SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Rock Melon Crop Yield Prediction using Supervised Classification Machine Learning on Cloud Computing

Mohamad Khairul Zamidi Zakaria[1], Sazlinah Hasan[1,*], Rohaya Latip [1], Indrarini Dyah Irawati[2], A.V. Senthil Kumar[3]

[1] Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[2] School of Applied Science, Telkom University, Kabupaten Bandung, Jawa Barat 40257, Indonesia
[3] Hindusthan College of Arts & Science, Coimbatore, Tamil Nadu 641028, India

ABSTRACT

Precision agriculture is a technology-driven approach to farmer to improve their crop yields and reduce costs. One of the major challenges facing farmers today is the lack of precise prediction which leads to decreased production and mismanagement of labour and resource. Precision technology is costly, and they only rely on manual observations which are less precise. Crop yield prediction systems on cloud computing can solve both problems by predicting the harvested fruit at earlier stages of farming and ease farmers to make decisions. In this study, we proposed a crop yield prediction system for farmers that utilizes cloud computing and machine learning techniques. The system uses data on the physical growth of the plant such as plant's height at 15 and 30 days after transplant, type of pollination treatment, condition of the leaves, and their variety to predict the crop yield at the early stage. Logistic regression, k-nearest neighbour, and random forest classifier were used to compare the accuracy of the model. Our result shows that by using a random forest classifier, it can achieve an accuracy of 91% which is higher than logistic regression which is only 73% of accuracy, and k-nearest neighbour with 82% accuracy. The study highlights the potential of precision agriculture, cloud computing, and machine learning to revolutionize the way farmers manage their crops and increase their efficiency and productivity, even with the limited resources and hardware that many farmers have.

Keywords:

Agriculture; Machine learning; Cloud computing; Logistic regression; Random forest; K-nearest neighbour

1. Introduction

Agriculture is an industry that has co-existed with human societal evolution and the advancement of our species. Resource management in agriculture plays a key role in tackling important issues we face today, such as overpopulation and food supply management. Toward better food security, Malaysia already has implemented National Food Policy Action 2021-2025. The development of technology, enabling research and studies, empowering food security data, expanding strategic

* *Corresponding author.*
E-mail address: khairulzamidi2000@gmail.com

https://doi.org/10.37934/araset.54.2.200217

collaboration, and bolstering departmental and agency governance were all described as part of the plan's five fundamental initiatives [1].

One of the problems is farmers experience decreased production due to a lack of understanding of their crop's potential yield. This leads to a lack of proper planning for labour and resource needs. Without accurate predictions, farmers cannot estimate how much fruit can be harvested in a season. In some scenarios, when the produce is more than the farmers expected, it can lead to storage problems or waste due to no buyers. Another problem is farmers spend more energy on monitoring their crops to estimate the yield. Some technologies are costly, so farmers need to rely on manual observations and inspections, which can be less precise. Manually observing the crops is time-consuming and physical strain as farmers need to evaluate all their plants. Some observations also happen at a late stage of the plant where the plant is already bearing fruit. This observation needs farmers to monitor for leaf discoloration, pests like aphids, and signs of diseases on leaves and fruits. They need to ensure proper pollination during flowering, monitor fruit growth, and maintain consistent soil moisture. Checking for ripe fruits and providing support for vining varieties are also essential. Regular observation helps identify problems early, allowing farmers to take timely actions to ensure healthy plant growth and a good harvest.

The primary goal of this project is to develop a Rock Melon Crop Yield Prediction system for rock melons planted in poly bags within a greenhouse environment. The system aims to achieve two key objectives: firstly, to conduct a study on models suitable for predicting rock melon crop yield, and secondly, to develop an accurate and effective rock melon crop yield prediction system utilizing the most suitable model identified during the study.

The use of cloud computing in agriculture greatly helps farmers to have storage and computer resources for them to analyse their plants. Cloud computing offers three services which are Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS). With cloud computing, everything becomes easier to acquire such as convenience and cloud storage which is dependable. Machine learning is a technology that enhances crop productivity. It assists in predicting the quantity of fruits that can be harvested for the season by analysing various farm data, such as plant height, fertilizer usage, pesticides, and pollination. This capability greatly aids farmers in monitoring and improving their farm management.

The project scope is divided into user and system scopes. User scope describes the user of the system and sources of the dataset used. Meanwhile, the system scope describes the limitations of the crop yield prediction system. This project was in collaboration with two students from the Faculty of Agriculture. There are two different types of users which are farmers and developers. Farmers are the students who plant rock melon crops and have great knowledge of agriculture. The developer is the person who designed and developed the rock melon crop yield prediction system. The rock melon plant has been planted by the students from the Faculty of Agriculture in a polybag in a greenhouse located at the Pusat Pertanian Putra, Universiti Putra Malaysia. The plants have been given recommended fertilizer, pesticide, and water as scheduled. There are around 100 rock melon plants in the greenhouse planted in a polybag. In conclusion, this project develops a rock melon crop yield prediction that can help farmers predict the estimated rock melon produced in a season. This project also adopts cloud computing technology as it is used as a storage and machine learning resource. Lastly, this project will greatly help the farmers with precision agriculture and help to decide on their labour management and marketing decision-making.

There are six sections in the paper's structure. The first section describes the project background, problem statements, objectives, and scope of the project. The second section is about a literature review of the research that has been done on crop yield prediction and models related to the project. Section 3 is about the methodology used for this project which is explained generally from one phase

to another phase. Section 4 describes the software and hardware requirements of the project, algorithms used for machine learning, and the description of the rock melon dataset. Section 5 is about project implementation where the development process of the project is described. Results and discussions will be discussed in this section. Section 6 is about project limitations and future works. It describes the weakness of the system and provides solutions for future development.

