

RECOMMENDATION SYSTEM FOR CROP, FERTILIZER AND DISEASE DETECTION

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in Information Technology

By

KALVA POOJITHA (REG.NO. 40120042)

S. KAVIYA (REG. NO. 40120045)



**DEPARTMENT OF INFORMATION TECHNOLOGY
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, TAMILNADU-600119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

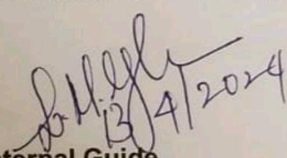
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

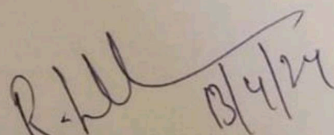
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KALVA POOJITHA (40120042)** and **S.KAVIYA (40120045)** who carried out the project entitled "**RECOMMENDATION SYSTEM FOR CROP, FERTILIZER AND DISEASE DETECTION**" under my supervision from January 2024 to April 2024.


Internal Guide

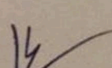
Dr. L. MARY GLADENCE, M.E., Ph.D.,

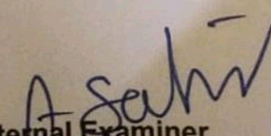

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph. D.,

Submitted for Viva-voce Examination held on

17/4/24


Internal Examiner

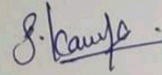

External Examiner

DECLARATION

I, **S.KAVIYA** hereby declare that the project report entitled "**RECOMMEDATION SYSTEM FOR CROP, FERTILIZER AND DISEASE DETECTION**" was done by me under the guidance of **Dr. L. MARY GLADENCE, M.E., Ph.D.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology Degree in Information Technology.

DATE: 17/4/24

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their 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, and **Dr. R. Subhashini, M.E., Ph.D.**, Head of the Department of Information Technology, for providing me with the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide, **Dr. L. Mary Gladence, M.E., Ph.D.**, for the valuable guidance, suggestions, and constant encouragement that paved the way for the successful completion of the project work.

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

ABSTRACT

India is the second largest populated country in the world and one of the largest food exporters in the world. Most people who live in India by inheritance or environmental choices opt for farming as their occupation. In general, rural farmers lack the knowledge and guidance required to cultivate crops that improve agricultural productivity. Farmers usually grow crops in the same pattern year after year, using the same fertilizers and crop types, this in turn results in the loss of nutrients and minerals required to create healthy crops. They typically struggle to recognize the crop diseases and lack knowledge in selecting the suitable to grow based on the local conditions. Government funded initiatives and efforts have been established recently to teach farmers about crops and assist them to become more technologically aware. This project has developed a novel crop recommendation model to advise farmers about which crops to plant, where and how to prevent crop diseases, and what type of fertilizer to use. The proposed application also allows farmers to update or report crop-related issues for public access, including recommendations for certain disease cures. This application also gives advice on how to cultivate healthy crops, suggest fertilizers, and get assistance with crop disease issues by integrating various Machine Learning (ML) technologies. The system also includes a community section where farmers can post or update their problems related to crops, which can be viewed by any individual and can suggest them in the process to cure the disease. This system also initiates to create a blog to connect farmers globally and allows them to exchange difficulties and information on various crop diseases by using Flask. This project is about a process which helps the farmers to know the right crop to grow in their location and to get more information about the disease and get suggestions about the right fertilizer to be used. With the help of machine learning algorithms and predictive analysis, the farmers are offered to grow healthy crops and recommend fertilizers and help through problems of the disease of the crops. With the help of Flask, a farmer-community-based blog can be developed, allowing farmers to communicate and share their problems.

TABLE OF CONTENTS

CHAPTER No	TITLE	PAGE No
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
	2.1 Open problems in Existing System	4
	2.2 Objective	5
3	REQUIREMENTS ANALYSIS	6
	3.1 Feasibility Studies/Risk Analysis of the Project	6
	3.2 Software Requirements Specification Document	7
	3.2.1 Functional Requirements	8
	3.2.2 Non-Functional Requirements	8
	3.2.3 Environmental Requirements	9
	3.3 Methods and Algorithms Used	10
	3.4 Algorithm Selection	13
	3.5 System Use Case	16
4	DESCRIPTION OF PROPOSED SYSTEM	19
	4.1 Methodology	19
	4.2 Architecture	20
	4.3 Description of software for implementation and testing plan of the proposed system	22
	4.4 Transition / Software to operations plan	25

5	IMPLEMENTATION DETAILS	26
5.1	Development and Deployment setup	26
5.1.1	Dataset Description	26
5.1.2	Data Preprocessing	27
5.2	Algorithms	28
5.3	Website	37
6	RESULTS AND DISCUSSION	41
7	CONCLUSION	48
7.1	Conclusion	48
7.2	Future Work	49
7.3	Implementation Issues	50
	REFERENCES	51
	APPENDIX	53
	A. SOURCE CODE	54
	B. RESEARCH PAPER	62

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.1	CLASS DIAGRAM	16
3.2	SEQUENCE DIAGRAM	17
3.3	USE CASE DIAGRAM	18
4.1	SYSTEM ARCHITECTURE -1	20
4.2	SYSTEM ARCHITECTURE - 2	21
4.3	CONCEPT DIAGRAM	23
5.1	HEATMAP OF FINAL DATASET	28
5.2	ACCURACY OF DECISION TREE	30
5.3	ACCURACY OF NAIVE BAYES	30
5.4	ACCURACY OF SVM	31
5.5	ACCURACY OF LOGISTIC REGRESSION	31
5.6	ACCURACY OF RANDOM FOREST	32
5.7	ACCURACY OF XGBOOST	32
5.8	COMPARISON OF ACCURACY	33
5.9	ACCURACY OF ENSEMBLE MODEL	33
5.10	DISEASE DATA COLLECTION	35
5.11	TRAINING OF DATASET	36
5.12	ACCURACY OF DISEASE MODEL	37
5.13	SAMPLE OF THE WEBSITE	40

LIST OF ABBREVIATIONS

SHORT FORM

FULL FORM

SVM

SUPPORT VECTOR MACHINE

CNN

CONVOLUTIONAL NEURAL NETWORK

XG BOOST

EXTREME GRADIENT BOOSTING

ML

MACHINE LEARNING

DL

DEEP LEARNING

PKL

PICKLE

CSV

COMMA SEPARATED VALUES

RESNET

RESIDUAL NEURAL NETWORK

CHAPTER 1

INTRODUCTION

India is a country with agribusiness and is the significant wellspring of the economy. 70% of the Indian populace straightforwardly depends on farming. The standard issue among youthful Indian ranchers is picking the correct yield dependent on the dirt prerequisites. This script makes novel by using simple parameters like State, district, season, and area and the user can predict the yield of the crop the user wants. Because of this, users face a genuine difficulty in efficiency. Our work proposes to assist ranchers with deciding the dirt quality by examining its different boundaries and recommending crops dependent on the outcomes acquired utilizing the information mining approach. So, the system is being designed using machine learning algorithms for the betterment of farmers. The system will suggest the best crop for land based on content like weather parameters. And the system provides information about the required range and quantity of fertilizers for cultivation. Hence, by utilizing this system, farmers can cultivate a new variety of crops, increase their profit margin, and avoid soil pollution. In addition, the model provides a feature in which the farmer can upload a diseased plant or leaves photo and get the necessary measures to take care of the plants and fertilizers for further process. And even this system provides the cause of disease and gives precautions to take care of the plants. This system even provides a farmer-community blog where the farmers can post their views on the recommended product or the diseases that crops are getting. Every individual can see that and reply to that. All these are comprised of a flask web app in which users will be provided the options to go to crop, fertilizer, disease detection or the community and get the desired outcomes.

CHAPTER 2

LITERATURE SURVEY

Researchers and automotive companies have extensively studied this problem statement over the past few years to create a solution. All their solutions vary from analyzing the crop or fertilizer prediction using different methods. Some of the researches were as follows.

In the research outlined in [1] in the year of 2021, P. A, S. Chakraborty, A. Kumar and O. R. Pooniwala discussed and the inference is that he used different types of algorithms for the recommendation of the system but there are other ways to increase the efficiency of the model.

Some studies were conducted using different ML algorithms, primarily, through Random Forest Classifier, SVM. It has been used to show differences in how the soil reacts to the rainfall and what kind of crop must be used, and the fertilizers can be used.

In this study of [2] in 2021, by Mrs. S. Vaishnavi., M. Shobana., R. Sabitha. and S. Karthik. This model is purely based on the season and the crops cultivated in that season but the system can also take many other attributes to get the crop efficiently.

Based on the study [3] in 2021 of Mr. Suresh, G., M Used various types of different algorithms to predict the type of fertilizer but failed in getting better accuracy. The system can even consider using crop and get it better.

The work demonstrated in [4] in 2021 by Mr. Kodimalar Palanivel and Surianarayanan, C. has developed a model that works on Prediction using Big Data Techniques in which it is a unique way to do these kinds of projects and it didn't show many good results in the accuracy and to improve that he can use the

mix of the ML and Big Data to yield still better results.

In the work demonstrated by [5] in 2021 On the inference of S. P. Raja, G. Mariammal, A. Suruliandi and E. Poongothai has given a different kind of approach based on soil with hardware using modified recursive feature elimination technique with various classifiers.

Based on the inferences by [6] in 2020 given by Nischitha, K., Mahendra N. Dhanush Vishwakarma, and Manjuraju MR Ashwini. This research recommends the crop by using different kinds of algorithms but mainly based on the factor rainfall factor which isn't much recommendable.

In this study of [7] in 2020 by Mr. Patil, P., Panpatil, V. and Kokate, S developed a model in based on KNN and various kinds of decision tree algorithms with processing in large data sets. But the final decision is to go with Random Forest Classifier which doesn't seem well, while the other models can also achieve better accuracy.

Based on the inferences by [8] in 2019 Mr. Ghadge, R., Kulkarni, J., More, P., Nene, S., & Priya, R. L. Prediction of crop yield using machine learning. Int. Res. J. Eng. Technol.(IRJET). Used different types of Techniques in solving the problem statement and yet did not succeed in finding the efficient way due to not used updated algorithms.

In the work demonstrated by [9] in 2018 by Militante, Sammy V., Bobby D. Gerardo, and Nanette V. Dionisio. The system is good with the algorithm and gives efficient results but detects only less no. of diseases and not that efficient in detecting problems.

In the research outlined in [10] in 2017 by Subramanian, Kanaga Suba & Rishi, R. & Sundaresan, E. & Valliappan, Srijit. Recommends only based on the demand but need to consider other factors to even decrease the cost of farming.

2.1 OPEN PROBLEMS IN EXISTING SYSTEM

Some of the drawbacks or problems with the existing system are:

- By using SVM in the model, overfitting may occur and gives incorrect predictions.
- While using Convolutional Neural Networks, the dataset must be more significant, if not, less accuracy is resulted.
- By using Data Augmentation, it produces both realistic as well as unrealistic images.
- While using the XG Boost method for smaller datasets, the predicted outcomes may be wrong.
- Selecting the appropriate machine learning algorithms and training them on the available data can be challenging. This is particularly true in areas where data is limited or incomplete, as this can lead to overfitting or inaccurate predictions.
- The user interface and experience of the app can affect its adoption and effectiveness. However, designing a user-friendly interface that is suitable for farmers with varying levels of technical expertise and literacy can be challenging.
- In some cases, similar projects may need to integrate with existing agricultural systems, such as irrigation or fertilization systems. Ensuring that these systems can communicate effectively with the app can be challenging, particularly if they were not designed with interoperability in mind.

