CHAT BOT FOR CROP YIELD PREDICTION AND CROP RECOMMEND USING K-NEAREST NEIGHBOUR ALGORITHM

A PROJECT REPORT

#### Submitted by

J Surya RA221200504001

#### Under the Guidance of

Dr. B. Prabha

(Assistant Professor, Department of Computing Science and Engineering)

### *in partial fulfilment of the requirements* *for the degree of*

## MASTER OF TECHNOLOGY

## in

## COMPUTER SCIENCE AND ENGINEERING

## of

## FACULTY OF ENGINEERING AND TECHNOLOGY

## 

## Department of Computer Science and Engineering

## Vadapalani Campus,Chennai

## MAY 2024

### 



# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

# BONAFIDE CERTIFICATE

Certified that 21CSC201J Major project Phase - I report titled **CHAT BOT FOR CROP YIELD PREDICTION AND CROP RECOMMEND USING K-NEAREST NEIGHBOUR ALGORITHM** is the bonafide work of **J SURYA [RA221200504001] of** II year M.Tech in Computer Science and Engineering who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SUPERVISPOR HEAD OF THE DEPARTMENT**

Dr. B. Prabha B.E., M.E., PhD, Dr. S Prasanna Devi M.E, PhD, PGDHRM, PDF(IISc)

Assistant Professor, Professor and Head,

Dept of Computer Science & Engg Dept of Computer Science & Engg

SRMIST, SRMIST,

Vadapalani Campus Vadapalani Campus



Department of Computer Science and Engineering

##### **SRM Institute of Science andTechnology**

##### **Own Work Declaration Form**

This sheet must be filled in and signed with dated along with student registration number, work will not be marked unless this is done.

To be completed by the student for all assessments

##### **Degree/ Course** **: M. Tech/CSE**

**Student Name : J. Surya**

##### **Registration Number** **: RA2212005040001**

**Title of Work : Chat bot for crop yield prediction and crop**

**recommend using k-nearest neighbour algorithm**

I / We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism\*\*, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

* Clearly referenced / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above. |
| If you are working in a group, please write your registration numbers and sign with the date for every student in your group. |

**ACKNOWLEDGEMENT**

We express our humble gratitude to our Honorable Chancellor **Dr. T. R. Paarivendhar**, Pro Chancellor (Administration), **Dr. Ravi Pachamoothoo**, Pro Chancellor (Academic) **Dr. P. Sathyanarayanan** for the facilities extended for the completion of the project work.

We would record our sincere gratitude to our Vice Chancellor, **Dr. C. Muthamizhchelvan and** Registrar, **Dr. S. Ponnusamy** for their support to complete our project work by giving us the best of academic excellence support system in place. We extend our sincere thanks to our Dean, **Dr. C V Jayakumar** and Vice Principal – Academics, **Dr. C. Gomathy** and Vice Principal - Examination - **Dr. S. Karthikeyan** for their invaluable support.

We wish to thank **Dr. S Prasanna Devi**, Professor & Head, Department of CSE, SRM Institute of Science and Technology, Vadapalani Campus for her valuable suggestions and encouragement throughout the period of the project work and the course.

We are extremely grateful to our Project Coordinator, **Dr. Sheeba Santha Kumari,**  Assistant Professor, Department of CSE, SRM Institute of Science and Technology, Vadapalani Campus, for leading and helping us to complete our course.

We extend our gratitude to our guide, **Dr. B. Prabha**, Assistant Professor, Department of CSE, SRM Institute of Science and Technology, Vadapalani Campus for providing us an opportunity to pursue our project under her mentorship. She provided us the freedom and support to explore the research topics of our interest.

We sincerely thank all faculty of DCSE, staff and students of the department who have directly or indirectly helped our project.

Finally, we would like to thank our parents, our family members and our friends for their unconditional love, constant support and encouragement.

**ABSTRACT**

Agriculture remains the basis of the country’s economy, providing the main source of livelihood for most citizenry such as food, employment, income and foreign exchange as well as raw materials for the manufacturing sectors. To meet market demands, farmers must control yield quality and quantity throughout crop cultivation. Farmers must therefore be knowledgeable in order to develop and preserve crops. India relies heavily on agriculture. India is the world’s largest producer of various crops. However, without the use of better methods, the yield from the agricultural sector remains inadequate. For Indian farmers, there is no existing reliable recommendation mechanism. They face a significant difficulty in determining the optimal crop for their farming region to maximize productivity and profit. Common challenges faced by Indian farmers include (i) inability to select suitable crops based on soil properties and (ii) inability to find the health of the soil. For this reason, they face a serious setback in productivity. Giving an address to this issue, Crop yield prediction and crop recommendation as plays a crucial role in agricultural decision-making processes, enabling farmers to optimize resource allocation and plan for potential risks. In recent years, machine learning algorithms have emerged as powerful tools for predicting crop yields accurately. This abstract focuses on the application of the Decision Tree algorithm to train for crop yield prediction. Once the Decision Tree model is constructed, it can be used to predict crop yields for unseen data. New input variables, such as weather forecasts or soil measurements, can be fed into the model to obtain yield predictions and crop recommend. The interpretability of Decision Trees allows farmers to understand which factors contribute most significantly to crop yield variations and make informed decisions accordingly. User can interact with the chat bot to give details then model is predicted crop and reply by chat bot. The trained model proved advantageous in catering the farmers with a ranked list of crops deployed along with an interface for better user experience.

**TABLE OF CONTENTS**

[**ABSTRACT**](#_bookmark1) **iii**

**TABLE OF CONTENTS iv**

[**1 INTRODUCTION**](#_bookmark4) **8**

* 1. Aim 8
  2. Synopsis 8
  3. Software Requirements Specification 9

[**2 LITERATURE SURVEY**](#_bookmark20) **10**

**3 SYSTEM ANALYSIS 22**

**4** **[SYSTEM A](#_bookmark30)RCHITECTURE AND DESIGN 25**

**5 MODULE DESCRIPTION 30**

**6 LANGUAGE DESCRIPTION 34**

**7 METHODOLOGY 45**

**8** **EXPERIMENTAL RESULT & ANALYSIS 49**

**9 CONCLUSION 59**

**10 REFERENCES 60**

**CHAPTER 1**

**INTRODUCTION**

**CHAPTER 1**

**INTRODUCTION**

* 1. **AIM:**

The project's main objective is to implement chat bot using a decision tree algorithm for crop yield prediction and crop recommendation

* 1. **SYNOPSIS:**

Agriculture remains the basis of the country’s economy, providing the main source of livelihood for most citizenry such as food, employment, income and foreign exchange as well as raw materials for the manufacturing sectors. To meet market demands, farmers must control yield quality and quantity throughout crop cultivation. Farmers must therefore be knowledgeable in order to develop and preserve crops. India relies heavily on agriculture. India is the world’s largest producer of various crops. However, without the use of better methods, the yield from the agricultural sector remains inadequate. For Indian farmers, there is no existing reliable recommendation mechanism. They face a significant difficulty in determining the optimal crop for their farming region to maximize productivity and profit. Common challenges faced by Indian farmers include (i) inability to select suitable crops based on soil properties and (ii) inability to find the health of the soil. For this reason, they face a serious setback in productivity.

Giving an address to this issue, Crop yield prediction and crop recommendation as plays a crucial role in agricultural decision-making processes, enabling farmers to optimize resource allocation and plan for potential risks. In recent years, machine learning algorithms have emerged as powerful tools for predicting crop yields accurately. This abstract focuses on the application of the Decision Tree algorithm to train for crop yield prediction. Once the Decision Tree model is constructed, it can be used to predict crop yields for unseen data. New input variables, such as weather forecasts or soil measurements, can be fed into the model to obtain yield predictions and crop recommend. The interpretability of Decision Trees allows farmers to understand which factors contribute most significantly to crop yield variations and make informed decisions accordingly. User can interact with the chat bot to give details then model is predicted crop and reply by chat bot. The trained model proved advantageous in catering the farmers with a ranked list of crops deployed along with an interface for better user experience.

* 1. **Software Requirements Specification**

**Hardware Configuration:**

* Processor - i5
* Speed - 3 GHz
* RAM - 8 GB(min)
* Hard Disk - 500 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**Software Configuration:**

* Operating System: Linux, Windows/7/10
* Tools: Anaconda, Jupiter, vscode
* Front End: HTML, CSS
* Server side Script: Python , AIML

**CHAPTER 2**

**LITERATURE SURVEY**

**CHAPTER 2**

**LITERATURE SURVEY**

# SUPPORT VECTOR MACHINE EXPERIMENTS FOR ROAD RECOGNITION IN HIGH RESOLUTION IMAGES.

**YEAR OF PUBLISHING:** 2005

# AUTHOR NAME: JY Lai, A Sowmya, J Trinder

# ABSTRACT:

# Support Vector Machines have received considerable attention from the pattern recognition community in recent years. They have been applied to various classical recognition problems achieving comparable or even superior results to classifiers such as neural networks. We investigate the application of Support Vector Machines (SVMs) to the problem of road recognition from remotely sensed images using edge-based features. We present very encouraging results from our experiments, which are comparable to decision tree and neural network classifiers.