2. Literature Review

Rock melon, also known as cantaloupe and musk melon or scientific name Cucumis melo L. belongs to the Cucurbitaceae family which comprises many other types of squashes and melons. Villanueva *et al.,* [2] claimed that rock melon has exceptional soil and temperature tolerance making it easier to cultivate in all different temperate parts of the world. It takes around 75 days for the fruit to be ready to be harvested starting from when the seed was planted. Rock melon fruit is greyish green in colour outside and striking golden yellow on the inside. This rock melon plant is planted in a poly bag inside a greenhouse and uses a drip irrigation system for fertigation. The fertilizer was mixed with water in a reservoir with a certain electrical conductivity (EC) and potential of Hydrogen (pH). Subsequently, the solution was conveyed to the plant through the piping lines according to the schedule. Figure 1 below shows the piping line and fertilizer injector of the plant.

Chen *et al.,* [3] proposed a fruit prediction system for strawberries on a deep neural network using high-resolution aerial orthoimages. Usually, the number of flowers and their distribution are estimated manually, which is time-consuming and labour-intensive. With the use of technology, a small unmanned aerial vehicle (UAV) is equipped with an RGB (red, green, blue) camera to capture images from above the farm and create orthoimages of the strawberry field. Then, the collected data was used for prediction. The identification and counting of the number of blooms, ripe strawberries, and immature strawberries were done using a state-of-the-art deep neural network model known as the region-based convolutional neural network (R-CNN), which is quicker. With an average occlusion of 13.5%, the deep learning counting accuracy on average was 84.1%. Another study in deep learning that combines background subtraction and deep learning also concludes that R-CNN is an effective method for quick deep learning in obstacle detection, as demonstrated in their work on agricultural field analysis (Christiansen *et al.,*) [4]. This technique could offer precise strawberry blossom counts that can be used to predict future yield. This method has been tested for strawberries only and different fruits may have different results. However, with the use of high-technology drones, orthoimages take much more time to train the data and use a lot of storage making it a more costly method. Also, farmers may not have the technical expertise to operate the drone and make use of the technology which could limit the adoption of this technology.

Gong *et al.,* [5] proposed a deep learning-based prediction on greenhouse crop yield by using a combined temporal convolutional network (TCN) and recurrent neural network (RNN). Another study by Lea *et al.,* [6] also stated that temporal convolutional networks (TCN) excel in accurately recognizing and segmenting fine-grained human actions within videos, making them valuable for applications like robotics, surveillance, and education. Greenhouses are currently commonly used for plant growth. In the contemporary greenhouse, environmental conditions may also be regulated to ensure the highest crop output. Accurate crop production prediction is a crucial prerequisite for controlling greenhouse environmental variables adequately. By fusing two cutting-edge neural networks, which are TCN and RNN; Gong *et al.,* [5] created a novel method for predicting crop production in greenhouses. The data that was needed included historical yields and greenhouse environmental parameters such as carbon dioxide (CO2) concentration, temperature, humidity, and other related data. A long short-term memory (LSTM) - RNN is used to pre-process an input temporal

sequence including historical yield and environmental data to extract representative features, which are then further processed by a sequential of residual blocks of the TCN module. A fully connected network is then supplied with the characteristics that were ultimately derived from the TCN to forecast future crop yields. The proposed approach has been proven for accurate greenhouse crop yield prediction, but the data used is only the environment of the greenhouses. The growth condition of the plant such as the plant's height and the condition of its leaves can be included to know if the plant has a problem or not. It is also mentioned in their future work that they also need to get more datasets collected from different growers on different sites to improve the effectiveness of the prediction.

Abd-Elrahman *et al.,* [7] claimed that close-range high-resolution images taken in the field throughout the strawberry season could be an effective tool for strawberry yield prediction at various periods. The development of prediction models and the testing of variables both used linear regression models. The least squares approach was used to estimate the models. With percentage prediction errors of 26.3% and 25.7%, real flower and fruit counts were predicted using canopy size characteristics such as canopy area, volume, height standard deviation, and visually interpreted fruit and flower counts that were retrieved from the recorded photos. The accuracy of forecasting out-of-sample yields at various time intervals was improved by 10-29% by using image-derived variables in the models as opposed to those without them. This method uses high-resolution images and drones which is costly for the farmers. A large training set of data consisting of images and plant parameters surely takes a big toll on the resources and time for machine learning to process all the data.

Hani *et al.,* [8] proposed a unique semantic segmentation-based strategy for fruit detection and undertook a detailed comparison against current state-of-the-art algorithms. It offers an end-to-end modular solution for estimating apple orchard production to find the fruit recognition and counting techniques that perform the best. Results for fruit detection show that in most data sets, the semi-supervised technique based on Gaussian Mixture Models outperforms the deep learning-based methods. However, for fruit counting, all the data sets show that the deep learning-based strategy performs better. By combining these two approaches, Hani *et al.,* [8] can estimate yields with accuracy levels between 95.56% and 97.83%. The limitation of this study is it requires a high-resolution image to recognize the number of fruits. The yield estimation occurs when the tree begins to bear fruit which is a bit late for the farmers to find their potential customers.

Cedric *et al.,* [9] proposed a crop yield prediction for decision-level for the farmers that predicts at country-level six crop yields which are bananas, yams, cassava, maize, rice, and seed cotton in West African countries. Combining agricultural, chemical, pesticide, and meteorological databases from the Food and Agriculture Organization for the United Nations and the Climate Knowledge Portal World Bank allowed for the completion of this work. To make the process easier, they combined all the various data sources into a single database using ETL techniques. On the merged dataset, they used preprocessing methods, analytics, and feature engineering approaches. Three models were used which are multivariate logistic regression with a coefficient of determination (R2) is 83.30%, k-nearest neighbour with R2 of 93.15%, and decision tree with R2 of 95.3%. The limitation of this research is this only focused on six crops in West African countries and the results may not be generalizable to other regions or crops.

3. Methodology

This section describes the approaches and techniques used to complete the work. There are several steps as in Figure 1 involving requirement, design, development and coding, integration and testing, implementation and deployment, and review.