2.2 OBJECTIVE:

- The proposed system examines data from the farmer or user. It gets some of the primary attributes from the user, like potassium, Nitrogen levels and rainfall, etc., and will predict the crop that is best suitable for that kind of soil.
- The above combined with fertilizer, prediction is also generated based upon the resultant predicted crop and one more another attribute, namely soil erosion. The system helps the farmer to get more yield and develop much business.
- Adding upon here comes the exciting part in which almost any type of disease is detected that a plant gets by just uploading a picture of that leaf and get the precautions for further development of the plant and re-iterates the fertilizer to be given there.
- Easy-to-use digital tool that can help them make informed decisions about crop selection, fertilizer management, and disease detection.
- The system aims to leverage machine learning algorithms and real-time data to provide accurate and personalized recommendations that can improve crop yields, reduce costs, and increase efficiency.
- The system also aims to promote sustainable agricultural practices and enhance the resilience of smallholder farmers to climate change and other environmental challenges.
- And carrying upon this it provides all these with a website developed with Flask, which will be more user-friendly for the end user or customer without any hurdles.
- With the help of Flask, a farmer-community-based blog can be developed, allowing farmers to communicate and share their problems.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES / RISK ANALYSIS OF THE PROJECT

The project plan and objective of the project is clear, and this project is implemented till application level without any problem. Also, the risk involved in this project is minimal, which is negligible. This project can further be carried out in next phase as well it has wide scope. Currently four classification algorithms are used in the project. The accuracy and performance of each algorithm is used further to predict with the client-side data, and it can also be improved in future. The applications is built with minimal risk. So, from this we can analyze that the project and feasible and it is submitted on time. The theme of the feasibility study is to find out if the application is technical, economical, operational and market feasible so that applications sustain for longer time and gives return on investment with good results. It can be used to inform future decisions.

Feasibility Studies:

- **Technical Feasibility:** The project's technical feasibility depends on the availability of necessary hardware, software, and data infrastructure. It is important to evaluate whether the required technology and data is readily available and can be obtained within a reasonable time and cost.
- **Operational Feasibility:** It is important to assess whether the project can be successfully integrated into existing farming operations and workflows. This includes evaluating whether farmers can easily use the system and whether the system can be scaled up to accommodate large numbers of users.
- **Economic Feasibility:** The economic feasibility of the project involves assessing whether the benefits of the project outweigh the costs. This includes evaluating the potential return on investment, the cost of hardware and software, and the cost of data acquisition.

Risk Analysis:

- **Technical Risks:** Technical risks include issues such as system crashes, data loss, and compatibility issues. It is important to ensure that the system is built using reliable and robust technology and that backup and recovery mechanisms are in place.
- **Data Risks:** The system relies heavily on accurate and relevant data. Risks related to data quality, data privacy, and data security must be addressed to ensure that the system provides reliable and trustworthy recommendations.
- **Adoption Risks:** Adoption risks include issues related to user acceptance and adoption of the system. Farmers may not trust the system or may prefer traditional farming practices. It is important to involve farmers in the design and testing of the system to ensure that it meets their needs.
- **Operational Risks:** Operational risks include issues related to system maintenance, hardware failure, and data quality. It is important to have a plan in place to address these risks and ensure that the system is reliable and functional.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT:

It is crucial to comprehend and record all forms of requirements to guarantee that the software system or application is created in a way that meets the expectations of the users and stakeholders and operates as anticipated within its designated environment. A project typically encompasses three types of requirements, namely functional requirements, non-functional requirements, and environmental requirements.

Purpose of this document:

The purpose of this document is to identify the essential requirements of the software and define the criteria to make sure the development is moving in the right path.

Scope of this document:

The project's main objective is to help the farmers to cultivate better crops, fertilizer and disease detection based on a particular area. All these will be accessed through a website that is completely developed using Flask.

Requirements are the primary constraints that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Environmental requirements

3.2.1 FUNCTIONAL REQUIREMENTS:

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists the requirements of a particular software system.

3.2.2 NON-FUNCTIONAL REQUIREMENTS:

Process of functional steps:

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction of the result
6. Integration of model with Flask
7. Deployment using Flask

3.2.3 ENVIRONMENTAL REQUIREMENTS:

1. Software Requirements:

Operating system: Windows

Tool: Anaconda with Jupiter notebook, spyder

Updated Python libraries

2. Hardware Requirements:

A. Internet connection to download and activate.

B. Minimum 10GB free disk space

C. Windows 8.1 or 10 (64-bit version only) is required.

D. Minimum System Requirements To run Office Excel 2013, your computer needs to meet.

the following minimum hardware requirements:

- 500-megahertz (MHz)
- 256 megabytes (MB) RAM
- 1.5 gigabytes (GB) available space
- 1024x768 or higher resolution monitor

It describes the environment in which the software system or application will operate, including the hardware, software, and network infrastructure. They typically specify the resources required to deploy and run the system, such as hardware specifications, operating systems, and software dependencies. Examples of environmental requirements include the operating system, database management system, web server, network protocols, and required software libraries. An operating system is windows7+ or mac OS 10.6+, anaconda and vs-code are the software requirement of this project to run it uninterruptable. As far as hardware is concerned a good SSD is recommended to run it smoothly but HDD is also fine to run the program. A minimum of 4GB RAM and 128GB storage is required.

3.3 METHODS AND ALGORITHMS USED:

The method attempted is to try includes a support vector machine, random Forest, XG Boost, Decision Tree, Naïve Bayes and Linear regression algorithms. In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observations.

This data set may simply be bi-class. (Like identifying whether the person is subscribed to a term deposit or not, whether a person is interested to subscribe) or it may be multi-class too. Some examples of classification problems are speech recognition, handwriting recognition, biometric identification, document classification, etc.

In Supervised Learning, algorithms are learned from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associates the patterns to the unlabeled new data.

Used Python Packages:

- **NumPy:**

It is an open-source numerical Python library. It contains a multidimensional array and matrix data structures and can be used to perform mathematical operations.

- **Scikit-learn:**

It is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

- **Matplotlib and Seaborn:**

Matplotlib is mainly deployed for basic plotting. Visualization using Matplotlib generally consists of bars, pies, lines, scatter plots, and so on. Seaborn, on the other hand, provides a variety of visualization patterns. It uses fewer syntax and has easily interesting de-fault themes.

- **Pandas:**

It is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language. It is a popular open-source library for data manipulation and analysis in Python. It provides tools for working with structured data, such as tabular data and time series data.

- **Flask:**

Flask provides configuration and conventions, with sensible defaults, to get started. This section of the documentation explains the different parts of the Flask framework and how they can be used, customized, and extended. Beyond Flask itself, look for community- maintained extensions to add even more functionality.

- **CNN:**

A convolutional neural network (CNN or convnet) is a subset of machine learning. It is one of the various types of artificial neural networks which are used for different applications and data types. A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data.

- **SQLAlchemy:**

SQLAlchemy is a popular Python library for working with relational databases. It provides a powerful set of tools for working with databases, including an Object Relational Mapping (ORM) layer, which allows developers to work with databases using Python classes and objects. SQLAlchemy also provides a SQL expression language for generating SQL queries in a Pythonic way.

- **PyTorch:**

Torchvision is a PyTorch package that provides a range of tools and utilities for working with computer vision tasks, including image classification, object detection, and segmentation. It includes a number of pre-trained models, as well as tools for loading and preprocessing image data. These models have been trained on large datasets, such as ImageNet, and can be fine-tuned on new datasets with relatively few additional training examples.

- **Scikit-learn:**

It is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction in Python.

3.4 ALGORITHM SELECTION:

When a structured dataset is present, the continuous or categorical outcome needs to be estimated, then the system uses supervised machine learning methodologies like regression and classification techniques.

When unstructured data is present and want to predict the clusters of items to which a particular input test sample belongs, unsupervised algorithms are used. An actual data scientist applies multiple algorithms to get a more accurate model.

- **SUPPORT VECTOR MACHINE (SVM):**

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression challenges. However, it is primarily used in classification problems. Support Vectors are simply the coordinates of individual observation.

A support vector machine's goal is to draw hyperplanes and divide data points and to draw the hyperplane that separates data points with the largest margin or with the most space between the dividing line and any given data point.

3.4.1 RANDOM FOREST CLASSIFICATION:

Random forest can also be used for classification. It is one of the most broadly used machine learning algorithms for classification. Whether the response variable is continuous or categorical, it works in both cases. According to Friedman et al. random forests start to become stable at around 200 trees, whereas at 1000 trees the boosting of this still keeps on improving.

If trees are much smaller or there is a presence of shrinkage, then the process of boosting starts to reduce. The important function of the RF

is the utilization of out-of-bag (OOB) samples. For each value $z_i = (x_i, Y_i)$, in which term z_i did not appear that makes the RF predictor by averaging only those trees which are consistent with the bootstrap samples.

The OOB error estimate is then nearly identical to that which is getting by N fold cross-validation. In contrast to the other nonlinear estimators, it is possible to fit the RF in one sequence with the cross-validation being completed. The training can be finished if the OOB error stabilizes itself.

3.4.2 XG BOOST:

XGBoost is an implementation of Gradient Boosted decision trees. XGBoost models majorly dominate in many Kaggle Competitions. In this algorithm, decision trees are created in sequential form.

Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results.

The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree.

These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

3.4.3 DECISION TREE:

Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems. Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. Decision Tree can

represent any Boolean function on discrete attributes.

3.4.4 NAIVE BAYES CLASSIFICATION:

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Using Bayes theorem, the probability of A happening can be found, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent.

3.4.5 LINEAR REGRESSION:

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables.

It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on the relationship between the dependent and independent variables they are considering and the number of independent variables getting used.

$$y = \theta_1 + \theta_2 \cdot x$$

While training the model, given are: x input training data (univariate – one input variable(parameter)) y: labels to data (supervised learning). When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ_1 and θ_2 values. θ_1 : intercept θ_2 : coefficient of x and once the best θ_1 and θ_2 values are found, the best fit line is found. So, when finally use the model for prediction, it will predict the value of y for the input value of x.

All these algorithms will be combined and shall create an ensemble model in

which it gives better accuracy than any individual project. This model is pickled and will be integrated with the flask website and will be accessed it throughout the website.