# EFFECTS OF CO2 AND TEMPERATURE ON GROWTH AND YIELD OF CROPS OF WINTER WHEAT OVER FOUR SEASONS.

**YEAR OF PUBLISHING:** 2021

# AUTHOR NAME: G R Batts

## ABSTRACT:

Crops of winter wheat ([*Triticum aestivum*](https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/triticum-aestivum) L. cv. Hereward) were grown in the field in four consecutive seasons from 1991/1992 to 1994/1995 at Reading, UK, within polyethylene-covered tunnels along which a temperature gradient was superimposed on the ambient temperature variation at normal atmospheric (ca. 370) or an increased [CO2] (ca. 700 μmol CO2 mol−1 air), producing many environments from one [sowing date](https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/sowing-date) in each season at one location. Mean seasonal temperatures varied by up to 4°C along the temperature gradient. Increased [CO2] had no effect on crop duration, or on the rate of reproductive development, which had the same temperature sensitivity across all years. A 2°C warming, on the 4-year ambient mean temperature (10°C), reduced crop duration by 42 days (from 254), and reduced the reproductive phase by 16 days (from 130). [Crop biomass](https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/crop-biomass) generally declined with increase in mean temperature, and was greater at increased [CO2], with the effect of increased [CO2] varying with temperature and between years (6–34% range in relative stimulation by increased [CO2]). Grain yield was substantially reduced by warmer temperatures, and increased by doubling [CO2], but the effect varied greatly between years and with temperature (7–168% range). There were both positive and negative interactions of temperature and increased [CO2] on biomass and grain yield. In all 4 years, the increase in grain yield from doubling [CO2] was negated by an increase in mean seasonal temperature of only 1.0–2.0°C. Year-to-year variation in the responses of biomass and grain yield to [CO2] and temperature resulted from differences in environmental conditions, influencing [biomass partitioning](https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/biomass-partitioning) and altering the role of different yield components.

# DATA MINING OF AGRICULTURAL YIELD DATA:

# A COMPARISON OF REGRESSION MODELS.

**YEAR OF PUBLISHING**: 2021

# AUTHOR NAME: G Rub

# ABSTRACT:

Nowadays, *precision agriculture* refers to the application of state-of-the-art GPS technology in connection with small-scale, sensor-based treatment of the crop. This introduces large amounts of data which are collected and stored for later usage. Making appropriate use of these data often leads to considerable gains in efficiency and therefore economic advantages. However, the amount of data poses a data mining problem – which should be solved using data mining techniques. One of the tasks that remains to be solved is *yield prediction* based on available data. From a data mining perspective, this can be formulated and treated as a multi-dimensional regression task. This paper deals with appropriate regression techniques and evaluates four different techniques on selected agriculture data. A recommendation for a certain technique is provided.

# SHORT TERM FORECASTING OF AIR POLLUTION EPISODES

**YEAR OF PUBLISHING**: 2021

# AUTHOR NAME: Jorquera H

ABSTRACT:

# OAuth 2.0 is a delegated authorization framework enabling secure authorization for applications running on various kinds of platforms. In healthcare services, OAuth allows the patient (resource owner) seeking real time clinical care to authorize automatic monthly payments from his bank account (resource server) without the patient being required to supply his credentials to the clinic (client app). OAuth 2.0 achieves this with the help of tokens issued by an authorization server which enables validated access to a protected resource. To ensure security, access tokens have an expiry time and are short-lived. So the clinical app may use a refresh token to obtain a new access token to cash monthly payments for rendering real time health care services. Refresh tokens need secure storage to ensure they are not leaked, since any malicious party can use them to obtain new access and refresh tokens. Since OAuth 2.0 has dropped signatures and relies completely on SSL/TLS, it is vulnerable to phishing attack when accessing interoperable APIs. In this paper, we develop an approach that combines JSON web token (JWT) with OAuth 2.0 to request an OAuth access token from authorization server when a client wishes to utilize a previous authentication and authorization. Experimental evaluation confirms that the proposed scheme is practically efficient, removes secure storage overhead by removing the need to have or store refresh token, uses signature and prevents different security attacks which is highly desired in health care services using an IOT cloud platform.

# A REAL TIME GRADING METHOD OF APPLES BASED ON FEATURES EXTRACTED FROM DEFECTS

**YEAR OF PUBLISHING**: 2021

# AUTHOR NAME: Leemans V

ABSTRACT:

# As grading results of apples based on the single feature such as size, shape or color are not accurate, this paper proposes a multi-feature information fusion method based on BP neural network and D-S evidential theory to improve the accuracy of apple grading. Firstly, size, shape and color features are extracted from the processed images of apples. Secondly, apples are classified with each kind of feature by BP network classifier and as independent evidences, the outputs of classifiers are combined to construct the basic probability assignment (BPA). Finally, using D-S fusion rules of evidences to make the decision and achieve the final grading result. The experimental results have shown that the decision information fusion method based on size, shape or color features has good performance on accuracy compared to the single feature-based method in apple grading.

# ANALYSIS, FEATURE EXTRACTION, MODELING AND TESTING TECHNIQUES FOR SPEAKER RECOGNITION.

**YEAR OF PUBLISHING**: 2021

# AUTHOR NAME: Welch Allyn Spot vital Signs

ABSTRACT:

# To validate a significantly lower priced, new-generation blood pressure device by using the ANSI/AAMI SP10:2002 Standard evaluation. The Spot Vital Signs device evaluation was performed using the procedure specified in the SP10: 2002 American National Standard. Both obese and hypertensive participants were overrepresented in the study population. The mean+/-standard deviation for the device readings compared with auscultation by paired, trained, blinded observers for systolic blood pressure was -1.0+/-4.1 mmHg; for diastolic blood pressure it was -0.4+/-6.1 mmHg. These passed AAMI criteria. The device, at a cost well below those of comparable professional-quality (vs. self measurement) electronic blood pressure devices, was extremely accurate. The Welch Allyn Spot Vital Signs device provides a cost-effective alternative for medical-care delivery sites where accuracy and cost are both high priorities. The study population was oversampled for obese and hypertensive patients, the major target groups in whom blood pressure must be able to be recorded accurately to optimize diagnosis and management of cardiovascular disease.

# PREDICTIVE ABILITY OF MACHINE LEARNING METHODS FOR MASSIVE CROP YIELD PREDICTION

**YEAR OF PUBLISHING**: 2021

# AUTHOR NAME: [Alberto González-Sanchez](https://www.researchgate.net/profile/Alberto-Gonzalez-Sanchez-2)

ABSTRACT:

# An important issue for agricultural planning purposes is the accurate yield estimation for the numerous crops involved in the planning. Machine learning (ML) is an essential approach for achieving practical and effective solutions for this problem. Many comparisons of ML methods for yield prediction have been made, seeking for the most accurate technique. Generally, the number of evaluated crops and techniques is too low and does not provide enough information for agricultural planning purposes. This paper compares the predictive accuracy of ML and linear regression techniques for crop yield prediction in ten crop datasets. Multiple linear regression, M5-Prime regression trees, perceptron multilayer neural networks, support vector regression and k-nearest neighbor methods were ranked. Four accuracy metrics were used to validate the models: the root mean square error (RMS), root relative square error (RRSE), normalized mean absolute error (MAE), and correlation factor (R). Real data of an irrigation zone of Mexico were used for building the models. Models were tested with samples of two consecutive years. The results show that M5- Prime and k-nearest neighbor techniques obtain the lowest average RMSE errors (5.14 and 4.91), the lowest RRSE errors (79.46% and 79.78%), the lowest average MAE errors (18.12% and 19.42%), and the highest average correlation factors (0.41 and 0.42). Since M5-Prime achieves the largest number of crop yield models with the lowest errors, it is a very suitable tool for massive crop yield prediction in agricultural planning.

# RECURRENT NEURAL NETWORKS FOR PREDICTION: LEARNING ALGORITHMS, ARCHITECTURES AND STABILITY

**YEAR OF PUBLISHING**: 2021

# AUTHOR NAME: Mandic D. P

ABSTRACT:

# An important issue for agricultural planning purposes is the accurate yield estimation for the numerous crops involved in the planning. Machine learning (ML) is an essential approach for achieving practical and effective solutions for this problem. Many comparisons of ML methods for yield prediction have been made, seeking for the most accurate technique. Generally, the number of evaluated crops and techniques is too low and does not provide enough information for agricultural planning purposes. This paper compares the predictive accuracy of ML and linear regression techniques for crop yield prediction in ten crop datasets. Multiple linear regression, M5-Prime regression trees, perceptron multilayer neural networks, support vector regression and k-nearest neighbor methods were ranked. Four accuracy metrics were used to validate the models: the root mean square error (RMS), root relative square error (RRSE), normalized mean absolute error (MAE), and correlation factor (R). Real data of an irrigation zone of Mexico were used for building the models. Models were tested with samples of two consecutive years. The results show that M5- Prime and k-nearest neighbor techniques obtain the lowest average RMSE errors (5.14 and 4.91), the lowest RRSE errors (79.46% and 79.78%), the lowest average MAE errors (18.12% and 19.42%), and the highest average correlation factors (0.41 and 0.42). Since M5-Prime achieves the largest number of crop yield models with the lowest errors, it is a very suitable tool for massive crop yield prediction in agricultural planning.

