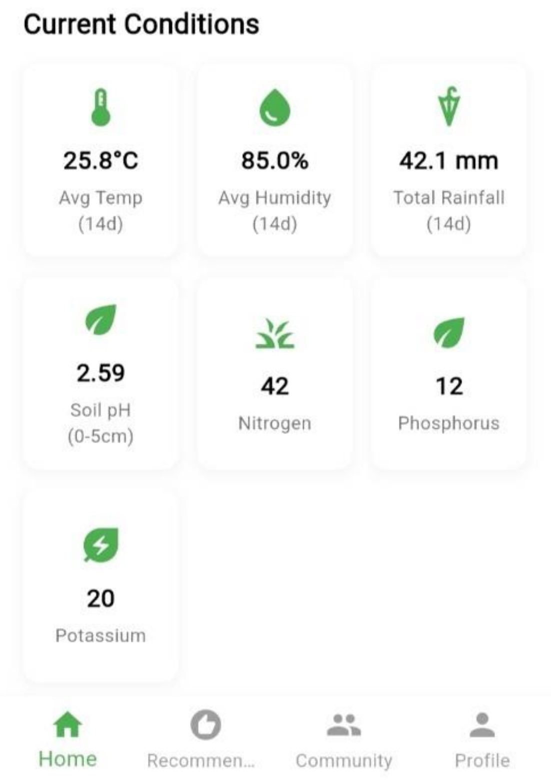



**Figure 4.8** Generated interface input environmental data from API

1) Temperature
2) Humidity
3) Rainfall

Figure 4.8 shows the home screen, which shows environmental data from the API and manual input from the user. Once all the required numerical input fields have been completed, the user can get their recommendations. The system performs initial validation on the input data, before forwarding the information to the trained Random Forest model to perform predictions. The user interface is designed to enable users with technical experience to use the application and receive accurate output.

### 4.2.2 Recommendation Output Results

Once the user has entered all of the necessary information, the system produces a recommendation output using the model that has previously been trained. The main recommendation output from the system includes the description and the expected price at harvest time.

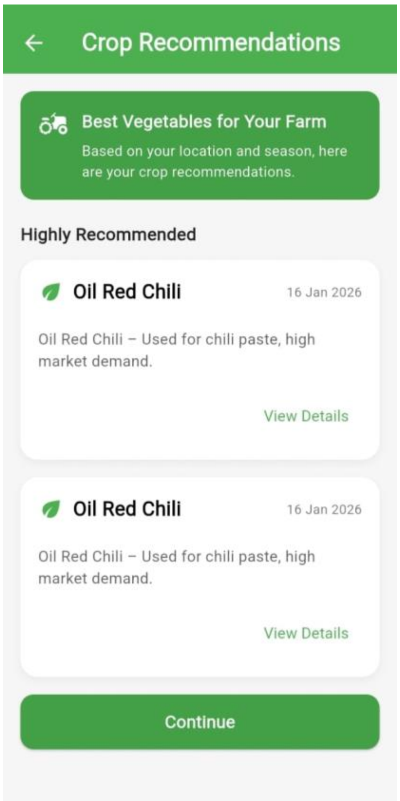



**Figure 4.9** Interface recommendation/output generated

Figure 4.9 shows the recommendation output screen, in this screen user sees the expected market price for the recommended crop variety, based on the environmental conditions at the time the request was made. For instance, the application suggests that the best crop for these conditions would be an Oil Red Chili, given the description information about the crop and the date recommendation was generated.

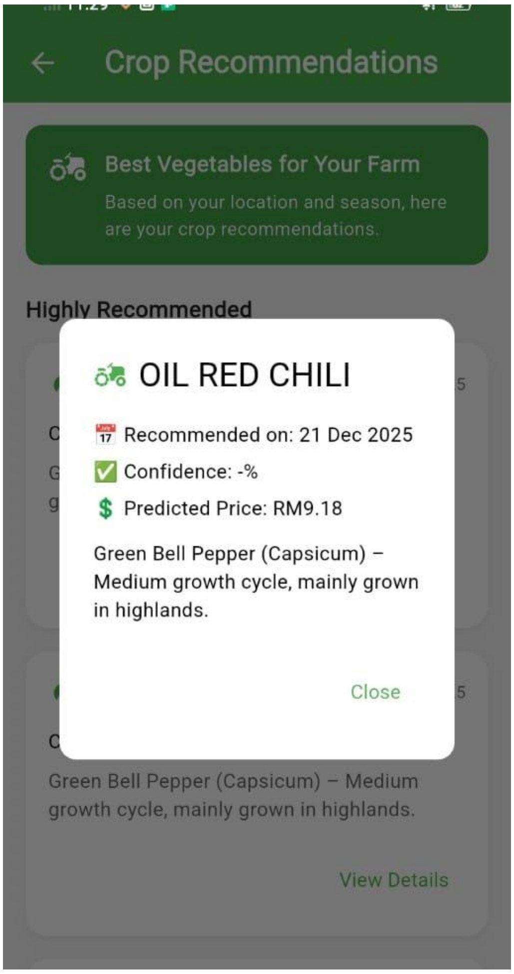



**Figure 4.10** Detail Pop-up Information Screen

Figure 4.10 shows the recommendation details screen. The application additionally allows users access to further details on their recommended action through the view details feature. Clicking on the view details option provides a detailed of the prediction results with additional context for the crop selected. The view details feature provides greater transparency for the user, thereby enhancing the user's understanding of how the recommendation was developed.

Overall, the output from the recommendation demonstrates how effectively the Machine Learning predictive models have been developed, providing insights into the prediction results that will be leveraged as decision support tools when planning for the future of vegetable crop planting. The combination of environmental variables, real-time weather information, and historic price trends provides the necessary decision support to allow a farmer to effectively plan for the future of farming activity.

## 4.4 Discussion

The section contains details concerning the overall results from evaluating the designed Crop Price Recommendation application. In general, it provides an interpretation of what the result represents in terms of technical and practical applications then it emphasises the advantages of the designed system based on extensive experimental findings and applications.

### 4.4.1 Interpretation of Findings

The findings of this study indicate that the Random Forest Regression Model developed has a very strong ability to predict vegetable crop prices from historical market data and contextual features. The $R^2$ score and prediction error of the Random Forest model remained consistent throughout all of the evaluation processes and none of them is below 70% $R^2$ score, which indicates the model had strong predictive power.

The results of the baseline cross-validation process showed that the Random Forest Regression model was explaining more than 93% of the variance in crop prices. This outcome indicates that the features chosen for the Random Forest Regression Model are appropriate for providing an explanation for the price behaviour of the crop, which is typically driven by a combination of seasonal effects, the characteristics of the crop, and fluctuations in the market. Additionally, because the MAE and RMSE were relatively low compared to the average price of RM 1.00 per kg, it is reasonable to conclude that prediction errors will be insignificant when considering the high volatility associated with agricultural products.

After applying hyperparameter tuning using the grid Search technique, it can be seen that the model has bold improvement. The mean $R^2$ score improve up to 0.9655 of the tuned model improved up to 0.9655, and the MAE, MSE, and RMSE error metrics decreased significantly. These results suggest that the tuning process helped to model to capture complex non-linear relationships in the data, while still allowing for a stable solution. Additionally, the small standard deviation of the $R^2$ score across the cross-validation folds suggests that the model provides similar performance on different data partitions and is not very dependent on the particular train split that was used.

