COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

BACHELOR OF COMPUTER SCIENCE (Hons.)

SPECIAL TOPICS IN COMPUTER SCIENCE

CSC649

GROUP PROJECT REPORT:

FarmAi: Integrating Machine Learning for Sustainable Farming Solutions

Prepared By:

NO.	NAME	STUDENT ID
1	HAZRUL MUHAMMAD IDHAM BIN ABU BAKAR	2021816444
2	ABDUL HADI BIN ABDUL HALIM	2021899622
3	MUHAMMAD ALIF SAFWAN BIN SYAMSUL SYAHAR	2021454272
4	MUHAMMAD HAIKAL HAIKIMI BIN HAIRUL ANUAR	2021476784

Prepared For:

DR MOHD FAAIZIE BIN DARMAWAN

Semester:

MAC – AUG 2023

TABLE OF CONTENTS

CONTENTS **PAGE**

LIST OF FIGURES

FIGURES **PAGE**

LIST OF TABLES

TABLES **PAGE**

1. INTRODUCTION

FarmAi aims to revolutionize traditional farming practices by integrating machine learning solutions to address critical challenges in agriculture. The current scenario depicts a gap in farmers' knowledge concerning optimal crop selection, suitable fertilizers, pesticides, and an accurate assessment of soil conditions. While traditional practices are deeply rooted in agricultural communities, they often fall short in addressing modern challenges, exacerbated by environmental factors such as climate change. These challenges, including the impact of increased atmospheric CO2, temperature fluctuations, and global warming, contribute to significant stress on crops, potentially reducing yields by up to 70%.

The consequences of this knowledge gap and reliance on outdated methods manifest in reduced crop yields, leading to diminished income and economic risks for countries heavily dependent on agriculture. Notably, disruptions in crop production, as seen in India's rice export restrictions, can have far-reaching consequences on global food security and prices. To mitigate these challenges, the FarmAi project proposes the development of a machine learning model, FarmAI, tailored to the agricultural sector. FarmAI aims to empower farmers by leveraging machine learning algorithms to predict optimal crops, suggest suitable fertilizers, and assess soil fertility and water quality accurately.

2. PROBLEM STATEMENT

Ideally, farmers should have plenty of knowledge of various aspects of agriculture. The farmers should have knowledge on choosing the suitable crop based on parameters like nitrogen, potassium, phosphorus, and pH level of the soil (Kansal et al., 2023). Besides that, farmers also should know on the condition of their soil whether it fertile enough to get high crop yield. The farmers also must know about choosing the suitable fertilizer for the crop and the suitable pesticides.

However, not all farmers have knowledge on choosing the optimal plants, fertilizers, pesticides according to the soil condition of their land. Moreover, farmers rely on traditional approaches that were passed down from many generations. On top of that, some of the traditional approaches may be irrelevant in today's era. This is because environmental factors such as global increase of atmospheric CO2, fluctuations in sea level, temperature changes, rainfall changes and global warming causes increasing climate stresses to the crops that would reduce crop yields by up to 70% (Liliane et al., 2020). Thus, practising traditional approaches may not be the most efficient way to increase crop yield.

The consequences of those problems are reduced or less crop yield which leads to less income and profit. Besides, crops production is very important, and it is a very large contributor to a country's economy (Gill et al., 2022). If crop production is not maximized, it will also put the country's economy at risk. As an example, in July 2020, India's government announced that the country would stop exporting non-basmati white rice to lower rice prices and ensure food security in the country. This causes a 'panic' in other countries such as the Philippines, Malaysia, Vietnam, Ivory Coast, and Senegal which rely on India's white rice exports. The ban also causes price hike on white rice in the effected countries (Wiener Bronner, 2023). The issue mentioned just now explains how much important of crop production in a country's economy,

Based on the problems and various challenges stated, the study proposed to develop a model using machine learning algorithm called FarmAI which beneficent in agriculture sector. This is because machine learning can help farmers by suggesting and predicting various aspects such as suitable fertilizers, suitable pesticides and suitable crop based on various elements such as the soil's nutrient content, pH level since its crucial to the fertility level of the soil. As a result, increase the crop production, saves cost, and saves time by integrating machine learning techniques to aid farmers.

3. OBJECTIVES

This project will primarily focus on the following objectives to guarantee the accuracy of the predicted object:

- To identify the techniques or algorithms that will be used to determine the optimal crops, suitable fertilizer, water quality and soil fertility in classifications and predictions machine learning model.

- To develop a precise agriculture decision support system that implements machine learning algorithms for crop yield prediction, fertilizer optimization, soil fertility prediction, and water quality classifications.
- To evaluate test the accuracy of developed model in predicting crop yield, fertilizer recommendations, soil moisture and optimal quality of water.

4. DATA COLLECTION

All the datasets that utilised in this study was obtained from the Kaggle.com website. Kaggle allows users to find datasets they want to use in building AI models, publish datasets, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges. Therefore, this study mainly used Kaggle as the source of getting datasets as it is the initial step in building machine learning model. There are 4 datasets used for building FarmAI model which are crop recommendation dataset, fertilizer prediction dataset, soil fertility dataset and water quality dataset.