3.5 SYSTEM USE CASE

CLASS DIAGRAM:

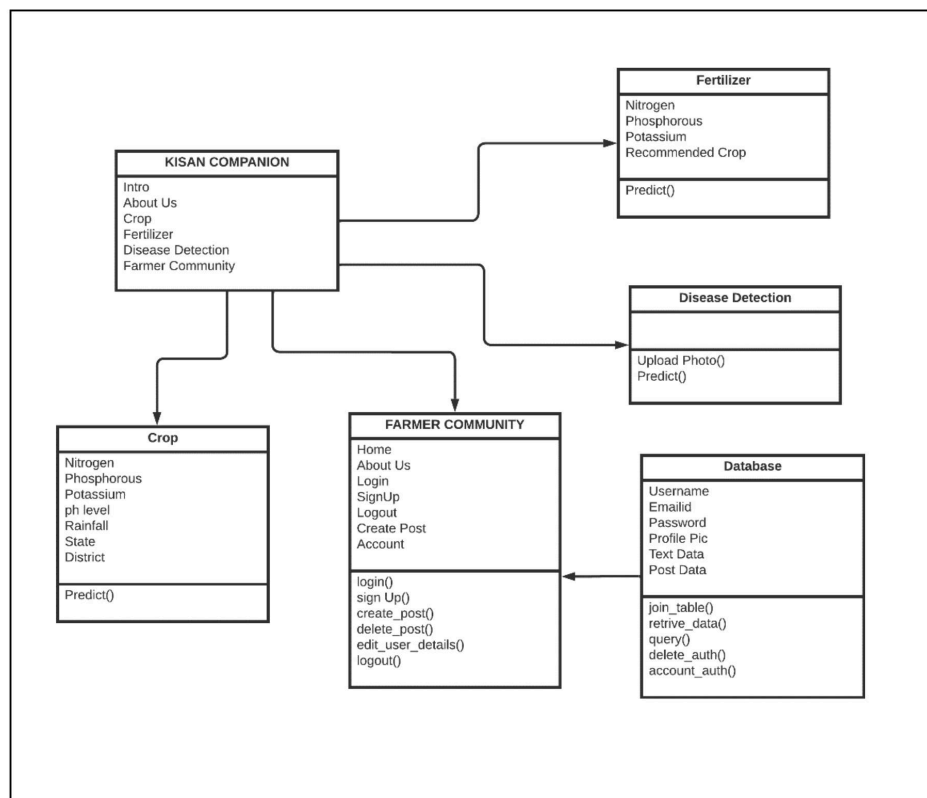


FIG 3.1 CLASS DIAGRAM

This class diagram shows the basic classes and their functions of the system. The Actor class represents the user or farmer, who can interact with the Main Website to access the Crop, Fertilizer, Disease, and Community pages. The Main Website class has functions to display each of these pages. The Machine Learning Module class has functions to provide recommendations and diagnoses for each of the three categories: crops, fertilizer, and disease. Finally, the Crop page, Fertilizer page, and Disease diagnosis page classes have functions to

display their respective pages. The Community Website has several attributes in order to save the data and retrieve the data according to user login and create posts and delete posts based on user authentication and sign up, login and logout.

SEQUENCE DIAGRAM:

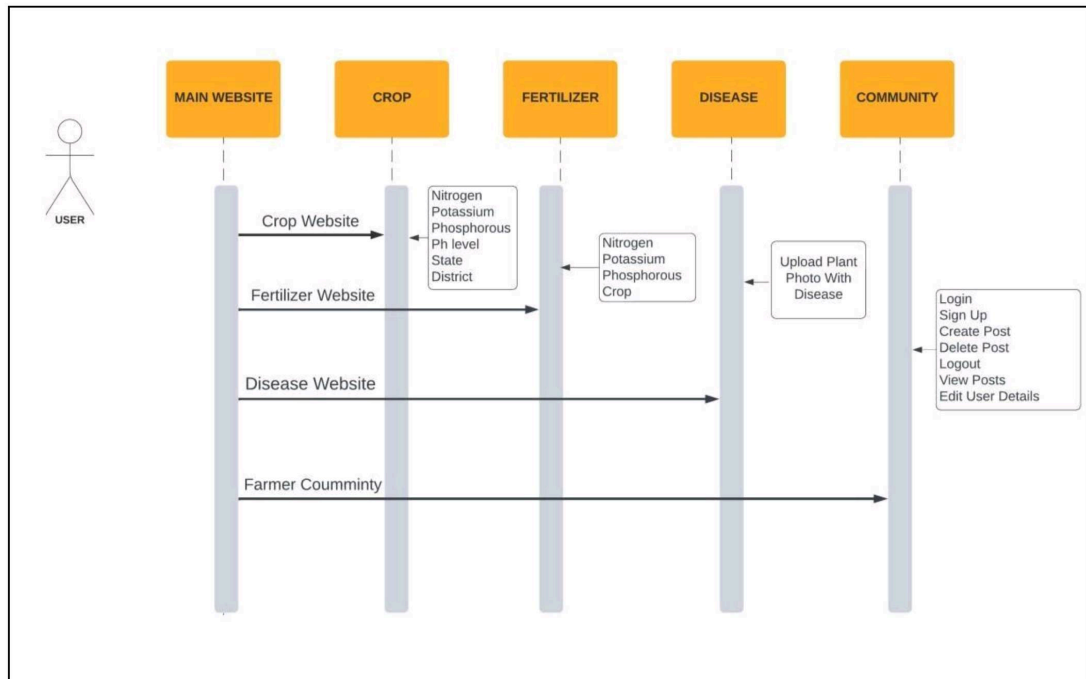


FIG 3.2 SEQUENCE DIAGRAM

This sequence diagram shows the basic flow of the system, where the actor opens the main website and can then access the Crop, Fertilizer, Disease, and Community websites. The user can request a recommendation from the machine learning module integrated with pickle file which suggests and then puts that into Fertilizer and get the recommendation.

Now can post that results in the community website which will help other fellow farmers to analyze the recommendation and proceed further.

USE CASE DIAGRAM:

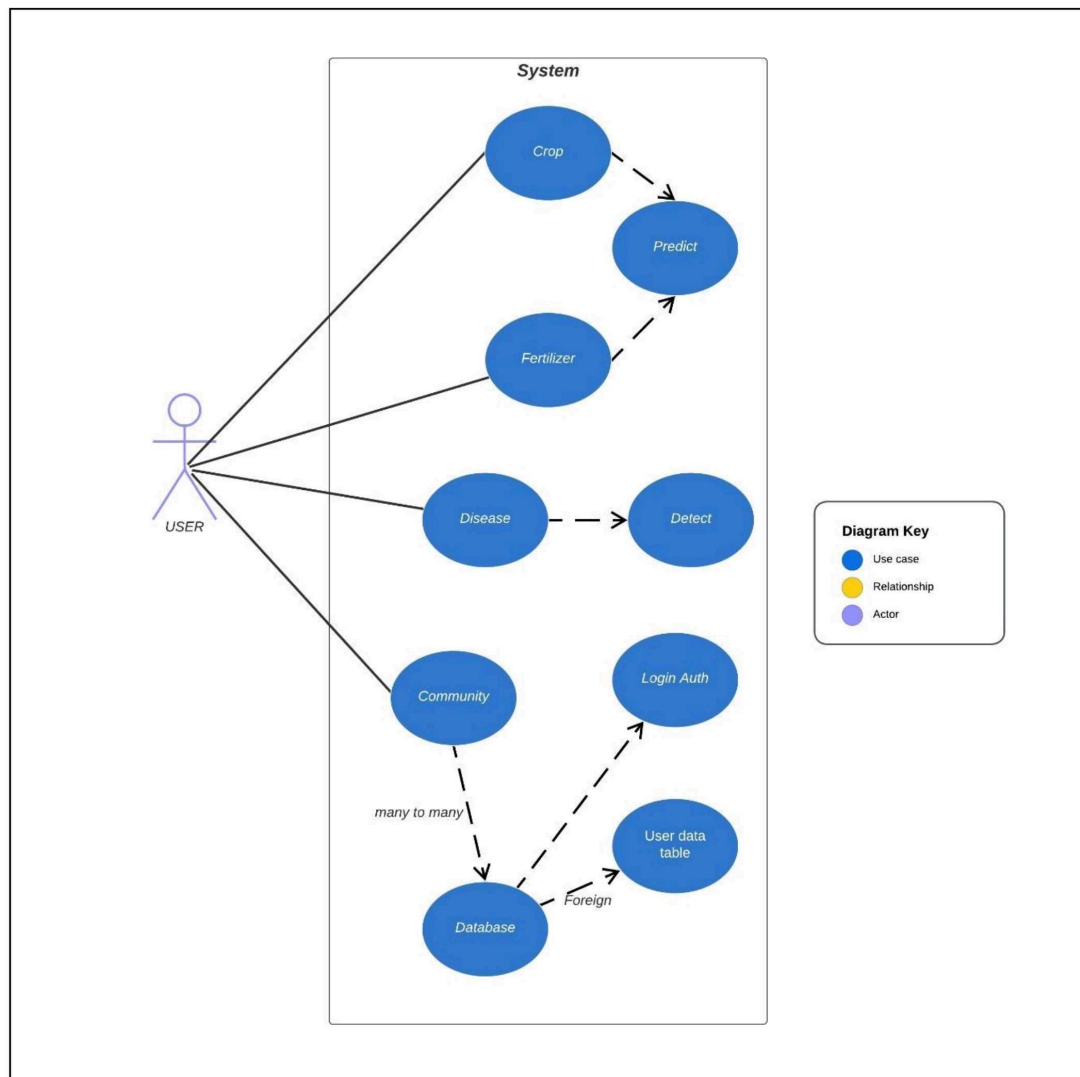


FIG 3.3 USE CASE DIAGRAM

As you can see, the use case diagram includes a user who access Crop, Fertilizer, Disease, Community, Database, and Website. The Crop, Fertilizer, and Disease use cases are responsible for recommending and predicting crops, fertilizers, and detecting diseases, respectively. The Community use case allows users to create and interact with posts, and the Database use case stores all the data. Finally, the Website use case is responsible for displaying data and interacting with user.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

The system is making use of classification algorithms to improvise the crop yields. It passes the different contributions to the regulator which utilizes the Help Vector Machine Calculation for characterization.

The proposed framework gives simple openness to the clients. They are additionally simple to utilize and comprehend by the trifier ranchers. It improves perception and understandability.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

- The system is working on various models like Random Forest Classifier, SVM, XGBoost, Decision Tree, Naive Bayes, and Logistic Regression.
- An ensemble model is created from it because it can give the most accurate value among the used algorithms.
- Ensemble Model is used in cases where accuracy plays the most important role, and it always provides better or mean among the used algorithms based on the types of the machine learning process. With the help of Flask, a farmer-community-based blog can be developed, allowing farmers to communicate and share their problems.
- This website is completely designed with flask in which it has several kinds of modules and authentications involved. This even can be an eye opener for farmers to cultivate crops using online help in order to improve the Decision making, Efficiency.
- The system will be accessible through a web application which is a responsive website which will be easy to use for the farmers. The system will also have a feedback system where farmers can give feedback about the recommendations and predictions provided by the system.