Additionally, the difference in training and testing results was small for the unseen test data. The close $R^2$ and RMSE values between training and testing suggest that the model has generalisation capabilities, and that it's not severely overfitting or underfitting the training data. This balance indicates that the model is capturing the underlying price patterns, rather than memorising the training examples.

The graphs that representation the results, specifically the scatter plots and regression lines, illustrates the predicted prices showed wide dispersion from actual prices before hyperparameter tuning. The closer clustering of the predicted prices to the ideal diagonal line after tuning highlights that predicted prices are more accurate, and improved the accuracy, also reducing the error.

Furthermore, the predicted prices and crop recommendations generated through the application layer of the machine learning demonstrate that the predictions of the machine learning models are both realistic and useful when implemented in real-world decision-making processes. As an illustration in the application result section, the ability of the application to make a recommendation of producing Bird's Eye Chili with a predicted sale price shows how the model could be integrated into real-world decision-making processes.

### 4.4.2 Strengths of the Proposed Model

One of the strengths of the Proposed System is its level of predictive accuracy and reliability as demonstrated by the model strong performance across multiple evaluation metrics ($R^2$, RMSE, MAE, MSE). Furthermore, a Comprehensive Hyperparameter Tuning Process using a 5-fold cross-validation was applied to ensure that the selected model is accurate and Stable across all different data splits, thereby minimising the risk of overfitting. The result between the training and testing also indicates that the model has good generalisation ability, which can perform on unseen data. Additionally, by combining Machine Learning-based Crop Price Prediction with Rule-based, it enables the model to provide practical and meaningful decision support for crop selection, demonstrating its applicability in real-world agricultural scenarios.

### 4.4.3 Limitations

While the proposed crop price prediction and recommendation model is able to perform well, it has several limitations. The model uses a historical market price dataset, and inaccuracies can impact the prediction accuracy of the model, even with pre-processing steps taken to clean the data. Besides, there are only a limited number of input features such as crop type, week, month and year to make predictions, and important factors that would influence the price such as pest outbreaks, fertilizer, transportation or logistics costs, government policies and the effects of unexpected events on the market are not included in this model. The weekly data used may miss short-term price variations that can occur on a daily basis.

Additionally, a rule-based approach was used for processing environmental data. While rule-based processing is straightforward, it has no ability to adapt or improve from newly available data. As such, rule-based processing may be overly simplistic in nature, leading to the oversimplification of complex interactions that occur between many variables in the environment and the inability of rule-based systems to effectively perform in atypical or rapidly shifting environmental states. Lastly, the crop recommendation is solely based on future predicted prices, which does not take into account actual farming limitations such as production costs, availability of land and farmer preference; therefore, this system should not be considered the only method of decision making when making crop recommendations.

## 4.5 Summary

In conclusion, this chapter presented the results and evaluation of the proposed crop price prediction and recommendation application. These findings included various methods of evaluation, including hyperparameter tuning and the final training/testing analysis of a Random Forest regression model to assess how well these model performed. As indicated by the results, the Random Forest regression model produced a high overall accuracy with a strong $R^2$ value, and consistently low error measures. With the completed hyperparameter tuning process applied to the model, additional improvements were made regarding prediction error reduction and increasing stability across different datasets. Visual analysis through graphs and application outputs demonstrated that the predicted prices closely followed actual market trends and provided meaningful recommendations. Overall, the findings confirm that the proposed model and application performs reliably and is suitable for supporting vegetable crop type selection.

# CHAPTER 5

# CONCLUSION AND RECOMMENDATIONS

This chapter wraps up the Final Year Project titled " Integrated Vegetable Crop Type Recommendation Using Rule-Based and Random Forest. It summarizes the project and provides an overall view of the research objective, main findings and limitations that were encountered during the project execution, as well as offering suggestions for future enhancement of the application. Additionally, these discussions reflect from Chapters 1 - 4, and it highlight the contributions of this project to smart and sustainable agricultural methods in Malaysia.

## 5.1 Summary

The purpose of this research was to assist farmers in selecting appropriate types of vegetables to grow based on the changing environmental conditions and fluctuating market prices. For many years, farming in Malaysia has relied primarily on the experience and gut feeling of the farmer to select what to grow. However, this approach will not work anymore because of uncertainty about future climate changes and the price fluctuations in products sold on the national market. Therefore, a data-driven approach was proposed where a mobile recommendation application would recommend vegetable crop types to farmers based on environmental data and analysis of current market trends of the vegetable crop prices.

The first objective of the research was to identify what environmental and market characteristics were most important to the farmers in their decision-making process of selecting vegetable crops to grow. In the process of literature review and preliminary study, there are several environmental characteristics that are critical to determining crop type suitability for vegetable crops, such as soil nutrient content (N, P, and K), soil pH, temperature, humidity, and rainfall. Furthermore, it also included market factors such as the weekly trends of vegetable price data provided by FAMA to ensure that the vegetable crops would have the potential to be profitable for farmers.

The second objective focused on developing a predictive machine learning model that predicts which appropriate vegetable crops are to be grown by farmers by creating a Random Forest regression model. The Random Forest regression model was selected because it is robust in its performance, can accommodate non-linear relationships, is less prone to overfitting and has a strong performance history in agricultural-based datasets. The model was trained using historical weekly market prices as a basis, with a growth cycle mapping technique, to predict the price for harvest time.

To evaluate the performance of the Random Forest regression model, a variety of numerical metrics R-squared, root mean squared error (RMSE), mean absolute error (MAE), mean square error (MSE) were used and determined that the model successfully captured price trends and supported profitable crop recommendations.

The third objective was to develop a mobile application prototype that integrate a hybrid by combining the rule-based and Random Forest known as “FarmSmart" to deliver the vegetable crop type recommendations to the user. The mobile application prototype was created using Flutter, and a Flask-based API was used to create a bridge to the machine-learning model based on the requirements of the user. The application prototype used the OpenWeatherMap API to retrieve real-time environmental conditions and utilized a rule-based filtering process to be sure that the environment was suitable before applying the machine-learning price prediction. Therefore, the overall application successfully combined the two key components of environmental suitability and predicted profitability in order to provide farmers with the most suitable vegetable crop recommendations, thus completing all three research objectives.

To summarize, this project successfully accomplished all three research objectives, including identifying key variables affecting crop production, creating a working crop recommendation model based on machine learning and creating a working prototype of a mobile application. Therefore, based on these accomplishments, it can be concluded that the proposed methodology has met the original research objectives and has provided an effective solution for the research questions listed in Chapter 1.

## 5.2 Limitations

While this project goals were supported by the outcomes from its processes, there are also some areas for improvement in terms of limitations. The first being the existence and quality of area-specific or localized agricultural datasets. The dataset used for this project comes from a public data source, meaning the data represents average and simple forms rather than the variety of soils and climates that may be distributed throughout the country. Similarly, the weekly pricing dataset provided by FAMA only provides average prices. Farmers may sell their product at a lower price due to market conditions.

