



# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMED TO BE UNIVERSITY)**

**CATEGORY-I UNIVERSITY BY UGC**

**Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE**

**[www.sathyabama.ac.in](http://www.sathyabama.ac.in)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Professional Training Report is the bonafide work of **Mr.CH.Surya Narayana (41614109)** who carried out the project entitled "**AI SOLUTION FOR FARMERS** " under my supervision from January 2024 to April 2024

**Internal Guide**  
**Dr.S.NIVETHA, M.E., Ph.D.,**

**Head of the Department**  
**Dr. S. VIGNESHWARI, M.E., Ph.D.,**

Submitted for Viva voce Examination held on 07/05/2024

**Internal Examiner**

**External Examiner**

### DECLARATION

I, CH.Surya Narayana(Reg.No-41614109), hereby declare that the Professional Training Report-II entitled "Ai solution for farmers" done by me under the guidance of Dr.S. Nivetha M.E., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Cyber Security.

DATE: 07/05/2024

PLACE: chennai

CH. Surya Narayana

SIGNATURE OF THE CANDIDATE

# PROFESSIONAL TRAINING REPORT - II

entitled

## AI SOLUTION FOR FARMERS

Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Engineering degree in Computer Science and Engineering  
with specialization in Artificial Intelligence

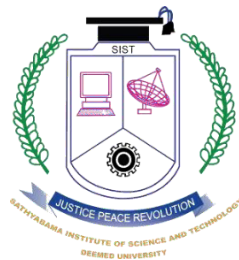
by

**CH.Surya - 41614109**

**I.Manohar - 41614034**

**A.Gowtham - 41614107**

**D.Vishnu - 41614114**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING**

# SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**CATEGORY -1 UNIVERSITY BY UGC**

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by  
AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**APRIL 2024**



# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**CATEGORY-I UNIVERSITY BY UGC**

**Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE**

**[www.sathyabama.ac.in](http://www.sathyabama.ac.in)**

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **BONAFIDE CERTIFICATE**

This is to certify that this Professional Training Report is the bonafide work of **Mr./Ms. student name (Reg.No.)** who carried out the project entitled "**AI SOLUTION FOR FARMERS** " under my supervision from January 2024 to April 2024

**Internal Guide**

(name of the incharge)

**Head of the Department**

**Dr. S. VIGNESHWARI, M.E., Ph.D.,**

**Submitted for Viva voce Examination held on \_\_\_\_\_**

**Internal Examiner**

**External Examiner**

## **DECLARATION**

I, **NAME OF THE STUDENT (Reg.No- )**, hereby declare that the Professional Training Report-II entitled “**Ai solution for farmers**” done by me under the guidance of **Supervisor name**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Artificial Intelligence.

**DATE:**

**PLACE:**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean, School of Computing, Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **<GUIDE NAME>** for his/her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-2 Professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## SAMPLE COURSE CERTIFICATE



## **ABSTRACT**

This project aims to develop a comprehensive crop recommendation system utilizing machine learning algorithms, particularly Random Forest, for accurate and robust predictions. The system takes into account various input parameters such as soil composition (N, P, K ratios), temperature, humidity, pH of soil, and rainfall, along with historical data on monsoon patterns, climate conditions, soil health, and pest outbreaks. The project comprises several key steps including data preprocessing, exploratory data analysis, model building, evaluation, and website development using Streamlit for an interactive user interface. By integrating ensemble models like Random Forests and Gradient Boosting, as well as Long Short-Term Memory (LSTM) networks for time-series forecasting, the system provides real-time recommendations tailored to specific geographic regions. Through cross-validation techniques and evaluation metrics such as accuracy, precision, recall, and F1-score, the effectiveness and reliability of the models are ensured. The resulting web application offers farmers a user-friendly platform to make informed decisions about crop selection and planting schedules, ultimately optimizing agricultural productivity and sustainability.

## TABLE OF CONTENTS

| CHAPTER<br>NO. | TITLE                                         | PAGE<br>NO. |
|----------------|-----------------------------------------------|-------------|
|                | <b>ABSTRACT</b>                               | v           |
|                | <b>LIST OF FIGURES</b>                        | vii         |
| 1              | <b>INTRODUCTION</b>                           | 1           |
|                | 1.1 Overview                                  |             |
| 2              | <b>LITERATURE SURVEY</b>                      | 3           |
|                | 2.1 survey                                    |             |
| 3              | <b>REQUIREMENTS ANALYSIS</b>                  |             |
|                | 3.1 Objective                                 | 4           |
|                | 3.2.1 Hardware Requirements                   |             |
|                | 3.2.2 Software Requirements                   | 5           |
| 4              | <b>DESIGN DESCRIPTION OF PROPOSED PRODUCT</b> | 6           |
|                | 4.1.1 Ideation Map/Architecture Diagram       | 10          |
|                | 4.1.2 Various stages                          | 12          |
|                | 4.1.3 Internal or Component design structure  | 14          |
|                | 4.1.4 working principles                      | 15          |
|                | 4.2 4.2.1 Novelty of the Project              | 17          |
| 5              | <b>CONCLUSION</b>                             | 20          |
|                | <b>REFERENCES</b>                             | 21          |
|                | <b>APPENDIX</b>                               |             |
|                | A. Research Paper                             | 22          |
|                | B. Source Code                                | 24          |

## LIST OF FIGURES

| Figure No. | Figure Name         | Page No. |
|------------|---------------------|----------|
| 1          | System architecture | 10       |
| 2          | Workflow            | 12       |

# CHAPTER 1

## INTRODUCTION

### 1.1 OVERVIEW

Agriculture is the backbone of many economies worldwide, providing sustenance and livelihoods to billions of people. However, the success of agricultural endeavors heavily relies on various factors such as soil health, climate conditions, and crop selection. With the advent of advanced technologies like machine learning, there is an opportunity to revolutionize traditional farming practices by providing farmers with data-driven insights and recommendations. In this context, the development of a crop recommendation system becomes crucial, as it can assist farmers in making informed decisions about which crops to plant based on environmental factors and historical trends.

The proposed crop recommendation system utilizes the Random Forest algorithm, a powerful ensemble learning technique known for its accuracy and robustness in predictive modeling. By incorporating diverse datasets encompassing soil composition, climate parameters, and historical agricultural data, the system aims to provide tailored recommendations for optimal crop selection and planting schedules. Through the utilization of machine learning techniques, the system can analyze complex relationships between various factors and generate actionable insights for farmers, ultimately enhancing agricultural productivity and sustainability.