- Moreover, having all the features for crop, fertilizer, and disease detection on a single website would be a convenient way for farmers to access all the information they need in one place.
- The website could have a section dedicated to providing crop and fertilizer recommendations based on the farmer's input of soil type, climate, and desired yield. Machine learning algorithms could be used to analyze this data and provide personalized recommendations.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

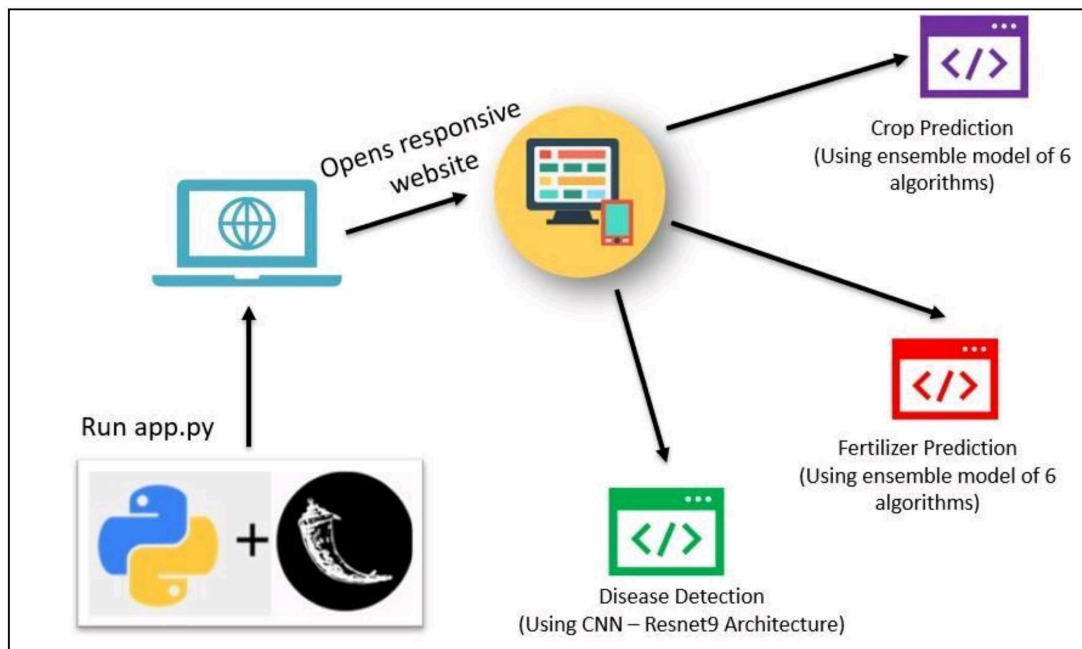


FIG 4.1 SYSTEM ARCHITECTURE – 1

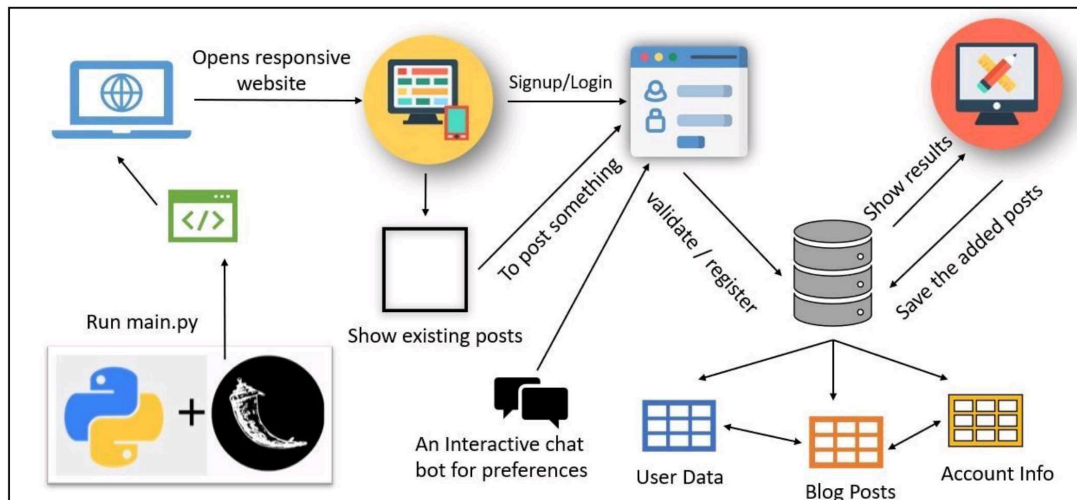


FIG 4.2 SYSTEM ARCHITECTURE – 2

- The very first step is to create a main website which users can access all the available options Crop, Fertilizer, Disease Detection and Farmer Community website.
- There are certain field in which user need to fill it and get detailed information about the crop that needs to be cultivated.
- Once the farmer gets the crop now, he can navigate to fertilizer page and conclude which fertilizer can be used based on the minerals in the soil and the type of crop it recommended.
- As the crop gets cultivated there are cases in which crop can get affected to any disease and then click a picture of the disease and upload it the Disease Detection form and get the precautions and cause of the disease and take certain measures based on it
- Once all this is over, he can post the results in the farmer community website where the fellow farmers can access it and take measures based on it. They can update all these through posts.
- It is like a social media platform for farmers where they can react to others posts and can watch individual posts too and grow together.

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

ANACONDA NAVIGATOR:

Anaconda Navigator is remembered for the Anaconda distribution and permits clients to send off applications and oversee condo packages, environments, and channels without using command-line commands. Navigator can search for packages, install them in an environment, run the packages and update them. With the help of the anaconda command prompt, all necessary packages are installed.

GIT-HUB:

Git-hub is used for cloning repositories like UCI and uploading the files into Git-hub and saving them there. GitHub also supports the development of open-source software, which is software with source code that anyone can inspect, modify, and enhance the models. Kaggle might also use for training the models as it supports faster execution as it supports GPU.

TESTING PLAN OF THE PROPOSED SYSTEM:

While coming evaluation to evaluating the performance of the model, it is needed to do manually as it deals with real-time data, a suitable dataset is found for the implementation of the model. It results in a lot of raw data and need to preprocess it using different techniques and get it ready for the next process and create the model and deploy it with a nice and user-friendly website.

- Performance testing: Test the performance of the system under different load and stress conditions to ensure it can handle expected traffic and usage.
- Security testing: Test the system for vulnerabilities and potential security issues to ensure sensitive data is protected.
- Usability testing: Test the system for ease of use and user experience

to ensure it is user-friendly and accessible to its intended audience.

- Compatibility testing: Test the system on different devices and browsers to ensure it works as expected across different platforms.

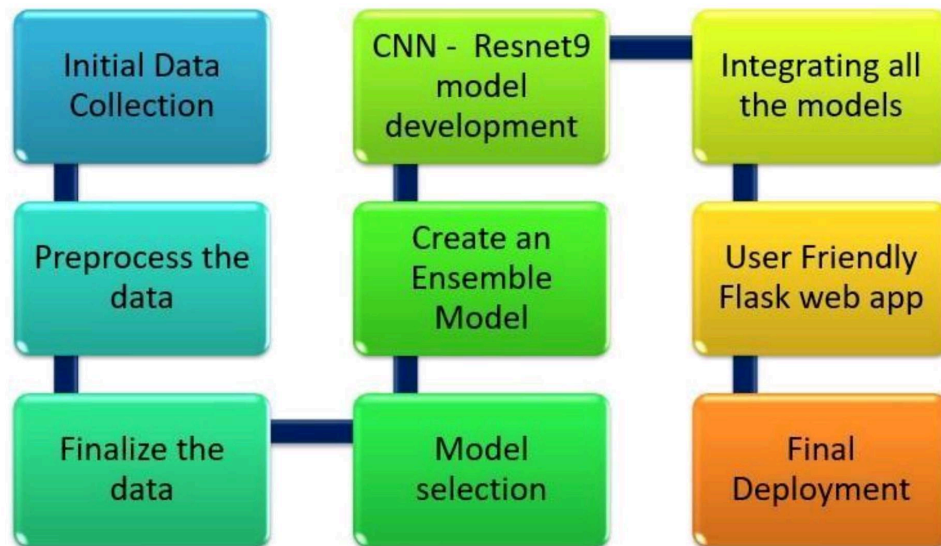


FIG 4.3 CONCEPT DIAGRAM

- Initially Data is collected from several websites like github, ucs, kaggle and there is lot's of raw data and that needs to be preprocessed very well before use.
- Once after finalizing the data let's go for model selection and use 6 different kinds of models and make a ensemble model out of it and use it asa primary prediction module.
- For disease detection the used Reset 9 architecture which is a high-level CNN model to detect the disease of crop.
- And now let's integrate all into one single unit into a flask website and then start integrate the farmer community based website into it.
- It has several other features which can be used effectively while cultivating crops.

- Initial dataset collection and preprocessing it completed
- The process is started the model selection with all the 6 algorithms namely SVM, RandomForest Classifier, XGBoost, NaiveBayes, Decision Tree and Linear Regression.
- An ensemble model is created with all those algorithms and get the required output.
- The model is tested with different types of inputs and again tested the model and re-iterated the model.
- And hence the system is ready to predict the crop and fertilizer to be used.
- Completed the Disease Detection part using CNN with other resources available to get the work done.
- Creation of forms for getting the input from the user for predicted using Flask WTF forms.
- Basic templates are ready and integration with pickle modules is done.
- The specialized website for farmer which is developed by flask is intact with all the user profiles, blog posts page, individual.
- It has different pages with login, signup, about, create post and delete post are in working.
- Every User can see all the posts and can view in particular about single user posts and delete their own posts.
- Every page contains 10 posts of page-nation on the main page and particular user profile it is of 5 posts of page-nation.
- Users can change the profile pic, username, mail etc.
- All this comprises of different navigations among the main page of the base page with different options in the navigation bar.
- The users can get all the details about the crop, fertilizer and diseases precautions and cure within detail based on the personalized detection.

4.4 TRANSITION/ SOFTWARE TO OPERATIONS PLAN

Every software project needs a transition strategy because it entails handing off the system from the development team to the operations team, which is in charge of its upkeep and support.

The operations team will receive instruction and knowledge transfer from the development team, ensuring they have the skills and information required to support and maintain the system. A user manual outlining the system's features, functions and operational needs will be made available.

The system will be deployed in stages, with testing and validation carried out at each step to make sure it's working correctly and effectively. The deployment will be carefully watched to make sure that any problems are dealt with right away.

To guarantee optimum performance and dependability, the operations team will monitor the system's operation and make any necessary adjustments. The operations team will be in charge of carrying out routine upkeep procedures, including installing updates and patches, creating backups, and guaranteeing data security.

The operations team will offer user support, which will include replying to user questions and resolving any problems that may occur. For the purpose of handling user questions and support requests, a helpdesk system will be created.

The operations team will be responsible for identifying opportunities for improvement and implementing changes to improve the system's performance, reliability, and usability.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP :

5.1.1 DATASET DESCRIPTION :

- Initially data is collected from various open source websites like Github, Kaggle, UCS and web scrapping methodologies. All these data are put together at a place and start preprocessing on it
- The process typically involves identifying the relevant sources of data, collecting the data in a systematic and organized manner, and then cleaning, preprocessing, and formatting the data to ensure that it is accurate, complete, and usable.
- Once the data has been gathered, it is important to carefully clean and preprocess the data to ensure that it is accurate, complete, and ready for analysis. This may involve removing duplicates or missing values, normalizing, or transforming the data, and formatting the data to ensure that it is compatible with the analysis tools that will be used.
- Here they collected a few datasets of Crop, Fertilizer, District wise data of crops and their nature towards the weather conditions and etc.
- Crop Data consists of 3101 rows of raw data in which the attributes are temperature, humidity, ph level, rainfall, crop type.
- There are 31 different types of crops in it which are unique in their own way.
- Fertilizer Data consists of 1842 rows of data with which the attributes are crop-type, nitrogen, phosphorous, potassium, and ph values.
- There are 97 types of crops in this dataset which consists of the same crops and many others with fertilizer's to be used.
- District wise data of crops dataset consists of 246092 rows of data with attributes as Crop, nitrogen, phosphorous, potassium, ph value, district, state.