1. LONG SHORT-TERM MEMORY. NEURAL COMPUTATION.

# YEAR OF PUBLISHING: 2021

# AUTHOR NAME: Hochreiter S.

ABSTRACT:

# Learning to store information over extended time intervals by recurrent backpropagation takes a very long time, mostly because of insufficient, decaying error backflow. We briefly review Hochreiter's (1991) analysis of this problem, then address it by introducing a novel, efficient, gradient based method called long short-term memory (LSTM). Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete-time steps by enforcing constant error flow through constant error carousels within special units. Multiplicative gate units learn to open and close access to the constant error flow. LSTM is local in space and time; its computational complexity per time step and weight is O. 1. Our experiments with artificial data involve local, distributed, real-valued, and noisy pattern representations. In comparisons with real-time recurrent learning, back propagation through time, recurrent cascade correlation, Elman nets, and neural sequence chunking, LSTM leads to many more successful runs, and learns much faster. LSTM also solves complex, artificial long-time-lag tasks that have never been solved by previous recurrent network algorithms.

# LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORK ARCHITECTURES FOR LARGE SCALE ACOUSTIC MODELING.

**YEAR OF PUBLISHING**: 2021

# AUTHOR NAME: H. Sak, F. Beaufays

ABSTRACT:

# Long Short-Term Memory (LSTM) is a specific recurrent neural network (RNN) architecture that was designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs. In this paper, we explore LSTM RNN architectures for large scale acoustic modeling in speech recognition. We recently showed that LSTM RNNs are more effective than DNNs and conventional RNNs for acoustic modeling, considering moderately-sized models trained on a single machine. Here, we introduce the first distributed training of LSTM RNNs using asynchronous stochastic gradient descent optimization on a large cluster of machines. We show that a two-layer deep LSTM RNN where each LSTM layer has a linear recurrent projection layer can exceed state-of-the-art speech recognition performance. This architecture makes more effective use of model parameters than the others considered, converges quickly, and outperforms a deep feed forward neural network having an order of magnitude more parameters.

# CLASSIFICATION AND REGRESSION BY RANDOMFOREST

**YEAR OF PUBLISHING**: 2021

# AUTHOR NAME: Wiener M., Andy Liaw

ABSTRACT:

# Recently there has been a lot of interest in “ensemble learning” — methods that generate many classifiers and aggregate their results. Two well-known methods are boosting (see, e.g., Shapire et al., 1998) and bagging Breiman (1996) of classification trees. In boosting, successive trees give extra weight to points incorrectly predicted by earlier predictors. In the end, a weighted vote is taken for prediction. In bagging, successive trees do not depend on earlier trees — each is independently constructed using a bootstrap sample of the data set. In the end, a simple majority vote is taken for prediction. proposed random forests, which add an additional layer of randomness to bagging. In addition to constructing each tree using a different bootstrap sample of the data, random forests change how the classification or regression trees are constructed. In standard trees, each node is split using the best split among all variables. In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node. This somewhat counterintuitive strategy turns out to perform very well compared to many other classifiers, including discriminant analysis, support vector machines and neural networks, and is robust against overfitting (Breiman, 2001). In addition, it is very user-friendly in the sense that it has only two parameters (the number of variables in the random subset at each node and the number of trees in the forest), and is usually not very sensitive to their values.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**CHAPTER 3**

**SYSTEM ANALYSIS**

**Existing System:**

The mechanism of the proposed system. We compared different combinations and their accuracy rates simultaneously. We have included Decision Tree Regressor, AdaBoost Regressor, Random Forest Regressor, KNN algorithms. The parameters we have considered constitute of Country, crop\_name, Year, Yield\_value, Avg\_rainfall, Pesticide\_tonns, Avg\_temp. By comparing the metric values of every ensemble, we found the best ensemble model among all the other models considered. After determining the best ensemble model, the system is ready for creation of prediction model.

**Disadvantage:**

* There is less accuracy compare to proposed system.
* Details give in the text box and prediction.
* Have not crop recommend.

**Proposed System:**

The system begins by collecting historical agricultural data, including variables such as weather conditions, soil characteristics, crop management practices, and previous yield records then suitable crops. The Decision Tree algorithm is then utilized to construct a predictive model. It partitions the data recursively, creating a tree structure where each internal node represents a decision based on a specific input variable, and each leaf node represents a predicted crop yield outcome. The integration of a Chatbot makes the crop yield prediction and crop recommend system easily accessible and user-friendly. Farmers, researchers, or any stakeholders can interact with the Chatbot, eliminating the need for specialized technical knowledge or complex data analysis.

**Advantage:**

* The decision tree algorithm gives more accuracy for prediction.
* User interacts with Chatbot to predict forming.

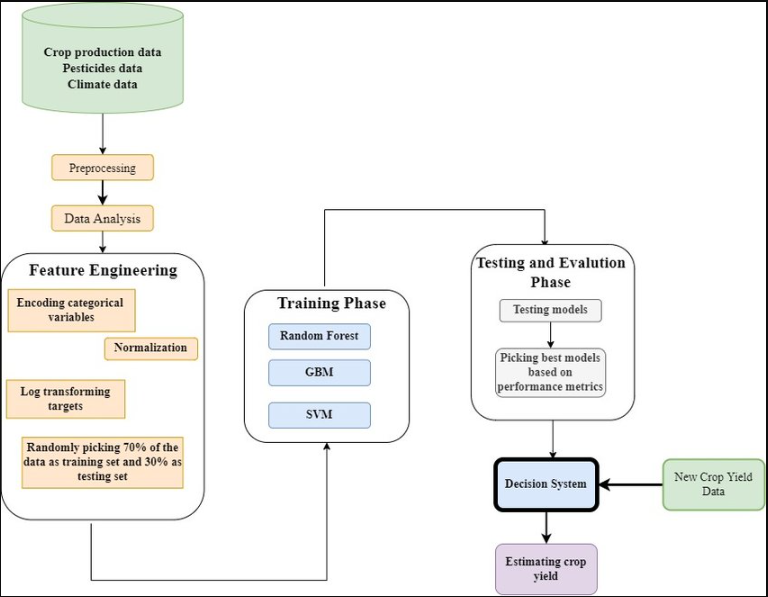
**CHAPTER 4**

**SYSTEM ARCHITECTURE AND DESIGN**

**CHAPTER 4**

**SYSTEM ARCHITECTURE AND DESIGN**

**System architecture diagram:**



**KNN**

**Decision Tree**

**Dataflow Diagram:**

Give data to chat bot for crop recommend

Give data to chat bot for crop yield

Pre-processing data

Load model

Prediction of crop

Prediction of crop

Show result

**CLASS DIAGRAM:**

Process

Pre-process raw data

-format: .jpg

image

-format: .jpg

image

-format: .jpg

image

Data give by user

image

Chat bot

recognition

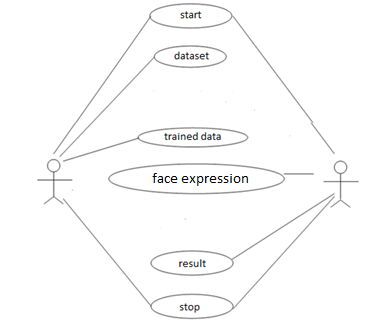
Model

Model predict

Result

Predict the crop yield and crop recommed

**USE-CASE DIAGRAM:**



**CHAPTER 5**

**Module Description**

**CHAPTER 5**

**MODULE DESCRIPTION**

In a software development project, it's crucial to organize the code into modular components to enhance maintainability, reusability, and collaboration among team members. Here are some key modules for the "Chat Bot for Crop Yield Prediction and Crop Recommendation" project:

1. **Data Collection and Preprocessing Module**

This module is responsible for collecting, cleaning, and preprocessing data from various sources, including crop yield data, soil data, and weather data. It includes submodules like:

Data Scraper: Collects data from external sources.

Data Cleaner: Handles missing values and data quality issues.

Feature Engineer: Extracts relevant features from the collected data.

Data Integrator: Combines data from different sources into a unified dataset.

1. **Decision Tree Model Module**

This module focuses on developing, training, and evaluating the decision tree models for crop yield prediction. It includes submodules like:

Model Trainer: Trains decision tree models on the prepared dataset.

Model Evaluator: Measures the model's performance using appropriate metrics.

Model Tuner: Fine-tunes the model's hyperparameters for better accuracy.

Model Serializer: Saves and loads trained models for future use.

1. **Chat Bot Interface Module**

This module deals with the user interface of the chat bot, enabling users to interact with the system. It includes submodules like:

User Input Processor: Handles user queries and extracts relevant information.

NLP (Natural Language Processing): Understands and interprets user messages.

Chat Bot Logic: Contains the core logic for responding to user queries and providing predictions and recommendations.