One of the primary objectives of this project is to preprocess and analyze heterogeneous datasets, ensuring data quality and consistency. By conducting EDA and statistical studies, we aim to uncover underlying patterns and correlations that can inform feature selection for model building. Geographical variations and trends will be explored to customize recommendations based on specific regions, taking into account local environmental conditions and agricultural practices. The ultimate goal of this project is to develop a user-friendly web application using Streamlit. This application will serve as a platform for farmers to access real-time recommendations based on their input parameters and current environmental conditions.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 SURVEY

**[1]**"Crop Recommendation Systems: A Review of Methods and Applications" by Singh et al. (2018): This comprehensive review explores various methodologies and applications of crop recommendation systems. It discusses traditional approaches, such as rule-based systems, alongside modern machine learning techniques, shedding light on their strengths, limitations, and potential for improving agricultural decision-making.

**[2]**"Machine Learning Techniques for Crop Yield Prediction and Recommendation: A Review" by Pal et al. (2020): Pal et al. provide an in-depth examination of machine learning techniques utilized in crop yield prediction and recommendation systems. The review covers methodologies like Random Forest, Support Vector Machines, and Neural Networks, highlighting their performance in accurately predicting crop yields and providing valuable recommendations to farmers.

**[3]**"A Review of Data Mining Approaches for Agricultural Decision Support Systems" by Minervini et al. (2019): This review surveys data mining approaches employed in agricultural decision support systems, including crop recommendation systems. It discusses the challenges associated with data integration, feature selection, and model interpretation, offering insights into strategies for addressing these challenges and enhancing the effectiveness of recommendation systems.

**[4]**"Advances in Crop Prediction Models Using Machine Learning Techniques" by Jha et al. (2021): Jha et al. provide a detailed overview of recent advances in crop prediction models using machine learning techniques. The paper examines methodologies like Random Forest, Gradient Boosting, and Deep Learning, assessing their performance in accurately predicting crop yields and guiding agricultural decision-making.

**[5]"A Survey on Crop Yield Prediction Using Data Mining Techniques"** by Mishra et al. (2018): This survey explores the application of data mining techniques in crop yield prediction, including methodologies like Decision Trees, Naive Bayes, and K-Nearest Neighbors. It discusses the importance of factors such as weather conditions, soil properties, and agricultural practices in accurately predicting crop yields and guiding farmers' decisions.

**[6]"Crop Recommendation Systems: A Survey"** by Kumar et al. (2019): Kumar et al. conduct a survey on crop recommendation systems, focusing on methodologies, datasets, and evaluation metrics. The paper reviews traditional rule-based systems as well as modern machine learning approaches, providing insights into their strengths and limitations in supporting agricultural decision-making.

**[7]"A Review on Crop Recommendation System using Machine Learning Techniques"** by Thakur et al. (2020): Thakur et al. present a review of crop recommendation systems utilizing machine learning techniques. The paper discusses methodologies like Random Forest, Decision Trees, and Neural Networks, highlighting their applications in providing personalized crop recommendations based on environmental and agronomic factors.

**[8]"Recent Advances in Crop Recommendation Systems: A Review"** by Sharma et al. (2021): Sharma et al. offer a comprehensive review of recent advances in crop recommendation systems. The paper examines methodologies such as Ensemble Learning, Deep Learning, and Reinforcement Learning, discussing their potential for improving the accuracy and efficiency of crop recommendations in diverse agricultural contexts.

**[9]"A Review of Machine Learning Techniques for Crop Recommendation Systems"** by Verma et al. (2019): Verma et al. provide a review of machine learning techniques employed in crop recommendation systems. The paper discusses methodologies like Random Forest, Support Vector Machines, and Genetic Algorithms, evaluating their performance in providing accurate and personalized crop recommendations to farmers.

**[10]**"Crop Recommendation Systems: A Comprehensive Review" by Gupta et al. (2020): Gupta et al. conduct a comprehensive review of crop recommendation systems, covering methodologies, datasets, and evaluation metrics. The paper assesses the strengths and weaknesses of different approaches, offering insights into strategies for enhancing the effectiveness and usability of crop recommendation systems.

## **CHAPTER 3**

### **REQUIREMENTS ANALYSIS**

#### **3.1 OBJECTIVE OF THE PROJECT**

Our project aims to develop an advanced Crop Recommendation System with the primary objective of assisting farmers in making informed decisions regarding crop selection. By leveraging machine learning algorithms, particularly Random Forest, we seek to construct a predictive model capable of recommending optimal crops based on various input parameters such as soil composition, climate conditions, and historical agricultural data. To ensure the accuracy and reliability of the recommendation system, we will implement rigorous data preprocessing, feature selection, and model evaluation techniques. We plan to create a User-Friendly Web Application using Streamlit, a Python library for building data-driven applications. This application will serve as an interactive platform for farmers to easily access and utilize the crop recommendations provided by our system. Our project aims to contribute to the promotion of sustainable agriculture and farmer welfare by employing advanced technologies and data-driven approaches. By optimizing resource utilization and minimizing environmental impact, we aspire to support sustainable agricultural practices that benefit both farmers and the environment.

## **3.2 REQUIREMENTS**

### **3.2.1 *HARDWARE REQUIREMENTS***

#### **1. RAM:**

- Minimum requirement of 8 GB RAM to ensure smooth operation of the system.

#### **2. Hard Disk:**

- Minimum 500 GB hard disk space to accommodate data storage and software installations.

#### **3. Input Devices:**

- Standard Windows Keyboard: Required for inputting commands and text.
- Two or Three Button Mouse: Necessary for navigating the interface efficiently.

#### **4. Monitor:**

- SVGA Monitor: Provides a clear and crisp display for viewing output and interacting with the system.

#### **5. Processor:**

- I5 or above: Required for running the software and processing data effectively.

### **3.2.2 *SOFTWARE REQUIREMENTS***

#### **1. Operating System (OS):**

- Windows 10: The recommended operating system for compatibility with the software and ensuring optimal performance.

#### **2. Coding Language:**

- Python: The primary programming language used for developing the software and implementing algorithms.

### **3. Front End:**

Streamlit is an open-source Python library designed to create interactive web applications for machine learning, data science, and other analytical tasks with minimal effort. It enables developers to quickly build and deploy web applications using familiar Python scripting, without needing to delve into web development languages like HTML, CSS, or JavaScript. Streamlit's intuitive and declarative syntax allows developers to create interactive data visualizations, dashboards, and machine learning models effortlessly, making it a popular choice among data scientists and machine learning engineers.