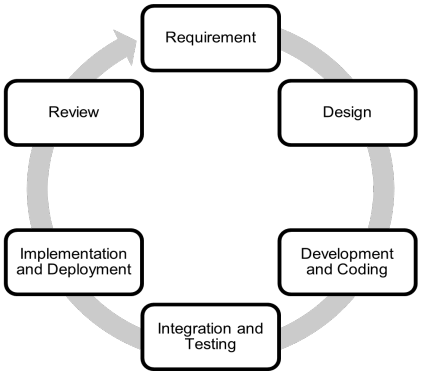

Fig. 1. Agile methodology

3.1 Requirement Phase

The first phase of the proposed framework involves the requirements phase, during which the developer discusses the problem and objectives of the project with stakeholders. The developer meets with the farmers at the rock melon farm in Pusat Pertanian Putra, Universiti Putra Malaysia. In this meeting, there was a discussion about what the farmers needed to obtain user requirements, which later became the objective of the project.

3.2 Design Phase

The design phase includes the developer's decision on the approach to solve the problem stated in the requirement phase. The machine learning algorithm added to the crop yield prediction system allowed for more accurate forecasts and better decision-making. Figure 2 shows the project area of the system which included Firebase for cloud storage and Google Collaboratory for cloud computation with scikit learn library for machine learning.

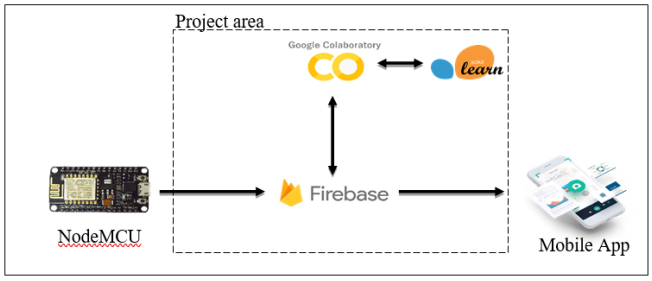

Fig. 2. Project area

3.3 Development and Coding Phase

In the development and coding phase, describe the hardware and software used to develop the system. There are two main development parts which are cloud computing for data storage and machine learning for the rock melon crop yield prediction. In this phase, it took the longest time as we translated design documentation into an actual functioning system.

3.4 Integration and Testing Phase

In the Integration and Testing phase, the product will go for testing to detect any errors in the coding or development. As we are using machine learning, we need to get the best model that can accurately predict the crop yield of the rock melon.

3.5 Implementation and Deployment Phase

During the Implementation and Deployment phase, the system can be deployed and made available to customers. Ongoing support was provided by the developer to ensure that the system functioned as planned. The prediction system was deployed for the rock melon farm in Pusat Pertanian Putra, and the results were based on the accuracy of the predictions. As this was an iterative process, updates to features or enhancements of the system were possible.

3.6 Review Phase

The review phase requires the developer to review the result with the client to make sure the requirements have been met. Take feedback from the client for improvement. The agile software development phases then restart, either with a fresh iteration or by progressing to the next step and scaling agile.

4. System Design

Cloud computing opens opportunities for everyone, including farmers, to leverage technology to enhance their crops. Krishna *et al.*, [10] stated that cloud computing, with its capacity to offer a diverse array of services over the Internet, has garnered widespread interest and success, turning utility computing into a practical reality. The system of rock melon crop yield prediction was built fully based on cloud computing by utilizing Google Firebase for cloud storage and Google Collaboratory for cloud computation.

4.1 Hardware

The hardware required for this project is only the laptop for development and coding, as machine learning was done in cloud computing; and this project required the cloud storage to act as the connecter between the Internet of Things sensors and the mobile application.

4.2 Software

In this project, there are two cloud-based software used in this project which are Google Firebase and Google Collaboratory. Both software can be accessed through the internet and do not require any installation.

Firebase can be used for many kinds of applications such as Android, iOS, JavaScript, Node.js, Java, Unity, etc. Firebase provides hosting services. It provides real-time hosting of databases, content, social authentication, and alerts, as well as other services like a real-time communication server, in addition to NoSQL hosting. Firebase offers Cloud Firestore that is suitable to be used as a cloud storage that can be connected to the sensor from the farm and the mobile application. Unlike a conventional SQL database, which uses SQL to store data in a table's columns and rows. Each piece

of information kept in the documents and collections formats is kept in the NoSQL Cloud Firestore. Each document has a set of key-value pairs that may be used to access data. However, the free version of Cloud Firestore only gives 1 Gigabytes of stored data limitation. This is already enough storage for this project.

Google Colab is a Jupyter Notebook environment that is entirely cloud-based. It manages all the program's setup and configuration needs. Collaboratory, known as "Colab," is a Google Research product. Colab is particularly well suited to machine learning, data analysis, and teaching. It enables anybody to create and execute arbitrary Python code through the browser. Google Colab used Python language for coding. The free version of Google Colab provides resources of Python 3 Google Compute Engine backend with 13GB RAM which is sufficient for this project. Scikit-learn is a machine-learning library in Python. It provides a variety of efficient tools for machine learning and statistical modelling including classification, regression, clustering, and dimensionality reduction. The user needs to import this library at the start of the code. It can be used with the Google Colab for data analysis and prediction.

4.3 Conceptual Design

This project is divided into two major parts which are cloud storage and machine learning for crop yield prediction.

4.3.1 Cloud storage

Looking at the first part, all the data collected from the Internet of Things sensor is stored in the real-time database. It provides synchronization with the NoSQL cloud database and all the clients connected in real-time, and it remains available when the app goes offline. This data can be accessed from the mobile app. Data and information from the mobile app such as user accounts, fertilizing schedules, and plants also was stored in the NoSQL cloud Firestore. Figures 3 and 4 show the Entity Relationship Diagram of the database.