4.1 Crop Recommendation Dataset

a) Dataset Description

Crop recommendation dataset comprises of 2200 samples and 8 columns. Each sample represents the type of the crop that most suitable to grow within various parameters. There are 7 parameters which are ratio of Nitrogen content in soil (N), ratio of Phosphorous content in soil (P), ratio of Potassium in soil (K), temperature, humidity, pH value of the soil (ph) and rainfail. These 7 parameters will be used as data input. As for the last column which is label consists of 22 different values represents the type of crop. There are rice, maize, chickpea, kidneybeans, pigeonpeas, mothbeans, mungbean, blackgram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute and coffee. These data on the label column used as data target.

Table 1 Description of data input for Crop Recommendation Dataset

No	Name	Description
1	N	The ratio of Nitrogen content in soil.
2	P	The ratio of Phosphorus content in soil.
3	K	The ratio of Potassium content in soil.
4	Temperature	Temperature of surrounding measure in degree Celsius (°C).
5	Humidity	Humidity in percentage (%)
6	pH	pH value of the soil.
7	Rainfail	Rainfall measured in mm

b) Training and Testing Dataset

The training and testing dataset is the splitted dataset that will be used for training the model and testing the model. There is a function to split the dataset from the sklearn library. However, there is also another way to split the dataset into training and testing dataset which in this dataset. The manual way to split the dataset has been applied in this dataset considering the accessibility of the training and testing dataset when it comes to comparing the same model with different parameters. For crop recommendation dataset, the ratio used to split the dataset in 80:20, meaning 80% of original dataset will be the training dataset and the remaining 20% of dataset will be the testing dataset. The number of samples will set as training data is 1760 samples while the number of samples will be set as testing data is 440 samples.

The step to prepare the training and testing data is quite easy. Starting with randomize the index original dataset, consider saving in another file so that the original dataset can be access in other time. After that, copy the 80% of data and save in csv file and similar with testing data, another 20% save in another csv file. Named it accordingly so that it will be easy to be access. As for this dataset, the name for training data is “Crop_recommendation_train.csv” while the name for testing data is "Crop_recommendation_test.csv”.

4.2 Fertilizer Prediction Dataset

a) Dataset Description

The dataset named Fertilizer Prediction were taken from https://www.kaggle.com/code/karthikreddy77/fertilizer-prediction. It provides 99 samples with 9 columns. Each samples represents the best fertilizer to grow based on the it's parameters which are suitable temperature for the fertilizer (Temperature), suitable humidity for the fertilizer (Humidity), moisture of the fertilizer (Moisture), type of soil used for the fertilizer (Soil Type), type of crop that will be used (Crop Type), ratio of nitrogen (Nitrogen), ratio of potassium (Potassium), ratio of phosphorous (Phosphorous) and name of the fertilizer (Fertilizer Name). For soil type columns, the data comprises options such as sandy, loamy, black, red and clayey. Meanwhile the data of crop type comprises options such as maize, sugarcane, cotton, tobacco, paddy, barley, wheat, millets, oil seeds, pulses, and ground nuts. Same goes to fertilizer name, there are data such as Urea, DAP, 14-35-14, 28-28, 17-17-17, 20-20, 10-26-26. The data target is shown below.

Table 2 Description of data input for Fertilizer Prediction Dataset

No	Name	Description
1.	Temperature	suitable temperature for the fertilizer
2.	Humidity	suitable humidity for the fertilizer
3.	Moisture	moisture of the fertilizer
4.	Soil Type	type of soil used for the fertilizer
5.	Crop Type	type of crop that will be used
6.	Nitrogen	ratio of nitrogen
7.	Potassium	ratio of potassium
8.	Phosphorous	ratio of phosphorous
9.	Fertilizer Name	name of the fertilizer

b) Training and Testing Dataset

First of all, splitting this dataset for training and testing is a bit different compared to others. Normally, the splitting technque that will be used for normal data is either manually split by using excel or by using function from the sklearn library. However, due to the number of samples of this dataset is critically low, cross vaidation by using

5 folds had been applied. The function of using cross validation is to prevent overfitting which occurs when a model is trained too well on the training data and performs poorly on new, unseen data. In 5- Folds Cross Validation, the dataset is splitted into 5 number of folds then the training model is performed on all the folds but leave 4 folds for te evaluation of the trained model. In this method, the model is iterating 5 times with a differents fold reserved for testing puurpose each time.

For the first training model, the rows used is from ([1-19]), a manual implementation of 5-fold cross-validation is carried out for a dataset. Each fold is defined by specifying a list of row indices to be excluded during training, and the process is repeated for five folds. For each fold, training sets are created by dropping the specified rows from the original dataset, and corresponding testing sets are selected using specific ranges of rows. The resulting training and testing sets for all folds are organized into lists. While this approach successfully divides the dataset for cross-validation, it's worth noting that libraries like scikit-learn offer built-in functions, such as sklearn.model_selection.StratifiedKFold or sklearn.model_selection.KFold, which provide a more standardized and efficient means of performing cross-validation, reducing the potential for errors and simplifying the implementation.