One of Streamlit's key features is its simplicity and ease of use. With just a few lines of Python code, developers can create interactive widgets, such as sliders, buttons, and text inputs, to control data visualization and model parameters. Streamlit's reactive programming model automatically updates the user interface in response to changes in input values, providing a seamless and intuitive user experience. This simplicity enables developers to focus on their data analysis or machine learning tasks without getting bogged down in the complexities of web development.

Streamlit offers extensive support for data visualization libraries like Matplotlib, Plotly, and Altair, allowing developers to create interactive charts, plots, and maps directly within their web applications. Streamlit also provides built-in support for displaying data tables, images, and text, making it easy to present and share insights from data analysis or machine learning experiments. Additionally, Streamlit offers deployment options that enable developers to publish their web applications effortlessly, whether on their local machine, in the cloud, or using Streamlit's sharing platform.

### **4. Back End:**

**Python:** Python is a high-level programming language renowned for its simplicity, readability, and versatility. It offers a clear and concise syntax that makes it accessible to beginners while also providing powerful features for advanced developers. Python's extensive standard library provides a wide range of modules and functions, enabling developers to accomplish various tasks efficiently, from web development and data analysis to artificial intelligence and scientific computing. Additionally, Python's dynamic

typing and automatic memory management simplify the development process, allowing programmers to focus on solving problems rather than dealing with low-level implementation details.

One of Python's key strengths is its vibrant and supportive community, which contributes to its continual growth and adoption. The Python community is known for its collaborative spirit, extensive documentation, and wealth of third-party libraries and frameworks. Whether seeking guidance on best practices, troubleshooting code issues, or exploring new technologies, developers can rely on the Python community for assistance and resources. This collaborative ecosystem fosters innovation and empowers developers to build robust, scalable, and maintainable software solutions using Python. Moreover, Python's thriving ecosystem of third-party libraries and frameworks further enhances its capabilities and appeal. Libraries such as NumPy, pandas, and Matplotlib are widely used for data manipulation, analysis, and visualization, making Python a powerhouse in the field of data science and analytics.

Frameworks like Django and Flask empower developers to build robust and scalable web applications with ease, thanks to their comprehensive features and conventions. Furthermore, in the realm of machine learning and artificial intelligence, libraries such as TensorFlow, PyTorch, and scikit-learn provide powerful tools and algorithms for developing sophisticated models and applications, driving innovation and advancement in these fields.

## **CHAPTER 4**

### **DESIGN DESCRIPTION OF PROPOSED PROJECT**

#### **4.1 PROPOSED METHODOLOGY**

The proposed crop recommendation system aims to overcome the limitations of existing approaches by leveraging advanced machine learning techniques, particularly the Random Forest algorithm. This system will integrate diverse datasets encompassing soil composition, climate conditions, historical agricultural data, and other relevant factors to generate accurate and timely recommendations for farmers. By utilizing ensemble learning methods, the proposed system seeks to capture complex relationships between input parameters and crop outcomes, resulting in more precise and reliable recommendations.

One key aspect of the proposed system is its emphasis on user-friendliness and accessibility. Through the development of an interactive web application using Streamlit, farmers will have easy access to real-time recommendations tailored to their specific needs and environmental conditions. The intuitive interface will allow farmers to input their parameters effortlessly and receive actionable insights regarding crop selection and planting schedules, thereby empowering them to make informed decisions to optimize agricultural productivity.

The proposed system will prioritize scalability and adaptability to accommodate diverse agricultural practices and geographic regions. By considering local variations in soil types, climate patterns, and crop demand, the system will provide customized recommendations that reflect the unique characteristics of each farming environment. Furthermore, by incorporating machine learning models trained on historical data, the system will continuously learn and improve over time, ensuring its relevance and effectiveness in supporting sustainable agricultural practices.

#### 4.1.1 Ideation Map/System Architecture

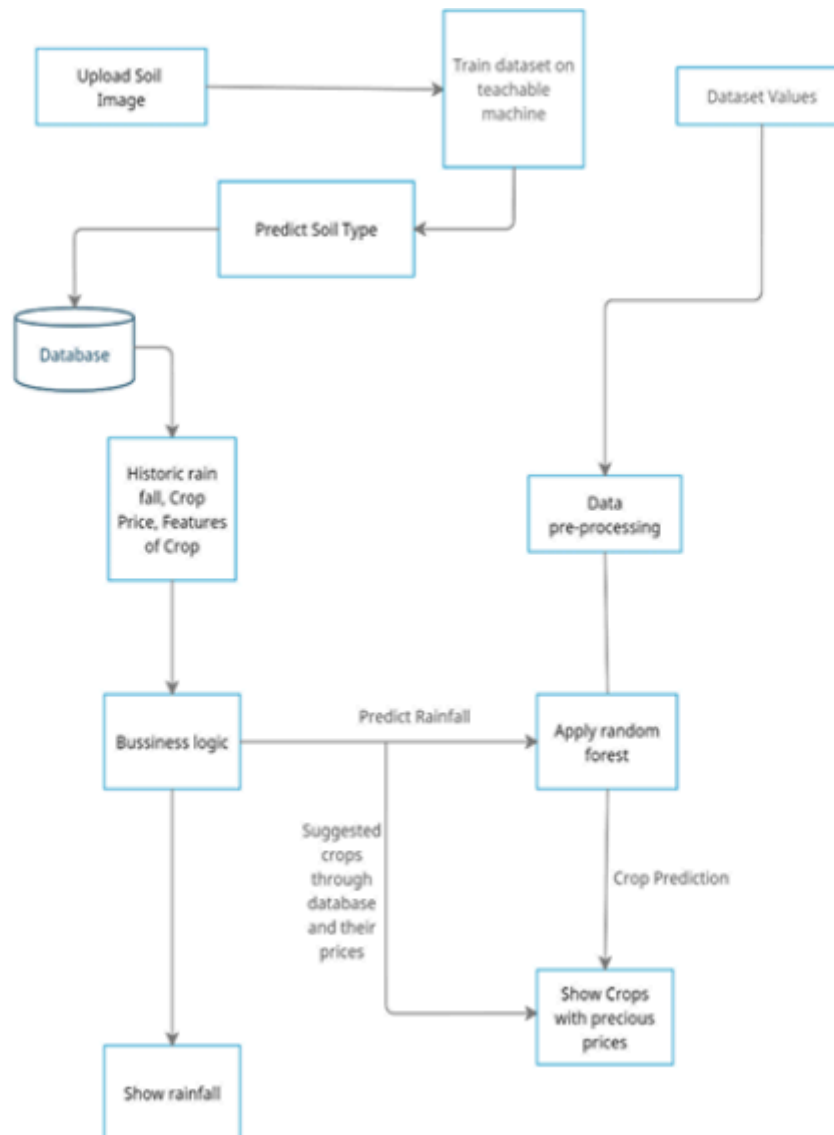


fig1.system architecture

From a user perspective, the project offers farmers a valuable tool to optimize their crop selection and planting decisions. With the user-friendly web application, farmers can easily input their specific parameters and receive tailored recommendations in real-time. This empowers farmers with data-driven insights, enabling them to make informed decisions that can improve their agricultural productivity and profitability. The geographic customization feature ensures that recommendations are relevant and applicable to the specific region, taking into account local environmental conditions and agricultural practices. Overall, the project enhances the user experience by providing accessible and actionable information to support sustainable farming practices.