Another limitation is related to choosing the selection factor for the model development. While the literature indicates several environmental, agronomic, economic, and sustainability influences upon vegetable crop selection, only a small segment was integrated into this project. The lack of comprehensive datasets prevents modelling some aspects of crop growth and production. Examples include pest/disease occurrence, fertiliser costs, workforce availability, government subsidy programs, and irrigation systems.

From a technical perspective, although Random Forest can produce accurate predictions, it lacks user-friendly interpretability for farmers who will use it. The prediction algorithm made a recommendation and predicted a price for the user. However, the results did not provide anything to indicate why a particular crop was selected over other crops. This lack of transparency may reduce how much trust users for the algorithms. In addition, the prototype version being tested currently relies upon a strong internet connection to obtain weather data in real-time and access the machine learning API. Since many rural areas have extremely limited or unreliable network connectivity, users in these regions may experience significant barriers when attempting to use machine learning to support their farming operations.

## 5.3 Future Work

In order to increase the efficiency and usability of the vegetables crop recommendation application, there are a number of future improvements that could be implemented. An enhancement to the overall capabilities of the application would be to allow for integration with the Explainable AI (XAI) tools that allow users to understand the recommended crop outputs. The use of SHapley Additive exPlanations (SHAP), or similar technology, would allow farmers to understand how several factors, including soil nutrients, temperature, rainfall and market price trends, affect the recommendations made to them by the application. The addition of this feature would allow farmers to understand the reasoning behind how the recommendations are generated, which would improve their trust in the recommendations and facilitate the making of informed decisions.

Another improvement for this project could be to find ways to create more precise localized recommendations. Examples of this could include collecting soil-specific regional data, yield data at the farm level, fertilizer, land size, and prices supported by local marketplaces instead of national averages. In addition, using things such as SoilGrids APIs or various forms of agricultural sensors would improve model accuracy and allow for recommendations to be tailored specifically to identified farming locations in Malaysia.

Finally, the project can be extended to include dynamic model updates and multi-objective recommendations, which Should allow for more flexibility in choosing options other than just the most profitable crop. In future, users will be able to specify additional priorities, including recommendations for minimizing risk, maximizing water efficiency, and Fertilizer recommendation. This would allow the future version of the commercial web-based platform to adjust as far as how long-term climate change will occur, emerging market trends, and changing agricultural practices.

## REFERENCES


Ahmed, B., Shabbir, H., Naqvi, S. R., & Peng, L. (2024). Smart Agriculture: Current State, Opportunities and Challenges. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3471647

Akkem, Y., Kumar, B. S., & Varanasi, A. (2023). Streamlit Application for Advanced Ensemble Learning Methods in Crop Recommendation Systems – A Review and Implementation. *Indian Journal Of Science And Technology*, *16*(48), 4688–4702. https://doi.org/10.17485/IJST/v16i48.2850

Begho, T., Glenk, K., Anik, A. R., & Eory, V. (2022). A systematic review of factors that influence farmers' adoption of sustainable crop farming practices: Lessons for sustainable nitrogen management in South Asia. In *Journal of Sustainable Agriculture and Environment* (Vol. 1, Issue 2, pp. 149–160). John Wiley and Sons Inc. https://doi.org/10.1002/sae2.12016

Behera, S., & Mishra, B. K. (n.d.). *Smart Crop Prediction Using Random Forest and Machine Learning Models*. https://ssrn.com/abstract=5220733

Bhat, S. A., Hussain, I., & Huang, N. F. (2023). Soil suitability classification for crop selection in precision agriculture using GBRT-based hybrid DNN surrogate models. *Ecological Informatics*, *75*. https://doi.org/10.1016/j.ecoinf.2023.102109

Dahiphale, D., Shinde, P., Patil, K., & Dahiphale, V. (2023). *Smart Farming: Crop Recommendation using Machine Learning with Challenges and Future Ideas*. https://doi.org/10.36227/techrxiv.23504496.v1

Dolaptsis, K., Pantazi, X. E., Paraskevas, C., Arslan, S., Tekin, Y., Bantchina, B. B., Ulusoy, Y., Gündoğdu, K. S., Qaswar, M., Bustan, D., & Mouazen, A. M. (2024). A Hybrid LSTM Approach for Irrigation Scheduling in Maize Crop. *Agriculture (Switzerland)*, *14*(2). https://doi.org/10.3390/agriculture14020210

Elbasi, E., Zaki, C., Topcu, A. E., Abdelbaki, W., Zreikat, A. I., Cina, E., Shdefat, A., & Saker, L. (2023). Crop Prediction Model Using Machine Learning Algorithms. *Applied Sciences (Switzerland)*, *13*(16). https://doi.org/10.3390/app13169288

Gunawan, M. I., Sitopu, J. W., Sechan, G., & Informasi, S. T. (2024). *International Journal of Enterprise Modelling Optimizing Crop Selection: A Multi-Criteria Decision Support System for Sustainable Agriculture* (Vol. 18, Issue 3). www.ieia.ristek.or.id

Hamadani, H., Rashid, S. M., Parrah, J. D., Khan, A. A., Dar, K. A., Ganie, A. A., Gazal, A., Dar, R. A., & Ali, A. (2021). Traditional farming practices and its consequences. In *Microbiota and Biofertilizers, Vol 2: Ecofriendly Tools for Reclamation of Degraded Soil Environs* (pp. 119–128). Springer International Publishing. https://doi.org/10.1007/978-3-030-61010-4_6

Jain, N., Kumar, A., Garud, S., Pradhan, V., & Kulkarni, P. (2017). Crop Selection Method Based on Various Environmental Factors Using Machine Learning. *International Research Journal of Engineering and Technology*. www.irjet.net

Jhajharia, K., Mathur, P., Jain, S., & Nijhawan, S. (2022). Crop Yield Prediction using Machine Learning and Deep Learning Techniques. *Procedia Computer Science*, *218*, 406–417. https://doi.org/10.1016/j.procs.2023.01.023

Kacprzyk, J., Kaynak, O., Pedrycz, W., Polycarpou, M. M., & Wang, J. (2022). *Lecture Notes in Networks and Systems Volume 680 Series Editor*.

Kephe, P. N., Siewe, L. C., Lekalakala, R. G., Kwabena Ayisi, K., & Petja, B. M. (2022). Optimizing Smallholder Farmers' Productivity Through Crop Selection, Targeting and Prioritization Framework in the Limpopo and Free State Provinces, South Africa. *Frontiers in Sustainable Food Systems*, *6*. https://doi.org/10.3389/fsufs.2022.738267

Kumar, S., Sohail, M. A., Jadhav, S., & Gupta, R. K. (2024). LIGHT GRADIENT BOOSTING MACHINE FOR OPTIMIZING CROP MAINTENANCE AND YIELD PREDICTION IN AGRICULTURE. *ICTACT Journal on Soft Computing*, *15*(2), 3551–3555. https://doi.org/10.21917/ijsc.2024.0495

Mallala, B., Ahmed, A. I. U., Pamidi, S. V., Faruque, M. O., & M, R. R. (2025). Forecasting global sustainable energy from renewable sources using random forest algorithm. *Results in Engineering*, *25*. https://doi.org/10.1016/j.rineng.2024.103789

Manish Lad, A., Mani Bharathi, K., Akash Saravanan, B., & Karthik, R. (2022). Factors affecting agriculture and estimation of crop yield using supervised learning algorithms. *Materials Today: Proceedings*, *62*, 4629–4634. https://doi.org/10.1016/j.matpr.2022.03.080

Özen, F. (2024). Random forest regression for prediction of Covid-19 daily cases and deaths in Turkey. *Heliyon*, *10*(4). https://doi.org/10.1016/j.heliyon.2024.e25746

Paithane, P. M. (2023). RANDOM FOREST ALGORITHM USE FOR CROP RECOMMENDATION. *Journal of Engineering and Technology for Industrial Applications*, *9*(43), 34–41. https://doi.org/10.5935/jetia.v9i43.906

Probst, P., Wright, M., & Boulesteix, A.-L. (2019). *Hyperparameters and Tuning Strategies for Random Forest*. https://doi.org/10.1002/widm.1301

Pudumalar, S., Ramanujam. E, & Harine Rajashree, R. (2017). *2016 Eighth International Conference on Advanced Computing (ICoAC)*. IEEE.