Visualizer: Generates graphs, charts, or visual aids for better user understanding.

1. **Crop Recommendation Module**

This module is responsible for suggesting suitable crops to users based on the decision tree predictions and environmental factors. It includes submodules like:

Recommendation Algorithm: Implements the logic for crop recommendations.

Customization Options: Allows users to customize recommendations based on their preferences.

Crop Database: Stores information about various crops and their characteristics.

**CHAPTER 6**

**LANGUAGE DESCRIPTION**

**CHAPTER 6**

**LANGUAGE DESCRIPTION**

**PYTHON:**

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.It ranges from simple automation tasks to gaming, web development, and even complex enterprise systems. These are the areas where this technology is still the king with no or little competence: Machine learning as it has a plethora of libraries implementing machine learning algorithms.Python is a one-stop shop and relatively easy to learn, thus quite popular now. What other reasons exist for such universal popularity of this programming language and what companies have leveraged its opportunities to the max? Let’s talk about that. Python technology is quite popular among programmers, but the practice shows that business owners are also Python development believers and for good reason. Software developers love it for its straightforward syntax and reputation as one of the easiest programming languages to learn. Business owners or CTOs appreciate the fact that there’s a framework for pretty much anything – from web apps to machine learning. Moreover, it is not just a language but more a technology platform that has come together through a gigantic collaboration from thousands of individual professional developers forming a huge and peculiar community of aficionados. So what is python used for and what are the tangible benefits the language brings to those who decided to use it? Below we’re going to discover that. Productivity and Speed It is a widespread theory within development circles that developing Python applications is approximately up to 10 times faster than developing the same application in Java or C/C++. The impressive benefit in terms of time saving can be explained by the clean object-oriented design, enhanced process control capabilities, and strong integration and text processing capacities. Moreover, its own unit testing framework contributes substantially to its speed and productivity.

**PYCHARM:**

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

Choose the best PyCharm for you﻿.

**PyCharm is available in three editions:**

* Community (free and open-sourced): for smart and intelligent Python development, including code assistance, refactorings, visual debugging, and version control integration.
* Professional (paid) : for professional Python, web, and data science development, including code assistance, refactorings, visual debugging, version control integration, remote configurations, deployment, support for popular web frameworks, such as Django and Flask, database support, scientific tools (including Jupyter notebook support), big data tools.
* Edu (free and open-sourced): for learning programming languages and related technologies with integrated educational tools.
* For details, see the editions comparison matrix.

**Supported languages:﻿**

To start developing in Python with PyCharm you need to download and install Python from python.org depending on your platform.

PyCharm supports the following versions of Python:

Python 2: version 2.7

Python 3: from the version 3.6 up to the version 3.10

Besides, in the Professional edition, one can develop Django, Flask, and Pyramid applications. Also, it fully supports HTML (including HTML5), CSS, JavaScript, and XML: these languages are bundled in the IDE via plugins and are switched on for you by default. Support for the other languages and frameworks can also be added via plugins (go to Settings | Plugins or PyCharm | Preferences | Plugins for macOS users, to find out more or set them up during the first IDE launch).

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Minimum** | **Recommended** |
| RAM | 4 GB of free RAM | 8 GB of total system RAM |
| CPU | Any modern CPU | Multi-core CPU. PyCharm supports multithreading for different operations and processes making it faster the more CPU cores it can use. |
| Disk space | 2.5 GB and another 1 GB for caches | SSD drive with at least 5 GB of free space |
| Monitor resolution | 1024x768 | 1920×1080 |
| Operating system | Officially released 64-bit versions of the following:   * Microsoft Windows 8 or later * macOS 10.13 or later * Any Linux distribution that supports Gnome, KDE, or Unity DE. PyCharm is not available for some Linux distributions, such as RHEL6 or CentOS6, that do not include [GLIBC](https://ftp.gnu.org/gnu/libc/) 2.14 or later.Pre-release versions are not supported. | Latest 64-bit version of Windows, macOS, or Linux (for example, Debian, Ubuntu, or RHEL) |

**SUPPORTED PLATFORMS﻿:**

[**Jupyter Notebook**](https://doc.cocalc.com/jupyter.html#id13)

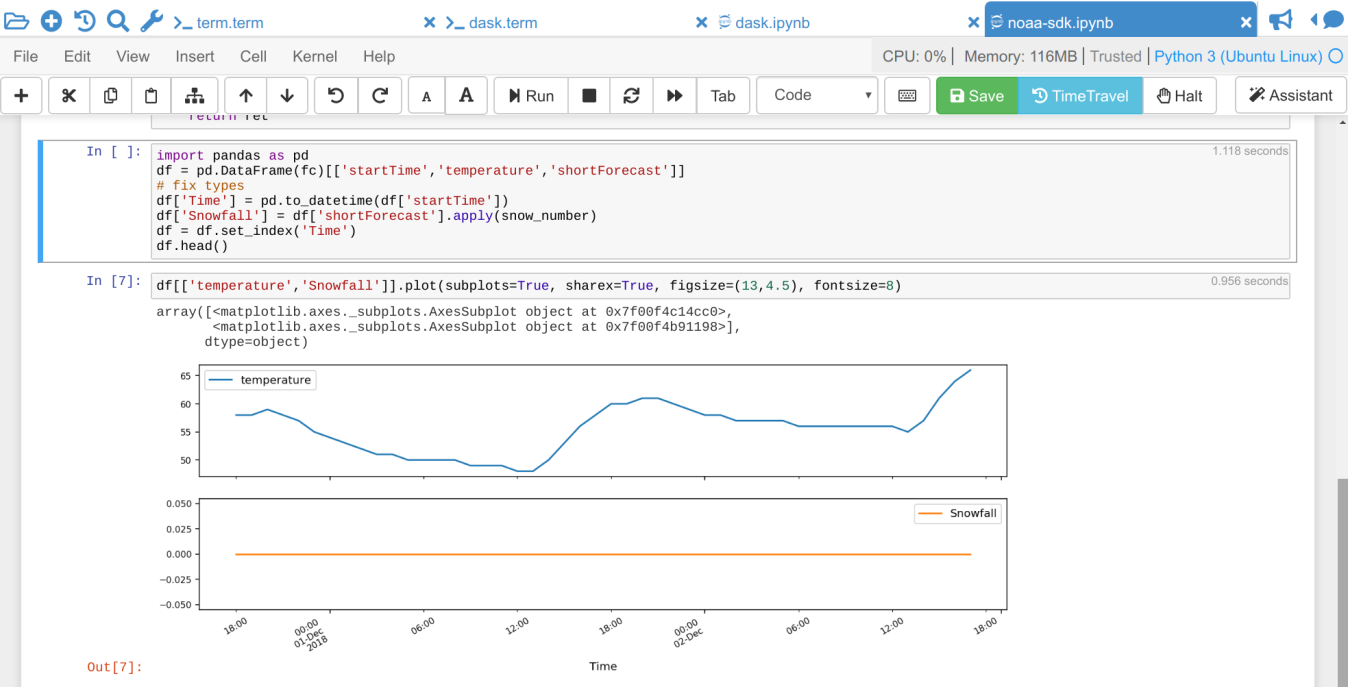
A Jupyter notebook is a specific filetype with the ending .ipynb, which records an interactive session with a **Kernel**. It made up of cells, which can either store one or more lines of code or formatted text. When you run a cell – which evaluates the piece of code in the cell via the active kernel session – you can see its output after the calculation is done. This combination of communicating back and forth with a kernel and adding descriptive text makes this form of document very attractive.

## [Jupyter Kernels](https://doc.cocalc.com/jupyter.html#id14)

You can choose the programming language and environment by selecting a Jupyter kernel for the notebook. Popular choices are [Python3](https://docs.python.org/3/), [SageMath](https://www.sagemath.org/), and [R](https://www.r-project.org/about.html). There many others. Our page on [Jupyter Kernel Selection](https://doc.cocalc.com/howto/jupyter-kernel-selection.html) shows how to set the kernel.

**JUPYTER NOTEBOOK BASIC:**

By default, a Jupyter notebook on CoCalc has all CoCalc’s core features, including real-time collaboration, side chat, and TimeTravel. Read more in our [blogpost](http://blog.sagemath.com/jupyter/2017/05/05/jupyter-rewrite-for-smc.html). The basic user interface looks like the following:

[](https://doc.cocalc.com/_images/jupyter-notebook-cocalc-1.png)

Above the main area is a menu bar and a button row:

* The **menu bar** contains all commands, and in particular the **Kernel** menu is for changing it if necessary.
* The **button row** gives you a one-click access to Run the current cell (otherwise press your Shift+Return keys), a way to restart the kernel (which clears the current session) and a Save button to make sure CoCalc has stored the file. The [Time Travel](https://doc.cocalc.com/time-travel.html) button allows you to see previous versions of that notebook, such that you can go back in time to recover from a bad change.
* **Active cell**: in the screenshot above, the blue bar on the left and a blue border around a cell indicates that this is the currently active one. Actions like Run, Delete Cell, etc. operate on the currently selected cell. It is also possible to select more than one cell.
* **Execution counter**: On the left of each cell, there is an execution counter  The number  increases each time a cell is being run. After the kernel stopped and restarted, that counter starts again at 1.
* The **output of code cells** is below the input cell. For example, is the output of cell  In the right hand corner of the input cell is some information about how long it took to calculate the result.
* **Text cells** are slightly different. Select “Markdown” in the dropdown menu in the button bar to change a code cell to such a markdown text cell. There, you can use [Markdown](https://www.markdownguide.org/basic-syntax) to format the text. Similar to code-cells, either Run these text cells to see the processed Markdown code or press Shift+Return. To edit a text cell, either double click it or press your Return key.
* **Saving**: more general, the nice things about Jupyter Notebooks is that they save all your intput and output in one single file. This means you can download or publish the notebook as it is, and everyone else sees it in exactly the same way.