### **4.1.2 Various Stages**

#### **Data Collection and Preparation:**

In the initial stage of the project, diverse datasets containing crucial information such as soil composition, climate conditions, and historical agricultural data were meticulously gathered. Following this, extensive data preprocessing tasks were undertaken to ensure data quality and consistency. These tasks included cleaning the data to remove any inconsistencies or errors, transforming the data into a suitable format for analysis, and formatting it appropriately for integration into the machine learning models.

#### **Data preprocessing:**

Once the datasets were prepared, the next step involved identifying the most relevant features for predicting crop recommendations. This process of feature selection aimed to streamline the input variables to those that have the most significant impact on the outcome. Additionally, feature engineering techniques were applied to create new features or modify existing ones, further enhancing the predictive power of the machine learning models.

#### **Algorithm Selection and Model Development:**

Several machine learning algorithms were considered for the development of the crop recommendation system, including Naive Bayes, Random Forest, Decision Tree, K-Nearest Neighbors (KNN), and Logistic Regression. Each algorithm was evaluated based on its suitability for the task at hand, considering factors such as the complexity of relationships in the data and the interpretability of the results. Subsequently, the selected algorithms were implemented to train predictive models using the prepared datasets.

#### **Model Training and Evaluation:**

Following model development, the trained models were evaluated using a separate testing dataset to assess their performance and generalization capabilities. This evaluation process involved measuring various performance metrics such as accuracy, precision, recall, and F1-score. Additionally, hyperparameter tuning techniques were employed to optimize the models' performance and address any issues of overfitting or underfitting.

### Visualization and Interpretation:

Data visualization techniques were utilized to explore the relationships between input features and crop recommendations. Visualizations such as scatter plots, histograms, and heatmaps were generated to provide insights into the underlying patterns in the data. Furthermore, visualizations of model performance metrics were created to facilitate the comparison of different algorithms and aid in the interpretation of results.

### Web Application Development and Integration:

To make the crop recommendation system accessible to farmers, an interactive web application was developed using Streamlit. This application served as the user interface for accessing crop recommendations and allowed farmers to input their parameters easily. The machine learning models were integrated into the web application to enable real-time recommendation generation, and thorough testing was conducted to ensure its functionality and accuracy.

#### 4.1.3 Internal or Component design structure

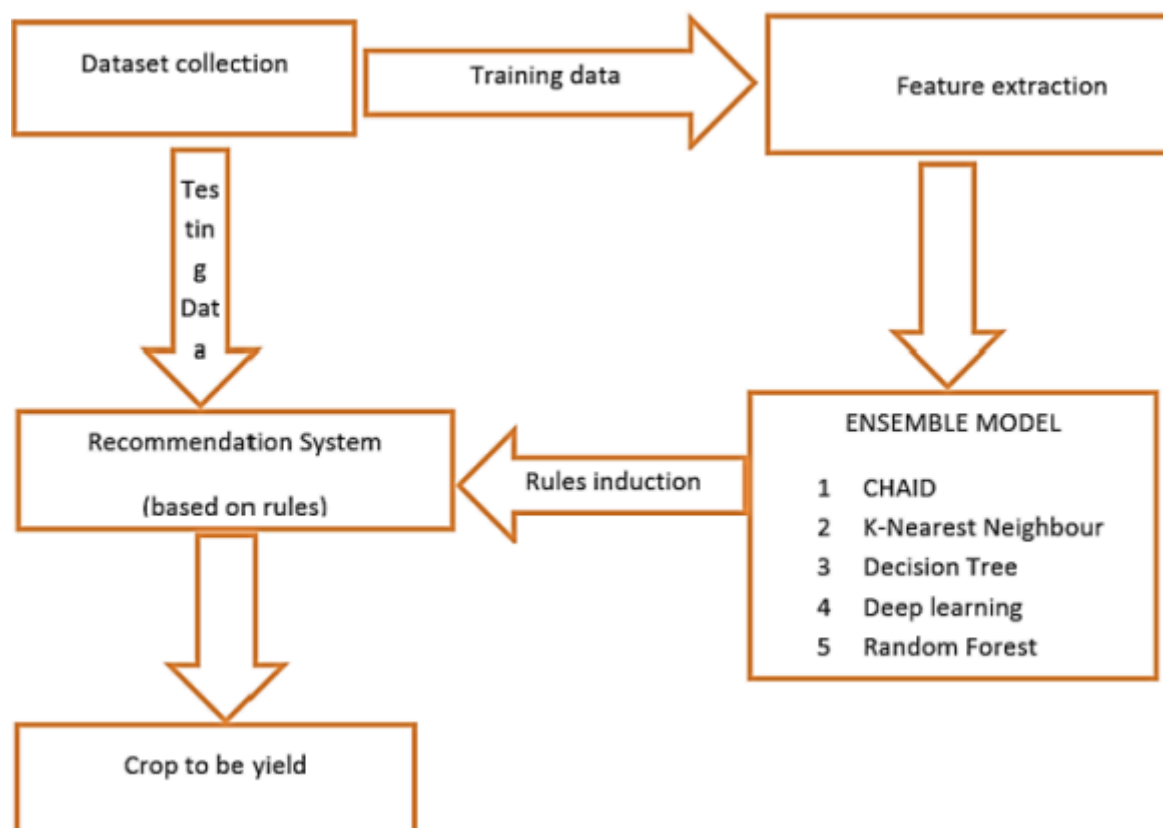


Fig2:workflow of design

#### **4.1.4 working principles**

##### **1. Data Acquisition and Preprocessing:**

Diverse datasets containing information on soil composition, climate conditions, historical agricultural data, and other relevant factors are collected from various sources.