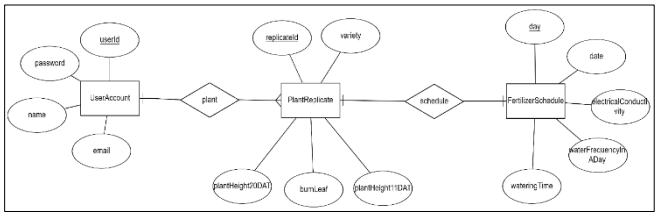

Fig. 3. Entity relationship diagram

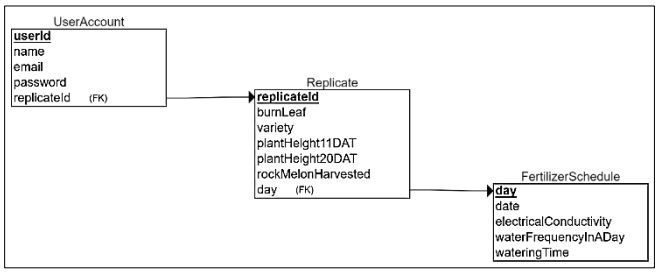

Fig. 4. Relational schema

4.3.2 Machine learning

For machine learning, the rock melon crop yield dataset is collected from students of the Faculty of Agriculture, Universiti Putra Malaysia who planted the rock melon in a polybag in the glass house. Wild pollination can enhance sustainability and improve crop yield, as demonstrated by Fijen *et al*., [11] underscore the broader potential of incorporating wild pollinators in larger agricultural settings, emphasizing the significant economic contributions of these natural pollination methods. However, this study primarily utilized human-assisted pollination for more control over the pollination process. This rock melon was planted in late August and early October. There are 7 features in this dataset which are:

i. Treatment. These include how the pollination is done. There are three different types which are T1 for self-pollination, T2 for cross-pollination, and T3 is the combination of both self-pollination and cross-pollination.
ii. Variety. There are two varieties of rock melon used which are Glamour and Chanchai.
iii. Plant height 15 days after transplant. Height is a good growth parameter for the plant as rock melon was grown vertically to save space. Ropes are being tied vertically to provide support for rock melon's vine sprawling. The unit used is in centimetres (cm).
iv. Plant height 30 days after transplant. The unit used is in centimetres (cm).
v. Growth rate in 15 days. Plant height 30 days after transplant minus plant height 15 days after transplant. The unit used is in centimetres (cm).
vi. Percentage of burned leaf. The burn leaf is the yellowish area of the leaves. More burned leaves make less green area which is not good for the plant. The percentage of burned leaf is classified into 3 classes 25%, 50%, and 75% burn leaf. The measurement was done through rough observation.
vii. Number of rock melons harvested. The number of rock melons harvested includes all the rock melon fruit that can be harvested although there are different sizes and qualities of the fruits. It is targeted that every one of the plants produce 1 good quality rock melon fruit.

Figure 5 shows a sample data from the dataset.

	Treatment	Variety	PlantHeight15DAT(cm)	PlantHeight30DAT(cm)	GrowthRateIn15Days	BurnLeaf	RockMelonHarvested
0	T1	Chanchai	17.0	78.0	61.0	75%	1
1	T1	Glamour	29.0	87.0	58.0	75%	1
2	T1	Glamour	20.0	49.0	29.0	50%	1
3	T1	Chanchai	20.0	34.0	14.0	75%	0
4	T1	Glamour	22.0	109.0	87.0	75%	1
5	T1	Chanchai	23.0	40.0	17.0	75%	0
6	T1	Glamour	20.0	96.0	76.0	75%	1
7	T1	Glamour	22.0	87.0	65.0	75%	1
8	T1	Chanchai	27.0	95.0	68.0	75%	1
9	T1	Chanchai	29.0	80.0	51.0	75%	1

Fig. 5. First 10 rows of the dataset

Machine learning is a type of artificial intelligence that allows systems to automatically improve their performance by learning and experiencing the data fed to it. Goodfellow *et al.,* [12] contend that machine learning solutions empower computers to acquire knowledge from experience, enabling them to comprehend the world through a hierarchy of concepts, each defined as simpler concepts. The effectiveness of these basic machine learning algorithms is significantly influenced by

how the data is represented. There are different types of machine learning, including supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

We focused on supervised learning where in this type of machine learning, the model is trained using labelled data, to predict new, unseen data. The model uses labelled training data to learn the relationship between the input and output variables. Under supervised machine learning, there are two different types of supervised learning algorithms:

i. Regression algorithms, which predict a continuous value, such as the price of a stock, and house value.
ii. Classification algorithms, which predict a discrete value, such as predicting whether the weather is rain or not, and image classification.

In this project, we used classification-supervised machine learning where we used different types of algorithms and compared the accuracy of the model. There are several popular classification algorithms such as logistic regression, decision tree, random forest, k-nearest neighbour and naïve Bayes. In this project, we only experiment with logistic regression, k-nearest neighbour and random forest classifier.

In supervised machine learning, the algorithm is trained with both input features and output labels. We used the classification model because the target feature of our dataset, 'RockMelonHarvested' is a discrete variable where it only takes values 0, 1, 2, and 3. Three different algorithm models used to compare the accuracy of the prediction which are:

i. Logistic regression: Used for classification problems. It is a type of generalized model that uses a logistic function also known as the sigmoid function to model the probability of an event occurring [13]. It is a statistical technique for looking at a dataset where one or more independent factors affect how things turn out. The output of the logistic regression is a probability value between 0 and 1, which can then be used to predict the outcome. In this project, multinomial logistic regression is used when there are more than two outcomes or classes exist. Zou *et al.,* [14] explained that to identify the actual class label, the logistic regression model finds the boundary line of the classification by calculating predicted probabilities for a linear combination of independent factors. Refer to Figure 6, the pseudocode explains how the logistic regression model works from the first iteration until the final prediction. Step 3 which is the most significant step is to calculate the linear combination of features and coefficient represented by the symbol 'z' in the logistic regression model. In the formula, 'y' is the target variable, 'P' is the predicted probabilities, and 'd' represents the number of features. Step 4 initializes the weight of each instance to indicate the importance of each feature to produce a prediction. Both coefficient and weight are finalized for the prediction function. The predicted probabilities were obtained from the sigmoid function, and the final class prediction was made by comparing these probabilities to a threshold in step 6.