- Disease Detection dataset is a public repository where the access can be made from online and can access it in the Kaggle website and make the model based on the architecture.

5.1.2 DATA PREPROCESSING:

- Now pre-processing is done based on all the data that has been collected.
- Crop data and fertilizer needs to be made as a single dataset since the recommendation work in such a way that the predicted crop needs to be placed in the fertilizer prediction
- This needs to be combined with the district wise data too and proceed for the model making
- Initially pre-processing is made with the crops that are present in the crop data
- All the data is made into smaller case in-order to ease of comparison. The crops in the fertilizer dataset are also printed to get to know the crops.
- By neglecting all the null values and empty cell and removing those or adding the median values in place of it.
- Once after finalizing the crops the very next moment the other crops in the fertilizer data must be deleted or changed into the primary crop data.
- This can be executed using different methods available in python scikit learn module.
- After executing all these data the left out data is of 2200 rows with 5 attributes namely crop, nitrogen, phosphorous, potassium, ph level.
- This will be combined with district wise data using the common crop data in both of the data.
- And hence we are left out with 2200 rows and 7 attributes with an extra of state and district values into it
- This file is saved with merge_file_data and hence we move to the model selection and make the optimal solution for required recommendations.

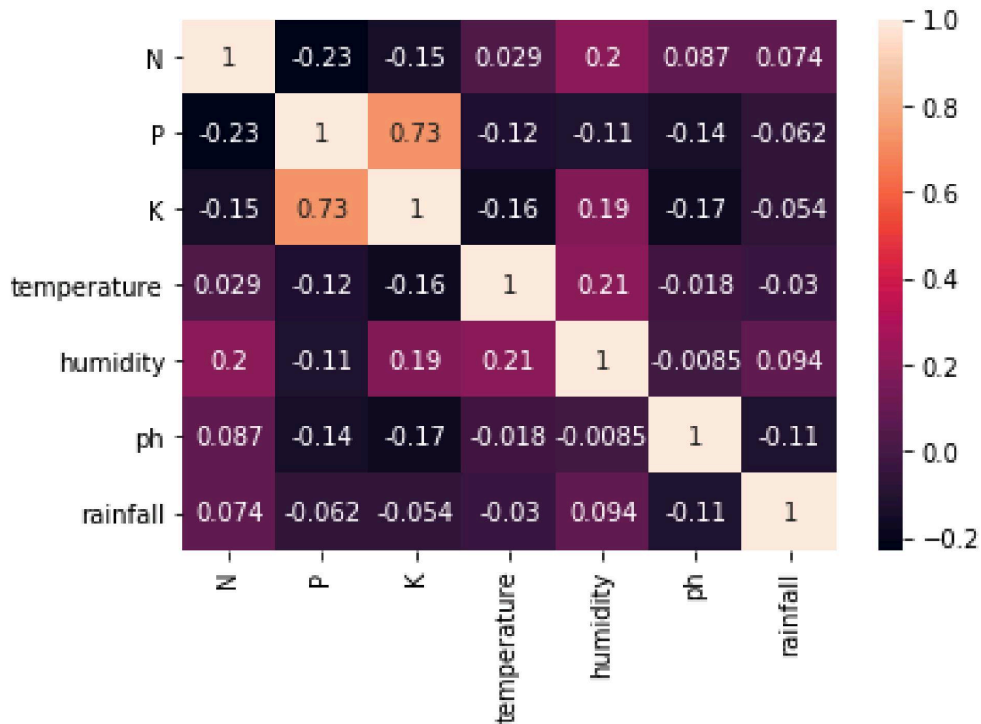


FIG 5.1 HEATMAP OF FINAL DATASET

5.2 ALGORITHMS:

Ensemble Model:

An ensemble model is a machine learning model that combines the predictions of multiple individual models in order to improve the overall accuracy and reliability of the predictions. The individual models can be of the same type, such as multiple decision trees or neural networks, or they can be of different types, providing different perspectives on the problem at hand. The idea behind ensemble modelling is that by combining the predictions of multiple models, we can reduce the impact of individual model errors and improve overall accuracy. This is done by taking the average or weighted average of the predictions of the individual models, or by using more complex techniques such as stacking, where the output of one model is used as input for another model.

In this project, we use a variety of machine learning algorithms for model selection in order to achieve the best possible accuracy and performance for crop, fertilizer, and disease detection. Specifically with the use of six different types of algorithms, and make an ensemble model including:

1. Decision Trees: These algorithms use a tree-like model of decisions and their possible consequences to make predictions based on input data.
2. Random Forests: Random forests are a type of ensemble learning method that combines multiple decision trees to improve accuracy and reduce overfitting.
3. K-Nearest Neighbors: K-NN algorithms classify new data points based on the classification of their nearest neighbors in the training data.
4. Support Vector Machines: SVMs are supervised learning models that analyze data and recognize patterns to classify new data points.
5. Naive Bayes: Naive Bayes algorithms use probability theory to classify data points based on their probability of belonging to a certain class.
6. XGBoost : XGBoost is a type of machine learning algorithm that are particularly effective at analyzing ,making them ideal for disease detection.

Decision trees are simple and easy to understand, but may over fit and suffer from high variance. Random forests address these issues by combining multiple decision trees to reduce overfitting and improve accuracy. K-nearest neighbors is a lazy learning algorithm that works well with small datasets but can be computationally expensive. Support vector machines are effective at finding decision boundaries between classes, but can be sensitive to the choice of kernel function. Naive Bayes is a probabilistic algorithm that works well with high- dimensional data, but assumes that features are independent of each other. Finally, CNNs are powerful deep learning algorithms that are particularly effective at analyzing image data, making them ideal for disease detection.

Decision Tree:

By using decision tree algorithm, the model achieved an accuracy of 81.8 %

DecisionTrees's Accuracy is: 81.818181818183				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.94	1.00	0.97	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	0.00	0.00	0.00	20
grapes	1.00	1.00	1.00	18
jute	1.00	0.11	0.19	28
kidneybeans	1.00	1.00	1.00	14
lentil	0.88	1.00	0.94	23
maize	1.00	0.90	0.95	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	0.79	0.88	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	0.00	0.00	0.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	0.37	1.00	0.54	17
rice	0.25	1.00	0.41	16
watermelon	1.00	1.00	1.00	15
accuracy			0.82	440
macro avg	0.84	0.85	0.81	440
weighted avg	0.83	0.82	0.79	440

FIG 5.2 ACCURACY OF DECISION TREE

NAÏVE BAYES:

By using Naïve bayes algorithm the model achieved an accuracy of 96.1%

Naive Bayes's Accuracy is: 0.9613636363636363				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.84	1.00	0.91	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	0.95	0.98	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.80	1.00	0.89	28
kidneybeans	0.93	1.00	0.97	14
lentil	0.96	0.96	0.96	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	0.84	0.91	19
mungbean	0.96	1.00	0.98	24
muskmelon	1.00	1.00	1.00	23
orange	0.97	1.00	0.98	29
papaya	1.00	0.74	0.85	19
pigeonpeas	1.00	0.94	0.97	18
pomegranate	1.00	1.00	1.00	17
rice	0.83	0.62	0.71	16
watermelon	0.94	1.00	0.97	15
accuracy			0.96	440
macro avg	0.96	0.96	0.96	440
weighted avg	0.96	0.96	0.96	440

FIG 5.3 ACCURACY OF NAIVE BAYES

SVM:

By using svm algorithm the model achieved an accuracy of 93.1 %

SVM's Accuracy is: 0.9318181818181818				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.88	0.94	0.91	16
chickpea	1.00	0.90	0.95	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.82	1.00	0.90	28
kidneybeans	0.70	1.00	0.82	14
lentil	1.00	0.96	0.98	23
maize	1.00	1.00	1.00	21
mango	0.81	1.00	0.90	26
mothbeans	0.82	0.47	0.60	19
mungbean	0.96	1.00	0.98	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	0.79	0.88	29
papaya	1.00	0.89	0.94	19
pigeonpeas	1.00	0.89	0.94	18
pomegranate	0.74	1.00	0.85	17
rice	0.91	0.62	0.74	16
watermelon	1.00	1.00	1.00	15
accuracy			0.93	440
macro avg	0.94	0.93	0.93	440
weighted avg	0.94	0.93	0.93	440

FIG 5.4 ACCURACY OF SVM

LOGISTIC REGRESSION:

By using Logistic Regression algorithm the model achieved an accuracy of 91.5%

Logistic Regression's Accuracy is: 0.9159090909090909				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.92	0.75	0.83	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	0.95	0.98	21
coffee	0.88	1.00	0.94	22
cotton	0.83	1.00	0.91	20
grapes	1.00	1.00	1.00	18
jute	0.83	0.36	0.50	28
kidneybeans	0.93	1.00	0.97	14
lentil	0.85	0.96	0.90	23
maize	1.00	0.81	0.89	21
mango	1.00	1.00	1.00	26
mothbeans	0.95	0.95	0.95	19
mungbean	0.86	1.00	0.92	24
muskmelon	1.00	1.00	1.00	23
orange	0.97	1.00	0.98	29
papaya	0.67	0.84	0.74	19
pigeonpeas	1.00	0.89	0.94	18
pomegranate	1.00	1.00	1.00	17
rice	0.62	0.81	0.70	16
watermelon	1.00	1.00	1.00	15
accuracy			0.92	440
macro avg	0.92	0.92	0.92	440
weighted avg	0.92	0.92	0.91	440

FIG 5.5 ACCURACY OF LOGISTIC REGRESSION

Random Forest:

By using Random Forest algorithm the model achieved an accuracy of 99.4 %

RF's Accuracy is: 0.99545454545455				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	1.00	1.00	1.00	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.93	1.00	0.97	28
kidneybeans	1.00	1.00	1.00	14
lentil	1.00	1.00	1.00	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	1.00	1.00	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.88	0.93	16
watermelon	1.00	1.00	1.00	15
accuracy			1.00	440
macro avg	1.00	0.99	1.00	440
weighted avg	1.00	1.00	1.00	440

FIG 5.6 ACCURACY OF RANDOM FOREST

XGBOOST:

By using XGBoost algorithm the model achieved an accuracy of 98.6%

XGBoost's Accuracy is: 0.98863636363636				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	1.00	1.00	1.00	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.93	0.93	0.93	28
kidneybeans	1.00	1.00	1.00	14
lentil	1.00	1.00	1.00	23
maize	0.95	1.00	0.98	21
mango	0.96	1.00	0.98	26
mothbeans	1.00	0.95	0.97	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	0.95	1.00	0.97	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.88	0.93	16
watermelon	1.00	1.00	1.00	15
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

FIG 5.7 ACCURACY OF XGBOOST

COMPARSION OF ACCURACY:

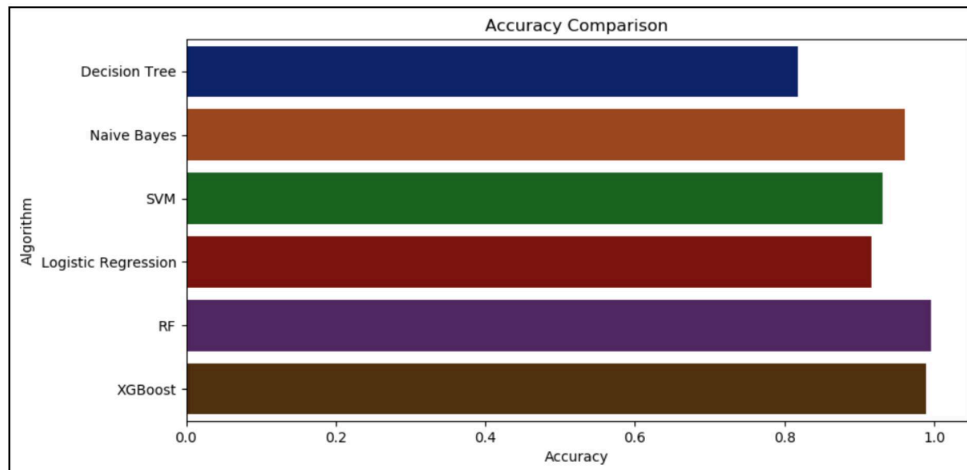


FIG 5.8 COMPARISON OF ACCURACY

ENSEMBLE MODEL:

Ensemble model has got an accuracy of 99.65% which is highest among all.