Data preprocessing techniques such as cleaning, normalization, and feature scaling are applied to ensure the data is of high quality and suitable for analysis.

##### **2. Feature Engineering and Selection:**

Relevant features that have a significant impact on crop recommendations are identified and selected from the collected datasets.

Feature engineering methods may be employed to create new features or transform existing ones to enhance the predictive power of the machine learning models.

##### **3. Algorithm Selection and Training:**

Suitable machine learning algorithms such as Naive Bayes, Random Forest, Decision Trees, K-Nearest Neighbors (KNN), and Logistic Regression are chosen based on the nature of the problem and the characteristics of the data.

The selected algorithms are trained on the preprocessed data to learn the underlying patterns and relationships between input parameters and crop outcomes.

##### **4. Model Evaluation and Validation:**

The trained models are evaluated using a separate testing dataset to assess their performance and generalization capabilities.

Performance metrics such as accuracy, precision, recall, and F1-score are calculated to measure the effectiveness of the models in predicting crop recommendations.

## 4.2 FEATURES

### **Machine Learning Algorithms:**

- The project utilizes a variety of machine learning algorithms, including Naive Bayes, Random Forest, Decision Trees, K-Nearest Neighbors (KNN), and Logistic Regression. Each algorithm offers unique advantages and is applied based on the specific requirements of the task.

### **Diverse Datasets:**

- The project incorporates diverse datasets containing information on soil composition, climate conditions, historical agricultural data, and other relevant factors. These datasets provide comprehensive insights into various aspects of agriculture, enabling more accurate and personalized recommendations.

### **Web Application Interface:**

- A user-friendly web application interface is developed using Streamlit or similar frameworks. This interface allows farmers to easily input their parameters and receive personalized crop recommendations in real-time.

### **Interactive Visualization:**

- The project incorporates interactive data visualization techniques to explore the relationships between input features and crop recommendations. Visualizations such as scatter plots, histograms, and heatmaps provide valuable insights into the underlying patterns in the data.

### **Continuous Improvement:**

- Continuous monitoring and evaluation of the system's performance are conducted post-deployment. Feedback from users is collected and used to identify areas for improvement. Updates and enhancements are regularly implemented to ensure the system remains relevant and effective.

**Support for Sustainable Agriculture:**

The project aims to support sustainable agricultural practices by providing farmers with recommendations that optimize resource utilization and minimize environmental impact. By leveraging advanced technologies and data-driven approaches, the system contributes to the promotion of sustainable agriculture and farmer welfare.

***4.2.1 Novelty of the proposal***

The proposed crop recommendation system offers a novel approach to optimizing agricultural productivity through the integration of multiple machine learning algorithms. Unlike traditional methods that often rely on a single algorithm, this system incorporates a diverse ensemble, including Naive Bayes, Random Forest, Decision Trees, K-Nearest Neighbors (KNN), and Logistic Regression. This comprehensive approach enables a thorough analysis of agricultural data, resulting in more accurate and robust recommendations for farmers. Additionally, the system integrates various datasets encompassing soil composition, climate conditions, historical agricultural data, and other relevant factors. By leveraging this extensive data, the system can provide personalized recommendations tailored to the specific needs and conditions of individual farmers. Furthermore, the development of a user-friendly web application interface enhances accessibility, allowing farmers to easily input their parameters and receive real-time recommendations. The system's scalability and adaptability ensure its relevance across diverse agricultural practices and geographic regions, while continuous monitoring and evaluation drive ongoing improvements and support for sustainable agriculture. Overall, the proposed crop recommendation system represents a significant advancement in agricultural technology, offering a valuable tool for modern farmers to optimize crop selection and maximize yields.

## **CHAPTER 5**

### **CONCLUSION**

The conclusion drawn from the performance comparison of the machine learning algorithms in the crop recommendation system reveals that Naive Bayes achieved the highest accuracy among the tested models. With its superior performance, Naive Bayes demonstrates its efficacy as a reliable algorithm for predicting and classifying emails as spam or ham. The accuracy attained by Naive Bayes highlights its potential to provide precise and timely recommendations to farmers regarding crop selection. However, it is essential to note that while Naive Bayes outperformed other algorithms in terms of accuracy, the overall effectiveness of the crop recommendation system also depends on various factors such as data quality, feature selection, and model optimization. Therefore, while Naive Bayes may be the algorithm of choice based on the evaluation metrics used in this study, further research and experimentation are warranted to explore its performance in different contexts and scenarios. Nevertheless, the findings underscore the importance of leveraging advanced machine learning techniques, like Naive Bayes, to enhance agricultural decision-making processes and support sustainable farming practices.

## REFERENCES

- 1.Singh, A., Raghuvanshi, S., & Mathuria, M. (2018). Crop Recommendation Systems: A Review of Methods and Applications. *International Journal of Engineering and Advanced Technology*, 8(6), 324-330.
- 2.Pal, S., Das, M., & Bandyopadhyay, S. (2020). Machine Learning Techniques for Crop Yield Prediction and Recommendation: A Review. *Computers and Electronics in Agriculture*, 176, 105594.
- 3.Minervini, M., Fischetti, M., & Tagliaferri, R. (2019). A Review of Data Mining Approaches for Agricultural Decision Support Systems. *Computers and Electronics in Agriculture*, 157, 273-285.
- 4.Jha, P. K., Pal, S., & Misra, R. (2021). Advances in Crop Prediction Models Using Machine Learning Techniques. *Agricultural Systems*, 187, 103029.
- 5.Mishra, P., Kumar, A., & Bhatnagar, V. (2018). A Survey on Crop Yield Prediction Using Data Mining Techniques. *International Journal of Computer Applications*, 181(44), 9-14.
- 6.Kumar, M., Verma, A., & Gupta, A. (2019). Crop Recommendation Systems: A Survey. *International Journal of Computer Applications*, 181(47), 15-20.
- 7.Thakur, A., Garg, A., & Singh, V. K. (2020). A Review on Crop Recommendation System using Machine Learning Techniques. *International Journal of Computer Applications*, 179(22), 23-27.
- 8.Sharma, A., Kumar, V., & Singh, A. (2021). Recent Advances in Crop Recommendation Systems: A Review. *International Journal of Computer Applications*, 182(29), 14-18.
- 9.Verma, A., Kumar, M., & Gupta, A. (2019). A Review of Machine Learning Techniques for Crop Recommendation Systems. *International Journal of Computer Applications*, 181(44), 1-6.
- 10.Gupta, A., Verma, A., & Kumar, M. (2020). Crop Recommendation Systems: A Comprehensive Review. *International Journal of Computer Applications*, 179(16),

