Plant Growth Recommendation System

Sowjanya Sailada, 211CS149
Sneha Sri Dulam, 211EE256
Prakriti Kunj, 211CV238
National Institute of Technology Karnataka
Surathkal, Mangalore, India

I. Abstract

The agriculture industry is vital for ensuring food security and economic stability, making it crucial to enhance crop productivity and optimize resource utilization,this project presents a Plant Growth Recommendation System (PGRS) that uses Machine Learning and Deep Learning techniques to offer precise recommendations for crop cultivation based on real-time weather conditions and soil patterns.

Keywords: Machine Learning, Random Forest Classifier ,Support Vector Machine, K-Nearest Neighbour,Hyperparameter Tuning.

II. Introduction

Crop Recommendation plays a vital role in agriculture ,empowering farmers to make decisions about suitable crops to grow. As technology advances there comes various crop predicting systems. We here are using machine learning algorithms for our crop recommendation system. It predicts suitable crop based on various parameters such as pH of soil, N,P,K contents in soil ,rainfall ,weather conditions such as temperatue ,humidity etc.

III. Literature Review

A. Historical Evolution of Plant growth recommendation systems

The evolution of plant growth recommendation systems reflects a fascinating journey from traditional farming practices to the cutting-edge, technology-driven solutions of today.Introduction of Precision Agriculture (Late 20th Century) precision agriculture began to gain traction with the use of GPS technology, allowing farmers to optimize field-level management with greater accuracy. Decision Support Systems (DSS) started to emerge, incorporating basic algorithms and databases to assist farmers in making more informed decisions about planting, irrigation, and fertilization.

Machine learning algorithms, such as decision trees and neural networks, began to be applied for predicting crop yields and optimizing farming practices.Advanced machine learning techniques, including Random Forest Classifiers, Support Vector Machines, and ensemble methods, have become integral to plant growth recommendation systems.

B. Activities and Actions of plant growth recommendation systems

Plant growth recommendation systems play a crucial role in modern agriculture by providing actionable insights and guidance to farmers. These systems analyze data and offer recommendations for optimizing plant growth. Key activities and actions performed by plant growth recommendation systems:

Soil analysis,climate monitoring,crop-specific Data gathering information on crop varieties, growth stages, and historical performance. By employing algorithms such as Random Forest, Support Vector Machines, or neural networks to analyze and process the collected data.

By performing these activities, plant growth recommendation systems contribute to more efficient, sustainable, and productive agricultural practices. They empower farmers with data-driven insights to make informed decisions, ultimately enhancing crop yields and resource management.

IV. Algorithms

1) Random Forest Classifier: The Random Forest Classifier is a key component of our Crop Recommendation System, designed to suggest multiple crops suitable for a specific type of soil. Instead of relying on a single decision tree, the Random Forest employs multiple decision trees, each trained on different subsets of our training data.

These subsets are randomly selected from our training data, creating a diverse set of decision trees. Each decision tree in the Random Forest provides its own prediction for a suitable crop based on the given soil properties and environmental conditions.

The Random Forest algorithm collects the votes from these individual decision trees. It considers the crops that receive the most votes from the decision trees as recommended options for cultivation. Unlike a traditional decision tree that might pick a single best crop, our system aims to provide flexibility and diversity. Using the Random Forest, we can suggest not just one, but several crops that are deemed suitable for the given soil conditions.

2) KNN : K-Nearest Neighbours (KNN) algorithm It works by finding similarities between new data and existing cases, and then assigns the new data to the category that is most similar to the existing categories.

Neighbour-Based Classification: KNN checks which existing data points are the most similar (nearest neighbours) to the new data. During the training phase, KNN doesn't make any assumptions about the underlying data. It simply stores the dataset, keeping all available data points in memory. When new data arrives, KNN looks at how the nearest neighbours are classified. If most of the nearest neighbours belong to a particular category (e.g., a specific crop type), the new data is classified into that same category.

Using KNN, we can classify different crops based on features they share. For example, if new data about a crop's characteristics is introduced, KNN assesses which existing crops are most similar based on those features and assigns the new crop to the category it closely resembles.

In our Crop Classification system, KNN uses the concept of similarity to group crops with similar characteristics. It's a flexible and adaptive approach, making it suitable for various agricultural scenarios where data characteristics can vary widely.

3) Support Vector Machine: A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for classification or regression tasks. The primary goal of an SVM is to find the hyperplane that best separates the data into different classes.

In a two-dimensional space, a hyperplane is a line that separates the data into two classes.In a higher-dimensional space, a hyperplane is a subspace of one dimension less than the input space. Support vectors are the data points that are closest to the hyperplane and have the maximum margin.The margin is the distance between the hyperplane and the nearest data point from each class.