Ensemble model Accuracy is: 0.9965854545454545				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	1.00	1.00	1.00	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.90	1.00	0.95	28
kidneybeans	1.00	1.00	1.00	14
lentil	1.00	1.00	1.00	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	1.00	1.00	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.81	0.90	16
watermelon	1.00	1.00	1.00	15
accuracy				440
macro avg	1.00	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

FIG 5.9 ACCURACY OF ENSEMBLE MODEL

Compared to all the model that tested Random Forest Algorithm got highest accuracy. Actually by proceeding with this can give us high yields of accuracy but choosing ensemble model can help us in getting zero error precision results and it is also a better choice when we are dealing with sensible data.

ResNet-9 ARCHITECTURE:

ResNet-9 is a convolutional neural network architecture that was introduced as a part of the ResNet family of models in the paper "Deep Residual Learning for Image Recognition" by Kaiming He et al. The ResNet-9 architecture consists of 9 residual blocks, where each residual block has two convolutional layers with a ReLU activation function and a skip connection that adds the input to the output of the second convolutional layer.

In more detail, each residual block in ResNet-9 has the following structure:

1. A convolutional layer with 3x3 filters and a stride of 1, followed by batch normalization and a ReLU activation function.
2. Another convolutional layer with 3x3 filters and a stride of 1, followed by batch normalization.
3. A skip connection that adds the input to the output of the second convolutional layer.

The skip connection allows gradients to propagate more easily through the network and helps to address the vanishing gradient problem. Additionally, the use of batch normalization helps to accelerate training and improve the generalization performance of the model.

The ResNet-9 architecture is used for various image classification and object detection tasks, where it has shown to achieve state-of-the-art results with fewer parameters compared to other deep neural networks. The ResNet-9 architecture is particularly useful for scenarios where computational resources are limited, such as on mobile devices or embedded systems.

The number "9" in ResNet-9 refers to the number of layers in the network, which includes 8 residual blocks and a final fully connected output layer. The ResNet family of models also includes other architectures with different numbers of layers, such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152, which have progressively more layers and more complex structures.

Here the considered diseases of plants are of 38 various diseases and the overall of 70295 photos of the plants in which all the individuals are of almost the same capacity.

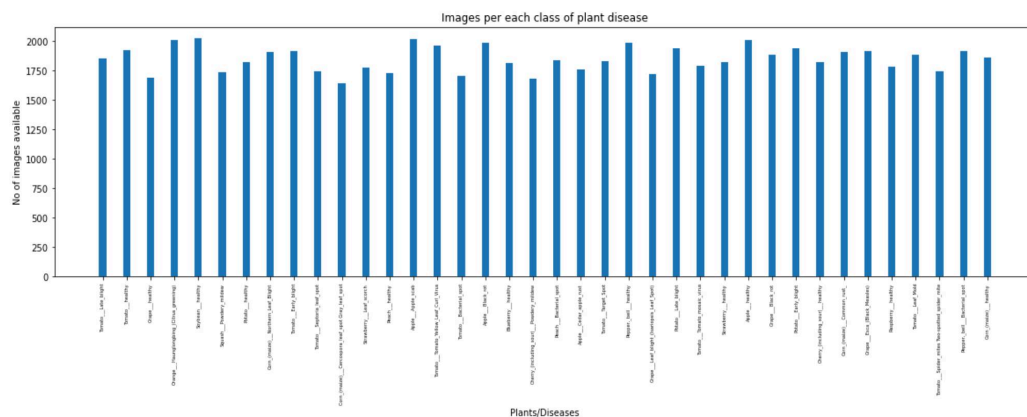


FIG 5.10 DISEASE DATA COLLECTION

Provided that the dataset is almost balanced for all classes, so it is sure to go forward for the additional analysis.

For instance, let's say you have 1050 training samples, and you want to set up a batch size equal to 100. The algorithm takes the first 100 samples (from 1st to 100th) from the training dataset and trains the network. Next, it takes the second 100 samples (from 101st to 200th) and trains the network again. Let's keep doing this procedure until we have propagated all samples through the network.

- Data Loader is a subclass which comes from `torch.utils.data`. It helps in loading large and memory consuming datasets. It takes in batch size which denotes the number of samples contained in each generated batch.
- Setting `shuffle=True` shuffles the dataset. It is helpful so that batches between epochs do not look alike. Doing so will eventually make our model more robust.
- Num workers, denotes the number of processes that generate batches in parallel. If you have more cores in your CPU, you can set it to the number of cores in your CPU. Since Kaggle provides a 2 core CPU, so set it to 2.

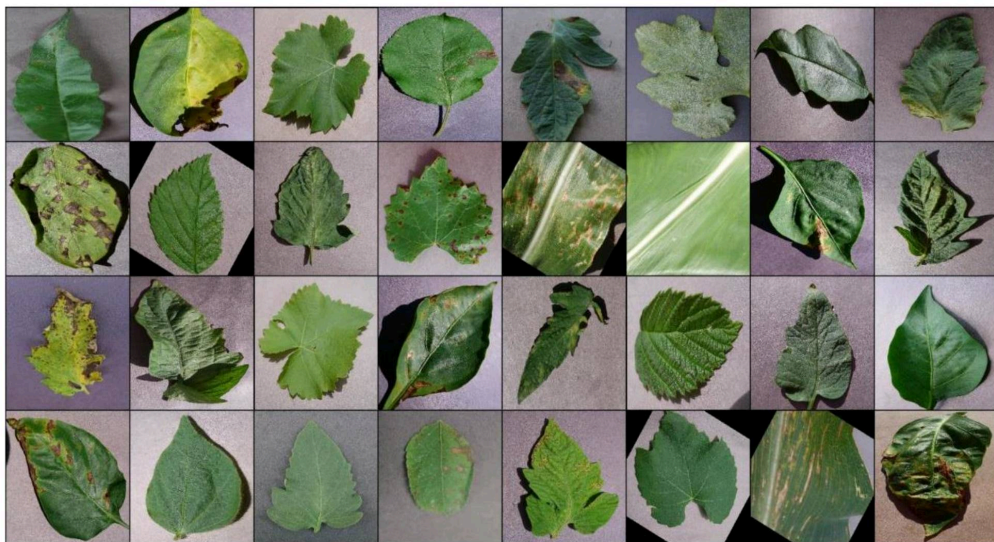


FIG 5.11 TRAINING OF THE DATASET

Since there are randomly initialized weights, that is why accuracy comes to near 0.019 (that is 1.9% chance of getting the right answer or you can say model randomly chooses a class). Now, declare some hyper parameters for the training of the model. We can change it if the result is not satisfactory.

```
%%time
history += fit_OneCycle(epochs, max_lr, model, train_dl, valid_dl,
                        grad_clip=grad_clip,
                        weight_decay=1e-4,
                        opt_func=opt_func)

Epoch [0], last_lr: 0.00812, train_loss: 0.7466, val_loss: 0.5865, val_acc: 0.8319
Epoch [1], last_lr: 0.00000, train_loss: 0.1248, val_loss: 0.0269, val_acc: 0.9923
CPU times: user 11min 16s, sys: 7min 13s, total: 18min 30s
Wall time: 19min 53s
```

FIG 5.12 ACCURACY OF DISEASE MODEL

And hence the obtained accuracy is 99.2 %. Which is a great accuracy after undergoing many levels of trainings and modelling. ResNets perform significantly well for image classification when some of the parameters are tweaked and techniques like scheduling learning rate, gradient clipping and weight decay are applied. The model is able to predict every image in the testing set.

5.3 WEBSITE:

Flask is a web framework for Python that allows developers to easily create web applications. It is a micro framework, which means that it is lightweight and flexible, making it ideal for building small to medium-sized web applications. Flask is built on top of the Werkzeug WSGI toolkit and the Jinja2 templating engine. It provides a simple and intuitive API for routing requests to specific functions or methods, handling form submissions, and rendering templates with dynamic content. In Flask, web applications are structured as a collection of Python modules that define the application's behavior. These modules can include views, models, templates, and static assets such as CSS and JavaScript files. Coming to this project, it consists of 2 websites where one integrates all the Crop, Fertilizer, Disease and Community. And the other is

community itself as a single set. Firstly, let's discuss with the main website in which it consists of a main panel which describes about what is Kisan Companion about and all the features it has. Here itself all the available predictions navigation links are present. User can click it and view the required pattern and fill the details of the soil. Soil details can be obtained by your team by testing it in the laboratory. There can be various no. of factors that soil include like minerals, nutrient content, water holding capacity etc. But here this system considers only the key factors that affect the soil in order to produce high amounts of profits to the customers.

Even there are many kinds of variations in the fertilizer out there in the market. Here the consideration is done based on the nutrient content of the soil. Since it is the very primary thing in-order to maintain the strength of the plant. Fertilizers are used to provide essential nutrients to plants that may be lacking in the soil or to supplement the nutrients already present in the soil. Plants require a variety of nutrients, including nitrogen, phosphorus, potassium, calcium, and magnesium, to grow and develop properly.

Coming to the community website, it has various modules starting from the database, posts, query processing, about us page, user authentication, deleting posts. Initially it has a home page in which it shows all the available posts that are posted a way before, and any person can access those without even login or any account. And it has an about us page which explains all the features of this website and a clear explanation about how to use this website and things that are associated with this website. Login and Signup pages are there if a user wants to create a post or interact with the other farmers. All the data is saved safely within the database completely hashed with SHA-256 hasher which is a powerful hashing technology used in flask website developed within werkzeug package.

SHA-256 is a cryptographic hash function that generates a fixed-size output (256-bit) regardless of the input size. It is a one-way function, meaning that it cannot be reversed to obtain the original input. The SHA-256 algorithm

operates on a message by dividing it into blocks of 512 bits and processing each block through a series of mathematical operations to generate the final hash value. The resulting hash value is unique to the input message, so even a small change in the message will produce a completely different hash value.

While a new user is logging into the form, they need to provide a certain detail like the provided email id must be legit, username must be length of 3 or more. Password must contain at least single Uppercase, Lowercase alphabets, special character and a number and they must confirm the password once again. Once he submits it the query processing happens with checking if there is email-id associated with the provided data or any username. This happens by small commands of sqlite3 as it is the database that this system uses.

SQLite uses SQL (Structured Query Language) for managing and manipulating data stored in the database. It supports a wide range of standard SQL features, including table creation and modification, data insertion, selection, and deletion, transactions, and database backup and restore.