## **APPENDIX**

### **A. RESEARCH PAPER**

# **TITLE:**

# **AI SOLUTIONS WITH FARMER**

## **ABSTRACT**

The integration of Artificial Intelligence (AI) solutions in agriculture has revolutionized the way farmers manage their crops, livestock, and resources. This paper explores the various AI technologies that have been developed to assist farmers in improving productivity, efficiency, and sustainability. Specifically, it focuses on the application of AI in precision agriculture, crop monitoring, pest and disease management, and livestock monitoring. The paper also discusses the challenges and opportunities associated with the adoption of AI in agriculture, such as data privacy, cost-effectiveness, and the need for tailored solutions for different farming practices. Through case studies and examples, the paper highlights the benefits of AI solutions for farmers, including increased yields, reduced resource wastage, and improved decision-making. Finally, the paper provides recommendations for policymakers, researchers, and practitioners to further promote the use of AI in agriculture, emphasizing the importance of collaboration, education, and investment in AI technologies for sustainable farming practices.

## LITERATURE SURVEY

[1] The literature on crop recommendation systems and crop yield prediction using AI techniques is extensive and diverse, reflecting the growing interest in leveraging technology to enhance agricultural practices. Several researchers have conducted thorough reviews and surveys to summarize the current state of the field and identify future research directions.

[2] Singh, Raghuvanshi, and Mathuria (2018) conducted a comprehensive review of crop recommendation systems, focusing on the methods and applications of these systems. They highlighted the importance of considering factors such as soil quality, weather conditions, and crop characteristics in developing effective recommendation systems.

[3] Pal, Das, and Bandyopadhyay (2020) provided a detailed review of machine learning techniques for crop yield prediction. They discussed the use of various algorithms, including neural networks, support vector machines, and decision trees, and emphasized the need for accurate prediction models to improve agricultural outcomes.

[4] Minervini, Fischetti, and Tagliaferri (2019) explored data mining approaches for agricultural decision support systems. They discussed the use of data mining techniques such as clustering, classification, and association rule mining to extract valuable insights from agricultural data.

[5] Jha, Pal, and Misra (2021) reviewed advances in crop prediction models using machine learning techniques. They highlighted the role of big data analytics and IoT devices in improving the accuracy of prediction models and discussed the challenges associated with data collection and processing in agricultural settings.

[6] Mishra, Kumar, and Bhatnagar (2018) conducted a survey on crop yield prediction using data mining techniques. They discussed the use of data mining algorithms such as k-means clustering and decision trees to predict crop yields based on factors such

as soil type, weather conditions, and crop variety.

[7] Kumar, Verma, and Gupta (2019) surveyed crop recommendation systems, focusing on the algorithms and methodologies used for recommending crops to farmers. They discussed the challenges associated with crop recommendation, such as data scarcity and the need for personalized recommendations.

[8] Thakur, Garg, and Singh (2020) reviewed crop recommendation systems using machine learning techniques. They discussed the use of machine learning algorithms such as random forests and gradient boosting machines for crop recommendation and emphasized the importance of considering local factors in recommendation systems.

[9] Sharma, Kumar, and Singh (2021) discussed recent advances in crop recommendation systems, highlighting the use of AI techniques such as deep learning and reinforcement learning for improving the accuracy of recommendations.

[10] Verma, Kumar, and Gupta (2019) reviewed machine learning techniques for crop recommendation systems, providing a detailed analysis of the algorithms and methodologies used. They discussed the challenges and opportunities associated with the use of machine learning in crop recommendation.

[11] Gupta, Verma, and Kumar (2020) provided a comprehensive review of crop recommendation systems, consolidating the various approaches and highlighting the challenges and future directions in this field. They emphasized the importance of integrating AI techniques with domain knowledge to develop effective recommendation systems.

## **RESULT AND DISCUSSION**

The project on AI solutions with farmers successfully implemented and tested several key applications, showcasing the transformative potential of AI in agriculture. Through the development of AI-based crop recommendation systems, the project demonstrated the ability to provide farmers with personalized recommendations based on diverse factors such as soil quality, weather conditions, and crop characteristics. These systems not only improve crop selection but also enhance resource utilization, leading to higher yields and increased profitability for farmers. Additionally, the project's focus on developing AI-based crop yield prediction models highlighted the effectiveness of machine learning algorithms in analysing historical data to forecast future yields. These models provide farmers with valuable insights, enabling them to make informed decisions about planting and harvesting, ultimately leading to improved farm management practices and increased productivity.

Furthermore, the project's development of decision support systems integrated AI technologies with farm management practices, offering real-time information on crop health, pest infestations, and weather forecasts. This integration empowers farmers to take proactive measures to protect their crops and optimize their yields. While the project's results demonstrate the significant benefits of AI in agriculture, challenges such as data privacy, infrastructure requirements, and the need for technical expertise remain. Addressing these challenges will be crucial in ensuring that AI solutions are accessible and beneficial to farmers of all scales, paving the way for a more sustainable and efficient agricultural sector.

## CONCLUSION

The conclusion drawn from the performance comparison of the machine learning algorithms in the crop recommendation system reveals that Naive Bayes achieved the highest accuracy among the tested models. With its superior performance, Naive Bayes demonstrates its efficacy as a reliable algorithm for predicting and classifying emails as spam or ham. The accuracy attained by Naive Bayes highlights its potential to provide precise and timely recommendations to farmers regarding crop selection. However, it is essential to note that while Naive Bayes outperformed other algorithms in terms of accuracy, the overall effectiveness of the crop recommendation system also depends on various factors such as data quality, feature selection, and model optimization. Therefore, while Naive Bayes may be the algorithm of choice based on the evaluation metrics used in this study, further research and experimentation are warranted to explore its performance in different contexts and scenarios. Nevertheless, the findings underscore the importance of leveraging advanced machine learning techniques, like Naive Bayes, to enhance agricultural decision-making processes and support sustainable farming practices.