4.3 Water Quality Dataset

a) Dataset Description

Water quality dataset obtained from (). It comprises of 3276 samples of water with its quality metric. There are 10 columns in the dataset indicating there are 9 parameters that determine whether the water is safe for human consumption or not. The parameters which are pH value, hardness, solids, chloramines, sulfate, conductivity, organic carbon, trihalomethanes and turbidity. Thes parameters will be used as the data input for Water Quality Classification Model. As for the last column which is potability, there are 0 and 1 indicate the water is potable or not potable respectively. Potability will be used as the data target for model development.

Table 3 Description of data input for Water Quality Dataset

No	Name	Description
1.	pH	pH value of water (0 to 14).
2.	Hardness	Capacity of water to precipitate soap in mg/L.

3.	Solids	Total dissolved solids in ppm.
4.	Chloramines	Amount of chloramines in ppm.
5.	Sulfate	Amount of Sulfates dissolved in mg/L.
6.	Conductivity	Electrical conductivity of water in μS/cm.
7.	Organic Carbon	Amount of organic carbon in ppm.
8.	Trihalomethanes	Amount trihalomethanes in ppm.
9.	Turbidity	Measure of light emiting property of water in NTU.

b) Training and Testing Dataset

For this dataset, the manual way of splitting has been applied. The ratio for training data is 80% from the original dataset and the testing data is the remaining 20% of the dataset. The process of splitting dataset starting from randomize the index of the samples. Next, assign the data from 0 to 2620 index of the sample into variable that will hold the values. The variable will be used as the training dataset. The remaining index until last index will be the testing data. Same as training data, assign a variable to hold the value of samples by using index of the testing data. The amount of training data will be 2619 samples and testing data will be 657 samples.

4.4 Soil Fertility Dataset

a) Dataset Description

Soil fertility dataset is a collection of data on soil based on the elemental analysis. It originally curated by Jaiswal, (2023). The dataset comprises of 880 samples with 13 columns and each sample characterized by the element contents in the soil. The element contents are Nitrogen (NH_4^+)), Phosphorus (P), Potassium (K), soil acidity (pH), electrical conductivity (EC), organic carbon (OC), Sulphur (S), Zinc (Zn), Iron (Fe), Copper (Cu), Manganese (Mn), and Boron (B). These 12 attributes of element soil content will be data input for the dataset. As the “output” column, it consists of 0, 1, and 2 which means “less fertile”, “fertile” and “highly fertile” respectively. This will be the data target to be used the soil fertility prediction model development.

Table 4 Content of elements in the soil

No	Name	Description
1	N	Ratio of Nitrogen (NH_4^+) content in the soil.
2	P	Ratio of Phosphorus (P) content in the soil.

3	K	Ratio of Potassium (K) content in the soil.
4	pH	Soil Acidity (pH)
5	EC	Electrical conductivity - The ability of soil to conduct (transmit) or attenuate electrical current.
6	OC	Organic Carbon
7	S	Sulphur (S)
8	Zn	Zinc (Zn)
9	Fe	Iron (Fe)
10	Cu	Copper (Cu)
11	Mn	Manganese (Mn)
12	B	Boron (B)

b) Training and Tesing Dataset

In this dataset, the “train_test_split” function is used to do the dataset splitting process. To ensure the same index of sample data in the training and testing dataset after running the model repeatedly, the parameter “random state” is being used. This will fix the randomness of the splitted dataset. Then, the comparisons between model can be done fairly because the data used to train, and test will be the same for each model. The ratio of splitting is 80:20. To use this ratio, there are test_size parameter in the syntax of train_test_split function and assigned the test_size with 0.2 which shows 20% will be the testing data. The remaining will be training which is 80%. There are 704 samples of soil will be the training data and the remaining 176 samples will be the testing data.

After splitting the dataset, there are 4 variables have been created. X_train assigned with the input variable of training data (706 sample of the 12 column) while y_train assigned with the output variable (706 sample of output column). X_test is the variable that holds the input variable of testing data (176 samples of 12 column) while y_test is the variable that holds the output variable of testing data (176 samples of output column).

5. MACHINE LEARNING ALGORITHM

The initial process of developing machine learning model by using Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbors (KNN) algorithms follows the same phase. It begins with data preprocessing, addressing missing values, handling outliers and scaling features to ensure uniformly. The datasets are then split into training and testing subsets. All these data preparation phases are important before the training model phase because it can affect the performance of each developed model. Anaconda Navigator is used as the model development platform because it can serve as an Integrated Development Environment (IDE) for managing Python environments, packages, and applications. Jupyter Notebook, accessible through Anaconda allowing for interactive development and documentation of code. Within Jupyter Notebook, code cells can be executed individually, facilitating step-by-step analysis and visualization of results.

5.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. The involvement of SVM algorithm in this project has potential in enhancing agricultural sector in prediction. SVM's strength lies in its ability to handle high-dimensional data, making it exceptionally suited for classifying crops based on a range of factors such as soil conditions, climate conditions, and historical data. The algorithm works to find the optimal hyperplane that separates classes in the feature space, maximizing the margin between them (Support Vector Machine SVM Algorithm, 2021). The features is the data input (X) while the classes is the data target (Y). The inclusion of different kernel functions, such as linear, polynomial, radial basis function (RBF), or sigmoid, enables SVM to handle both linear and non-linear decision boundaries, further enhancing its effectiveness in meeting the study's objectives ( ). This versatile capability contributes significantly to the overall success of the study in classification and prediction task. Below is the detailed process of the model development using SVM algorithm. Figure 1 represents the flowchart of SVM model development.