**ANACONDA PYTHON**

Anaconda® is a package manager, an environment manager, a Python/R data science distribution, and a collection of [over 7,500+ open-source packages](https://docs.anaconda.com/anaconda/packages/pkg-docs/). Anaconda is free and easy to install, and it offers [free community support](https://groups.google.com/a/anaconda.com/forum/?fromgroups#!forum/anaconda).

Get the Anaconda Cheat Sheet and then [download Anaconda](https://www.anaconda.com/downloads).

Want to install conda and use conda to install just the packages you need? Get [Miniconda](http://conda.pydata.org/miniconda.html).

**Anaconda Navigator or conda?**

After you install Anaconda or Miniconda, if you prefer a desktop graphical user interface (GUI) then use [Navigator](https://docs.anaconda.com/anaconda/navigator/). If you prefer to use Anaconda prompt (or terminal on Linux or macOS), then use that and conda. You can also switch between them.

You can install, remove, or update any Anaconda package with a few clicks in Navigator, or with a single conda command in Anaconda Prompt (terminal on Linux or macOS).

* **To try Navigator**, after installing Anaconda, click the Navigator icon on your operating system’s program menu, or in Anaconda prompt (or terminal on Linux or macOS), run the command anaconda-navigator.
* **To try conda**, after installing Anaconda or Miniconda, take the [20-minute conda test drive](https://conda.io/projects/conda/en/latest/user-guide/getting-started.html) and download a [conda cheat sheet](https://docs.conda.io/projects/conda/en/latest/user-guide/cheatsheet.html).

**Packages available in Anaconda**

* Over [250 packages](https://docs.anaconda.com/anaconda/packages/pkg-docs/) are automatically installed with Anaconda.
* Over 7,500 additional open-source packages (including R) can be individually installed from the Anaconda repository with the conda install command.
* Thousands of other packages are available from [Anaconda.org](https://anaconda.org/).
* You can download other packages using the pip install command that is installed with Anaconda. [Pip packages](https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-pkgs.html#installing-non-conda-packages) provide many of the features of conda packages and in some cases they can work together. However, the preference should be to install the conda package if it is available.
* You can also make your own [custom packages](https://conda.io/projects/conda-build/en/latest/) using the conda build command, and you can share them with others by uploading them to [Anaconda.org](http://anaconda.org/), PyPI, or other repositories.

**Previous versions**

Previous versions of Anaconda are available in the [archive](https://repo.anaconda.com/archive/). For a list of packages included in each previous version, see [Old package lists](https://docs.anaconda.com/anaconda/packages/oldpkglists/).

Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it does not matter which one you download, because you can create new environments that include any version of Python packaged with conda. See [Managing Python with conda](https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-python.html).

**CHAPTER 7**

**METHODOLOGY**

**CHAPTER 7**

**METHODOLOGY**

**1. Data Collection:**

**1.1. Crop Data Collection**

Collect historical crop yield data for various crops from reliable sources, such as agricultural departments, research institutions, and online databases.

Gather information on crop characteristics, growth patterns, and environmental requirements.

**1.2. Soil and Weather Data Collection**

Acquire soil data including pH levels, nutrient content, and texture for the target geographical region.

Retrieve weather data including temperature, precipitation, humidity, and seasonal patterns for the region.

**2. Data Preprocessing:**

**2.1. Data Cleaning**

Handle missing values, outliers, and inconsistencies in the collected data.

Ensure data uniformity and compatibility.

**2.2. Feature Engineering**

Extract relevant features from the collected data, such as growing season length, frost dates, and soil moisture content.

Create a dataset that combines crop, soil, and weather data.

**3. Decision Tree Model Development:**

**3.1. Training Data Preparation**

Split the dataset into training and validation sets.

Encode categorical variables and normalize numerical variables as necessary.

**3.2. Decision Tree Training**

Develop a decision tree model using the training data.

Experiment with different decision tree algorithms (e.g., ID3, C4.5, CART) and hyperparameter settings.

Evaluate model performance using appropriate metrics like accuracy, F1-score, and RMSE (Root Mean Square Error) for yield prediction.

**3.3. Model Validation and Testing**

Validate the decision tree model using the validation dataset.

Fine-tune the model if necessary based on validation results.

Test the model's predictive capability on unseen data to ensure generalization.

**4. Chat Bot Development:**

**4.1. User Interface**

Create a user-friendly chat interface for farmers to interact with the chat bot.

Allow users to input their geographical location, soil type, and any relevant weather conditions.

**4.2. Integration**

Integrate the trained decision tree model into the chat bot's backend for yield prediction and recommendation.

**4.3. Natural Language Processing (NLP)**

Implement NLP techniques to understand user queries and respond appropriately.

Provide clear explanations and visualizations to help users interpret results.

**5. Crop Recommendation**

**5.1. Recommendation Algorithm**

Develop a recommendation algorithm that considers the decision tree predictions and suggests suitable crops based on predicted yields and environmental conditions.

**5.2. Customization**

Allow users to customize recommendations based on their priorities (e.g., profit, sustainability, crop rotation).

**CHAPTER 8**

**EXPERIMENTAL RESULT & ANALYSIS**

**CHAPTER 8**

**EXPERIMENTAL RESULT & ANALYSIS**

1. **Decision Tree Algorithm:**
   1. **Introduction**

Decision trees are a popular and versatile machine learning algorithm used for both classification and regression tasks. They are widely employed in various fields, including finance, healthcare, marketing, and more. In this document, we will explore the decision tree algorithm, its functioning, construction, pruning, and applications.

* 1. **Overview of Decision Trees**

A decision tree is a hierarchical structure that resembles an upside-down tree, with a single root node branching into multiple decision nodes, which in turn lead to leaves representing class labels or numerical values. Each internal node in the tree represents a decision based on a feature's value, and each leaf node holds the predicted output.

* 1. **How Decision Trees Work**

The decision tree algorithm works by recursively splitting the dataset into subsets based on the feature that best separates the data at each node. The algorithm aims to minimize impurity or maximize information gain (entropy reduction) with each split. Common impurity measures include Gini impurity and entropy. The algorithm continues to split the data until a stopping criterion is met, such as reaching a maximum depth or a minimum number of samples per leaf.

1. **Random Forest Algorithm:**
   1. **Introduction**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

* 1. **Overview of Random Forest**

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

* 1. **How Random Forest Work**

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase. The algorithm aims to eliminate overfitting as the result is based on a majority vote or average. Each decision tree formed is independent of the others, demonstrating the parallelization property. For classification and regression, accordingly, the final output of the algorithm is based on Majority Voting or Averaging.

1. **KNN (K- Nearest Neighbour):**
   1. **Introduction**

The K-Nearest Neighbors (KNN) algorithm is a robust and intuitive machine learning method employed to tackle classification and regression problems. By capitalizing on the concept of similarity, KNN predicts the label or value of a new data point by considering its K closest neighbours in the training dataset. It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

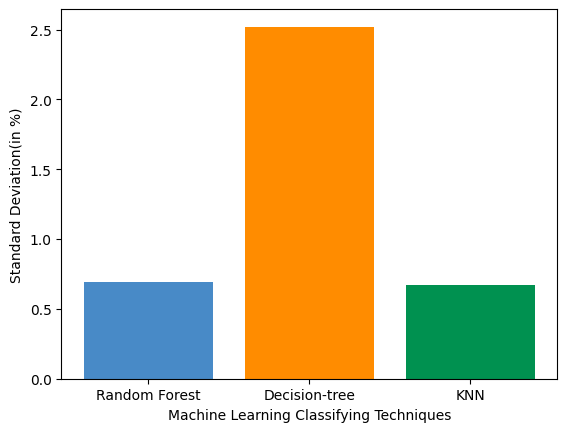
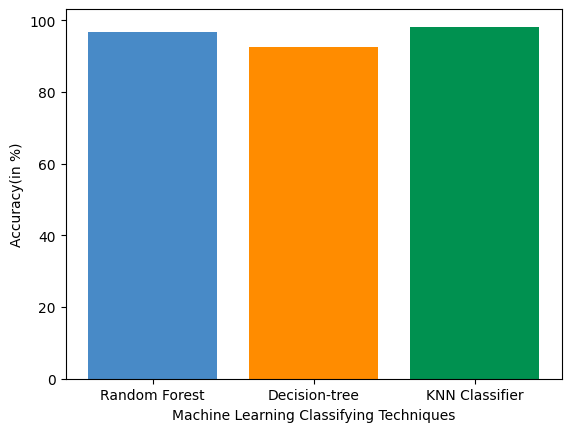
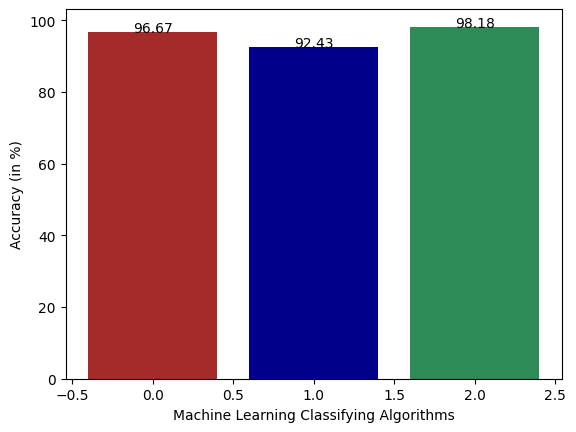
* 1. **Overview of KNN**