Input: Training data

1. For i←1 to k
2. For each training data instance d_i:
3. Set the target value for the regression to
   $z_i \leftarrow \frac{y_j - P(1 \mid d_j)}{[P(1 \mid d_j) . (1 - P(1 \mid d_j))]}$
4. nitialize the weight of instance d_j to $P(1 \mid d_j). (1 - P(1 \mid d_j)$
5. finalize a f(j) to the data with class value (z_j) & weights (wj)
   Classification Label Decision
6. Assign (class label:1) if $P(1|d_j) > 0.5$, otherwise (class label: 2)

Fig. 6. Logistic regression pseudocode

ii. K-nearest neighbours (KNN): This algorithm works by finding the k training examples that are physically closest to the input data and predicting the outcome based on their average value or majority class is the algorithm's main premise. Every crop data point that is close to another in proximity is assumed by KNN to belong to the same category. Since it is non-parametric, no assumptions are made on the underlying data. It is well known that KNN is a straightforward technique to use and is resistant to noisy training data. Referring to Figure 7, the algorithm started by setting the value of 'k,' representing the number of nearest neighbours used for prediction. Next, it calculated the distance between the test sample and each training sample, where 'N' represented the total samples of the dataset. The most common distance metric used in KNN was the Euclidean distance. From the input data, it found the 'k' closest nearest neighbours based on the calculated distance. Finally, it predicted the target value by choosing the majority class among the 'k' nearest neighbours. Figure 8 provides the visualization of the pseudocode.

Input: Data

1. Set the value of k
2. Loop: 1 to N // To get predicted class
   2.1. Calculate the distance Di (Euclidian/Cosine/Chebyshev) between data instance in training data and test data.
3. Increasingly arrange the computed distances (Di)
4. Populate the upper k results from the arranged list
5. Pick up the most frequent class from the list

Output: resultant class

Fig. 7. KNN pseudocode

Cunningham and Jane [15] indicated that the utilization of approximate nearest-neighbour techniques has the potential to significantly enhance retrieval times, often without a substantial compromise on accuracy.

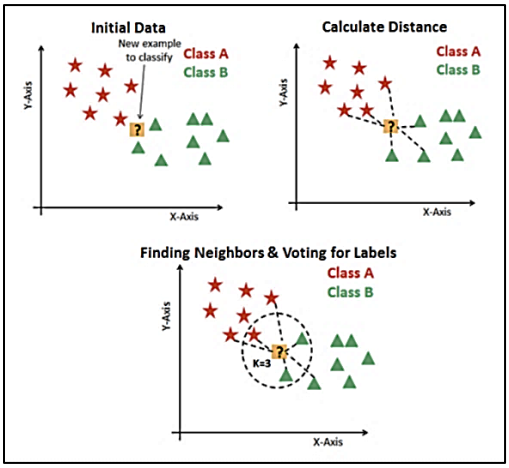

Fig. 8. Visualization of KNN algorithm

iii. Random forest classifier: Jackins *et al.,* [16] explained that a random forest is a collection of tree-based classifiers, effective for datasets with many dimensions and numerous irrelevant attributes. The algorithm uses a simple probability approach to select the most important attributes for classification. It is an ensemble machine-learning algorithm that is composed of multiple decision trees. It is called a random forest because it creates random decision trees during the training process. Each tree in the ensemble is built from a random sample of the data and a random subset of the features. This process increases the diversity of the ensemble and helps reduce the correlation between the trees. During the prediction phase, each tree in the forest makes a prediction, and the final prediction is made by taking the mode of the predictions. Figure 9 describes the pseudocode of a random forest classifier to create multiple decision trees from random features. Symbol 'p' represents the number of input features, 'F' represents the total number of features, 'n' represents the number of decision trees, and 'd' represents the depth of the tree. Steps 1 until 4 will loop until it reaches the number of decision trees wanted. From that multiple decision tree, the majority class voted will be the final prediction class for the input.

Input: Training set S with F features

1. Randomly pick 'p' features in 'F' features, ∀ p <F
2. Using 'p' features, find the node 'd' by the best split
3. Break the node into child nodes by applying the best split method
4. Iterate steps: 1 to 3 until 'l' number of nodes has been reached
5. Repeat steps: 1 to 4 and build the forest by generating 'n' number of decision trees

Output: Random Forest Trees (RFTs)

Fig. 9. Random forest classifier pseudocode

Using Google Colab, the dataset will be going through a pre-processing process before it can be trained for machine learning. Below is the flowchart on how the process from the start till getting the result of the rock melon crop yield prediction.

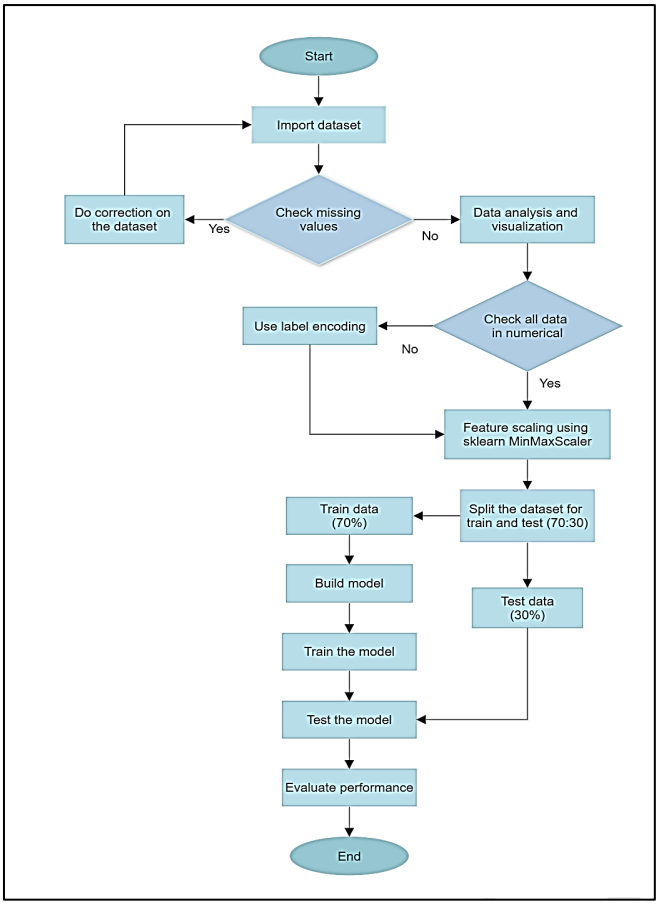

Fig. 10. Machine learning modelling flowchart

5. Implementation

In this section, the code development involved is about the preprocessing of the data, analysis, modelling, and evaluating performance for each of the algorithms.