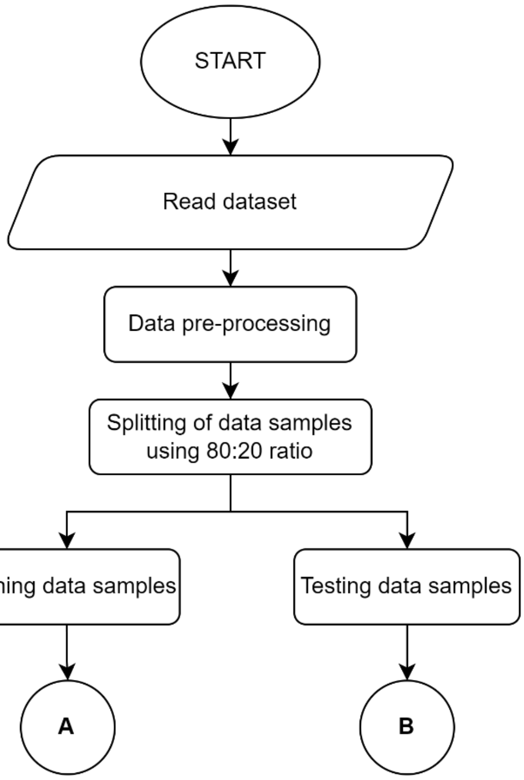
- **Initialization and Dataset Import:** Launch the Jupyter Notebook through Anaconda Navigator. Import the pandas library to enable the function of reading dataset in csv file. The dataset can be read by importing the csv file in same folder of the project file using pd.read_csv syntax. Pandas library also enables data manipulation and analysis.

- **Data Pre-processing:** Assigning the data input and data target into different variables. For example, for soil fertility dataset, variables named as X_soil and y_soil holds the value of data input and data target respectively. Do this to all dataset and named it as easiest way to use the variables.

- **Data Splitting:** Manually split the crop and fertilizer dataset by creating two different new csv file for training and testing. For water quality dataset, it also splitted manually but just using index syntax and declare in the coding. For soil fertility dataset, the train_test_split function is imported from sklearn library so that it can be split using the function. The ratio of the splitting dataset is also the same which are 80% of data will be training data and 20% of remaining data will be testing data.

- **Support Vector Machine Model:** Import the Support Vector Classification and Regression library from the Scikit-learn and used it according to task of the model. Instantiate the SVC and SVR model with a 'linear' kernel.

- **Model Training and Evaluation:** Fit the model to the training data (X_train, y_train) using the fit method. Evaluate the model's performance on the testing dataset based on the predicted output (y_pred, y_test) using Scikit-learn's accuracy_score, confusion_matrix and classification report for classification model while mean squared error (MSE) value for regression model. Repeat the process with alternative kernels ('poly,' 'rbf,' and 'sigmoid') to explore their impact on classification accuracy.



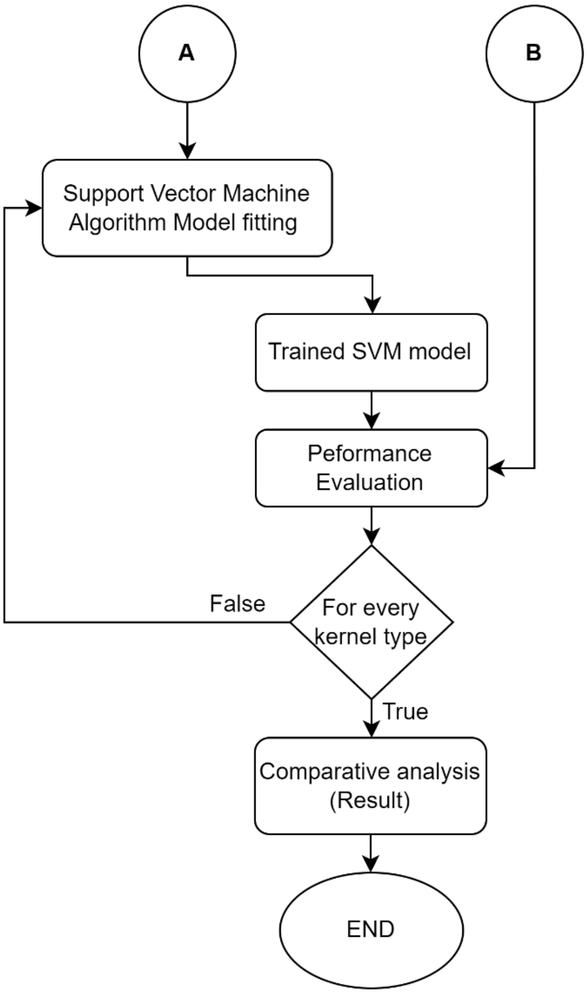

A
B
Support Vector Machine Algorithm Model fitting
Trained SVM model
Peformance Evaluation
False
For every kernel type
True
Comparative analysis (Result)
END