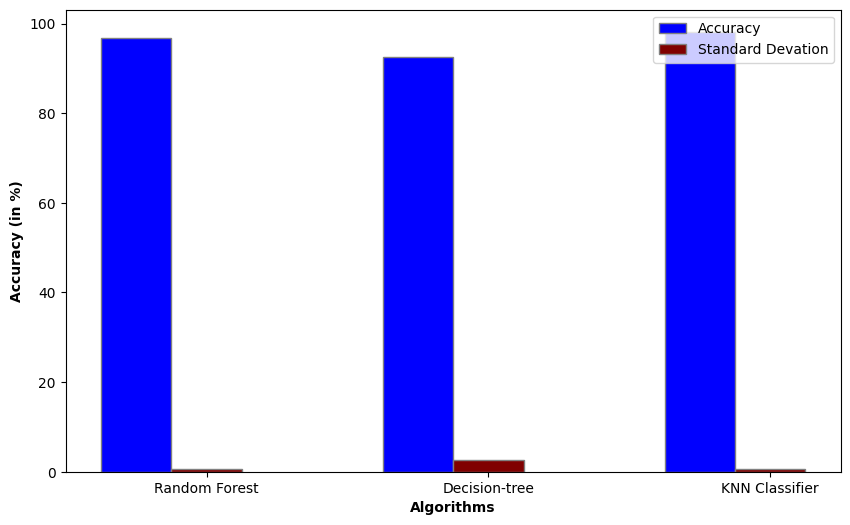
(K-NN) algorithm is a versatile and widely used machine learning algorithm that is primarily used for its simplicity and ease of implementation. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection. It does not require any assumptions about the underlying data distribution. It can also handle both numerical and categorical data, making it a flexible choice for various types of datasets in classification and regression tasks. It is a non-parametric method that makes predictions based on the similarity of data points in a given dataset. K-NN is less sensitive to outliers compared to other algorithms.

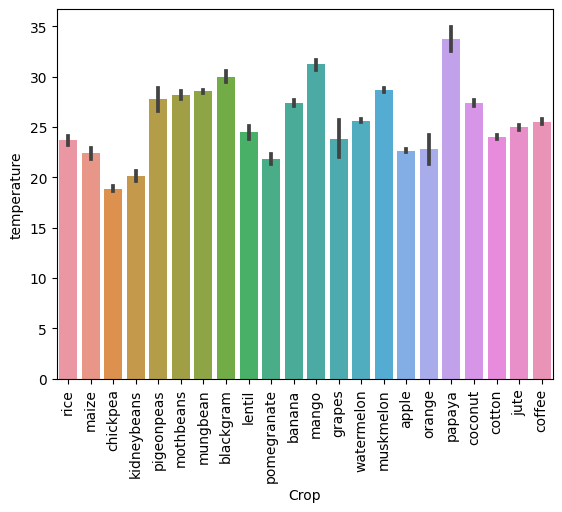
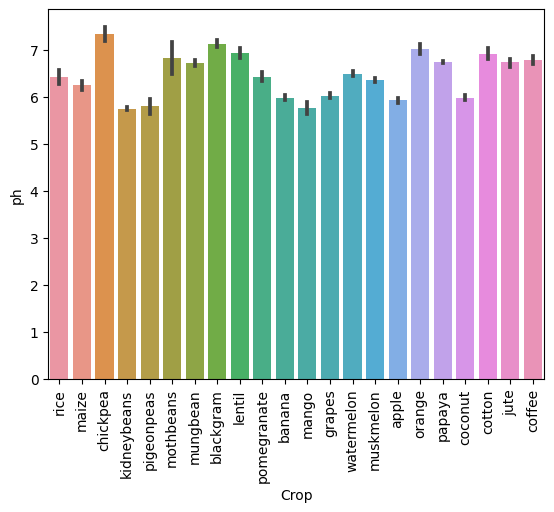
* 1. **How KNN Works**

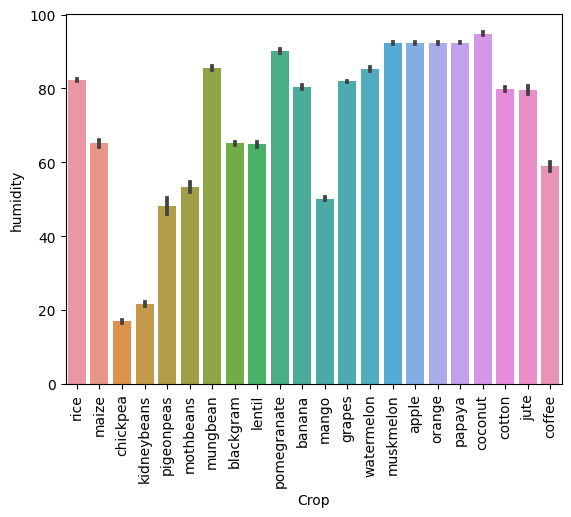
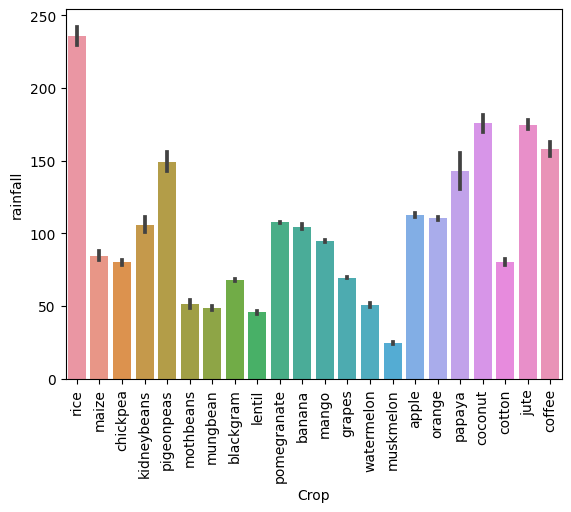
The K-NN algorithm works by finding the K nearest neighbors to a given data point based on a distance metric, such as Euclidean distance. The class or value of the data point is then determined by the majority vote or average of the K neighbors. To make predictions, the algorithm calculates the distance between each new data point in the test dataset and all the data points in the training dataset. This approach allows the algorithm to adapt to different patterns and make predictions based on the local structure of the data. Once the distances between the new data point and all the data points in the training dataset are calculated, the algorithm proceeds to find the K nearest neighbours based on these distances. After identifying the K nearest neighbors, the algorithm makes predictions based on the labels or values associated with these neighbors. For classification tasks, the majority class among the K neighbors is assigned as the predicted label for the new data point. For regression tasks, the average or weighted average of the values of the K neighbors is assigned as the predicted value.