5.1 Machine Learning

Pandas' libraries are imported to read the dataset. Google Drive is connected to the Google Colab making it easier to read the dataset directly from the Google Colab. Before the dataset can be modelled and used for machine learning, the dataset needs to be complete without any missing values. Figure 11 shows there are 108 rows and 7 columns. There are no missing values, and it also describes its datatypes for each feature.

```
# check for the data types, memory usage, etc
display(df.info())
```

```
(108, 7)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 108 entries, 0 to 107
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Treatment             108 non-null    object
 1   Variety               108 non-null    object
 2   PlantHeight11DAT(cm)  108 non-null    float64
 3   PlantHeight20DAT(cm)  108 non-null    float64
 4   GrowthRateIn10Days    108 non-null    float64
 5   BurnLeaf              108 non-null    object
 6   RockMelonHarvested    108 non-null    int64
```

Fig. 11. Dataset information

Data visualizations are used to display statistics of the dataset in a picture or graph. Libraries such as Matplotlib and Seaborn are used for data visualization. In the dataset, there are also object data types. Machine learning cannot read this object data type and it needs to be changed into an integer data type. 4 features need to be labelled and encoded to numeric values which are 'Treatment', 'Variety', 'BurnLeaf', and 'RockMelonHarvested'.

Feature scaling is a method used to standardize the range of independent variables or features of a data set. In machine learning, it is important to scale the features so that they have comparable ranges because algorithms often operate under the assumption that the input features have similar scales. In this project, MinMaxScaler from the sklearn library was used. Min-Max normalization is a method of scaling features in which the minimum value of a feature is transformed to 0 and the maximum value is transformed to 1, while all other values are normalized between 0 and 1. This method is used to standardize the range of independent variables or features of a data set in machine learning to improve the performance of the model.

The dataset was split into 70% for training and 30% of the dataset for testing purposes. The ratio of 70:30 has the best performance for most machine learning models [17]. The algorithm model is taken from the scikit-learn libraries. Some enhancements made are by modifying the hyperparameter of the algorithms.

A logistic regression classifier is a simple classification algorithm to predict the outcome occurring, based on one or more predictor variables. The classifier works by estimating the coefficients for the predictor variables in the logistic function, which is used to model the probability of the outcome occurring.

Next, the KNN algorithm has also been tested to get the accuracy. Okamura *et al.,* [18] data scientist who graduated from Flatiron School suggested hyperparameter tuning to increase the accuracy of a model. A small k value, such as k=1, can lead to overfitting if the nearest neighbour is an outlier or an extreme value. If the k value is set too high, such as k=30 the model may underfit the data which means it may not capture the true underlying pattern in the data. We used hyperparameter tuning by using GridSearchCV where it will test a range value of k from 1 to 30 and find the best fit for the dataset. As shown below, the best k value is 3.

Next, we model the KNN classifier algorithm with a k-nearest neighbour value is 3 that we get from the GridSearchCV result. Fitting the classifier to training data and predicting the training data. The process of splitting the dataset, building a model, fitting the model, making predictions, and accuracy performance are the same as the logistic regression algorithm. The only difference is some of the algorithms have hyperparameter tuning when building the model. The accuracy score will be evaluated.

The Random Forest Classifier is configured to consist of 10 decision trees, determined by setting the parameter n_estimators to 10, and the criterion = 'entropy' as it is for the classification problem. According to [19] Meinert (2019), a data scientist from the United States that random forest is already good for classification by using the default value 'n_estimators' of 10. In each of the decision trees, a random number of samples will be used to create the decision tree. Based on the result from the 10 different decision trees, the majority class will be predicted as the output.

The code for visualizing data and making predictions for rock melons can be obtained from the GitHub repository, https://github.com/khairul006/RockMelonCropYieldPredictionML.

5.2 Results and Discussion

There are three models trained and tested with the rock melon yield dataset. The performance will be compared through the accuracy of the classification report for each model. For each model,

the process of train and testing is done about 10 times as suggested by Aporia [20] and this result is among the best performance by each model.

Table 1
Logistic regression performance

Model: Logistic Regression				
Accuracy: 0.7272727272727273				
	precision	recall	f1-score	support
0	0.00	0.00	0.00	0
1	0.78	0.88	0.82	16
2	0.91	0.59	0.71	17
3	0.00	0.00	0.00	0
accuracy			0.73	33
macro avg	0.42	0.37	0.38	33
weighted avg	0.85	0.73	0.77	33

Table 2
K-Nearest Neighbour performance

Model: K-Nearest Neighbour				
Accuracy: 0.79				
	precision	recall	f1-score	support
1	0.75	0.86	0.80	14
2	0.88	0.74	0.80	19
3	0.00	0.00	0.00	0
accuracy			0.79	33
macro avg	0.54	0.53	0.53	33
weighted avg	0.82	0.79	0.80	33

Table 3
Random forest performance

Random Forest Classifier				
Accuracy: 0.9090909090909091				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
1	0.85	1.00	0.92	17
2	1.00	0.77	0.87	13
3	1.00	1.00	1.00	2
accuracy			0.91	33
macro avg	0.96	0.94	0.95	33
weighted avg	0.92	0.91	0.91	33

These models were assessed on their ability to classify instances within the dataset. The KNN model exhibited an accuracy of 0.79, indicating that it correctly predicted nearly 79% of instances in the dataset. However, upon closer examination of precision and recall values, a nuanced performance emerged. While achieving an admirable precision of 0.82, suggesting a high proportion of accurately predicted positive instances, the model's recall values showed variability across classes.

Notably, class 3 lacked any true positive predictions, which significantly impacted the overall performance of recall and F1-score, highlighting potential limitations in correctly identifying instances from this class.

In contrast, the Logistic Regression model demonstrated a slightly lower accuracy of 0.73. Despite achieving relatively high precision for classes 1 and 2 (0.78 and 0.91, respectively), the model displayed lower recall values for these classes. Notably, class 0 and class 3 lacked any predicted instances, resulting in null precision, recall, and F1-score for these classes. This model's performance suggests a certain level of imbalance and difficulty in correctly classifying instances across multiple classes.

