A NUTRIENT RECOMMENDATION SYSTEM WITH CLIMATE PREDICTION USING RANDOM FOREST ALGORITHM

PROJECT WORK1(REVIEW1)

Submitted by

DIVYA S 212221040043

in partial fulfillment of the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SAVEETHA ENGINEERING COLLEGE, THANDALAM

An Autonomous Institution Affiliated to

ANNA UNIVERSITY - CHENNAI 600 025

NOVEMBER 2024

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project report "**A NUTRIENT RECOMMENDATION SYSTEM WITH CLIMATE PREDICTION USING RANDOM FOREST ALGORITHM**" is the bonafide work of **DIVYA S(212221040043)** who carried out this project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. Anish Kumar	**Dr. G. Nagappan, M.E., PhD**
SUPERVISOR	**Professor**
Assistant Professor, CSE	**HEAD OF THE DEPARTMENT**
Dept of Computer Science and Engineering,	Dept of Computer Science and Engineering,
Saveetha Engineering College, Thandalam, Chennai 602105	Saveetha Engineering College, Thandalam, Chennai 602105

DATE OF THE VIVA VOCE EXAMINATION: …………………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to our esteemed Founder President **Dr. N. M. Veeraiyan**, our President **Dr. Saveetha Rajesh**, our Director **Dr. S. Rajesh**, and the entire management team for providing the essential infrastructure.

I extend my sincere appreciation to our principal, **Dr. V. Vijaya Chamundeeswari, M.Tech., Ph.D**., for creating a supportive learning environment for this project.

I am very thankful to our Dean of ICT, **Mr. Obed Otto**, M.E., for facilitating a conducive atmosphere that allowed me to complete my project successfully.

My thanks go to **Dr. G. Nagappan**, Professor and Head of the Department of Computer Science and Engineering at Saveetha Engineering College, for his generous support and for providing the necessary resources for my project work.

I would also like to express my profound gratitude to my supervisor, **Mr. Premnath K**, Assistant Professor and my Project Coordinator **Dr. N.S. Gowri Ganesh**, Associate Professor at Saveetha Engineering College, for their invaluable guidance, suggestions, and constant encouragement, which were instrumental in the successful completion of this project. Their timely support and insights during the review process were greatly appreciated.

I am grateful to all my college faculty, staff, and technicians for their cooperation throughout the project. Finally, I wish to acknowledge my loving parents, friends, and well-wishers for their encouragement in helping me achieve this milestone.

ABSTRACT

Fertilizer use is typically under the limited control of farmers. For the farmers to achieve higher yields and reduce fertilizer loss, competent guidance is required for the best use of these fertilizers. Additionally, there is a connection between rainfall volume and nutrient loss for various fertilizer applications after each rainfall event. Rainfall that is moderate and falls at the right moment can help nutrients penetrate the soil's rooting zone and dissolve dry fertilizer. However, too much rain can increase the possibility of runoff and the pace at which nutrients like nitrogen (N) which is quintessential, phosphorus (P), and potassium (K) which are crucial, manganese (Mn), and boron (B) that are present in the soil. This research presents nutrient recommendations using an updated iteration of the random forest algorithm which is based on time-series data to forecast the required quantity of nutrients for various crops by examining rainfall patterns and crop fertility. The method suggested in this study, comes in handy for improving soil fertility by providing nutrients recommendations for optimum conditions for crop growth and reducing leaching and runoff potential.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Agriculture plays a very important role in national economic growth. Agriculture contributes 17-18% to India's GDP and ranks second worldwide in farm outputs. Plants need fertilizers and fertilizers replace the nutrients which crops take from the top layer of the soil. The absence of fertilizers can cause a drastic reduction in the volume of crop output. But fertilization requires precise action. Rainfall patterns and the amount of nutrients needed for a certain crop must be considered when using fertilizers. Machine learning is the current technology that can solve this problem by using available data for crop fertility and rainfall. Farmers can greatly benefit from the support of robust information about crops. The proposed model also uses a machine learning algorithm (random forest algorithm with k-fold cross validation technique) and takes two inputs from the user that are crop and location. After applying the algorithm, the model predicts the amount of nutrients required along with the best time to use fertilizers. The website is built using Flask Python (web framework) to provide access on all platforms and can be shared among users.

1.2 PROBLEM DEFINITION

The main motive behind Eco-Fertilization is to reduce farmers losses by providing useful insights about the amount and use of fertilizers, and to reduce water pollution by slowing down the process of leaching. It serves as a link between farmers and modern technology and enables them to increase yields while using less inputs. The system is designed as a website to provide platform-independent functionality, so that the user can access it from any device. The user interface has been kept simple with more emphasis on functionality and can be used by any naive user. It takes inputs such as crop, state and city using the drop-down menus provided in the website and applies ma In modern agriculture, efficient fertilization is crucial to maintaining soil health and maximizing crop output. However, traditional methods of determining fertilizer use often fail to account for essential factors like rainfall patterns and specific nutrient requirements for different crops and regions. This inefficiency can lead to either under-fertilization, which reduces crop yield, or over-fertilization, which harms the environment and depletes soil quality.

To optimize fertilizer use, there is a need for a system that provides accurate recommendations based on crop type, location, and environmental factors such as rainfall. Machine learning offers a powerful solution by analyzing large datasets to predict the precise nutrient requirements and optimal fertilization timing for various crops.

CHAPTER 2

LITERATURE SURVEY

Fertilizer Usage Prediction

The management of fertilizer usage is essential in modern agriculture to balance between achieving high crop yields and preventing environmental degradation due to over-fertilization. Fertilizer application involves not only supplying nutrients to plants but also ensuring that these nutrients are applied in quantities that prevent toxicity and deficiency. Paper [1] emphasizes that optimizing fertilizer usage leads to better yield with minimal waste. Over-application of fertilizers can lead to nutrient leaching, pollution of water bodies, and reduced soil fertility over time. Conversely, under-application can result in nutrient deficiencies, stunted growth, and low yields. By predicting the optimal amount of fertilizer needed for different crops and soils, farmers can avoid these negative consequences while maximizing production.

Fuzzy logic systems, as proposed in [2], are a promising approach to managing fertilizer use. These systems take into account the imprecise and variable nature of agricultural conditions—such as soil type, crop type, and environmental conditions—by using rule-based logic that mimics human reasoning. This approach allows for more flexible and adaptive fertilizer application strategies, reducing overall usage while maintaining or improving productivity.

However, fertilizer efficiency can be compromised by various factors. Paper [10] explores the limitations posed by soil compaction, which restricts root growth and nutrient uptake. This phenomenon can significantly reduce the effectiveness of fertilizers, even if they are applied optimally. Therefore, improving soil management practices, such as tilling and aeration, is equally important as managing fertilizer inputs.

Papers [11] and [4] focus on establishing a quantifiable relationship between fertilizer use, agricultural yield, and environmental factors like nitrogen and phosphate availability. By creating mathematical models that link these variables, researchers can provide farmers with data-driven recommendations for fertilizer application. These models can take into account the nitrogen need of crops, the remnant nitrate levels in the soil, and the potential environmental impacts of different fertilizer regimes. The ability to predict these outcomes can lead to more sustainable farming practices by reducing both economic and environmental costs associated with fertilizer use.

Challenges in Crop Yield Prediction

Predicting crop yield is a complex task due to the myriad of factors influencing plant growth, including unpredictable weather patterns, soil conditions, and pest pressure. Papers [3] and [6] highlight that crop yield is highly sensitive to variations in rainfall and temperature, which makes accurate prediction challenging. Rainfall is particularly critical in rain-fed agricultural systems, where the amount and distribution of rainfall can directly affect soil moisture levels, plant growth, and final yields. As climate change leads to more erratic weather patterns, traditional methods of predicting crop yield, which often rely on historical data, may become less reliable.