Figure 1 SVM Model Flowchart

5.2 Decision Tree (DT)

A decision tree algorithm is a supervised machine learning algorithm used for both classification and regression tasks. The algorithm creates a tree-like model of decisions based on features present in the training data. It is a predictive modeling tool that recursively splits the dataset into subsets based on the most significant attribute at each node of the tree. Incorporating the Decision Tree algorithm into this project can greatly enhance decision-making processes in agriculture. Decision Trees are particularly renowned for their ease of interpretation and ability to handle both numerical and categorical data. In the agricultural context, they can be utilized to create decision support systems that guide farmers in crucial areas such as crop rotation, fertilization, and irrigation practices. The clarity with which Decision Trees present information can help analyse complex agricultural data, enabling farmers to make more informed and effective decisions. Below is the detailed process of implementing Decision Tree algorithm in machine learning model development. Figure 2 shows the Decision Tree model flowchart.

- **Initialization and Dataset Import:** Launch the Jupyter Notebook through Anaconda Navigator. Import the pandas library to enable the function of reading dataset in csv file. The dataset can be read by importing the csv file in same folder of the project file using pd.read_csv syntax. Pandas library also enables data manipulation and analysis.

- **Data Pre-processing:** Assigning the data input and data target into different variables. For example, for water quality dataset, variables named as water_input and water_target holds the value of data input and data target respectively. Do this to all dataset and named it as the easiest way to use the variables.

- **Data Splitting:** Manually split the crop and fertilizer dataset by creating two different new csv file for training and testing. For water quality dataset, it also splitted manually but just using index syntax and declare in the coding. For soil fertility dataset, the train_test_split function is imported from sklearn library so that it can be split using the function. The ratio of the splitting dataset is also the same which are 80% of data will be training data and 20% of remaining data will be testing data.

- **Decision Tree Model:** Import the Decision Tree Classification and Regression library from the Scikit-learn and used it according to task of the model.

- **Model Training and Evaluation:** Fit the model to the training data (X_train, y_train) using the fit method. Evaluate the model's performance on the testing dataset based on the predicted output (y_pred, y_test) using Scikit-learn's accuracy_score, confusion_matrix and classification report for classification model while mean squared error (MSE) value for regression model.

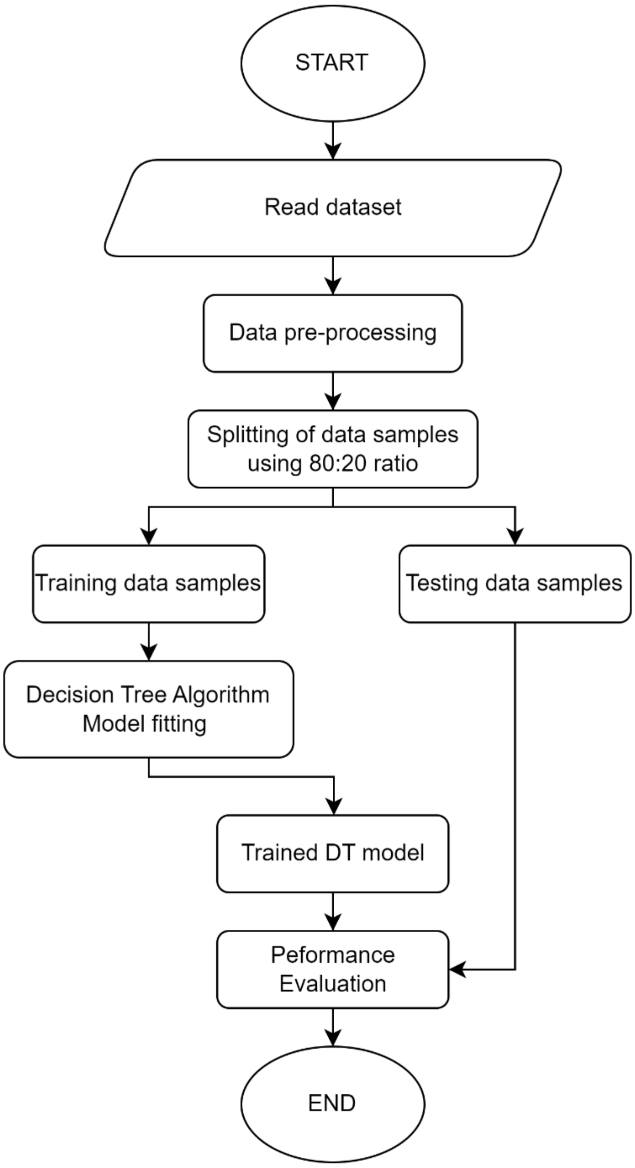


Figure 2 DT Model Flowchart

5.3 K-Nearest Neighbors (KNN)

The k-Nearest Neighbors algorithm is a supervised machine learning algorithm used for classification and regression tasks. It is a simple and intuitive algorithm that makes predictions based on the majority class or average of the k-nearest data points in the feature space. When determining the k-nearest neighbors for a given data point, KNN calculates the Euclidean distance between that point and all other points in the dataset and selects the k-nearest neighbors based on the smallest distances. This algorithm is another valuable addition to the ML techniques in the project because of it effectiveness in pattern recognition make it a prime choice for applications like predicting soil nutrient levels or identifying patterns in pest infestations. The algorithm's non-parametric nature allows it to adapt to various types of data, which is a significant advantage given the diverse data sets in agriculture. Below is the details on how KNN be implemented in model development.