Conversely, the Random Forest Classifier model yielded the highest accuracy of 0.91, signifying a robust predictive capability in this context. With precision values ranging from 0.85 to 1.00 across different classes, the model demonstrated a consistent ability to accurately classify instances, particularly in classes 0, 1, and 3. The recall values showed a relatively high sensitivity in identifying positive instances across most classes, contributing to a balanced F1-score across all classes.

Based on the Random Forest Classifier has the best performance with 0.91 accuracy among the three models. As a result, a random forest classifier will be used in predicting the crop yield of the rock melon.

For example, in Figure 12, a decision tree was extracted from the model. It randomly chose 49 samples from the dataset with entropy of 1.6. Entropy is used to measure the impurity of a set of data. A set of data is pure if all the data points in the set belong to the same class but considered impure if it belongs to multiple classes. The higher value of entropy is considered impure, and the lower value of entropy is considered pure. The arrow on the left side means the conditions in the box are true, while the arrow on the right considers otherwise. The value = [3, 34, 29, 9] means there are 4 different classes referred to 'RockMelonHarvested' class which are plants with 0, 1, 2, and 3 numbers of rock melon harvested. In this example, there are 3 samples from class 0, 34 from class 1, 29 from class 2, and 9 from class 3. Going down to the box, based on the condition it will arrive at its predicted output with an entropy is 0.0, meaning the sample tested belongs to its predicted class. The same process will happen at the other 9 decision trees and the output of the decision tree will be observed. The majority output from the decision tree will be considered as the predicted class for the sample.

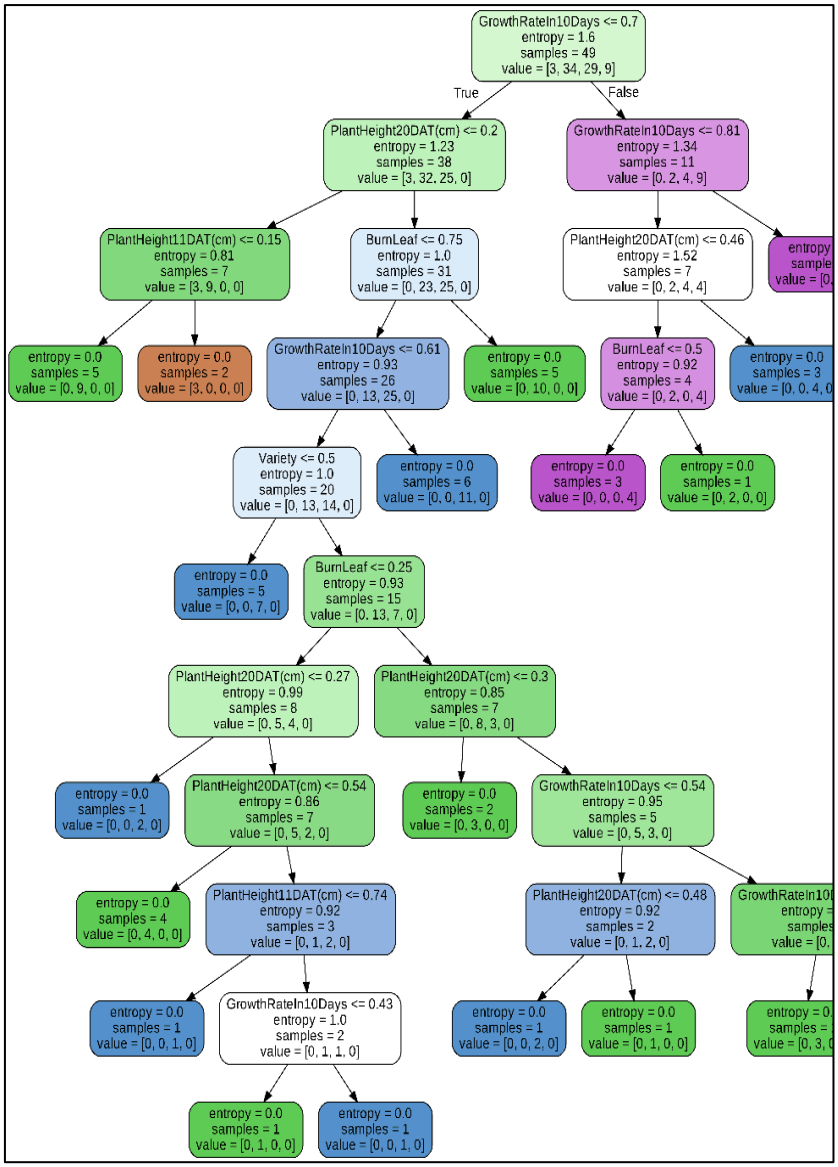

Fig. 12. Random forest classifier

6. Conclusions

Agriculture industries are going toward precision agriculture where the Internet of Things, sensors, and technology have been applied to the farm system. Rock melon crop yield prediction system is one the systems that can be managed to help farmers determine their crop yield in a season. The farmers usually predict their crop yield manually based on their experience, but it is not accurate. Problems like mismanagement such as some of the harvested fruit becoming rotten because cannot be sold off to the customer or an undersupply of fruit harvested to the clients.

Rock melon crop yield prediction system offers a logic-based prediction using machine learning. This system will get their average height on certain days, leaf conditions, and some other features to determine what their average fruit can be harvested by the plant. In this project, we are comparing the logistic regression model, k-nearest neighbour, and random forest classifier where the result is random forest classifier performs better in predicting the crop yield of the rock melon harvested per plant. In contrast to existing machine learning models highlighted in the literature review that predominantly centre on image processing, my project relies on numerical data, resulting in lower

data processing requirements. These image-focused models often incur higher processing costs and longer processing times due to the intricacies of handling visual information. By utilizing numerical data for prediction in rock melon crop yield, my project streamlines the processing load, offering a more efficient and cost-effective approach compared to the resource-intensive image-based models found in the current literature.