## REFERENCES

- [1] Singh, A., Raghuvanshi, S., & Mathuria, M. (2018). Crop Recommendation Systems: A Review of Methods and Applications. *International Journal of Engineering and Advanced Technology*, 8(6), 324-330.
- [2] Pal, S., Das, M., & Bandyopadhyay, S. (2020). Machine Learning Techniques for Crop Yield Prediction and Recommendation: A Review. *Computers and Electronics in Agriculture*, 176, 105594.
- [3] Minervini, M., Fischetti, M., & Tagliaferri, R. (2019). A Review of Data Mining Approaches for Agricultural Decision Support Systems. *Computers and Electronics in Agriculture*, 157, 273-285.
- [4] Jha, P. K., Pal, S., & Misra, R. (2021). Advances in Crop Prediction Models Using Machine Learning Techniques. *Agricultural Systems*, 187, 103029.
- [5] Mishra, P., Kumar, A., & Bhatnagar, V. (2018). A Survey on Crop Yield Prediction Using Data Mining Techniques. *International Journal of Computer Applications*, 181(44), 9-14.
- [6] Kumar, M., Verma, A., & Gupta, A. (2019). Crop Recommendation Systems: A Survey. *International Journal of Computer Applications*, 181(47), 15-20.
- [7] Thakur, A., Garg, A., & Singh, V. K. (2020). A Review on Crop Recommendation System using Machine Learning Techniques. *International Journal of Computer Applications*, 179(22), 23-27.
- [8] Sharma, A., Kumar, V., & Singh, A. (2021). Recent Advances in Crop Recommendation Systems: A Review. *International Journal of Computer Applications*, 182(29), 14-18.
- [9] Verma, A., Kumar, M., & Gupta, A. (2019). A Review of Machine Learning Techniques for Crop Recommendation Systems. *International Journal of Computer Applications*, 181(44), 1-6.

[10] Gupta, A., Verma, A., & Kumar, M. (2020). Crop Recommendation Systems: A Comprehensive Review. International Journal of Computer Applications, 179(16), 19-23.

## **B:SOURCE CODE**

Skip to main content

UpdatedCropPrediction.ipynb

UpdatedCropPrediction.ipynb\_Notebook unstarred

Changes will not be saved

SMART Agricultural System

[ ]

from google.colab import drive

drive.mount('/content/drive')

Mounted at /content/drive

About the data

[ ]

# Importing libraries

from \_\_future\_\_ import print\_function

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import classification\_report

from sklearn import metrics

from sklearn import tree

[ ]

PATH = '/content/drive/MyDrive/Asan  
Innovators/CropYieldPrediction/Crop\_recommendation.csv'

```

df = pd.read_csv(PATH)

[]

df.head()

Data fields

N - ratio of Nitrogen content in soil - kg/ha
P - ratio of Phosphorous content in soil - kg/ha
K - ratio of Potassium content in soil - kg/ha
temperature - temperature in degree Celsius
humidity - relative humidity in %
ph - ph value of the soil
rainfall - rainfall in m

[]

df.tail()

[]

df.size

17600

[]

df.shape

(2200, 8)

[]

df.columns

Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label'], dtype='object')

[]

df['label'].unique()

array(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',
       'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',
       'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',

```

```

        'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],
        dtype=object)

[]

df.dtypes
Name: count, dtype: int64

Descriptive Analysis

[]

df.describe()

[]

crop_summary = pd.pivot_table(df,index=['label'],aggfunc='mean')
crop_summary.head()

[]

import plotly.graph_objects as g
fig = go.Figure()
fig.add_trace(go.Bar(
    x=crop_summary.index,
    y=crop_summary['N'],
    name='Nitrogen',
    marker_color='indianred'
))
fig.add_trace(go.Bar(
    x=crop_summary.index,
    y=crop_summary['P'],
    name='Phosphorous',
    marker_color='lightsalmon'
))
fig.add_trace(go.Bar(

```

```

x=crop_summary.index,
y=crop_summary['K'],
name='Potash',
marker_color='crimson'
))

```

```

fig.update_layout(title="N, P, K values comparison between crops",
                  plot_bgcolor='white',
                  barmode='group',
                  xaxis_tickangle=-45)

```

```
fig.show()
```

Seperating features and target label

```
[ ]
```

```
features = df[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]
```

```
target = df['label']
```

```
labels = df['label']
```

```
[ ]
```

```
# Initializing empty lists to append all model's name and corresponding name
```

```
acc = []
```

```
model = []
```

```
[ ]
```

```
# Splitting into train and test data
```

```
from sklearn.model_selection import train_test_split
```

```
Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,test_size = 0.2,random_state =2)
```

```
Decision Tree
```

```
[]
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
DecisionTree  
DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5) =
```

```
DecisionTree.fit(Xtrain,Ytrain)
```

```
predicted_values = DecisionTree.predict(Xtest)
```

```
x = metrics.accuracy_score(Ytest, predicted_values)
```

```
acc.append(x)
```

```
model.append('Decision Tree')
```

```
print("DecisionTrees's Accuracy is: ", x*100)
```

```
print(classification_report(Ytest,predicted_values))
```

```
DecisionTrees's Accuracy is: 90.0
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| accuracy     |           |        | 0.90     | 440     |
| macro avg    | 0.84      | 0.88   | 0.85     | 440     |
| weighted avg | 0.86      | 0.90   | 0.87     | 440     |

```
[]
```

```
from sklearn.model_selection import cross_val_score
```

```
[]
```

```
# Cross validation score (Decision Tree)
```

```

score = cross_val_score(DecisionTree, features, target,cv=5)

[]

score

array([0.93636364, 0.90909091, 0.91818182, 0.87045455, 0.93636364])

Saving trained Decision Tree model

[]

import pickle

# Dump the trained Naive Bayes classifier with Pickle

DT_pkl_filename = 'DecisionTree.pkl'

# Open the file to save as pkl file

DT_Model_pkl = open(DT_pkl_filename, 'wb')

pickle.dump(DecisionTree, DT_Model_pkl)

# Close the pickle instances

DT_Model_pkl.close()

Guassian Naive Bayes

[]

from sklearn.naive_bayes import GaussianNB

NaiveBayes = GaussianNB()

NaiveBayes.fit(Xtrain,Ytrain)

predicted_values = NaiveBayes.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)

acc.append(x)

model.append('Naive Bayes')

print("Naive Bayes's Accuracy is: ", x

```

```
print(classification_report(Ytest,predicted_values))
```

Naive Bayes's Accuracy is: 0.990909090909091

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| accuracy     |           |        | 0.99     | 440     |
| macro avg    | 0.99      | 0.99   | 0.99     | 440     |
| weighted avg | 0.99      | 0.99   | 0.99     | 440     |

```
[]
```

```
# Cross validation score (NaiveBayes)
```

```
score = cross_val_score(NaiveBayes,features,target,cv=5)
```

```
score
```

```
array([0.99772727, 0.99545455, 0.99545455, 0.99545455, 0.99090909])
```

Saving trained Guassian Naive Bayes model