- **Initialization and Dataset Import:** Launch the Jupyter Notebook through Anaconda Navigator. Import the pandas library to enable the function of reading dataset in csv file. The dataset can be read by importing the csv file in same folder of the project file using pd.read_csv syntax. Pandas library also enables data manipulation and analysis.

- **Data Pre-processing:** Assigning the data input and data target into different variables.

- **Data Splitting:** Manually split the crop and fertilizer dataset by creating two different new csv file for training and testing. For water quality dataset, it also splitted manually but just using index syntax and declare in the coding. For soil fertility dataset, the train_test_split function is imported from sklearn library so that it can be split using the function. The ratio of the splitting dataset is also the same which are 80% of data will be training data and 20% of remaining data will be testing data.

- **K-Nearest Neighbors Model:** Import the KNeigborClassifier or KNeighborRegressor library from the Scikit-learn and used it according to task of the model. Instantiate the KNN model with a specified value of K (n_neighbors).

- **Model Training and Evaluation:** Fit the model to the training data (X_train, y_train) using the fit method. Evaluate the model's performance on the testing dataset based on the predicted output (y_pred, y_test) using Scikit-learn's accuracy_score, confusion_matrix and classification report for classification model while mean squared error (MSE) value for regression model. Repeat the process with alternative kernels ('poly,' 'rbf,' and 'sigmoid') to explore their impact on classification accuracy.

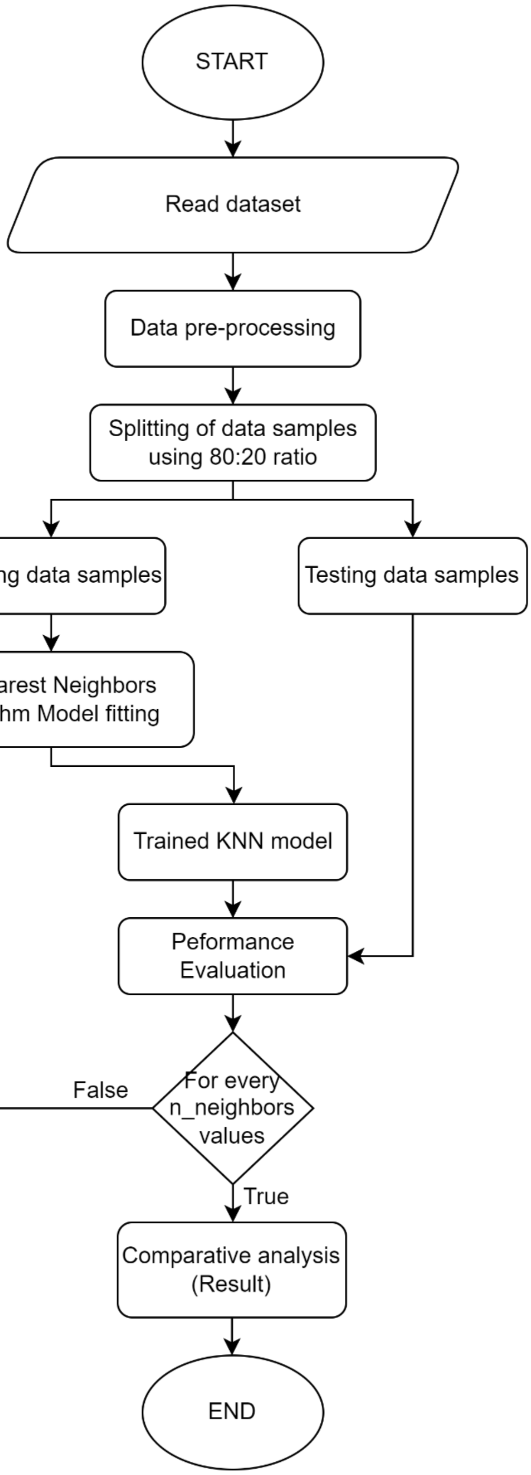

START
Read dataset
Data pre-processing
Splitting of data samples using 80:20 ratio
Training data samples
Testing data samples
K-Nearest Neighbors Algorithm Model fitting
Trained KNN model
Peformance Evaluation
False
For every n_neighbors values
True
Comparative analysis (Result)
END

Figure 3 KNN Model Flowchart

6. RESUTLS AND DISCUSSIONS

The results obtained are organized in a tabular format for clarity. Each model's performance outcomes are scrutinized, and through a comparative analysis, the model with the highest accuracy is selected to represent each specific algorithm. The emphasis is placed on the model that achieves the utmost accuracy, encapsulating the central findings and insights derived from this study. The ensuing discussion elaborates on the key takeaways from the chosen model, highlighting its significance within the context of the study's objectives.