To address these challenges, researchers have turned to advanced data mining techniques. Paper [3] discusses the use of data mining algorithms that analyze large datasets of historical crop yields, weather patterns, and soil characteristics to identify trends and patterns. These techniques enable more accurate yield predictions by incorporating a wide range of variables and learning from past outcomes. By applying predictive models to real-time data, farmers can make more informed decisions about planting, irrigation, and fertilization, improving the resilience of their operations in the face of environmental variability.

Papers [13] and [6] further analyze the relationship between specific environmental factors—such as rainfall—and crop yield. They propose models that forecast yields based on the amount and distribution of rainfall received during the growing season. These models can be particularly useful in regions that rely heavily on rain-fed agriculture, where water availability is often the limiting factor for crop growth. By providing region-specific predictions, these studies offer practical tools for farmers to plan their planting schedules and resource allocation more effectively.

Soil Health and Fertility

Soil health is a fundamental component of sustainable agriculture, as it directly impacts crop productivity, water retention, and nutrient cycling. The quality of soil can be influenced by a range of factors, including long-term management practices, fertilizer use, and environmental conditions. Paper [7] introduces a novel metric for assessing soil health, which includes indicators such as organic matter content, microbial activity, and soil structure. This metric allows for a more holistic evaluation of soil quality, going beyond simple chemical analyses to include biological and physical factors that are essential for maintaining soil fertility over the long term.

Preparation history, including the type and frequency of fertilizer application, also plays a critical role in shaping soil health. Paper [8] examines how long-term fertilization and rainfall patterns affect the resistance of soil microorganisms to environmental stresses. Soil microorganisms are vital for nutrient cycling, decomposition of organic matter, and overall soil fertility. Understanding how these organisms respond to different management practices can help farmers adopt more sustainable approaches to soil management.

Nitrogen leaching, which occurs when excess nitrogen from fertilizers is washed out of the soil and into groundwater, is another major concern for soil health and environmental sustainability. Paper [5] discusses the risk of nitrogen leaching in areas with no-till management practices, which are often adopted to reduce soil erosion and improve water retention. However, no-till systems can also lead to nutrient stratification, where nutrients accumulate in the upper soil layers, increasing the risk of leaching. Paper [14] offers solutions to these challenges by proposing supervised algorithms that optimize fertilizer application based on soil parameters. These algorithms help to balance nutrient inputs with plant needs, reducing the risk of leaching while maintaining high crop yields.

Machine Learning and Advanced Techniques in Agriculture

The advent of machine learning and artificial intelligence (AI) has brought significant advancements to agricultural research and practice. Several papers in this review explore the application of these technologies for improving crop management, predicting yields, and optimizing resource use. Paper [12] suggests the use of transfer learning techniques, where models trained on one dataset are adapted to new datasets, to improve the accuracy of crop yield predictions. This approach is particularly useful in agriculture, where data availability can be limited or inconsistent across regions and crops.

Paper [16] demonstrates the utility of machine learning models in long-term fertilizer studies. By analyzing large datasets of crop responses to different fertilizer regimes, machine learning algorithms can identify the most effective strategies for maximizing yields while minimizing inputs. These models can also be applied to other long-term agricultural experiments, providing valuable insights into sustainable farming practices.

Decision-based systems, as proposed in [17] and [18], offer another avenue for integrating AI into agriculture. These systems use data from various sources—such as climate data, crop characteristics, and pest information—to assist farmers in making informed decisions about planting, fertilization, and pest control. By automating the decision-making process, these systems can help farmers optimize their operations, reduce costs, and improve productivity.

Paper [19] explores the use of the Extreme Learning Machine (ELM) technique for soil nutrient classification. ELM is a rapid learning algorithm that can process large amounts of data efficiently, making it well-suited for agricultural applications where timely decisions are crucial. By classifying soil nutrients based on their chemical properties, ELM can help farmers tailor their fertilizer application to the specific needs of their soil, improving nutrient use efficiency and reducing environmental impacts.

Crop Disease Prediction

Crop diseases are one of the primary factors that can negatively impact agricultural productivity. Early detection and prevention are key to minimizing the damage caused by diseases, which can spread rapidly and devastate entire crops. Paper [15] addresses this issue by proposing an IoT-based apple disease prediction model in the Kashmir Valley. This model uses sensors and data analysis techniques to monitor environmental conditions and detect early signs of disease outbreaks. By integrating modern technology with traditional agricultural practices, the model helps farmers respond more quickly to disease threats, reducing crop losses and improving overall yield.

The integration of IoT systems with machine learning algorithms enables real-time monitoring and prediction of crop diseases. These systems can continuously collect data on temperature, humidity, soil moisture, and other factors that influence disease development. By analyzing this data, machine learning models can identify patterns that indicate the onset of disease, allowing for timely intervention. This approach not only improves disease management but also reduces the need for chemical pesticides, contributing to more sustainable agricultural practices.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The current agricultural practices for determining crop nutrient needs are predominantly based on traditional methods such as soil testing and generalized recommendations from agronomists. Farmers often rely on laboratory analysis of soil samples to measure essential nutrients like nitrogen (N), phosphorus (P), and potassium (K), along with pH levels. Although soil tests provide a baseline understanding of the soil's nutrient composition, they represent a static, one-time snapshot, which may not reflect real-time conditions.

Moreover, these recommendations are not tailored to specific farms but are based on average local data. Factors like seasonal nutrient leaching, variations in weather patterns, and localized climate conditions are not adequately addressed. Farmers also use weather data based on historical trends rather than real-time updates, which reduces the accuracy of fertilization timing. As a result, nutrient management becomes suboptimal, leading to problems such as under-application or over-application of fertilizers. Manual data entry introduces another layer of error, contributing to inconsistent crop yields and environmental challenges, such as nutrient runoff and soil degradation.

3.2 DISADVANTAGES

Limited Specificity: Soil tests provide only a one-time assessment, making it difficult to adapt to changing environmental conditions.

Generalized Recommendations: Agronomic advice is often based on broad, regional data, which may not apply to a farm's unique needs.

Human Error: Manual processes for tracking crop yields, weather conditions, and nutrient applications increase the risk of inaccuracies.

Lack of Real-Time Data: Existing methods do not account for real-time weather variations, potentially leading to poor fertilization timing and inefficient use of resources.

Environmental Impact: Over-application or under-application of nutrients can harm both crops and the environment by causing nutrient runoff, soil degradation, and inefficient resource use.

3.3. PROPOSED SYSTEM

The proposed system leverages machine learning, specifically a Random Forest model with K-Fold Cross Validation, to predict optimal nutrient requirements based on both crop type and environmental conditions. This approach allows for real-time data-driven insights, enhancing the precision and efficacy of nutrient application. By integrating weather forecasts via a Weather API, the system evaluates conditions such as temperature, humidity, and rainfall, issuing alerts if heavy rainfall is anticipated. This helps reduce nutrient loss due to leaching and runoff.

The model requires user inputs such as crop type and location, which are then used to provide personalized recommendations for nutrient applications (specifically nitrogen, phosphorus, and potassium levels). The platform, built using the Flask Python framework, is web-based, ensuring accessibility across devices. This system represents a significant improvement over traditional methods by providing timely, location-specific guidance for fertilization, ultimately increasing crop yield while reducing environmental impact.

3.4 ADVANTAGES

Real-Time Data Integration: The proposed system incorporates real-time weather data via a Weather API, allowing for more accurate nutrient recommendations tailored to current conditions such as temperature, rainfall, and humidity.

Personalized Recommendations: By using machine learning algorithms like Random Forest, the system generates personalized nutrient recommendations based on crop type and location, optimizing nutrient application.