Rizzo, G., Agus, F., Batubara, S. F., Andrade, J. F., Rattalino Edreira, J. I., Purwantomo, D. K. G., Anasiru, R. H., Maintang, Marbun, O., Ningsih, R. D., Syahri, Ratna, B. S., Yulianti, V., Istiqomah, N., Aristya, V. E., Howard, R., Cassman, K. G., & Grassini, P. (2023). A farmer data-driven approach for prioritization of agricultural research and development: A case study for intensive crop systems in the humid tropics. *Field Crops Research*, *297*. https://doi.org/10.1016/j.fcr.2023.108942

Sabirin, N. H. A., Mohd Fadhil, N. F., & Arifin, J. (2022). Information Technology (IT) in Agriculture Sector: Issues and Challenges. *Social & Management Research Journal*, *19*(2), 111–137. https://doi.org/10.24191/smrj.v19i2.19307

Shams, M. Y., Gamel, S. A., & Talaat, F. M. (2024). Enhancing crop recommendation systems with explainable artificial intelligence: a study on agricultural decision-making. *Neural Computing and Applications*, *36*(11), 5695–5714. https://doi.org/10.1007/s00521-023-09391-2

Singhal, S., Suthar, O. P., & Mishra, A. (2025). State-of-the-Art and Emerging Trends in Crop Recommendation Systems: A Comprehensive Review. *Procedia Computer Science*, *259*, 443–452. https://doi.org/10.1016/j.procs.2025.03.346

Ullah, A., Saqib, S. E., & Kächele, H. (2022). Determinants of Farmers' Awareness and Adoption of Extension Recommended Wheat Varieties in the Rainfed Areas of Pakistan. *Sustainability (Switzerland)*, *14*(6). https://doi.org/10.3390/su14063194

Wee, G. W. E., & Lim, A. S. S. (2022). Factors Influencing the Behavioral Intention for Smart Farming in Sarawak, Malaysia. *Journal of Agribusiness Marketing*, *9*(1), 37–56. https://doi.org/10.56527/jabm.9.1.4

Wee Wei En, G., & Shou Hui, I. T. (2023). Development of Smart Farming Technologies in Malaysia - Insights from Bibliometric Analysis. *Journal of Agribusiness Marketing*, *10*(1). https://doi.org/10.56527/fama.jabm.10.1.3

Wong, T.-T., & Yeh, S.-J. (2019). Weighted Random Forests for Evaluating Financial Credit Risk. In *Proceedings of Engineering and Technology Innovation* (Vol. 13).

Yash Gupta, & Garima Srivastava. (2024). Crop recommendation system for growing best suitable crop. *International Journal of Science and Research Archive*, *12*(1), 2928–2936. https://doi.org/10.30574/ijsra.2024.12.1.1111

# APPENDICES

## APPENDIX A: Datasets

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | N | P | K | temperatu | humidity | ph | rainfall | label |
| 2 | 90 | 42 | 43 | 20.87974 | 82.00274 | 6.502985 | 202.9355 | rice |
| 3 | 85 | 58 | 41 | 21.77046 | 80.31964 | 7.038096 | 226.6555 | rice |
| 4 | 60 | 55 | 44 | 23.00446 | 82.32076 | 7.840207 | 263.9642 | rice |
| 5 | 74 | 35 | 40 | 26.4911 | 80.15836 | 6.980401 | 242.864 | rice |
| 6 | 78 | 42 | 42 | 20.13017 | 81.60487 | 7.628473 | 262.7173 | rice |
| 7 | 69 | 37 | 42 | 23.05805 | 83.37012 | 7.073454 | 251.055 | spinach |
| 8 | 69 | 55 | 38 | 22.70884 | 82.63941 | 5.700806 | 271.3249 | spinach |
| 9 | 94 | 53 | 40 | 20.27774 | 82.89409 | 5.718627 | 241.9742 | rice |
| 10 | 89 | 54 | 38 | 24.51588 | 83.53522 | 6.685346 | 230.4462 | rice |
| 11 | 68 | 58 | 38 | 23.22397 | 83.03323 | 6.336254 | 221.2092 | spinach |
| 12 | 91 | 53 | 40 | 26.52724 | 81.41754 | 5.386168 | 264.6149 | rice |
| 13 | 90 | 46 | 42 | 23.97898 | 81.45062 | 7.502834 | 250.0832 | enggplant |
| 14 | 78 | 58 | 44 | 26.8008 | 80.88685 | 5.108682 | 284.4365 | rice |

**Environmental Data (Kaggle)**

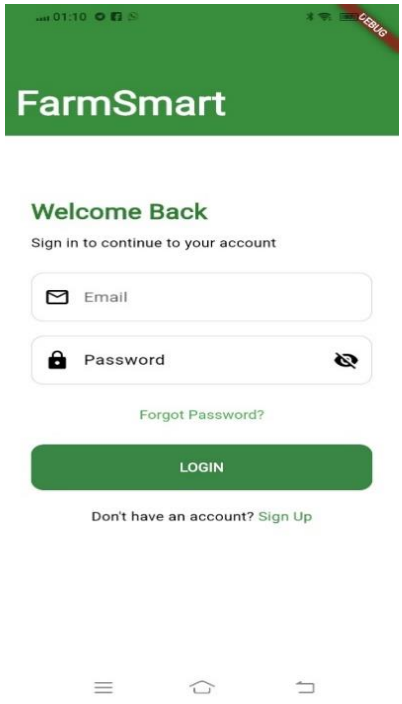
LAPORAN HARGA PURATA KOMODITI
JANUARI 2024 - DISEMBER 2024 (MINGGUAN)
FAMA MALAYSIA