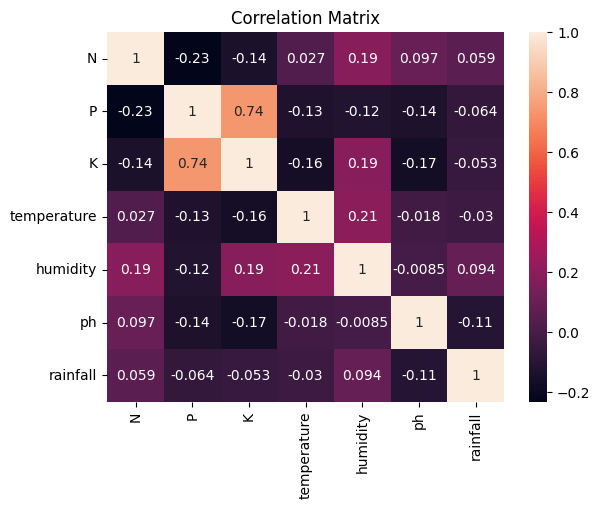
|  |  |  |  |
| --- | --- | --- | --- |
|  | Algorithms | Accuracy | Standard Deviation |
| 0 | Random Forest | 96.67 | 0.691015 |
| 1 | Decision-tree | 92.43 | 2.520343 |
| 2 | KNN Classifier | 98.18 | 0.668450 |

**** ****

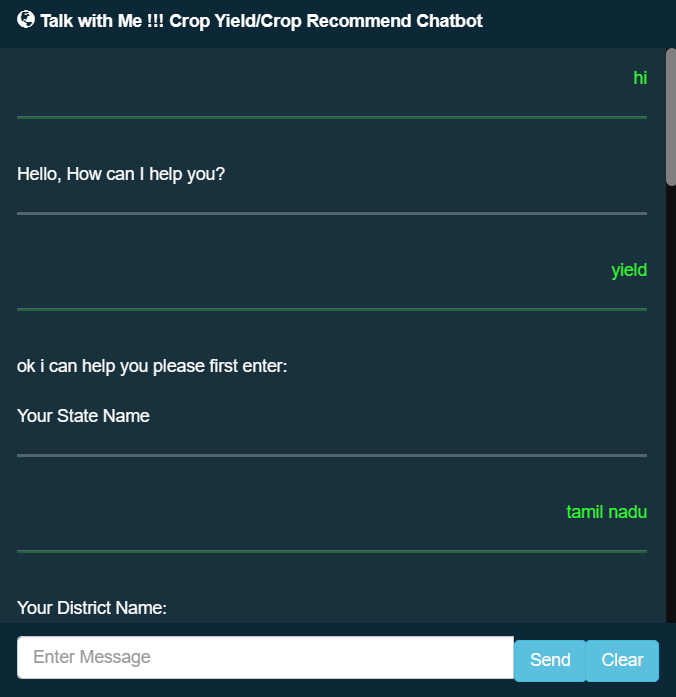
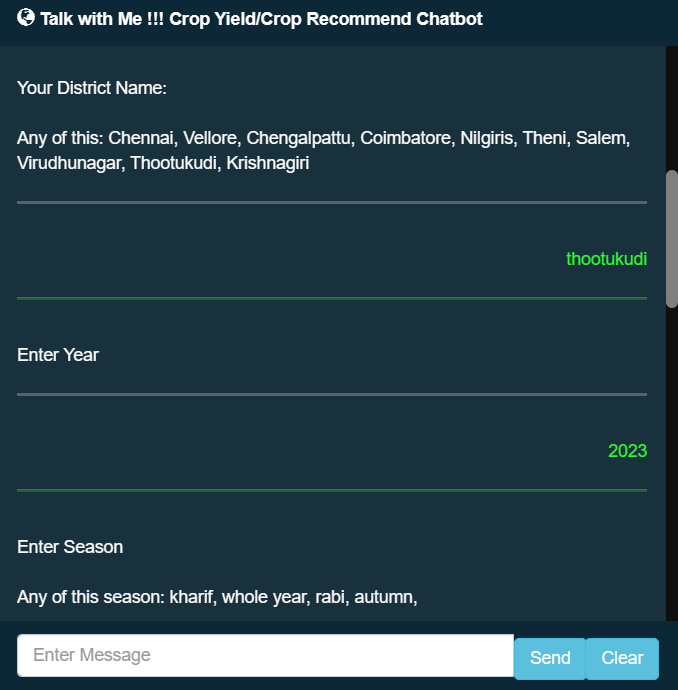
****

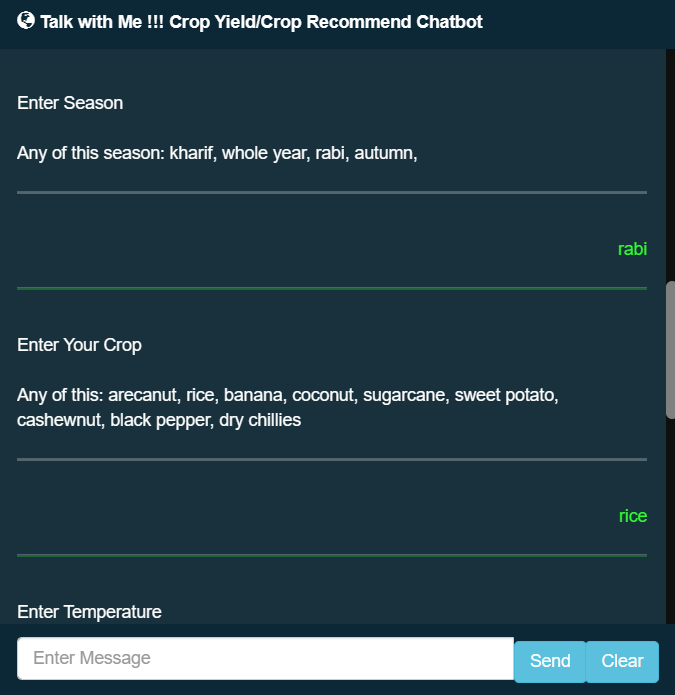
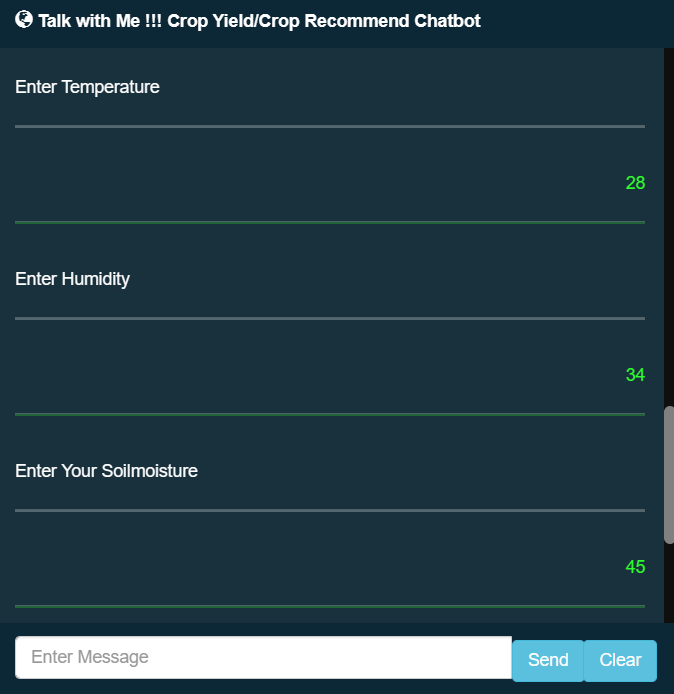
** **