Machine Learning-Based Insights: The use of Random Forest with K-Fold Cross Validation enhances prediction accuracy, ensuring that farmers receive optimal nutrient management guidance.

Automation and Precision: Automating the process reduces the risk of human error, leading to better data accuracy and improved nutrient management.

Environmental Sustainability: By reducing unnecessary fertilizer usage and mitigating nutrient runoff, the system minimizes environmental harm and promotes sustainable farming practices.

Accessibility: As a web-based platform developed using Flask, the system is accessible from various devices, enabling widespread usage and easy access to recommendations.

3.5 HARDWARE ENVIRONMENT

- Processor: Intel(R) Core (TM) i3-4005U CPU @ 1.70GHz
- RAM: 4.00 GB
- System type: 64-bit operating system, x64-based processor
- Network Interface Card
- Keyboard
- Mouse

3.6 SOFTWARE ENVIRONMENT

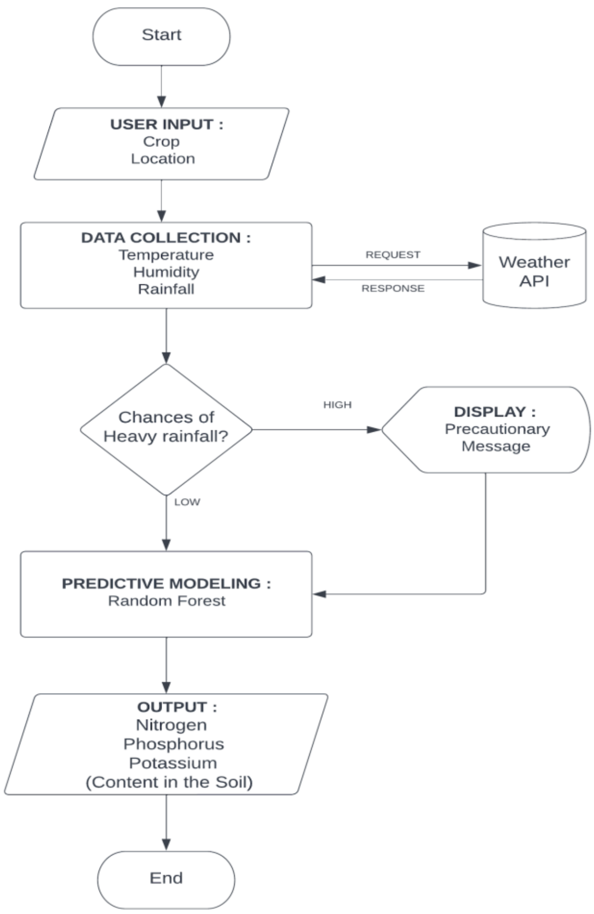
- Operating System (any)
- Google Chrome (web browser)
- Visual Studio Code
- Jupyter Notebook

CHAPTER 4

SYSTEM DESIGN

4.1 Data Flow Diagram

A data flow diagram is a visual representation of how data "flows" throughout a data system, simulating certain features of its operation. It is frequently used as an initial stage to develop, without going into great depth, an overview of the system that may then be expanded upon. They may also be utilized to display data processing.

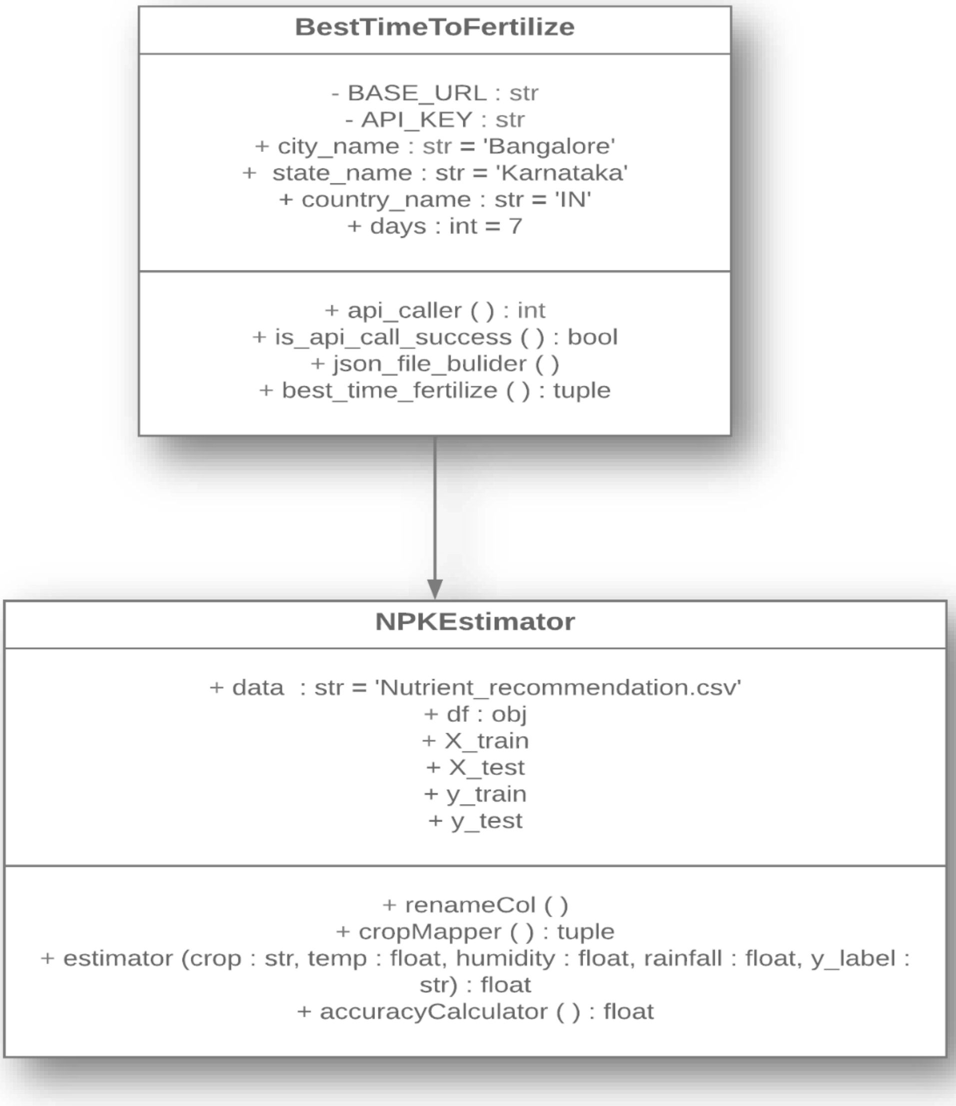


4.2 UML DIAGRAM

4.2.1 CLASS DIAGRAM

Static diagrams include class diagrams. It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualizing, explaining, and documenting various elements of systems.