The margin is a measure of how well-separated the classes are.SVM aims to maximize the margin, i.e., finding the hyperplane that maximizes the distance between the support vectors of the two classes. SVM can be used for both linear and non-linear classification tasks. Linear SVM is suitable for linearly separable data, while non-linear SVM, with the help of kernels, can handle more complex decision boundaries.

To implement SVM,machine learning libraries such as Scikit-Learn in Python, which provides a 'SVC' (Support Vector Classification) class for SVM classification tasks and an 'SVR' (Support Vector Regression) class for regression tasks are used.

V. Methodology

A. Data

Data Exploration:We collected dataset and analyzed the features available and their types plotting statistics, distributions, and correlations between variables to gain insights into the data.

Data Preprocessing: Handle missing data: Impute missing values or remove instances with missing values.Encode categorical variables: Convert categorical variables into a numerical format suitable for machine learning models. Scale or normalize features: Ensure that numerical features are on a similar scale to prevent certain features from dominating others during model training.

Feature Engineering:Create new features if necessary, based on domain knowledge or insights from the data exploration phase.Select relevant features that contribute the most to the prediction task. Train-Test Split:Split your dataset into training and testing sets. This helps evaluate the model's performance on unseen data.

B. Algorithm Implentation

Implementation of Random Forest, k-Nearest Neighbors (KNN), and Support Vector Machines (SVM) for a plant growth recommendation system:

Random Forest Classifier:

The Random Forest classifier is an ensemble learning method that builds a multitude of decision trees during training and merges them together to get a more accurate and stable prediction. the implementation involves the following steps:

(i)Data Splitting:The dataset is split into training and testing sets,using an 80-20 ratio.

(ii)Model Creation:A Random Forest classifier is instantiated with a specified number of trees (n$_estimators), anditistrainedonthetrainingdata.$

(iii)Prediction: The trained model is used to make predictions on the test set.

k-Nearest Neighbors (KNN):

The k-Nearest Neighbors algorithm is a simple,effective, instance-based learning method that classifies data points based on the majority class of their k-nearest neighbors.

(i)Data Splitting: Similar to Random Forest, the dataset is split into training and testing sets.

(ii)Feature Scaling:Features are standardized using methods like z-score normalization to ensure that all features contribute equally to the distance calculation.

(iii)Model Training: The KNN classifier is created and trained on the scaled training data.

(iv)Prediction: The model predicts the class labels of the test set based on the majority class among its k-nearest neighbors.

Support Vector Machine (SVM):

Support Vector Machines are powerful supervised learning models that find the optimal hyperplane to separate classes in feature space.

(i)Data Splitting:As before, the dataset is split into training and testing sets.

(ii)Feature Scaling:Standardization is applied to ensure that all features have a similar scale, preventing certain features from dominating the others.

(iii)Model Creation:An SVM classifier with a linear kernel is instantiated, and it is trained on the standardized training data.

(iv)Prediction:The trained SVM model predicts the class labels of the test set.

Performance Evaluation: Accuracy and other metrics are calculated to evaluate how well the SVM ,KNN and Random Forest Classifier model performs on the test set.

C. Recommendation Model

We created a recommendation model that suggests the farmers to add fertilisers or reduce fertiliser based on the soil ,and what type of nutrients are required whether it is N,P,K or increase irrigation considering the rainfall.We also recommend pH action to increase or reduce pH .We can see in the output has a column of recommended action on the predicted crop by the models used.

VI. Results

A. Model Evaluation Metrics

Model Evaluation Metrics

Accuracy:Accuracy is the proportion of correctly classified instances out of the total number of instances. It is a commonly used metric in classification problems, including crop recommendation systems. A high accuracy score indicates that the model is correctly predicting crop recommendations.

Precision:Precision is the proportion of true positives (correctly predicted positive instances) out of the total number of predicts positive instances. Precision is an important metric when the cost of false positives is high, such as in medical diagnosis or financial fraud detection.

F1-score:The F1-score is the harmonic mean of precision and recall. It is a useful metric when the classes are imbalanced, meaning one class has significantly more instances than the other. The F1-score provides a balanced measure of precision and recall.