6.1 Results

a) SVM Model Results

Table 5 SVM result for Crop Classification Model

Kernel	Accuracy Score (%)
Linear	96.59
Polynomial	96.83
Radial Basis Function (RBF)	96.15
Sigmoid	9.75

The results presented in Table 4 illustrates the accuracy score of crop classification model using SVM algorithm with 4 different kernels. The Polynomial kernel exhibited the highest accuracy score, reaching 96.83%. In contrast, both Linear and RBF kernels displayed slightly lower accuracy score with 96.59% and 96.15% respectively. The Sigmoid kernel yield a lowest accuracy score with 9.75%. Based from the interpretation, the Polynomial kernel is chosen to represent the SVM model for crop classification task.

Table 6 SVM result for Fertilizer Classification Model

Kernel	Accuracy Score (%)
Linear	84.21
Polynomial	78.95
Radial Basis Function (RBF)	57.89
Sigmoid	84.21

Table 5 summarizes the accuracy scores of the SVM algorithm applied to a Fertilizer Classification Model using four different kernels. The Linear and Sigmoid kernels achieved the highest accuracy at 84.21%, followed by the Polynomial kernel with 78.95%. The Radial Basis Function (RBF) kernel showed the lowest accuracy of

57.89%. In this context, the Linear kernel emerges as the representative of kernel for SVM model in fertilizer classification task.

Table 7 SVM result for Water Quality Classification Model

Kernel	Accuracy Score (%)
Linear	62.65
Polynomial	62.35
Radial Basis Function (RBF)	62.35
Sigmoid	53.20

Table 6 illustrates the accuracy scores for the Water Quality Classification Model employing SVM with various kernels. The results showcase comparable performance across the kernels, with the Linear, Polynomial, and RBF kernels exhibiting accuracy scores of 62.65%, 62.35%, and 62.35%, respectively. The Sigmoid kernel recorded a slightly lower accuracy at 53.20%. Despite the marginal differences, the Linear kernel is selected as the representative for the SVM model in the Water Quality Classification task due to its slightly higher accuracy, emphasizing its suitability for this specific classification context.

Table 8 SVM result for Soil Fertility Classification Model

Kernel	Accuracy Score (%)
Linear	83.52
Polynomial	84.66
Radial Basis Function (RBF)	85.80
Sigmoid	41.48

Table 7 delineates the accuracy scores for the SVM algorithm applied to the Soil Fertility Classification Model using various kernels. Notably, the Radial Basis Function (RBF) kernel demonstrated the highest accuracy at 85.80%, followed closely by the Polynomial kernel with 84.66%. The Linear kernel also performed well, achieving an accuracy score of 83.52%. In contrast, the Sigmoid kernel exhibited a lower accuracy of 41.48%. Consequently, the RBF kernel is chosen to represent the SVM model for the Soil Fertility Classification task, emerging as the most effective kernel based on its higher accuracy in this specific classification scenario.

b) DT Model Results

Table 9 DT result for each Classification Model

Model	Accuracy Score (%)
Crop	98
Fertilizer	100
Water Quality	61.43
Soil Fertility	83.52

Table 8 presents the accuracy scores for Decision Tree (DT) models applied to various classification tasks. Notably, the Crop Classification Model achieved an accuracy score of 98%, showcasing the model's effectiveness in accurately classifying crops. The Fertilizer Classification Model demonstrated perfect accuracy at 100%, indicating precise predictions in this domain. For Water Quality Classification, the DT model achieved an accuracy of 61.43%, while the Soil Fertility Classification Model achieved an accuracy of 83.52%. These results emphasize the robust performance of the Decision Tree model, particularly in tasks such as crop and fertilizer classification.

c) KNN Model Results

Table 10 KNN result for Crop Classification Model

N_neighbors	Accuracy Score (%)
10	96.37
20	95.24
30	94.33
40	93.88

Table 9 summarizes the KNN Model Results for the crop classification task, showcasing accuracy scores for different values of N_neighbors. The highest accuracy is achieved with 10 neighbors at 96.37%, followed by 20 neighbors at 95.24%, 30 neighbors at 94.33%, and 40 neighbors at 93.88%. These results provide insights into the performance of the KNN model for crop classification, with a decreasing trend in accuracy as the number of neighbors increases. Hence, 10 neighbors is the chosen to represent the KNN model for crop classification task.

Table 11 KNN result for Fertilizer Classification Model

N_neighbors	Accuracy Score (%)
10	75.25
20	70.08
30	56.87
40	51.53

Table 10 presents the KNN Model Results for the fertilizer classification task, displaying accuracy scores for different values of N_neighbors. The accuracy decreases with an increasing number of neighbors, with the highest accuracy at 10 neighbors (75.25%), followed by 20 neighbors (70.08%), 30 neighbors (56.87%), and 40 neighbors (51.53%). Therefore, Fertilizer Classification Model with KNN algorithm will be using 10 neighbors to represent the KNN model.