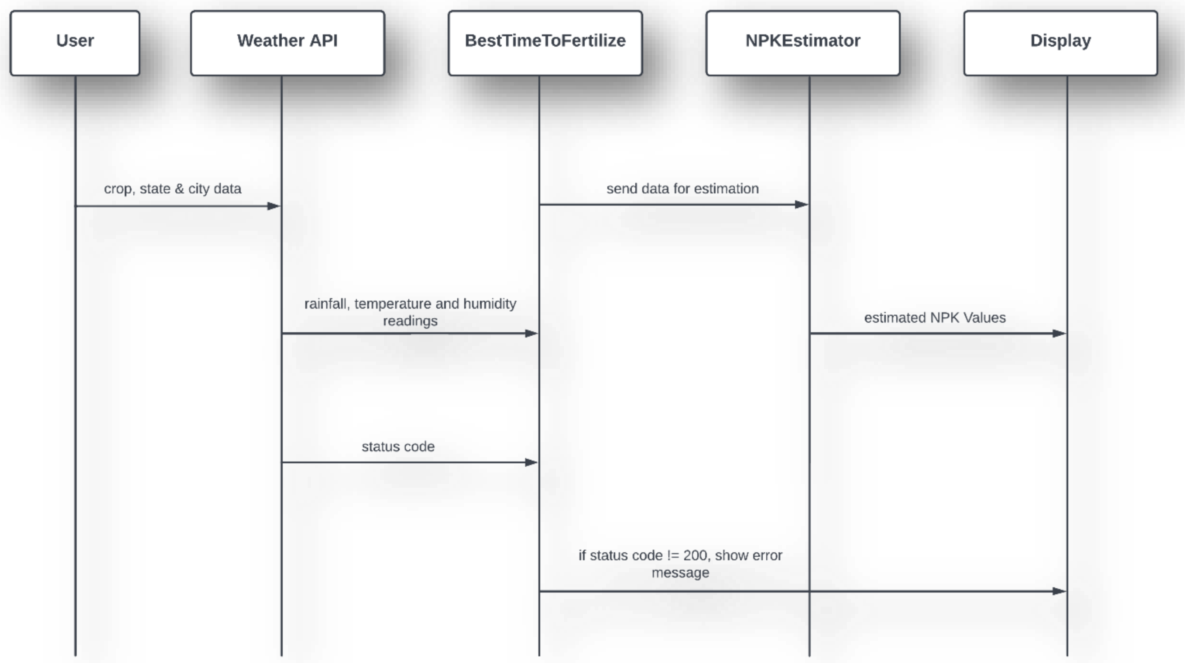
A collection of classes, interfaces, affiliations, collaborations, and constraints are displayed in a class diagram. A structural diagram is another name for it.



4.2.2 SEQUENCE DIAGRAM

Object interactions are arranged in temporal sequence in a sequence diagram. It shows the classes and objects engaged in the scenario as well as the flow of messages that must be exchanged for the objects to work as intended. Inside the logical view of the system being developed, sequence diagrams are often connected to use case realizations. Event diagrams and event scenarios are other names for sequence diagrams.

A sequence diagram is made up of vertical parallel lines (called "lifelines") that represent several processes or things that exist at the same time and horizontal arrows that represent the messages sent between them in the chronological order in which they take place. This enables the graphical specification of straightforward runtime scenarios.

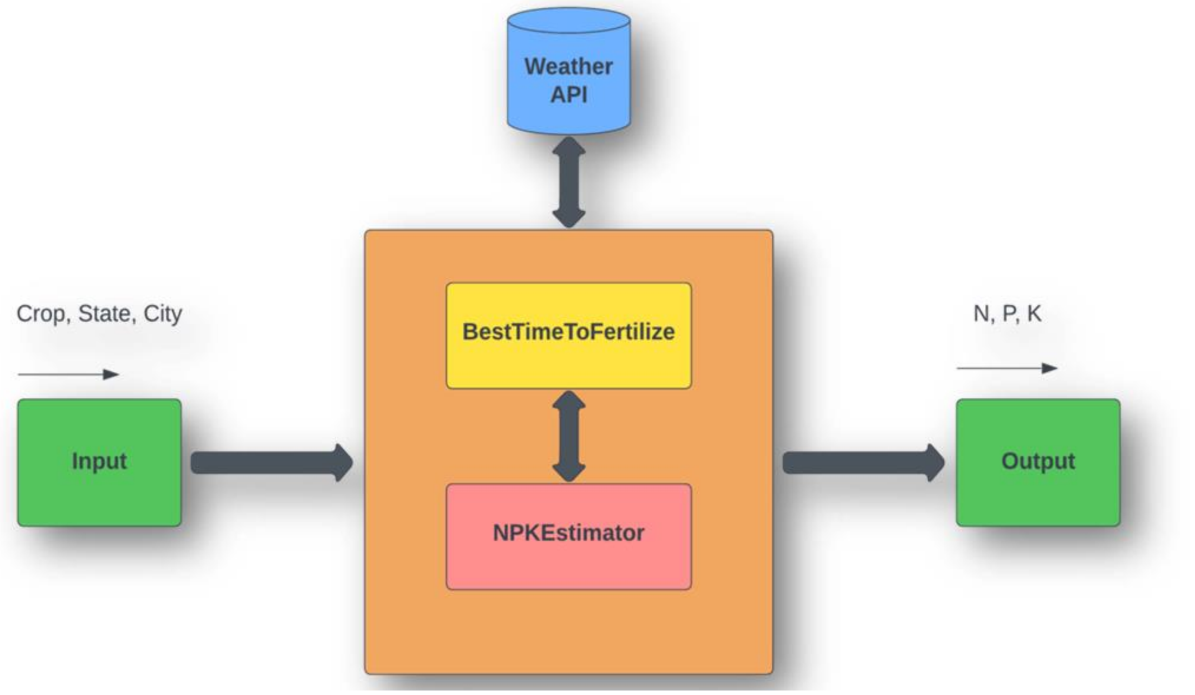


CHAPTER 5

SYSTEM ARCHITECTURE

5.1 ARCHITECTURAL DIAGRAM

A conceptual model known as system architecture describes the structure and behaviour of the system. It consists of the system's elements and the connections between them that explain how the whole system is implemented. The figure below shows the system's architecture and the various components added to them.



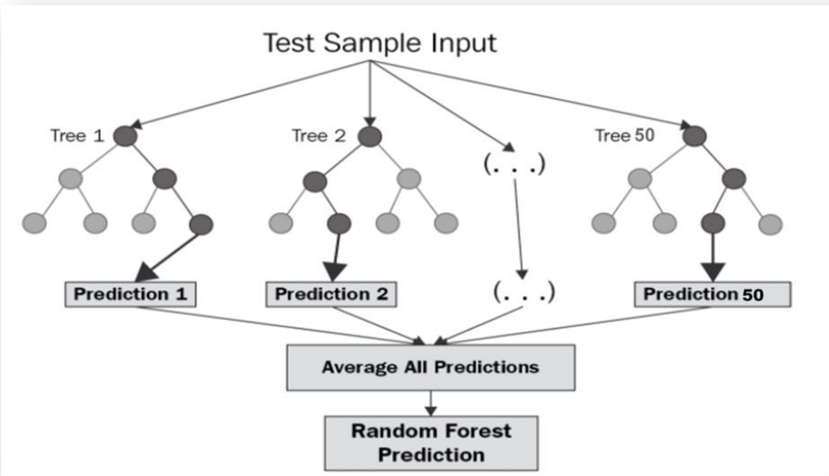
The description of each component from the block diagram above and their major functionalities with respect to the Eco-Fertilization as a complete unit is described in the table below.

Sl No.	Block Name	Functions
1	Input	User provides data such as crop, state and city using drop-down menu.
2	Weather API	Weather details like temerature, humidity, rainfall etc. is fetched from the weather API.
3	BestTimeToFertilize	This module provides the functionality to determine the best time to fertilize using fetched weather data and provides warning for heavy rain.
4	NPKEstimator	This module estimates the required ratio of NPK contents in the soil.
5	Output	Nitrogen, Phosphorus and Potassium content displayed on the website.

5.2 ALGORITHMS

5.2.1 RANDOM FOREST REGRESSION

A group of several decision trees called a random forest (RF) are trained using different subsets of data and have changeable hyper-parameters. In our project, we are going to take crop and location as input, and based on it, we will predict the value of N, P, and K. First, we will divide our dataset into training and test datasets, where the training dataset is 80% of the original data and the rest 20% is test data. Then we will create three different random forests of size 50 (decision tree) for each N, P, and K and produces the average of the classes as the overall tree projection.