Random Forest Classifier

RFC Accuracy: 0.9435064935064935
RFC Classification Report:

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	72
banana	1.00	1.00	1.00	63
blackgram	0.96	0.97	0.96	67
chickpea	1.00	1.00	1.00	75
coconut	1.00	1.00	1.00	71
coffee	0.99	0.99	0.99	68
cotton	0.97	1.00	0.99	67
grapes	1.00	1.00	1.00	69
jute	0.80	0.98	0.88	65
kidneybeans	0.92	0.71	0.80	79
lentil	0.88	0.87	0.87	68
maize	1.00	0.97	0.99	72
mango	1.00	1.00	1.00	65
mothbeans	0.81	0.71	0.76	77
mungbean	0.77	0.89	0.83	65
muskmelon	1.00	1.00	1.00	72
orange	1.00	1.00	1.00	66
papaya	1.00	1.00	1.00	71
pigeonpeas	0.74	0.93	0.82	68
pomegranate	1.00	1.00	1.00	69
rice	1.00	0.80	0.89	76
watermelon	1.00	1.00	1.00	75
accuracy			0.94	1540
macro avg	0.95	0.95	0.94	1540
weighted avg	0.95	0.94	0.94	1540

Random Forest after hypertuning parameters

Best Hyperparameters: {'n_estimators': 100, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_depth': 20}
RFC Accuracy: 0.9636363636363636
Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	1.00	1.00	1.00	21
2	0.75	0.90	0.82	20
3	1.00	1.00	1.00	26
4	1.00	1.00	1.00	27
5	1.00	1.00	1.00	17
6	1.00	1.00	1.00	17
7	1.00	1.00	1.00	14
8	0.91	0.91	0.91	23
9	1.00	1.00	1.00	20
10	0.62	0.91	0.74	11
11	1.00	1.00	1.00	21
12	1.00	1.00	1.00	19
13	1.00	0.96	0.98	24
14	1.00	1.00	1.00	19
15	1.00	1.00	1.00	17
16	1.00	1.00	1.00	14
17	1.00	1.00	1.00	23
18	1.00	0.65	0.79	23
19	1.00	1.00	1.00	23
20	0.89	0.89	0.89	19
21	1.00	1.00	1.00	19
accuracy			0.96	440
macro avg	0.96	0.96	0.96	440
weighted avg	0.97	0.96	0.96	440

KNN algorithm

KNN Accuracy: 0.9022727272727272
KNN Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	1.00	1.00	1.00	21
2	0.71	0.85	0.77	20
3	1.00	1.00	1.00	26
4	1.00	1.00	1.00	27
5	0.94	0.94	0.94	17
6	0.94	1.00	0.97	17
7	1.00	1.00	1.00	14
8	0.76	0.57	0.65	23
9	1.00	1.00	1.00	20
10	0.45	0.82	0.58	11
11	1.00	0.95	0.98	21
12	0.95	1.00	0.97	19
13	0.91	0.83	0.87	24
14	1.00	1.00	1.00	19
15	0.76	0.94	0.84	17
16	1.00	1.00	1.00	14
17	1.00	1.00	1.00	23
18	0.91	0.43	0.59	23
19	1.00	1.00	1.00	23
20	0.64	0.84	0.73	19
21	0.93	0.74	0.82	19
accuracy			0.90	440
macro avg	0.91	0.91	0.90	440
weighted avg	0.92	0.90	0.90	440

KNN algorithm after hypertuning parameters

Best Hyperparameters: {'weights': 'distance', 'p': 2, 'n_neighbors': 5}
KNN Accuracy: 0.975
KNN Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	1.00	1.00	1.00	21
2	0.95	1.00	0.98	20
3	1.00	1.00	1.00	26
4	1.00	1.00	1.00	27
5	1.00	0.94	0.97	17
6	0.85	1.00	0.92	17
7	1.00	1.00	1.00	14
8	0.85	1.00	0.92	23
9	0.95	1.00	0.98	20
10	0.85	1.00	0.92	11
11	1.00	0.86	0.92	21
12	1.00	1.00	1.00	19
13	1.00	0.92	0.96	24
14	1.00	1.00	1.00	19
15	1.00	1.00	1.00	17
16	1.00	1.00	1.00	14
17	1.00	1.00	1.00	23
18	1.00	0.91	0.95	23
19	1.00	1.00	1.00	23
20	1.00	0.84	0.91	19
21	1.00	1.00	1.00	19
accuracy			0.97	440
macro avg	0.98	0.98	0.97	440
weighted avg	0.98	0.97	0.98	440

SVM algorithm

SVM Accuracy: 0.977
SVM Classification Report:

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	23
banana	1.00	1.00	1.00	21
blackgram	0.95	1.00	0.98	20
chickpea	1.00	1.00	1.00	26
coconut	1.00	1.00	1.00	27
coffee	1.00	0.94	0.97	17
cotton	0.94	1.00	0.97	17
grapes	1.00	1.00	1.00	14
jute	0.79	1.00	0.88	23
kidneybeans	1.00	1.00	1.00	20
lentil	0.85	1.00	0.92	11
maize	1.00	0.95	0.98	21
mango	1.00	1.00	1.00	19
mothbeans	1.00	0.92	0.96	24
mungbean	1.00	1.00	1.00	19
muskmelon	1.00	1.00	1.00	17
orange	1.00	1.00	1.00	14
papaya	1.00	1.00	1.00	23
pigeonpeas	1.00	0.96	0.98	23
pomegranate	1.00	1.00	1.00	23
rice	1.00	0.74	0.85	19
watermelon	1.00	1.00	1.00	19
accuracy			0.98	440
macro avg	0.98	0.98	0.98	440
weighted avg	0.98	0.98	0.98	440