PERINGKAT: LADANG

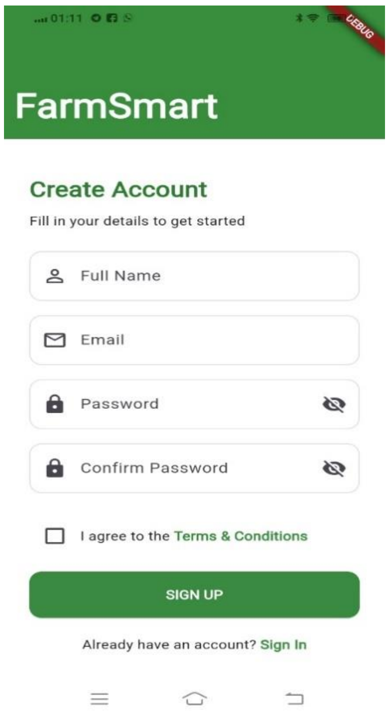
| KOMODITI | GRED | UNIT | MINGGU 1 TAHUN 2024 | MINGGU 2 TAHUN 2024 | MINGGU 3 TAHUN 2024 | MINGGU 4 TAHUN 2024 | MINGGU 5 TAHUN 2024 | MINGGU 6 TAHUN 2024 | MINGGU 7 TAHUN 2024 | MINGGU 8 TAHUN 2024 | MINGGU 9 TAHUN 2024 | MINGGU 10 TAHUN 2024 | MINGGU 11 TAHUN 2024 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BAYAM | F.A.Q | KG | 2.43 | 2.71 | 2.91 | 2.64 | 2.44 | 2.33 | 2.17 | 2.05 | 1.71 | 1.74 | 1.85 |
| CILI HIJAU | F.A.Q | KG | 7.63 | 7.66 | 7.55 | 7.04 | 7.32 | 7.56 | 7.75 | 7.62 | 6.91 | 6.80 | 7.27 |
| CILI MERAH KULAI / KULAI HIBRID | F.A.Q | KG | 13.15 | 12.61 | 12.45 | 11.97 | 12.09 | 11.94 | 12.00 | 10.80 | 10.63 | 9.74 | 9.94 |
| CILI MERAH MINYAK | F.A.Q | KG | 11.81 | 11.73 | 11.31 | 10.10 | 9.48 | 9.54 | 9.94 | 8.91 | 9.26 | 9.05 | 8.64 |
| CILI PADI | F.A.Q | KG | 18.00 | 18.40 | 17.10 | 18.55 | 18.55 | 19.50 | 19.50 | 19.55 | 19.27 | 20.27 | 19.55 |
| DAUN SUP (TANAH TINGGI) | F.A.Q | KG | 10.33 | 8.36 | 5.83 | 5.58 | 6.10 | 5.55 | 7.79 | 7.36 | 4.75 | 4.75 | 6.71 |
| HALIA MUDA | F.A.Q | KG | 10.50 | 10.50 | 10.50 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 7.67 | 7.67 | 7.67 |
| KACANG BENDI | F.A.Q | KG | 3.90 | 4.39 | 4.73 | 4.71 | 5.14 | 5.19 | 4.72 | 4.08 | 4.09 | 4.15 | 4.54 |
| KACANG BUNCIS | F.A.Q | KG | 5.86 | 6.75 | 6.70 | 6.03 | 6.04 | 5.74 | 4.72 | 5.46 | 5.11 | 4.83 | 4.77 |
| KACANG PANJANG HIJAU | F.A.Q | KG | 4.02 | 4.04 | 4.37 | 4.06 | 4.51 | 4.36 | 4.43 | 4.22 | 4.14 | 3.57 | 4.20 |
| KAILAN | F.A.Q | KG | 6.00 | 4.88 | 4.88 | 4.75 | 4.25 | 4.38 | 5.75 | 5.00 | 4.50 | 4.55 | 4.38 |
| KANGKUNG | F.A.Q | KG | 2.51 | 2.50 | 2.63 | 2.52 | 2.35 | 2.37 | 2.02 | 1.83 | 1.64 | 1.63 | 1.76 |
| KUBIS BULAT (TANAH TINGGI) | F.A.Q | KG | 1.90 | 1.70 | 1.70 | 1.29 | 1.16 | 1.51 | 1.54 | 0.96 | 0.89 | 0.93 | 0.96 |
| LABU MANIS | F.A.Q | KG | 2.09 | 1.96 | 1.95 | 1.93 | 1.88 | 1.91 | 1.54 | 1.94 | 1.94 | 1.92 | 1.89 |
| LADA BENGGALA / CAPSICUM HIJAU | F.A.Q | KG | 9.00 | 9.25 | 7.25 | 8.26 | 8.63 | 9.88 | 10.70 | 8.10 | 7.13 | 7.10 | 7.00 |
| LIMAU KASTURI | F.A.Q | KG | 3.65 | 3.76 | 3.63 | 3.44 | 3.17 | 3.30 | 3.71 | 3.71 | 3.50 | 3.79 | 3.80 |
| LIMAU NIPIS | F.A.Q | KG | 4.27 | 3.98 | 4.12 | 3.78 | 3.86 | 4.06 | 4.34 | 4.31 | 3.92 | 4.28 | 4.56 |
| PERIA | F.A.Q | KG | 3.95 | 4.47 | 4.94 | 4.78 | 4.83 | 4.54 | 4.50 | 5.00 | 4.72 | 4.77 | 4.80 |
| SALAD BULAT TANAH TINGGI | F.A.Q | KG | 5.83 | 6.38 | 5.13 | 4.63 | 3.33 | 3.33 | 3.38 | 2.30 | 2.15 | 2.05 | 2.00 |

**Weekly Market Price Data (FAMA)**

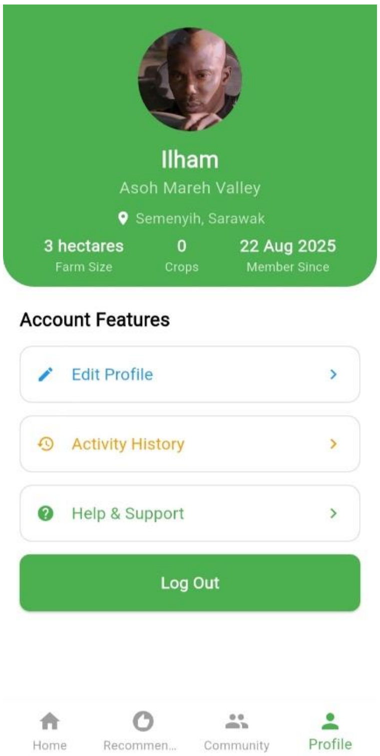
APPENDIX B: Complete Mobile Application Prototype