CHAPTER 6

SYSTEM IMPLEMENTATION

System implementation builds system pieces that adhere to user requirements of the system requirements founded in the early life cycle stages using the framework generated throughout architectural design and the outcomes of system analysis. These system components are then combined to create intermediate aggregates, which ultimately result in the entire system-of-interest (SoI). The system hierarchy's lowest-level system components are really produced by the implementation process (system breakdown structure). System components are created, purchased, or recycled. Production includes the forming, removing, connecting, and finishing processes used in hardware fabrication, the coding and testing processes used in software realization, or the processes used to build operating procedures for the duties of operators.

A design method known as "modular design," sometimes known as "modularity in design," separates a system into smaller components known as modules or skids that may be independently produced and then used in multiple systems. Functional division into distinct, scalable, reusable modules; strict usage of very well modular interfaces; and adherence to industry norms for interfaces are characteristics of a modular system.

6.1 RANDOM FOREST REGRESSION

A group of several decision trees called a random forest (RF) are trained using different subsets of data and have changeable hyper-parameters. In our project, we are going to take crop and location as input, and based on it, we will predict the value of N, P, and K. First, we will divide our dataset into training and test datasets, where the training dataset is 80% of the original data and the rest 20% is test data. Then we will create three different random forests of size 50 (decision tree) for each N, P, and K and produces the average of the classes as the overall tree projection, shown in Table 6.1.

Fig.6.1.1: Random Forest Regression

BEGIN:
Step 1: The dataset of size n = 2200 is divided into training and test dataset (where the raining set is 80% and the test set is 20% that is training set=1,760 and the test set=240).
Step 2: Apply random forest regression to each N, P and K (Nitrogen, Phosphorus & Potassium) value with n estimators=50 (n estimators is the number of decision trees).
Step 3: Train the N Label, P Label and K Label with the training dataset and dependent variable (Where the dependent variable is N for N Label, P for P Label and K for K Label).
Step 4: Each N Label, P Label and K Label generates a 50-decision tree as an output based on the training dataset.
END

Table 6.1: Random Forest Regression Algorithm

Why we selected 50 decision trees (n_estimator = 50) for each label?

We have tested for different n_estimator values, but the upmost accuracy achieved for N_Label is 0.87 for two decimal digit precision. As shown in below figure Fig 6.1.1.

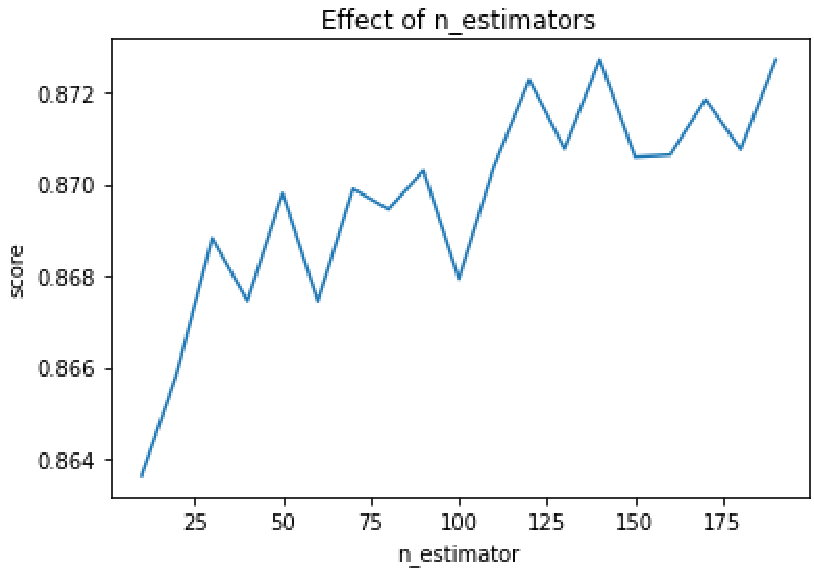


Fig 6.1.2: Effect of n_estimator

Why is a Random Forest chosen instead of a Decision Tree for this project?

Decision trees are trees that show all possible consequences of a selection using a branched technique, which is a key distinction between them and the random forest algorithm. In contrast, a set of decision trees that follow the output are produced by the random forest method.

In general, adding more trees will increase performance and predictability while decreasing calculation speed. The end solution for regression issues is the mean of all the trees. The samples in the tree target cell is the initial level of means in a random forest method regression model, followed by all trees. In contrast to linear regression, it estimates values beyond the observed range using prior observations.

The accuracy in the decision tree depends on the number of right predictions made divided by the total number of predictions, since it uses huge value attributes at each node, and it produces less accurate results when we apply an algorithm to handle the regression problem in a random forest. Decision trees are greedy and may be deterministic, meaning they produce different answers if we add or remove any additional rows. So, compared to decision trees, random forest forecasts outcomes with higher accuracy.

The main problem with machine learning is overfitting. Overfitting may be viewed as a generic bottleneck in machine learning and occurs when we apply algorithms. When machine learning models are unable to perform well on unknown datasets, this is a sign of overfitting. This is especially true if the problem is detected mostly on testing or validation datasets and is significantly larger than the error on the training dataset. Overfitting occurs when models gain knowledge non - constant data in the training data, which has a negative effect on the performance on the new data model. Due to the employment of several decision trees in the random forest, the danger of overfitting is lower than that of the decision tree.

The accuracy increases when we employ a decision tree classifier on a data set since it contains more splits, which makes it easier to overfit the dataset and validate it.

So, that's why we decided to select random forest as our machine learning model rather than decision tree to predict the required nutrients (Nitrogen, Phosphorus, Potassium) for the given crop. Random Forest performs well in terms of computation if we adjust the n_estimator value carefully. In our case we have used n_estimator = 50 after taking readings of our model's accuracy with different n_estimator values, same is shown in Fig 6.1.2.

6.2 CROSS VALIDATION

In order to evaluate machine learning algorithm on a small set of data, cross-validation is a re - sampling technique. The algorithm's sole parameter, k, indicates how many groups should be formed from a given data sample. As a result, the technique is frequently referred to as k-fold cross-validation. When a precise value for k is given, it can be substituted for k in the model's regard, such as k=4 for cross-validation that is performed four times.

In applied machine learning, cross-validation is mostly used to gauge how well a machine learning model performs on untrained data. That is, to use a small sample to assess how the model will generally perform when used to generate predictions on data that was not utilized during the model's training.

It is a well-liked technique since it is easy to comprehend and typically yields a less biased or overly optimistic assessment of the model ability than other techniques, including a straightforward train/test split.

Following is the general process:

1. Randomly shuffle the dataset.
2. Create k groups from the dataset.
3. For every distinct group:
    a) The group should be used as a holdout or test data set.
    b) As a training dataset contains, use the remaining groupings.
    c) Adapt a model to the training set, then evaluate it against the test set.
    d) Keep the evaluation result, but discard the model.
4. Using a sample of quality assessment ratings, summaries the model's skill.

It's significant that every observation in the sample data is given a unique group and remains there throughout the process. This indicates that each sample has the chance to be used k times to train the model and k times in the hold out set.

It is crucial that all data preparation done before fitting the model takes place on the loop's CV-assigned training dataset rather than the larger data set. This also holds true for any hyperparameter adjustment. Data leakage and an exaggerated assessment of the model's skill may occur from failing to carry out these procedures within the loop.

The mean of the model skill scores is frequently used to sum up the outcomes of a k-fold cross-validation run. A measure of the skill scores' volatility, like the standard error or standard deviation, should also be included.