Results of SVM algorithm with hyperparameter tuning

```
Best Hyperparameters: {'weights': 'distance', 'p': 2, 'n_neighbors': 5}
KNN Accuracy: 0.975
KNN Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        23
           1       1.00      1.00      1.00        21
           2       0.95      1.00      0.98        20
           3       1.00      1.00      1.00        26
           4       1.00      1.00      1.00        27
           5       1.00      0.94      0.97        17
           6       0.85      1.00      0.92        17
           7       1.00      1.00      1.00        14
           8       0.85      1.00      0.92        23
           9       0.95      1.00      0.98        20
          10       0.85      1.00      0.92        11
          11       1.00      0.86      0.92        21
          12       1.00      1.00      1.00        19
          13       1.00      0.92      0.96        24
          14       1.00      1.00      1.00        19
          15       1.00      1.00      1.00        17
          16       1.00      1.00      1.00        14
          17       1.00      1.00      1.00        23
          18       1.00      0.91      0.95        23
          19       1.00      1.00      1.00        23
          20       1.00      0.84      0.91        19
          21       1.00      1.00      1.00        19

    accuracy                           0.97       440
   macro avg       0.98      0.98      0.97       440
weighted avg       0.98      0.97      0.98       440
```

Model	Accuracy	Accuracy(optimised)
Random Forest	94.3	96.3
KNN	90.2	97.5
SVM	97.7	98.0

TABLE I
COMPARISON OF ACCURACY OF MODELS WITH AND WITHOUT HYPERPARAMETER TUNING OPTIMIZATION

B. Recommendation Model

Recommendation model tells user for that predicted crop what action should be taken to enhance the crop growth. What nutrients should be added or reduced, irrigation,pH action etc.

```
        N   P   K  temperature   humidity        ph    rainfall   label  \
0      90  42  43    20.879744  82.002744  6.502985  202.935536    rice
1      85  58  41    21.770462  80.319644  7.038096  226.655537    rice
2      60  55  44    23.004459  82.320763  7.840207  263.964248    rice
3      74  35  40    26.491096  80.158363  6.980401  242.864034    rice
4      78  42  42    20.130175  81.604873  7.628473  262.717340    rice
...   ...  ..  ..          ...        ...       ...         ...     ...
2195  107  34  32    26.774637  66.413269  6.780064  177.774507  coffee
2196   99  15  27    27.417112  56.636362  6.086922  127.924610  coffee
2197  118  33  30    24.131797  67.225123  6.362608  173.322839  coffee
2198  117  32  34    26.272418  52.127394  6.758793  127.175293  coffee
2199  104  18  30    23.603016  60.396475  6.779833  140.937041  coffee

     Reduce_Nutrient Increase_Nutrient Water_Action   pH_Action
0                [K]            [N, P]      Optimal  Optimal pH
1             [P, K]               [N]      Optimal  Optimal pH
2             [P, K]               [N]       Reduce      Reduce
3                [K]            [N, P]      Optimal  Optimal pH
4                [K]            [N, P]       Reduce      Reduce
...              ...               ...          ...         ...
2195          [N, K]               [P]      Optimal      Reduce
2196          [N, K]               [P]     Increase  Optimal pH
2197          [N, K]               [P]      Optimal  Optimal pH
2198          [N, K]               [P]     Increase      Reduce
2199          [N, K]               [P]      Optimal      Reduce

[2200 rows x 12 columns]
```

+ Code + Markdown

VII. FUTURE WORK

1)Climate-Responsive Recommendations: Develop plant growth recommendation systems that dynamically adapt to changing climate conditions, providing farmers with actionable insights to mitigate the impact of climate change on crops.

2)Multi-Modal Recommendations:Combining Models to investigate the synergies of combining different machine learning models, such as ensemble models or hybrid models, to leverage the strengths of each approach.

3)Dynamic Crop Rotation Planning: Develop systems that dynamically recommend crop rotation plans based on historical data, soil health, and pest prevalence to optimize yields and soil fertility.

REFERENCES

[1] https://ieeexplore.ieee.org/document/10084211/

[2] Dataset : https://typeset.io/papers/plant-growing-system-32rm6vnbnm

[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6409995/

[4] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10084211

[5] https://link.springer.com/chapter/10.1007/978-981-99-0085-5$_6$8