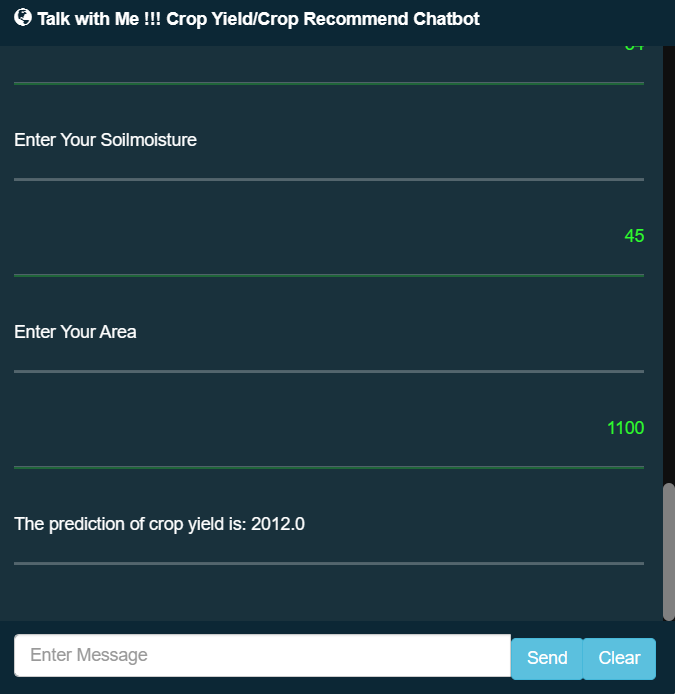
** **

****

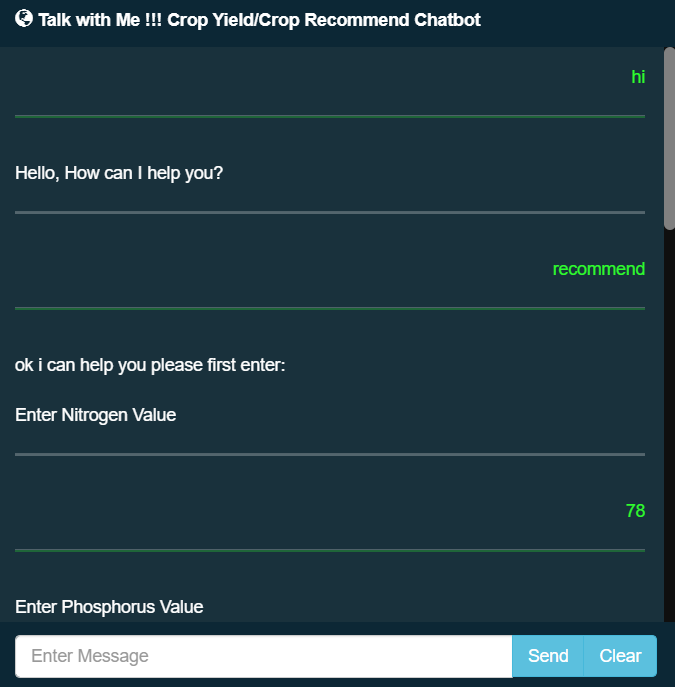
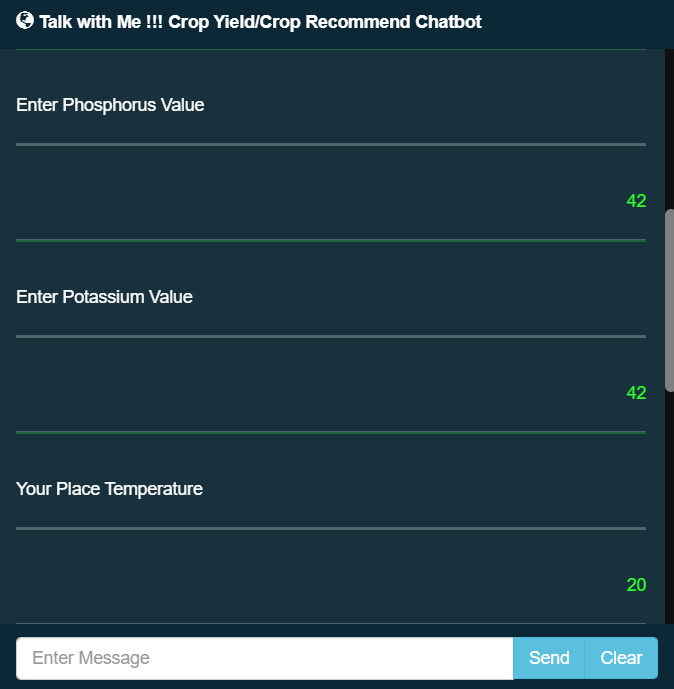
1. **Yield:**

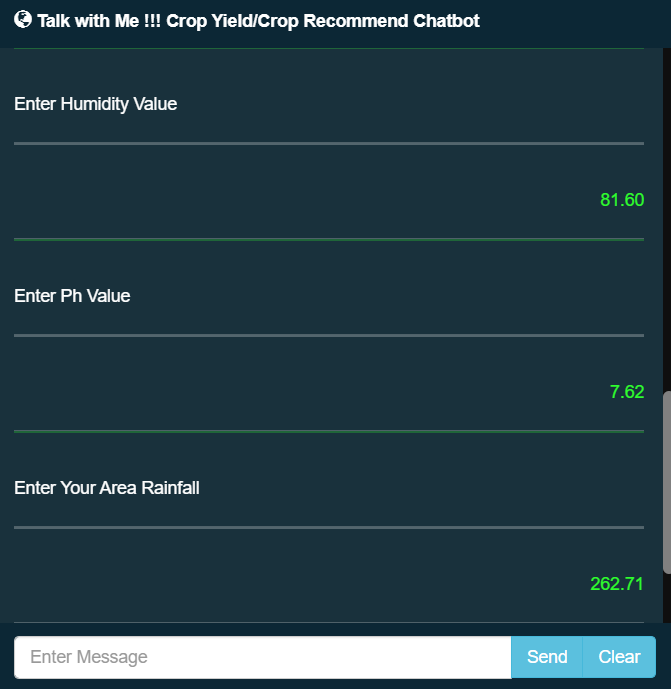
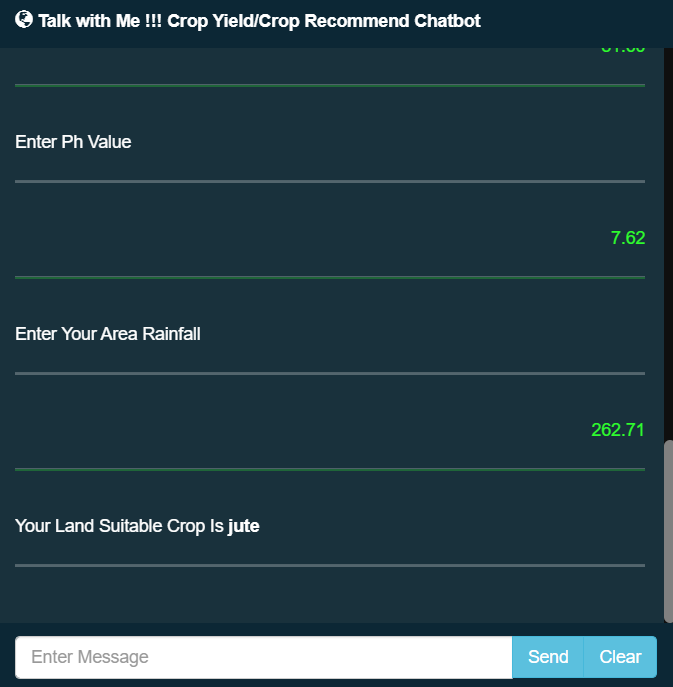
 

** **

****

1. **Recommend:**

** **

** **

**CHAPTER 9**

**CONCLUSION**

**CHAPTER 9**

**CONCLUSION**

The development of the "Chat Bot for Crop Yield Prediction and Crop Recommendation" project represents a significant advancement in leveraging technology to assist farmers in making informed decisions about crop selection and cultivation practices. Farmers can gain knowledge through this chatbot in order to develop and preserve crops. The interpretability of Decision Trees allows farmers to understand which factors contribute most significantly to crop yield variations and make informed decisions accordingly. Through a systematic methodology and well-structured modular design, this project aims to provide valuable support to the agricultural community. This system helps the farmer to choose the right crop by providing insights that ordinary farmers don’t keep track of thereby decreasing the chances of crop failure and increasing productivity. It also prevents them form incurring losses. The system can be extended to the web and can be accessed by millions of farmers across the country.

**CHAPTER 10**

**REFERENCE**

**CHAPTER 10**

**REFERENCE**

[1] Lai J.Y., Sowmya A., Trinder J., (2005). Support vector machine experiments for road recognition in high resolution images, Machine Learning and Data Mining in Pattern Recognition, 4th International Conference, MLDM 2005, Leipzig, Germany, July 9-11, 2005.

[2] Batts G.R., (1997). Effects of co2 and temperature on growth and yield of crops of winter wheat over four seasons, journal ISSN:1161-0301.

[3] Rub G., (2009). Data mining of agricultural yield data: a comparison of regression models, Advances in Data Mining. Applications and Theoretical Aspects, 9th Industrial Conference, ICDM 2009, Leipzig, Germany, July 20-22, 2009.

[4] Jorquera, H., JR Pérez-Correa, A. Cipriano, and G. Acuña, (2001). Short Term Forecasting of Air Pollution Episodes, in Environmental Modeling (vol. 4), P. Zannetti (ed.), WIT Press.

[5] Leemans V., (2004). a real time grading method of apples based on features extracted from defects, Journal of Food Engineering 61(1).

[6] H.S. Jayanna, S.R.M Prasanna, (2009). analysis, feature extraction, modeling and testing techniques for speaker recognition, IETE Technical Review 26(3).

[7] Alberto González-Sanchez, (2014). predictive ability of machine learning methods for massive crop yield prediction, Spanish Journal of Agricultural Research 2014 12(2): 313-328.

[8] Mandic D. P., (2001). recurrent neural networks for prediction: learning algorithms, architectures and stability, ISBN: 978-0-471-49517-8.

[9] Hochreiter S., (1997). long short-term memory. neural computation, Neural Computation 9(8):1735-80.

[10] H. Sak, F. Beaufays, (2014). long short-term memory recurrent neural network architectures for large scale acoustic modelling, arXiv:1402.1128 [cs.NE].

[11] Wiener M., Andy Liaw, (2002). classification and regression by randomforest, R News, 2 (3): 18--22 (2002).

[12] Kumar, Y. J. N., Spandana, V., Vaishnavi, V. S., Neha, K., & Devi, V. G. R. R. (2020). Supervised Machine learning Approach for Crop Yield Prediction in Agriculture Sector. In 2020 5th International Conference on Communication and Electronics Systems (ICCES) (pp. 736-741). IEEE.

[13] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. Sensors, 18(8), 2674.

[14] Chandgude, A., Harpale, N., Jadhav, D., Pawar, P., & Patil, S. M. (2018). A Review on Machine Learning Algorithm Used For Crop Monitoring System in Agriculture. International Research Journal of Engineering and Technology (IRJET), 5(04), 1470.

[15] Patil, A., Kokate, S., Patil, P., Panpatil, V., & Sapkal, R. (2020). Crop Prediction using Machine Learning Algorithms. International Journal of Advancements in Engineering & Technology, 1(1), 1-8.

[16] Khaki, S., & Wang, L. (2019). Crop yield prediction using deep neural networks. Frontiers in plant science, 10, 621.

[17] Batts G. R. (2021). Support vector machine experiments for road recognition in high resolution images.