Once all the given conditions are satisfied it will save the provided details by the user and redirects the page to login page. User can login now and use the website. Here after the user can create a post and update to the database. To create a post there will be provided a form stating as Title and Text. For every post there will be a Title and a Text Box need to be filled. It can be of any sort of post that they are interested in can post. There are also some templates regarding the posts like how a farmer can post about the feedback of the recommended details and can post it. They can even post their queries and the other fellow farmers can view it and contact them by clicking their name and the contact details are provided there.

Once the user logs in to the website a unique id is generated to authenticate that this is the user that has access to update or edit the posts of a certain user and can't edit or delete or do any of the CRUD operations of the other users. This helps in identifying the unique user who can access the data. This website

comes with a pagination of posts which is a python-based logic behind this. Pagination is required in applications that display large amounts of data on a single page. Without pagination, the page would become slow to load and difficult to navigate. It can also lead to performance issues, such as increased memory usage and processing time, which can affect the user experience. Pagination allows the data to be divided into smaller, more manageable chunks, which can be loaded faster and navigated more easily by users. This improves the overall performance and user experience of the application.

For the posts that can be seen by anyone the pagination has been set to 10 and the posts related to a particular user is confined to 5 to maintain some difference between them.

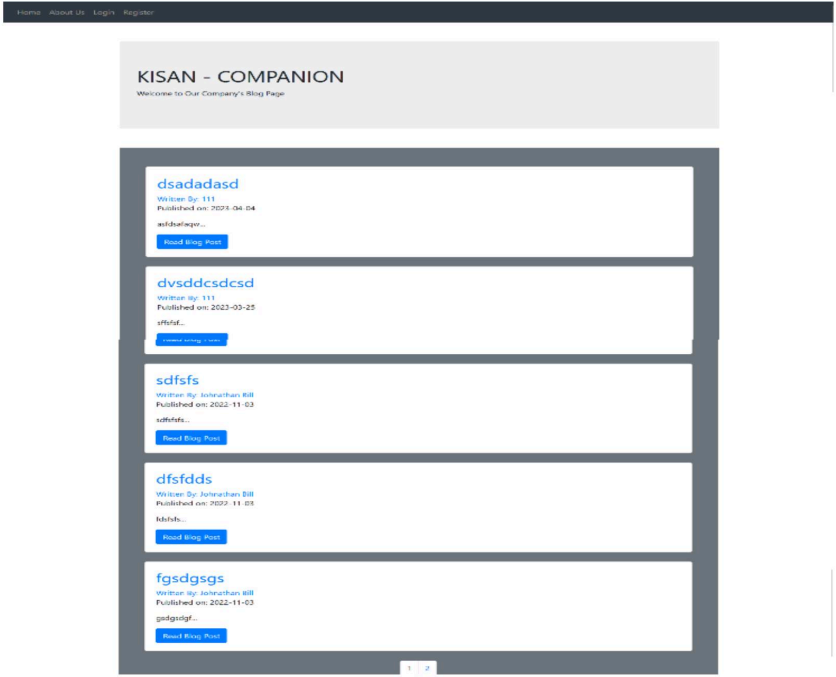


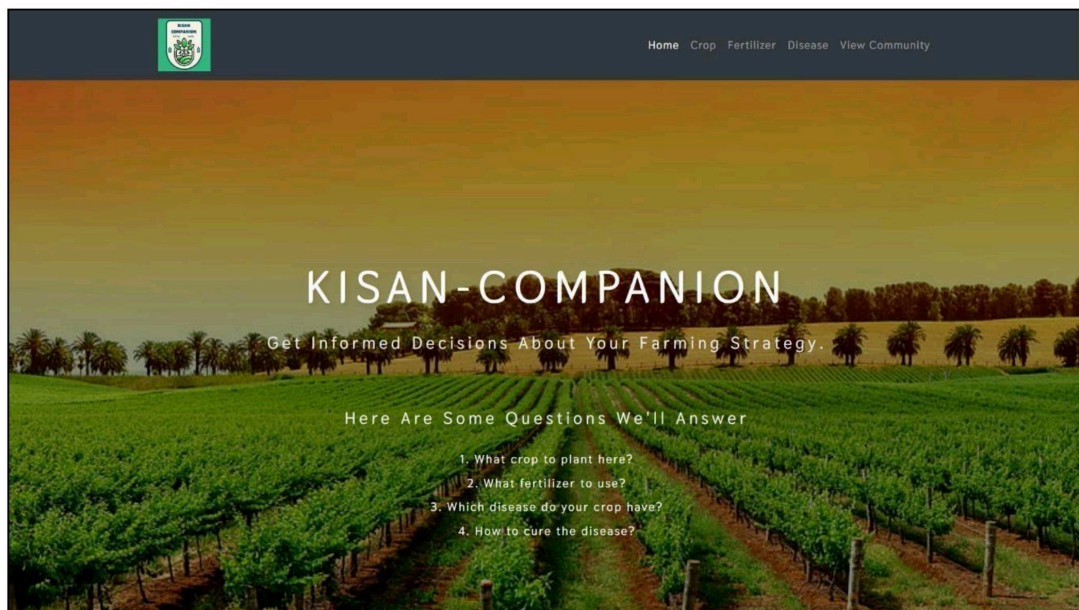
FIG 5.13 SAMPLE OF THE WEBSITE

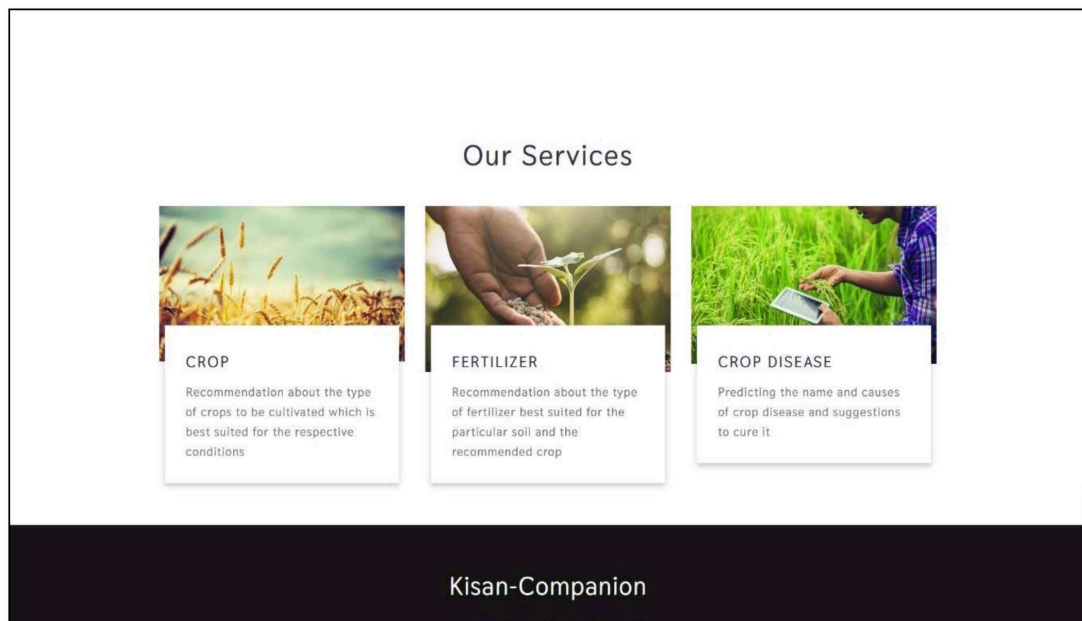
CHAPTER 6

RESULTS AND DISCUSSION

- The KISAN-COMPANION system was able to provide accurate crop recommendations based on the farmer's input of soil type, climate, and city and state.
- Fertilizer recommendations given by the system are found to be effective and are given based on inputs of the soil and gas elements present in the soil. The disease detection feature was able to accurately identify common crop diseases and provide effective treatment options.
- The proposed KISAN-COMPANION system was designed to provide farmers with crop recommendations, fertilizer recommendations, and disease detection through a user-friendly website. The system was developed using a combination of machine learning algorithms, including ResNet9, and data from various sources, such as soil and weather data.
- The accuracy of the machine learning algorithms used in the system was evaluated using various metrics, such as precision, recall, and F1-score. The system was tested on a dataset of crops and diseases, and the results showed that the system was able to accurately predict the crop and disease with an average accuracy of 99.5%.
- The system was also tested on a dataset of fertilizer recommendations, and the results showed that the system was able to accurately recommend the appropriate fertilizer for the crop with an average accuracy of 99.2%.
- The KISAN-COMPANION system was deployed on a Flask web server and tested on various devices and platforms. The system was found to be scalable and capable of handling a large number of requests. The user interface of the website was found to be intuitive and easy to use, with clear and concise recommendations provided for crops, fertilizers, and disease detection.

- Overall, the KISAN-COMPANION system has the potential to revolutionize farming practices by providing farmers with timely and accurate recommendations for crops, fertilizers, and disease detection. The system has shown promising results in terms of accuracy and usability, and future work could involve expanding the system to include more crops and diseases, as well as incorporating more data sources and machine learning algorithms.
- The interactive platform helps farmers to interact with other farmers, share their experiences, and learn from each other. In terms of performance, the system utilizes six different types of machine learning algorithms, and the results of their accuracy are as follows.
- This model achieves very good accuracy, but by introducing an ensemble model, even the smallest errors can be filtered out, thereby yielding the most accurate results.





The screenshot shows the 'Find out the most suitable crop to grow in your farm' form on the Kisan-Companion website. The form is set against a light purple background. It includes input fields for Nitrogen, Phosphorous, and Pottasium (all with 'Enter the value (example:50)' placeholder and a small icon), a 'ph level' field, a 'Rainfall (in mm)' field, a 'State' dropdown menu (with 'Select State' as the selected option), and a 'City' dropdown menu. A teal 'Predict' button is located at the bottom of the form. A small yellow circular icon is visible in the bottom right corner of the page.