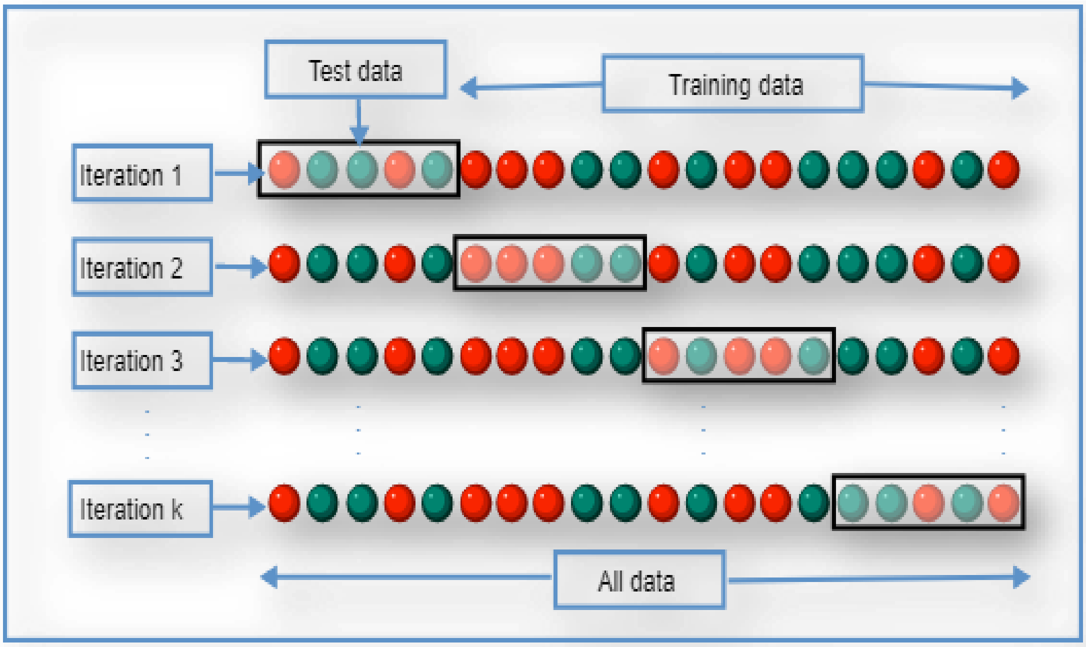


Fig.6.2.1: K-Cross Fold Validation

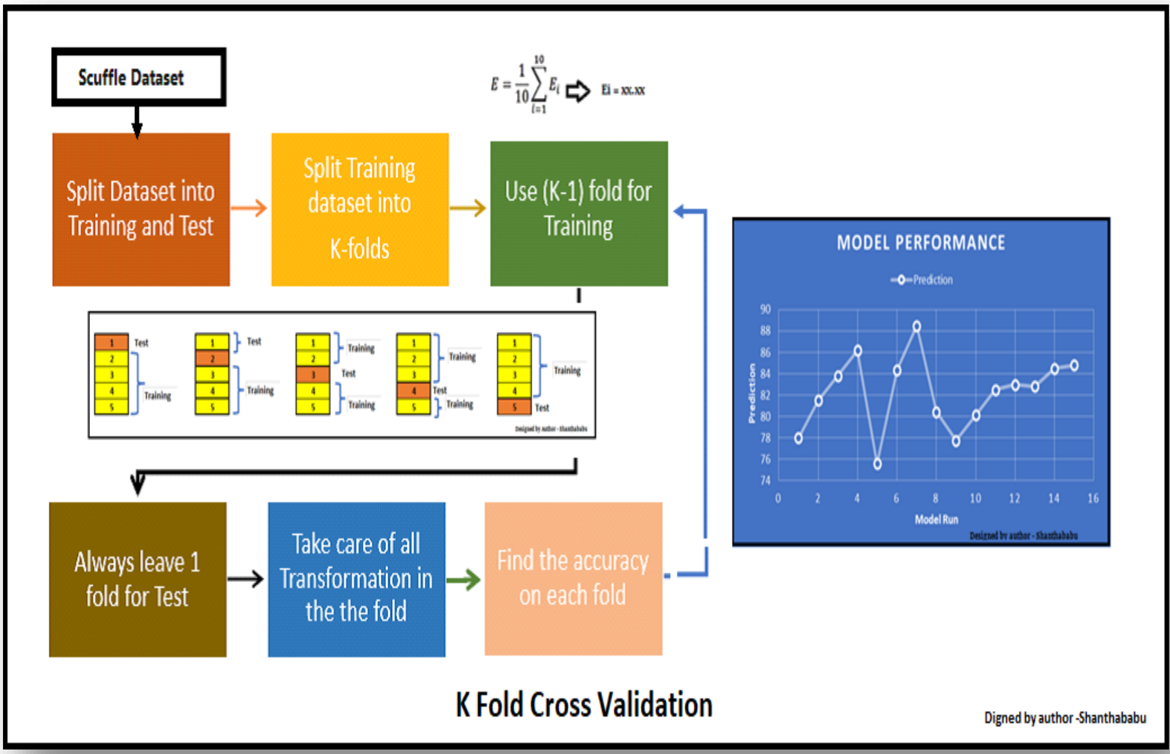


Fig 6.2.2: Cross-Validation procedure

6.3 DATASET

Dataset used in our proposed system:

	A	B	C	D	E	F	G	H
1	**N**	**P**	**K**	**temperature**	**humidity**	**ph**	**rainfall**	**label**
2	90	42	43	20.8797437	82.0027442	6.50298529	202.935536	rice
3	85	58	41	21.7704617	80.3196441	7.03809636	226.655537	rice
4	60	55	44	23.0044592	82.3207629	7.84020714	263.964248	rice
5	74	35	40	26.4910964	80.1583626	6.98040091	242.864034	rice
6	78	42	42	20.1301748	81.6048729	7.62847289	262.717341	rice
7	69	37	42	23.0580487	83.3701177	7.0734535	251.055	rice
8	69	55	38	22.708838	82.6394139	5.70080568	271.32486	rice
9	94	53	40	20.2777436	82.8940862	5.71862718	241.974195	rice
10	89	54	38	24.5158807	83.5352163	6.68534642	230.446236	rice
11	68	58	38	23.2239739	83.0332269	6.33625353	221.209196	rice
12	91	53	40	26.5272351	81.4175385	5.38616779	264.61487	rice
13	90	46	42	23.9789822	81.450616	7.50283396	250.083234	rice
14	78	58	44	26.800796	80.8868482	5.10868179	284.436457	rice
15	93	56	36	24.0149762	82.0568718	6.98435366	185.277339	rice
16	94	50	37	25.6658521	80.6638505	6.94801983	209.586971	rice
17	60	48	39	24.2820942	80.3002559	7.04229907	231.086335	rice
18	85	38	41	21.5871178	82.7883708	6.24905066	276.655246	rice
19	91	35	39	23.7939196	80.4181796	6.97085975	206.261186	rice
20	77	38	36	21.8652524	80.1923008	5.95393328	224.555017	rice
21	88	35	40	23.5794363	83.5876032	5.85393208	291.298662	rice
22	89	45	36	21.3250416	80.474764	6.44247538	185.497473	rice
23	76	40	43	25.1574553	83.1171348	5.07017567	231.384316	rice
24	67	59	41	21.9476674	80.973842	6.01263259	213.356092	rice
25	83	41	43	21.0525355	82.6783952	6.25402845	233.107582	rice
26	98	47	37	23.4838134	81.3326507	7.37548285	224.058116	rice
27	66	53	41	25.0756354	80.5238915	7.77891515	257.003887	rice
28	97	59	43	26.3592716	84.0440359	6.28650018	271.358614	rice
29	97	50	41	24.5292268	80.5449858	7.07096	260.263403	rice
30	60	49	44	20.7757615	84.497744	6.24484149	240.081065	rice

Fig.6.3: Actual Dataset

Crop Recommendation Dataset [20]

6.4 DATA PREPARATION

Actual dataset contains eight features. All of the features are not useful for the proposed model. Therefore, a dimension reduction technique called feature selection is applied and seven features, then selected for evaluation.