The rock melon crop yield prediction system achieves a 91% accuracy rate through the implementation of a random forest classifier. Despite this high accuracy, the dataset used for training and testing is limited to 109 plants within a single season. Expanding the dataset to encompass a broader range of conditions, including different seasons, would likely enhance accuracy, considering variables like dry season planting affecting fruit production. Currently, the system assumes uniform fertilization and pesticide schedules across all plants from the same farm. However, future iterations should incorporate diverse fertilizers and pesticides, recognizing their varying impacts on plant growth and yield. Including this information as features in the predictive model could significantly improve accuracy and effectiveness in predicting crop yield.

With the accurate prediction of crop yield, farmers can have better management of finding possible buyers and customers for their rock melon before it becomes rotten and cannot sell anymore. In conclusion, rock melon crop yield prediction using a Random Forest Classifier successfully predicts the class of rock melon harvested per plant with an accuracy of 91%.

Acknowledgment
This research was not funded by any grant.

References

[1] Ministry of Agriculture and Food Industries. “Pelan Tindakan Dasar Sekuriti Makanan Negara 2021-2025.” (2021). www.mafi.gov.my

[2] Villanueva, M. J., M. D. Tenorio, M. A. Esteban, and M. C. Mendoza. "Compositional changes during ripening of two cultivars of muskmelon fruits." *Food chemistry* 87, no. 2 (2004): 179-185. https://doi.org/10.1016/j.foodchem.2003.11.009

[3] Chen, Yang, Won Suk Lee, Hao Gan, Natalia Peres, Clyde Fraisse, Yanchao Zhang, and Yong He. "Strawberry yield prediction based on a deep neural network using high-resolution aerial orthoimages." *Remote Sensing* 11, no. 13 (2019): 1584. https://doi.org/10.3390/rs11131584

[4] Christiansen, Peter, Lars N. Nielsen, Kim A. Steen, Rasmus N. Jørgensen, and Henrik Karstoft. "DeepAnomaly: Combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field." *Sensors* 16, no. 11 (2016): 1904. https://doi.org/10.3390/s16111904

[5] Gong, Liyun, Miao Yu, Shouyong Jiang, Vassilis Cutsuridis, and Simon Pearson. "Deep learning based prediction on greenhouse crop yield combined TCN and RNN." *Sensors* 21, no. 13 (2021): 4537. https://doi.org/10.3390/s21134537

[6] Lea, Colin, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. "Temporal convolutional networks for action segmentation and detection." In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 156-165. 2017. https://doi.org/10.1109/CVPR.2017.113

[7] Abd-Elrahman, Amr, Feng Wu, Shinsuke Agehara, and Katie Britt. "Improving strawberry yield prediction by integrating ground-based canopy images in modeling approaches." *ISPRS International Journal of Geo-Information* 10, no. 4 (2021): 239. https://doi.org/10.3390/ijgi10040239

[8] Häni, Nicolai, Pravakar Roy, and Volkan Isler. "A comparative study of fruit detection and counting methods for yield mapping in apple orchards." *Journal of Field Robotics* 37, no. 2 (2020): 263-282. https://doi.org/10.1002/rob.21902

[9] Cedric, Lontsi Saadio, Wilfried Yves Hamilton Adoni, Rubby Aworka, Jérémie Thouakesseh Zoueu, Franck Kalala Mutombo, Moez Krichen, and Charles Lebon Mberi Kimpolo. "Crops yield prediction based on machine learning models: Case of West African countries." *Smart Agricultural Technology* 2 (2022): 100049. https://doi.org/10.1016/j.atech.2022.100049

[10] Reddy, V. Krishna, B. Thirumala Rao, L. S. S. Reddy, and P. Sai Kiran. "Research issues in cloud computing." *Global Journal of Computer Science and Technology* 11, no. 11 (2011): 59-64.

[11] Fijen, Thijs PM, Jeroen A. Scheper, Timo M. Boom, Nicole Janssen, Ivo Raemakers, and David Kleijn. "Insect pollination is at least as important for marketable crop yield as plant quality in a seed crop." *Ecology letters* 21, no. 11 (2018): 1704-1713. https://doi.org/10.1111/ele.13150
[12] Heaton, Jeff. "Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618." *Genetic programming and evolvable machines* 19, no. 1 (2018): 305-307. https://doi.org/10.1007/s10710-017-9314-z
[13] Robles-Velasco, Alicia, Pablo Cortés, Jesús Muñuzuri, and Luis Onieva. "Prediction of pipe failures in water supply networks using logistic regression and support vector classification." *Reliability Engineering & System Safety* 196 (2020): 106754. https://doi.org/10.1016/j.ress.2019.106754
[14] Zou, Xiaonan, Yong Hu, Zhewen Tian, and Kaiyuan Shen. "Logistic regression model optimization and case analysis." In *2019 IEEE 7th international conference on computer science and network technology (ICCSNT)*, pp. 135-139. IEEE, 2019. https://doi.org/10.1109/ICCSNT47585.2019.8962457
[15] Cunningham, Padraig, and Sarah Jane Delany. "k-Nearest neighbour classifiers: (with Python examples)." *arXiv preprint arXiv:2004.04523* (2020).
[16] Jackins, V., S. Vimal, Madasamy Kaliappan, and Mi Young Lee. "AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes." *The Journal of Supercomputing* 77, no. 5 (2021): 5198-5219. https://doi.org/10.1007/s11227-020-03481-x
[17] Nguyen, Quang Hung, Hai-Bang Ly, Lanh Si Ho, Nadhir Al-Ansari, Hiep Van Le, Van Quan Tran, Indra Prakash, and Binh Thai Pham. "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil." *Mathematical Problems in Engineering* 2021 (2021): 1-15. https://doi.org/10.1155/2021/4832864
[18] Okamura, Scott. "GridSearchCV for Beginners." *Towards Data Science* (2020).
[19] Meinert, Reilly. "Optimizing hyperparameters in random forest classification." *Towards Data Science* (2019).
[20] Aporia. "Machine Learning in Real Life: Models, Use Cases & Operations." (2023). https://www.aporia.com/learn/machine-learning-model/machine-learning-models-use-cases-operations/