Table 12 KNN result for Water Quality Classification Model

N_neighbors	Accuracy Score (%)
5	57.01
25	60.82
50	62.28
100	61.13

Table 11 delineates the KNN Model Results for the Water Quality Classification Model, presenting accuracy scores for different values of N_neighbors. The highest accuracy is achieved with 50 neighbors at 62.28%, followed by 100 neighbors at 61.13%, 25 neighbors at 60.82%, and 5 neighbors at 57.01%. Therefore, 50 number of neighbors is chosen to represent this KNN model.

Table 13 KNN result for Soil Fertility Classification Model

N_neighbors	Accuracy Score (%)
10	81.82
20	83.52
30	84.09
40	84.66

Table 12 outlines the Soil Fertility Classification Model with KNN algorithm, showcasing accuracy scores for varying values of N_neighbors. The highest accuracy is observed with 40 neighbors at 84.66%, closely followed by 30 neighbors at 84.09%, 20 neighbors at 83.52%, and 10 neighbors at 81.82%. Based on the analysis, 40 number of neighbors is chosen to represent the KNN model.

6.2 Discussions

Table 14 Summary of the results

Model	Algorithm	Accuracy Score (%)
Crop Classification	SVM (kernel = poly)	96.83
	DT	98
	KNN(n_neighbors = 10)	96.37
Fertilizer Classification	SVM (kernel = linear)	84.21
	DT	100
	KNN (n_neighbors = 10)	75.25
Water Quality Classification	SVM (kernel = linear)	62.65
	DT	61.43
	KNN (n_neighbors = 50)	62.28
Soil Fertility Classification	SVM (kernel = rbf)	85.80
	DT	83.52
	KNN (n_neighbors = 40)	84.66

The summary in Table 12 provides an overview of the results for each classification model and the corresponding algorithms with their accuracy scores. Based on the highest accuracy achieved for each task, the best algorithms are determined as follows:

i. **Crop Classification Model:**
   - Best Algorithm: Decision Tree (DT)
   - Accuracy Score: 98%

ii. **Fertilizer Classification Model:**
   - Best Algorithm: Decision Tree (DT)
   - Accuracy Score: 100%

iii. **Water Quality Classification Model:**
   - Best Algorithm: SVM with linear kernel
   - Accuracy Score: 62.65%

iv. **Soil Fertilizer Classification Model:**
   - Best Algorithm: SVM with RBF kernel
   - Accuracy Score: 85.80%

7. CONCLUSIONS

In conclusion, the comprehensive evaluation of various classification algorithms within the FarmAI model has provided valuable insights into their performance across different agricultural tasks. The optimal algorithmic choices, tailored to specific classification models, showcase the versatility and adaptability required for successful implementation in precision agriculture. The Decision Tree algorithm consistently excelled, demonstrating its efficacy in both Crop and Fertilizer Classification Model, achieving accuracy scores of 98% and 100%, respectively. The SVM algorithm, particularly with a polynomial kernel for Crop Classification and an RBF kernel for Soil Fertility Classification, showcased robust performance, emphasizing the importance of algorithm selection for specific tasks. KNN demonstrated competitive results, with its effectiveness dependent on the nature of the classification task. As we amalgamate these findings into the FarmAI model, it becomes evident that a nuanced approach to algorithmic selection is crucial, leveraging the strengths of each algorithm to optimize the accuracy and reliability of predictions across diverse facets of agriculture. This holistic evaluation provides a solid foundation for the implementation of the FarmAI model, offering farmers valuable insights for informed decision-making in crop selection, fertilizer application, water quality assessment, and soil fertility management.

8. REFERENCES

Gill, A., Kaur, T., & Devi, Y. K. (2022). Application of Machine Learning Techniques in Modern Agriculture: A Review. *ACM International Conference Proceeding Series*, 263–270. https://doi.org/10.1145/3549206.3549255

Kansal, L., Pandey, A., Shukla, S. M., & Dhaliwal, P. (2023). Review of Machine Learning Techniques for Crop Recommendation. *Proceedings of the 2023 Fifteenth International Conference on Contemporary Computing*, 443–449. https://doi.org/10.1145/3607947.3608045

Jaiswal, R. (2023). *Soil Fertility Dataset*. Kaggle.com. https://www.kaggle.com/datasets/rahuljaiswalonkaggle/soil-fertility-dataset

Aditya Kadiwal. (2021). *Water Quality*. Kaggle.com. https://www.kaggle.com/datasets/adityakadiwal/water-potability

Sharma, S. (2021). *Crop Recommendation Dataset*. Kaggle.com. https://www.kaggle.com/datasets/siddharthss/crop-recommendation-dataset

karthikreddy77. (2023, October 8). *Fertilizer Prediction*. Kaggle.com; Kaggle. https://www.kaggle.com/code/karthikreddy77/fertilizer-prediction

Support Vector Machine SVM Algorithm. (2021, January 20). GeeksforGeeks; GeeksforGeeks. https://www.geeksforgeeks.org/support-vector-machine-algorithm/

Wiener Bronner, D. (2023, August 3). *India's recent rice ban sent people into a panic. Here's what's going on now*. CNN Business. Retrieved November 22, 2023, from https://www.cnn.com/2023/08/03/business/india-rice-export-ban