	A	B	C	D	E	F	G
1	**Crop**	**Temperature**	**Humidity**	**Rainfall**	**Label_N**	**Label_P**	**Label_K**
2	rice	20.87974371	82.00274423	202.9355362	90	42	43
3	rice	21.77046169	80.31964408	226.6555374	85	58	41
4	rice	23.00445915	82.3207629	263.9642476	60	55	44
5	rice	26.49109635	80.15836264	242.8640342	74	35	40
6	rice	20.13017482	81.60487287	262.7173405	78	42	42
7	rice	23.05804872	83.37011772	251.0549998	69	37	42
8	rice	22.70883798	82.63941394	271.3248604	69	55	38
9	rice	20.27774362	82.89408619	241.9741949	94	53	40
10	rice	24.51588066	83.5352163	230.4462359	89	54	38
11	rice	23.22397386	83.03322691	221.2091958	68	58	38
12	rice	26.52723513	81.41753846	264.6148697	91	53	40
13	rice	23.97898217	81.45061596	250.0832336	90	46	42
14	rice	26.80079604	80.88684822	284.4364567	78	58	44
15	rice	24.01497622	82.05687182	185.2773389	93	56	36
16	rice	25.66585205	80.66385045	209.5869708	94	50	37
17	rice	24.28209415	80.30025587	231.0863347	60	48	39
18	rice	21.58711777	82.7883708	276.6552459	85	38	41
19	rice	23.79391957	80.41817957	206.2611855	91	35	39
20	rice	21.8652524	80.1923008	224.5550169	77	38	36
21	rice	23.57943626	83.58760316	291.2986618	88	35	40
22	rice	21.32504158	80.47476396	185.4974732	89	45	36
23	rice	25.15745531	83.11713476	231.3843163	76	40	43
24	rice	21.94766735	80.97384195	213.3560921	67	59	41
25	rice	21.0525355	82.67839517	233.1075816	83	41	43
26	rice	23.48381344	81.33265073	224.0581164	98	47	37
27	rice	25.0756354	80.52389148	257.0038865	66	53	41
28	rice	26.35927159	84.04403589	271.3586137	97	59	43
29	rice	24.52922681	80.54498576	260.2634026	97	50	41
30	rice	20.77576147	84.49774397	240.0810647	60	49	44

Fig 6.4: Customized Dataset

6.5 Input Features

Below are the input features of our system:

- **Crop**: rice, cotton, mango, orange, lentil, etc.
- **Temperature**: temperature measured in Celsius
- **Humidity**: measured relatively in percentages
- **Rainfall**: rainfall in mm

6.6 Output Features

Below are the output features of our system:

- **Label_N:** ratio of soil Nitrogen content
- **Label_P:** ratio of soil Phosphorus content
- **Label_K:** ratio of soil Potassium content

CHAPTER 7

SYSTEM TESTING

7.1 BLACK BOX TESTING

By giving the crop name,city name and state name we get to know the nutrition values as well as the weather report for the next 7 days.

```
Enter crop name: rice
Enter city name: karnataka
Enter state name: bangalore
Precipitation over 2 days: 0.0859375 mm, Chance: 10%

7-Day Weather Report:
Date: 2024-10-11, Temperature: 29.3°C, Relative Humidity: 78%, Rainfall: 0.0859375 mm, Probability of Precipitation: 20%
Date: 2024-10-12, Temperature: 29.6°C, Relative Humidity: 76%, Rainfall: 0 mm, Probability of Precipitation: 0%
Date: 2024-10-13, Temperature: 29.6°C, Relative Humidity: 75%, Rainfall: 0.015625 mm, Probability of Precipitation: 20%
Date: 2024-10-14, Temperature: 29.5°C, Relative Humidity: 75%, Rainfall: 0.11720276 mm, Probability of Precipitation: 20%
Date: 2024-10-15, Temperature: 29.8°C, Relative Humidity: 76%, Rainfall: 0.04686737 mm, Probability of Precipitation: 20%
Date: 2024-10-16, Temperature: 27.3°C, Relative Humidity: 75%, Rainfall: 0.0625 mm, Probability of Precipitation: 20%
Date: 2024-10-17, Temperature: 27°C, Relative Humidity: 82%, Rainfall: 8.875 mm, Probability of Precipitation: 75%

Nutrition Recommendation for rice:
N: 100.14, P: 22.06, K: 48.56
```

7.2 WHITE BOX TESTING

The weather report is taken by making an api call to the weather API and the nutrition values are received from dataset training.


```
Enter crop name: rice
Enter city name: karnataka
Enter state name: bangalore
Precipitation over 2 days: 0.0859375 mm, Chance: 10%

7-Day Weather Report:
Date: 2024-10-11, Temperature: 29.3°C, Relative Humidity: 78%, Rainfall: 0.0859375 mm, Probability of Precipitation: 20%
Date: 2024-10-12, Temperature: 29.6°C, Relative Humidity: 76%, Rainfall: 0 mm, Probability of Precipitation: 0%
Date: 2024-10-13, Temperature: 29.6°C, Relative Humidity: 75%, Rainfall: 0.015625 mm, Probability of Precipitation: 20%
Date: 2024-10-14, Temperature: 29.5°C, Relative Humidity: 75%, Rainfall: 0.11720276 mm, Probability of Precipitation: 20%
Date: 2024-10-15, Temperature: 29.8°C, Relative Humidity: 76%, Rainfall: 0.04686737 mm, Probability of Precipitation: 20%
Date: 2024-10-16, Temperature: 27.3°C, Relative Humidity: 75%, Rainfall: 0.0625 mm, Probability of Precipitation: 20%
Date: 2024-10-17, Temperature: 27°C, Relative Humidity: 82%, Rainfall: 8.875 mm, Probability of Precipitation: 75%

Nutrition Recommendation for rice:
N: 100.14, P: 22.06, K: 48.56
```

TEST CASE ID	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
01	Give state name,city name and crop name	7-day weather report and nutrition values	As expected.	Pass

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

The proposed system is able to achieve 92% of accuracy, which is quite good for any predictive model. It provides information about the use and the amount of nutrients required by the crops for satisfactory crop growth and production with respect to weather conditions. It provides weather alerts and messages. Alerts are displayed in the output of this application in case of bad weather conditions. The accuracy can be improved further with development in technologies.

8.2 APPLICATION

- This project is useful in the agriculture sector.
- Can be used to reduce the wastage of fertilizers.
- Used to suggest nutrient recommendations for the crops.
- It reduces water pollution by slowing down the process of soil leaching as fertilizers can reach the water table and contaminate shallow groundwater and deep aquifers.
- Reduce leaching and runoff potential.
- It provides weather alerts and messages. Alerts are displayed in the output of this application in case of bad weather conditions.
- Seven-day weather forecasts to timely plan the fertilization.

8.3 FUTURE WORKS

The proposed system provides a helping hand to our farmers. It gives information about the use and quantity of nutrients required by the crops. There is scope for improvement in the system by providing user interface in the native language, so that the user can operate the system easily if he or she is unfamiliar with the English language. In addition, speech recognition systems can be added to handle illiterate users.