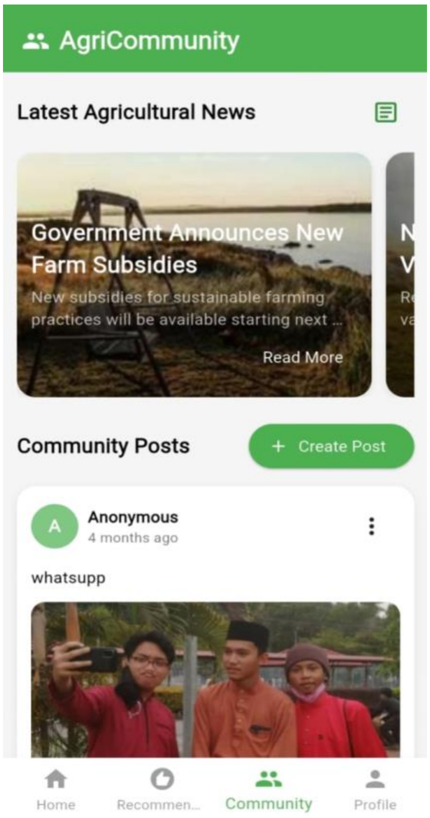

Sig in Screen



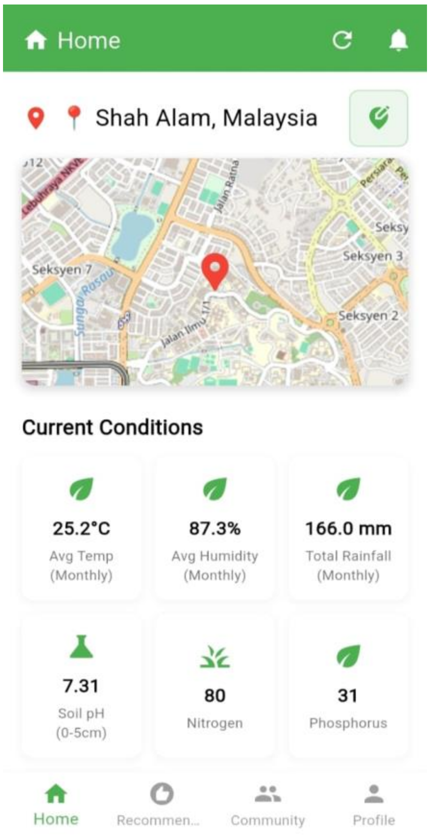

Sign Up Screen



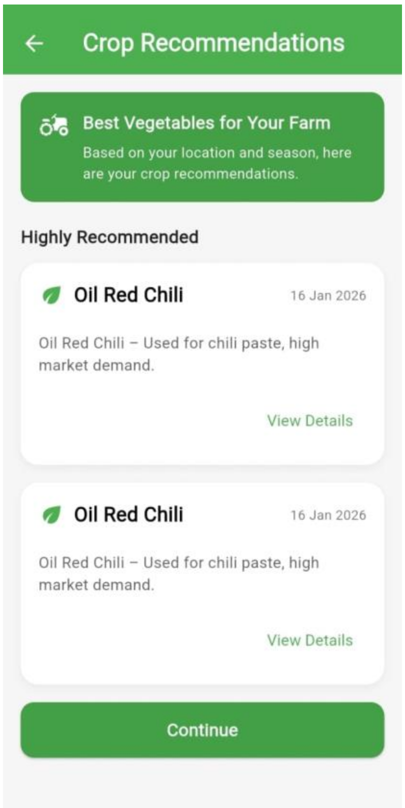

Profile Screen



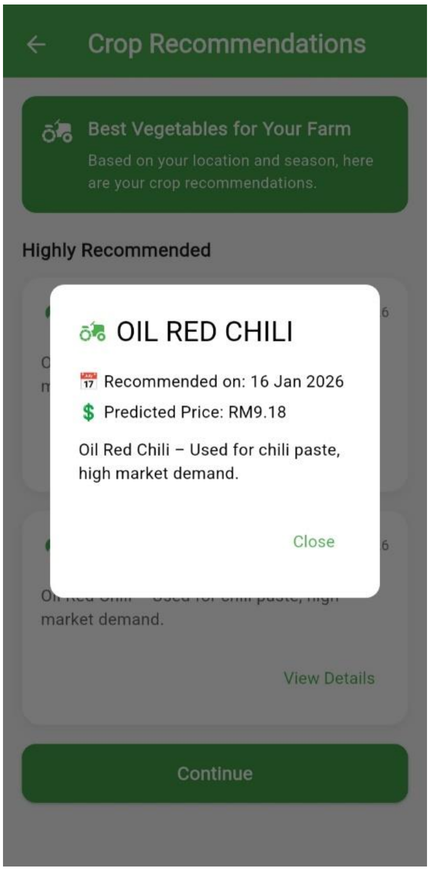

Community Screen



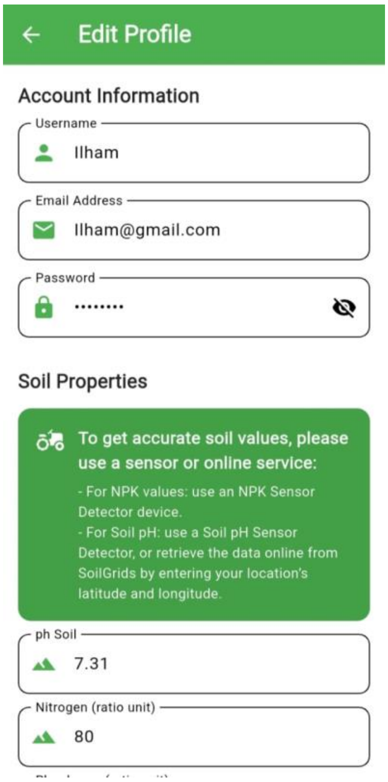

Home Screen



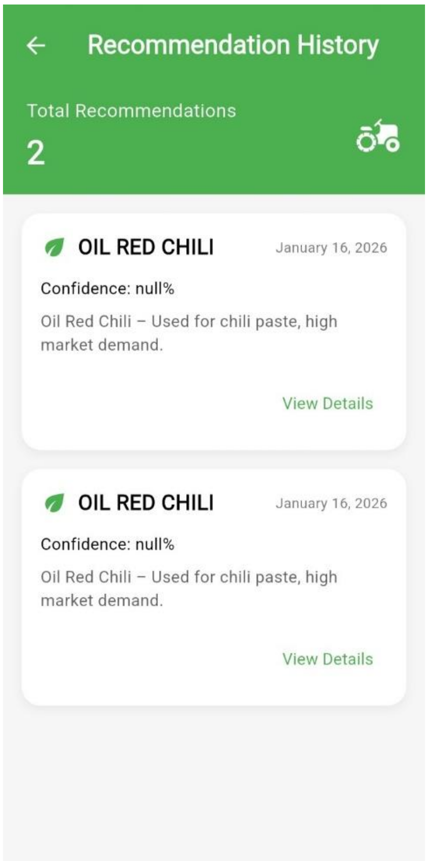

Recommendation Screen



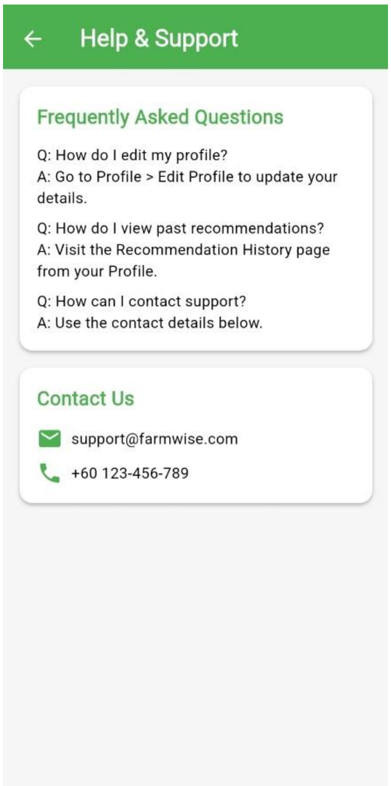

Recommendation Detail Screen




Edit Profile Screen




History Screen




Help & Support Screen

# APPENDIX C: Ruled-Based Filtering Code Block


```
{

  "Spinach": {
    "N": {
      "min": 36.0,
      "max": 113.0
    },
    "P": {
      "min": 9.0,
      "max": 57.0
    },
    "K": {
      "min": 8.0,
      "max": 41.0
    },
    "temperature": {
      "min": 20.88774309967587,
      "max": 30.666361414123244
    },
    "humidity": {
      "min": 53.57346595943703,
      "max": 87.8700291145872
    },
    "ph": {
      "min": 6.797763957910607,
      "max": 7.180904900864785
    },
    "rainfall": {
      "min": 127.32248045316362,
      "max": 246.5574239501257
    }
  },
  "Green Chili": {
    "N": {
      "min": 10.0,
      "max": 114.0
    },
    "P": {
      "min": 5.0,
      "max": 59.0
    },
    "K": {
      "min": 5.0,
      "max": 49.0
    },
    "temperature": {
      "min": 21.58194672756273,
      "max": 31.172528588194787
    },
    "humidity": {
      "min": 50.36779438269837,
```

```
      "max": 87.80232889731917
    },
    "ph": {
      "min": 5.583549338956908,
      "max": 7.287386729247952
    },
    "rainfall": {
      "min": 56.49103631890857,
      "max": 297.82295894967217
    }
  },
  "Red Chili (Kulai / Hybrid)": {
    "N": {
      "min": 11.0,
      "max": 106.0
    },
    "P": {
      "min": 11.0,
      "max": 54.0
    },
    "K": {
      "min": 10.0,
      "max": 44.0
    },
    "temperature": {
      "min": 20.015375415812464,
      "max": 31.83971270308716
    },
    "humidity": {
      "min": 50.60218695882205,
      "max": 85.89049756376033
    },
    "ph": {
      "min": 5.51702530222525,
      "max": 7.4840564260628994
    },
    "rainfall": {
      "min": 64.8080036174666,
      "max": 287.74727678950995
    }
  },
  "Oil Red Chili": {
    "N": {
      "min": 11.0,
      "max": 119.0
    },
    "P": {
      "min": 7.0,
      "max": 55.0
    },
    "K": {
      "min": 7.0,
      "max": 47.0
    },
```

```
    "temperature": {
      "min": 20.410159337998515,
      "max": 31.30706529873605
    },
    "humidity": {
      "min": 50.6718192546896,
      "max": 88.98272463324507
    },
    "ph": {
      "min": 5.529599548709987,
      "max": 7.452517914766303
    },
    "rainfall": {
      "min": 75.22145661738426,
      "max": 297.3249339300861
    }
  },
  "Birds Eye Chili": {
    "N": {
      "min": 10.0,
      "max": 104.0
    },
    "P": {
      "min": 5.0,
      "max": 49.0
    },
    "K": {
      "min": 7.0,
      "max": 48.0
    },
    "temperature": {
      "min": 20.03470786554801,
      "max": 30.96817687793317
    },
    "humidity": {
      "min": 52.055386217768216,
      "max": 85.92936906108079
    },
    "ph": {
      "min": 5.552434176408391,
      "max": 7.293351246630364
    },
    "rainfall": {
      "min": 60.03532954626812,
      "max": 291.0577411952387
    }
  },
  "Celery Leaves (Highland)": {
    "N": {
      "min": 19.0,
      "max": 99.0
    },
    "P": {
      "min": 8.0,