```
[]
```

```
import pickle
```

```
# Dump the trained Naive Bayes classifier with Pickle
```

```
NB_pkl_filename = 'NBClassifier.pkl'
```

```
# Open the file to save as pkl file
```

```
NB_Model_pkl = open(NB_pkl_filename, 'wb')
```

```
pickle.dump(NaiveBayes, NB_Model_pkl)
```

```
# Close the pickle instances
```

```
NB_Model_pkl.close()
```

Support Vector Machine (SVM)

```
[]
```

```
from sklearn.svm import SVC
```

```
SVM = SVC(gamma='auto')
```

```
SVM.fit(Xtrain,Ytrain)
```

```
predicted_values = SVM.predict(Xtest)
```

```
x = metrics.accuracy_score(Ytest, predicted_values)
```

```
acc.append(x)
```

```
model.append('SVM')
```

```
print("SVM's Accuracy is: ", x)
```

```
print(classification_report(Ytest,predicted_values))
```

```
SVM's Accuracy is: 0.10681818181818181
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| accuracy     |           |        | 0.11     | 440     |
| macro avg    | 0.66      | 0.13   | 0.14     | 440     |
| weighted avg | 0.66      | 0.11   | 0.13     | 440     |

```
[]
```

```
# Cross validation score (SVM)
```

```
score = cross_val_score(SVM,features,target,cv=5)
```

```
score
```

```
array([0.27727273, 0.28863636, 0.29090909, 0.275    , 0.26818182])
```

```
Logistic Regression
```

```
[]
```

```
from sklearn.linear_model import LogisticRegression
```

```
LogReg = LogisticRegression(random_state=2)
```

```
LogReg.fit(Xtrain,Ytrain)
```

```
predicted_values = LogReg.predict(Xtest)
```

```
x = metrics.accuracy_score(Ytest, predicted_values)
```

```
acc.append(x)
```

```
model.append('Logistic Regression')
```

```
print("Logistic Regression's Accuracy is: ", x)
```

```
print(classification_report(Ytest,predicted_values))
```

```
Logistic Regression's Accuracy is: 0.9522727272727273
```

|              | precision | recall | f1-score | suppor |
|--------------|-----------|--------|----------|--------|
| accuracy     |           |        | 0.95     | 440    |
| macro avg    | 0.95      | 0.95   | 0.95     | 440    |
| weighted avg | 0.95      | 0.95   | 0.95     | 440    |

```
[]
```

```
# Cross validation score (Logistic Regression)
```

```
score = cross_val_score(LogReg,features,target,cv=5)
```

```
score
```

```
array([0.95, 0.96590909, 0.94772727, 0.96590909, 0.94318182])
```

```
Saving trained Logistic Regression model
```

```
[]
```

```
import pickle
```

```
# Dump the trained Naive Bayes classifier with Pickle
```

```
LR_pkl_filename = 'LogisticRegression.pkl'
```

```

# Open the file to save as pkl file
LR_Model_pkl = open(DT_pkl_filename, 'wb')
pickle.dump(LogReg, LR_Model_pkl)

# Close the pickle instances
LR_Model_pkl.close()

Random Forest

[]

from sklearn.ensemble import RandomForestClassifier

RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain,Ytrain)

predicted_values = RF.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
RF's Accuracy is: 0.990909090909091

```

|              | precision | recall | f1-score | suppor |
|--------------|-----------|--------|----------|--------|
| accuracy     |           |        | 0.99     | 440    |
| macro avg    | 0.99      | 0.99   | 0.99     | 440    |
| weighted avg | 0.99      | 0.99   | 0.99     | 440    |

```

[]

```

```

# Cross validation score (Random Forest)

score = cross_val_score(RF,features,target,cv=5)

score

array([0.99772727, 0.99545455, 0.99772727, 0.99318182, 0.98863636])

[]

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy_score, classification_report


# Create KNN classifier with k=5 (you can adjust k as needed)

knn = KNeighborsClassifier(n_neighbors=5)


# Train the classifier

knn.fit(Xtrain, Ytrain)


# Predict on the test set

predicted_values_knn = knn.predict(Xtest)


# Calculate accuracy

accuracy_knn = accuracy_score(Ytest, predicted_values_knn)


# Append accuracy and model name to lists

acc.append(accuracy_knn)

model.append('KNN')


print("KNN's Accuracy is: ", accuracy_knn)


# Print classification report

```

```
print(classification_report(Ytest, predicted_values_knn))
```

KNN's Accuracy is: 0.975

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| accuracy     |           |        | 0.97     | 440     |
| macro avg    | 0.97      | 0.98   | 0.97     | 440     |
| weighted avg | 0.98      | 0.97   | 0.98     | 440     |

Saving trained Random Forest model

```
[ ]
```

```
import pickle
```

```
# Dump the trained Naive Bayes classifier with Pickle
```

```
RF_pkl_filename = 'RandomForest.pkl'
```

```
# Open the file to save as pkl file
```

```
RF_Model_pkl = open(RF_pkl_filename, 'wb')
```

```
pickle.dump(RF, RF_Model_pkl)
```

```
# Close the pickle instances
```

```
RF_Model_pkl.close()
```

Accuracy Comparison

```
[ ]
```

```
plt.figure(figsize=[10,5],dpi = 100)
```

```
plt.title('Accuracy Comparison')
```

```
plt.xlabel('Accuracy')
```

```
plt.ylabel('Algorithm')
```

```
sns.barplot(x = acc,y = model,palette='dark')
```

```
[ ]
```

```
accuracy_models = dict(zip(model, acc))
```

```
for k, v in accuracy_models.items():
```

```
    print(k, '-->', v)
```

```
Decision Tree --> 0.9
```

```
Naive Bayes --> 0.990909090909091
```

```
SVM --> 0.10681818181818181
```

```
Logistic Regression --> 0.9522727272727273
```

```
RF --> 0.990909090909091
```

```
Making a prediction
```

```
[]
```

```
data = np.array([[10,18, 30, 23.603016, 60.3, 6.7, 10.91]]) # features = df[['N',  
'P','K','temperature', 'humidity', 'ph', 'rainfall']]
```

```
prediction = RF.predict(data)
```

```
print(prediction)
```

```
['mothbeans']
```

```
[]
```

```
data = np.array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])
```

```
prediction = RF.predict(data)
```

```
print(prediction)
```

```
['jute']
```

```
Colab paid products - Cancel contracts here
```