CHAPTER 9

APPENDIX 1

9.1 SAMPLE CODING

```
import warnings

import numpy as np

import pandas as pd

from sklearn import metrics

import category_encoders as ce

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestRegressor


warnings.filterwarnings('ignore')



class NPKEstimator:

    def __init__(self, data = 'Nutrient_recommendation.csv', ):

        self.df = pd.read_csv(data, header=None)

        self.X_train = None

        self.X_test = None

        self.y_train = None

        self.y_test = None
```

```
    def renameCol(self):
        self.df.columns = ['Crop', 'Temperature', 'Humidity', 'Rainfall', 'Label_N', 'Label_P', 'Label_K']
        self.df.drop(self.df.index[:1], inplace=True)


    def cropMapper(self):
        # create mapping of crop(string) to int type
        mapping = dict()

        with open("mapped_crops.csv", "w") as fh:
            fh.write("Crops,Key\n")
            for i, crop in enumerate(np.unique(self.df[['Crop']]), 1):
                mapping[crop] =  i
                fh.write("%s,%d\n" % (crop, i))
            mapping['NA'] = np.nan
            fh.write("NA,nan")
        # print(mapping)

        ordinal_cols_mapping = [{"col": "Crop", "mapping": mapping}, ]
        encoder = ce.OrdinalEncoder(cols = 'Crop', mapping = ordinal_cols_mapping, return_df = True)
        return mapping, encoder
```

```
    def estimator(self, crop, temp, humidity, rainfall, y_label):
        X = self.df.drop(['Label_N', 'Label_P', 'Label_K'], axis=1)
        y = self.df[y_label]

        self.X_train, self.X_test, self.y_train, self.y_test = train_test_split(X, y, test_size = 0.20,
random_state = 42)

        mapping, encoder = self.cropMapper()
        self.X_train = encoder.fit_transform(self.X_train)
        self.X_test = encoder.transform(self.X_test)

        regressor = RandomForestRegressor(n_estimators = 50, random_state = 0)
        regressor.fit(self.X_train, self.y_train)

        # y_pred = regressor.predict(self.X_test)
        query = [mapping[crop.strip().lower()], temp, humidity, rainfall]
        y_pred = regressor.predict([query])
        return y_pred[0]


    def accuracyCalculator(self):
        model = RandomForestRegressor(n_jobs=-1)
        estimators = np.arange(10, 200, 10)
        scores = []
```

```
for n in estimators:
    model.set_params(n_estimators=n)
    model.fit(self.X_train, self.y_train)
    scores.append(model.score(self.X_test, self.y_test))

scores_arr = [round(sc, 3) for sc in scores]
unique, counts = np.unique(scores_arr, return_counts = True)

max_count = max(counts)
accuracy = -1
for uni, count in zip(unique, counts):
    # print(uni, count)
    if count == max_count:
        accuracy = uni

# print("Model accuracy: %.2f" % (accuracy))
return accuracy
```

10. REFERENCES

[1] Krutika Hampannavar, Vijay Bhajantri, Shashikumar G. Totad “Prediction of Crop Fertilizer Consumption,” Fourth International Conference on Computing Communication Control and Automation (ICCUBEA),2018, PP.1-5

[2] G. Prabakaran, D. Vaithiyanathan, Madhavi Ganesa, “Fuzzy decision support system for improving the crop productivity and efficient use of fertilizers,” Computers and Electronics in Agriculture, vol-150, 2018, PP. 88-97

[3] Shital Bhojani, Nirav Bhatt, “Data Mining Techniques for Crop Yield Prediction,” Computers and Electronics in Agriculture, vol-6, 2018, PP. 357-358

[4] Yulong Yin, Hao Ying, Huifang Zhen, Q ingsong Zhang, Y anfang Xue, Zhenling I, “Estimation of NPK requirements for rice production in diverse Chinese environments under optimal fertilization rate,” Agricultural and Forest Meteorology, vol-279, 2019, PP. 1-6

[5] Laura J.T. Hess, Eve-Lyn S. Hinckley, G. Philip Robertson, Pamela A. Matson, “Rainfall intensification increases nitrate leaching from tilled but not no-till cropping systems in the U.S. Midwest,” Agriculture, Ecosystems & Environment, vol-290, 2020, PP. 1-10

[6] Potnuru Sai Nishant,Pinapa Sai Venkat,Bollu Lakshmi Avinash,B. Jabber, “Crop Yield Prediction Based on Indian Agriculture using Machine Learning,” 2020 International Conference for Emerging Technology (INCET), 2020, PP. 1-4

[7] Tony Yang, Kadambot H.M., Siddique, Kui Liu, “Cropping systems in agriculture and their impact on soil health,” Global Ecology and Conservation, vol-23, year, PP. 1-13

[8] János Kátai, Ágnes Oláh Zsuposné, Magdolna Tállai, Tarek Alshaal, “Would fertilization history render the soil microbial communities and their activities more resistant to rainfall fluctuations? ,” Ecotoxicology and Environmental Safety, vol-201, 2020, PP. 1-11

[9] Usman Ahmed, Jerry Chun-Wei Lin, Gautam Srivastava, Youcef Djenouri, “A nutrient

recommendation system for soil fertilization based on Evolutionary Computation,” Computers and Electronics in Agriculture, vol-189, 2021, PP. 1-7

[10] A.Hussein, Diogenes L. Antille, Shreevatsa Kodur, GuangnanChen, Jeff N.Tullberg, “Controlled traffic farming effects on productivity of grain sorghum, rainfall and fertilizer nitrogen use efficiency,” Journal of Agriculture and Food Research, vol-3, 2021, PP. 1-17

[11] Zujiao Shi, Donghua Liu, Miao Liu, Muhammad Bilal Hafeez, Pengfei Wen, Xiaoli Wang, Rui Wang, Xudong Zhang, Jun Li, “Optimized fertilizer recommendation method for nitrate residue control in a wheat–maize double cropping system in dryland farming,” Field Crops Research, vol-271, 2021, PP. 1-10

[12] Janmejay Pant, R.P. Pant, Manoj Kumar Singh, Devesh Pratap Singh, Himanshu Pant, “Analysis of agricultural crop yield prediction using statistical techniques of machine learning,” Materials Today: Proceedings, vol-46, 2021, PP. 1-10

[13] Benny Antony, “Prediction of the production of crops with respect to rainfall.,” Environmental Research, vol-202, 2021, PP. 1-5

[14] Akash Manish Lad, K. Mani Bharathi, B. Akash Saravanan, R. Karthik, “Factors affecting agriculture and estimation of crop yield using supervised learning algorithms,” Materials Today: Proceedings, 2022, PP. 1-10

[15] Raves Akhtar, Shabbir Ahmad Sofi, “Precision agriculture using IoT data analytics and machine learning,” Journal of King Saud University - Computer and Information Sciences,

[16] Saheed Garnaik, Prasanna Kumar Samant, Mitali Mandal, Tushar Ranjan Mohanty, Sanat Kumar Dwibedi, Ranjan Kumar Patra, Kiran Kumar Mohapatra, R.H. Wanjari, Debadatta Sethi, Dipaka Ranjan Sena, Tek Bahadur Sapkota, Jagmohan Nayak, Sridhar Patra, Chiter Mal Parihar, Hari Sankar Nayak, “Untangling the effect of soil quality on rice productivity under a 16-years long-term fertilizer experiment using conditional random

forest," Computers and Electronics in Agriculture, vol-197,2022, PP. 1-10

[17] Rubby Aworka, Lontsi Saadio Cedric, Wilfried Yves Hamilton Adoni, Jeremie Thouakesseh Zoueu, Franck Kalala Mutombo, Charles Lebon Mberi Kimpolo, Tarik Nahhal, Moez Krichen, "Agricultural decision system based on advanced machine learning models for yield prediction: Case of East African countries," Smart Agricultural Technology, vol-3, 2022, PP. 1-9

[18] Senthil Kumar Swami Durai, Mary Divya Shamili, "Smart farming using Machine Learning and Deep Learning techniques," Decision Analytics Journal, vol-2, 2022, PP. 1-30

[19] M.S. Suchithra, Maya L. Pai, "Improving the prediction accuracy of soil nutrient classification by optimizing extreme learning machine parameters," Information Processing in Agriculture, vol-7, 2022, PP. 1-11

[20] Kaggle, " https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset " (accessed on 16th November 2021)