```

"max": 59.0
},
"K": {
"min": 6.0,
"max": 44.0
},
"temperature": {
"min": 20.634380716214025,
"max": 29.161253717499815
},
"humidity": {
"min": 50.15068157819812,
"max": 89.07829972733263
},
"ph": {
"min": 5.546341499517618,
"max": 7.420482046443284
},
"rainfall": {
"min": 50.11760909066466,
"max": 186.65902699989297
}
},
"Young Ginger": {
"N": {
"min": 11.0,
"max": 118.0
},
"P": {
"min": 12.0,
"max": 58.0
},
"K": {
"min": 5.0,
"max": 49.0
},
"temperature": {
"min": 20.40508238969082,
"max": 31.09293640880074
},
"humidity": {
"min": 51.5160640182507,
"max": 85.0759819528763
},
"ph": {
"min": 5.568852491000122,
"max": 7.468383241241727
},
"rainfall": {
"min": 97.82503392769388,
"max": 289.5112585131193
}
},
"Okra (Lady's Finger)": {

"N": {
"min": 71.0,
"max": 117.0
},
"P": {
"min": 13.0,
"max": 57.0
},
"K": {
"min": 7.0,
"max": 48.0
},
"temperature": {
"min": 21.65321258238182,
"max": 31.626561887256344
},
"humidity": {
"min": 68.45024196775907,
"max": 82.12497573628808
},
"ph": {
"min": 5.666506226407206,
"max": 7.377439378570406
},
"rainfall": {
"min": 58.07516656212545,
"max": 223.3545584753745
}
},
"French Beans": {
"N": {
"min": 13.0,
"max": 112.0
},
"P": {
"min": 5.0,
"max": 57.0
},
"K": {
"min": 11.0,
"max": 41.0
},
"temperature": {
"min": 20.24880487394718,
"max": 31.250574765404377
},
"humidity": {
"min": 54.65379645835593,
"max": 88.8726174411529
},
"ph": {
"min": 5.802989659930135,
"max": 6.810684253263768
},

```
    "rainfall": {
      "min": 58.64298754534926,
      "max": 270.9568640310869
    }
  },
  "Long Beans (Green)": {
    "N": {
      "min": 11.0,
      "max": 96.0
    },
    "P": {
      "min": 5.0,
      "max": 51.0
    },
    "K": {
      "min": 7.0,
      "max": 47.0
    },
    "temperature": {
      "min": 21.78791404049243,
      "max": 30.694215773276863
    },
    "humidity": {
      "min": 54.53297444612316,
      "max": 73.59160028591748
    },
    "ph": {
      "min": 5.507945301601311,
      "max": 7.393093305395064
    },
    "rainfall": {
      "min": 72.44759920591852,
      "max": 294.4792778029027
    }
  },
  "Chinese Broccoli (Kai-lan)": {
    "N": {
      "min": 14.0,
      "max": 119.0
    },
    "P": {
      "min": 9.0,
      "max": 58.0
    },
    "K": {
      "min": 16.0,
      "max": 47.0
    },
    "temperature": {
      "min": 20.505495916521447,
      "max": 29.808324356553467
    },
    "humidity": {
      "min": 53.26460941146578,
```

```
      "max": 88.30023981260632
    },
    "ph": {
      "min": 5.813334351796562,
      "max": 7.481020590628168
    },
    "rainfall": {
      "min": 58.60551651811073,
      "max": 297.7495947923697
    }
  },
  "Water Spinach": {
    "N": {
      "min": 21.0,
      "max": 89.0
    },
    "P": {
      "min": 44.0,
      "max": 57.0
    },
    "K": {
      "min": 18.0,
      "max": 44.0
    },
    "temperature": {
      "min": 21.69530372175281,
      "max": 31.446743968258065
    },
    "humidity": {
      "min": 56.02222804583128,
      "max": 82.19896822499825
    },
    "ph": {
      "min": 5.994901532847357,
      "max": 7.460933001445717
    },
    "rainfall": {
      "min": 122.46548197667744,
      "max": 209.01309128057736
    }
  },
  "Round Cabbage (Highland)": {
    "N": {
      "min": 10.0,
      "max": 115.0
    },
    "P": {
      "min": 20.0,
      "max": 47.0
    },
    "K": {
      "min": 12.0,
      "max": 49.0
    },
```

"temperature": {
"min": 20.40572065739013,
"max": 31.52850538766408
},
"humidity": {
"min": 59.556271470680215,
"max": 88.997014541699
},
"ph": {
"min": 5.606340519926278,
"max": 7.095459107994152
},
"rainfall": {
"min": 73.3723703726505,
"max": 278.7206955147124
}
},
"Pumpkin": {
"N": {
"min": 16.0,
"max": 117.0
},
"P": {
"min": 5.0,
"max": 51.0
},
"K": {
"min": 8.0,
"max": 49.0
},
"temperature": {
"min": 21.20478016290671,
"max": 31.211880258201692
},
"humidity": {
"min": 50.67316317470262,
"max": 86.86405364638381
},
"ph": {
"min": 5.558524085273925,
"max": 7.17257216760095
},
"rainfall": {
"min": 65.15644706573019,
"max": 263.37838458460936
}
},
"Green Bell Pepper (Capsicum)": {
"N": {
"min": 13.0,
"max": 93.0
},
"P": {
"min": 8.0,

```
        "max": 58.0
      },
      "K": {
        "min": 5.0,
        "max": 45.0
      },
      "temperature": {
        "min": 21.110544831453936,
        "max": 31.77459944135421
      },
      "humidity": {
        "min": 50.93372530562756,
        "max": 88.23274605487853
      },
      "ph": {
        "min": 5.551397656078429,
        "max": 7.368733619702783
      },
      "rainfall": {
        "min": 69.31489147885974,
        "max": 289.3309086769532
      }
    },
    "Calamansi Lime": {
      "N": {
        "min": 11.0,
        "max": 118.0
      },
      "P": {
        "min": 5.0,
        "max": 59.0
      },
      "K": {
        "min": 5.0,
        "max": 46.0
      },
      "temperature": {
        "min": 21.433536580990623,
        "max": 31.40397748444365
      },
      "humidity": {
        "min": 52.55840294052931,
        "max": 89.75738280566787
      },
      "ph": {
        "min": 5.532524312571007,
        "max": 7.249634753099558
      },
      "rainfall": {
        "min": 68.31310982670028,
        "max": 243.95945757370484
      }
    },
    "Key Lime": {
```

```
    "N": {
      "min": 14.0,
      "max": 111.0
    },
    "P": {
      "min": 7.0,
      "max": 59.0
    },
    "K": {
      "min": 6.0,
      "max": 47.0
    },
    "temperature": {
      "min": 20.618086795572477,
      "max": 30.915047063342328
    },
    "humidity": {
      "min": 53.547802300724165,
      "max": 88.76702729760905
    },
    "ph": {
      "min": 5.621262320607999,
      "max": 7.46604772239977
    },
    "rainfall": {
      "min": 53.93488277584597,
      "max": 282.9553679023353
    }
  },
  "Bitter Gourd (Bitter Melon)": {
    "N": {
      "min": 28.0,
      "max": 113.0
    },
    "P": {
      "min": 11.0,
      "max": 59.0
    },
    "K": {
      "min": 9.0,
      "max": 48.0
    },
    "temperature": {
      "min": 20.10020360071901,
      "max": 31.30829734892994
    },
    "humidity": {
      "min": 57.180251233240334,
      "max": 89.15457744519904
    },
    "ph": {
      "min": 5.605952460119078,
      "max": 7.42307640542217
    },
```

```
    "rainfall": {
      "min": 71.6188799453192,
      "max": 283.72785018021654
    }
  },
  "Head Lettuce (Highland)": {
    "N": {
      "min": 10.0,
      "max": 106.0
    },
    "P": {
      "min": 7.0,
      "max": 59.0
    },
    "K": {
      "min": 12.0,
      "max": 47.0
    },
    "temperature": {
      "min": 21.687819243117996,
      "max": 29.59800999792957
    },
    "humidity": {
      "min": 51.62323244699542,
      "max": 87.73925355021248
    },
    "ph": {
      "min": 6.478279649976185,
      "max": 7.463679069025463
    },
    "rainfall": {
      "min": 94.53003432751468,
      "max": 299.1853155238912
    }
  },
  "Green Mustard (Lowland)": {
    "N": {
      "min": 23.0,
      "max": 109.0
    },
    "P": {
      "min": 19.0,
      "max": 47.0
    },
    "K": {
      "min": 16.0,
      "max": 48.0
    },
    "temperature": {
      "min": 21.18601161098409,
      "max": 29.33419234919331
    },
    "humidity": {
      "min": 56.26409162469932,
```

```
      "max": 78.98942286120086
    },
    "ph": {
      "min": 5.654811007071865,
      "max": 7.122248474565279
    },
    "rainfall": {
      "min": 53.31934116276081,
      "max": 184.7671938022311б
    }
  },
  "Round Eggplant": {
    "N": {
      "min": 11.0,
      "max": 116.0
    },
    "P": {
      "min": 5.0,
      "max": 55.0
    },
    "K": {
      "min": 7.0,
      "max": 48.0
    },
    "temperature": {
      "min": 20.21589652120556,
      "max": 31.27881784085184
    },
    "humidity": {
      "min": 51.44562036775511,
      "max": 87.40293091935754
    },
    "ph": {
      "min": 5.696072074681567,
      "max": 7.167374942906156
    },
    "rainfall": {
      "min": 62.2827812052923,
      "max": 294.3348937407038
    }
  },
  "Long Eggplant (Lowland)": {
    "N": {
      "min": 35.0,
      "max": 114.0
    },
    "P": {
      "min": 10.0,
      "max": 54.0
    },
    "K": {
      "min": 8.0,
      "max": 44.0
    },
```

"temperature": {
"min": 20.814973206528585,
"max": 31.970948749404076
},
"humidity": {
"min": 56.80907867902791,
"max": 89.42392252017936
},
"ph": {
"min": 5.560065739666565,
"max": 7.4287946909491485
},
"rainfall": {
"min": 55.29622840629751,
"max": 274.9146398997826
}
},
"Cucumber": {
"N": {
"min": 14.0,
"max": 108.0
},
"P": {
"min": 6.0,
"max": 45.0
},
"K": {
"min": 19.0,
"max": 47.0
},
"temperature": {
"min": 20.07003138274292,
"max": 31.503439465116266
},
"humidity": {
"min": 50.72670418794343,
"max": 84.56941840723259
},
"ph": {
"min": 5.661599173094071,
"max": 7.004856697072718
},
"rainfall": {
"min": 155.60648311991082,
"max": 259.47181429053086
}
},
"Tomato (Highland)": {
"N": {
"min": 17.0,
"max": 105.0
},
"P": {
"min": 11.0,

```
            "max": 56.0
        },
        "K": {
            "min": 6.0,
            "max": 43.0
        },
        "temperature": {
            "min": 20.71508989692236,
            "max": 30.689339444323785
        },
        "humidity": {
            "min": 50.60023632884217,
            "max": 87.26893175451008
        },
        "ph": {
            "min": 5.849002936754927,
            "max": 7.428276579325765
        },
        "rainfall": {
            "min": 99.13266221215198,
            "max": 271.8024145612211
        }
    }
}
```