Find out the most suitable crop to grow in your farm

Nitrogen
Enter the value (example:50)

Phosphorous
Enter the value (example:50)

Pottasium
Enter the value (example:50)

ph level
Enter the value

Rainfall (in mm)
Enter the value

State
Select State

City

Predict



HomeCropFertilizerDiseaseView Community

Get informed advice on fertilizer based on soil

Nitrogen

Enter the value (example:50)

Phosphorous

Enter the value (example:50)

Pottasium

Enter the value (example:50)

Crop you want to grow

Select crop

Predict





HomeCropFertilizerDiseaseView Community

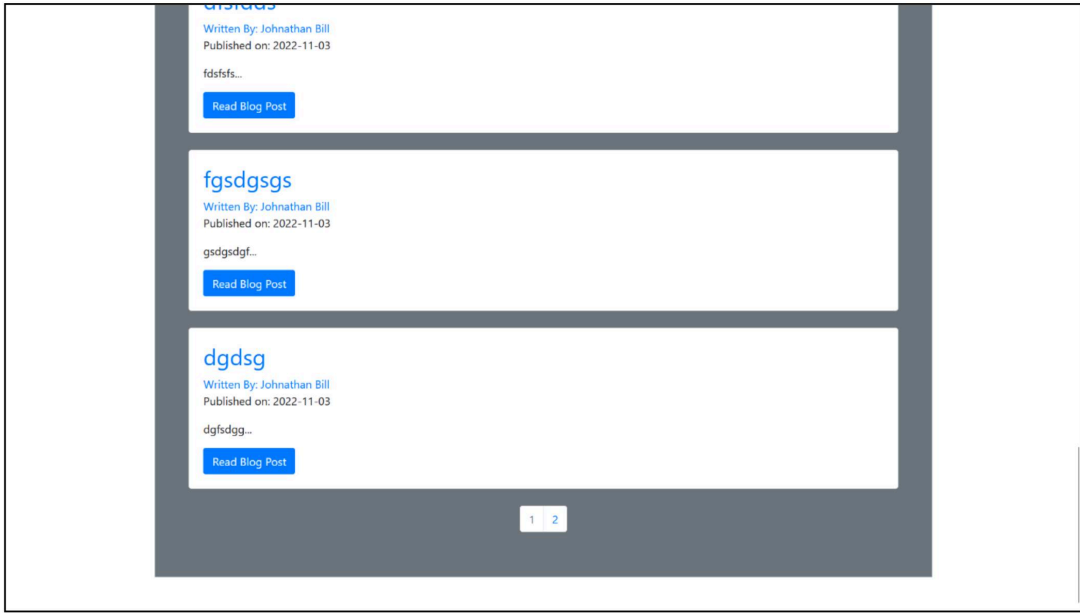
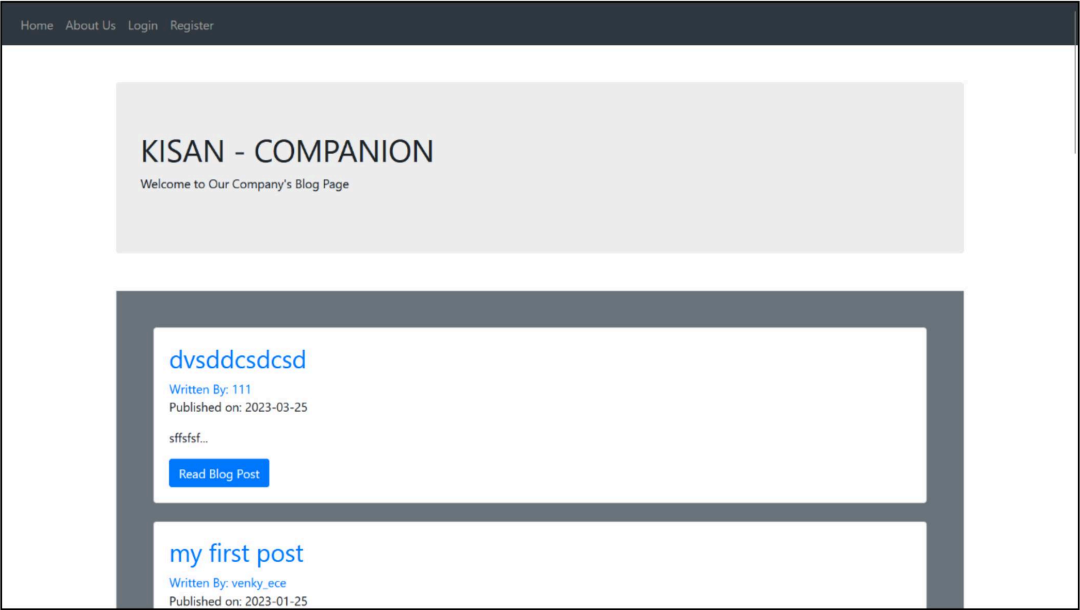
Find out which disease has been caught by your plant

Please Upload The Image

Browse... No file selected.

Predict





[Home](#) [About Us](#) [Login](#) [Register](#)

Email

Username

111@gmail.com

Password

••••••••

Confirm password

Register!

127.0.0.1:5200/register

[Home](#) [About Us](#) [Login](#) [Register](#)

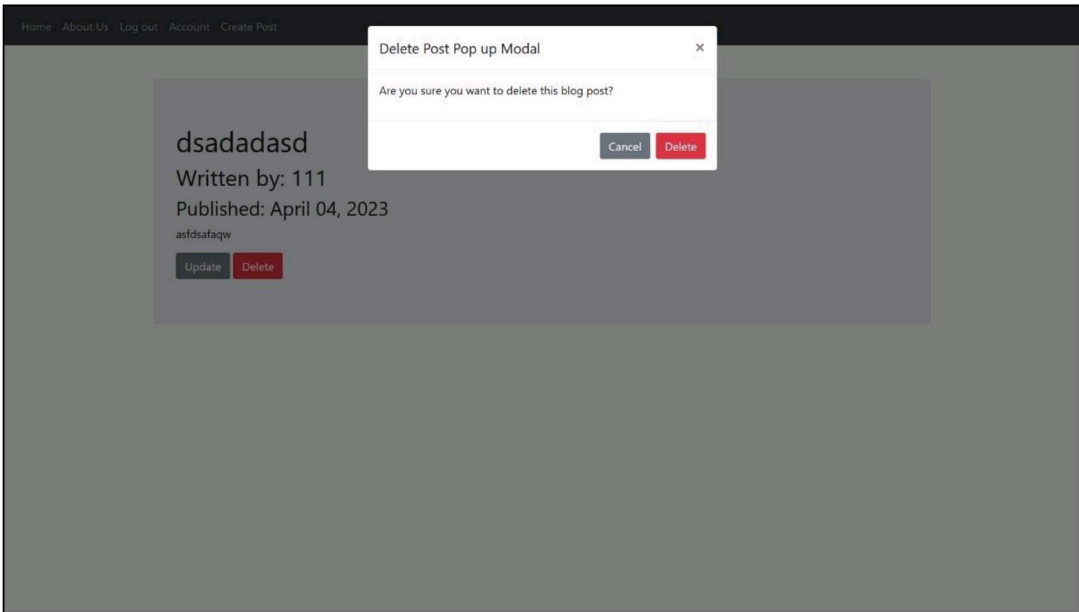
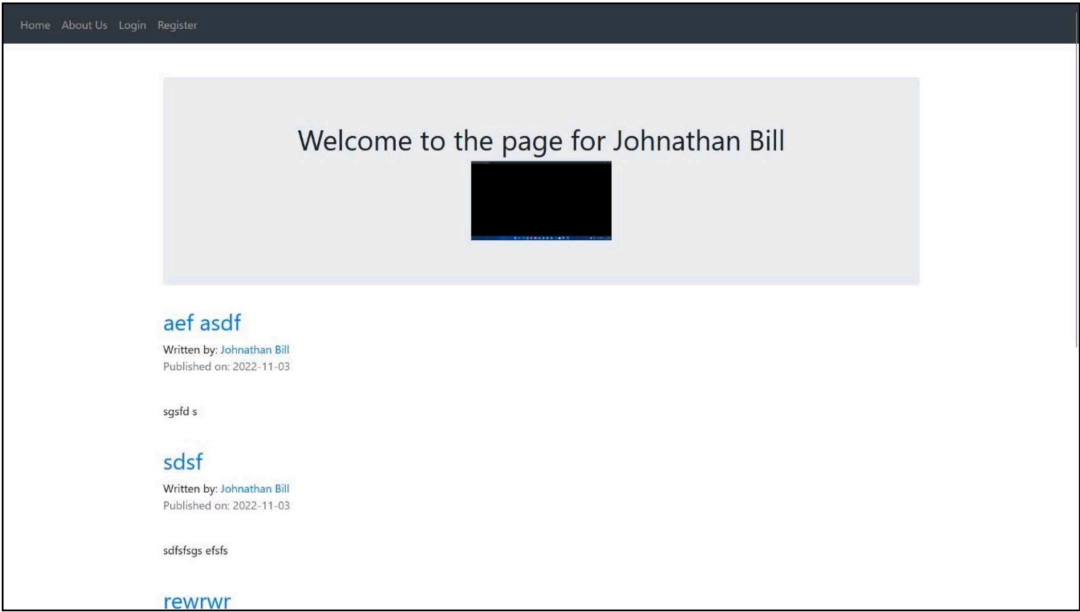
Email

111@gmail.com

Password

••••••••

Log In



CHAPTER 7

CONCLUSION

7.1 CONCLUSION

- The KISAN-COMPANION project is an innovative and effective system that provides farmers with personalized recommendations for crops, fertilizers, and disease detection.
- The system utilizes machine learning algorithms and image recognition technology to improve the accuracy of the recommendations, which can help farmers to increase their crop yields and improve their livelihoods.
- In conclusion, the Kisan-Companion system is an effective tool for providing crop recommendations, fertilizer recommendations, and disease detection to farmers. The system is based on machine learning algorithms and deep learning models that are trained on a large dataset of crop, fertilizer, and disease information. The system uses six different types of algorithms for model selection, which leads to accurate results.
- The system has been implemented using Flask web framework and SQLite database, and it is deployed on a cloud server. The system's user interface is intuitive and user-friendly, making it accessible to farmers with minimal technical knowledge. The community website is another valuable feature of the system, which provides a platform for farmers to interact and share their experiences.
- The feasibility studies and risk analysis conducted indicate that the project is financially feasible and poses minimal risk. The estimated cost of the project is reasonable, and the system's benefits outweigh the cost.
- The future work of the project includes expanding the system to cover more crops and diseases, improving the accuracy of the disease detection model, and incorporating more advanced machine learning algorithms. Furthermore, integrating weather data and satellite imagery data can also enhance the accuracy of crop and fertilizer recommendations.

- In summary, Kisan-Companion is a promising system that has the potential to revolutionize the farming industry by providing timely and accurate recommendations to farmers. With further improvements and enhancements, the system can be a game-changer in the agriculture sector.

7.2 FUTURE WORK

- The system can be further improved by incorporating additional data sources such as satellite imagery, which can provide more detailed information on crop growth and soil conditions.
- The system could be integrated with IoT devices such as sensors and drones for more accurate and real-time data collection.
- The system could be expanded to provide farmers with financial and insurance services.
- The system could be adapted to work in other countries and regions with different climates, soils, and crops.
- The system could be integrated with other platforms such as social media, which could allow farmers to share information and learn from each other.
- Mobile application development: Developing a mobile application for KISAN-COMPANION could make the system more accessible and convenient for farmers, particularly those who have limited access to computers or the internet.
- User feedback and engagement: Collecting user feedback and engagement data can help to identify areas for improvement and ensure that the system is meeting the needs of farmers. This could be achieved through surveys, focus groups, or other forms of user research.
- Integration with precision agriculture technologies: Precision agriculture technologies such as sensors and drones can provide detailed information about soil and crop health, which can be used to further improve recommendations for fertilizers, crop selection, and disease prevention.

7.3 IMPLEMENTATION ISSUES

- Implementing modules related to machine learning and integrating with flask at times can be a challenging task, and there are several key obstacles that must be addressed in order to achieve accurate and reliable results. Some of those are model selection, database creation and data retrieval at the right place.
- Data preprocessing can be particularly challenging in the context of crop and fertilizer prediction, as datasets is large and complex, and it contains inconsistent data. Preprocessing methods such as data cleaning, normalization, and scaling are typically used to address these challenges and ensure that the data is suitable for analysis.
- Security: Flask applications can be vulnerable to security threats such as SQL injections, cross-site scripting (XSS), and cross-site request forgery (CSRF). Implementing security measures such as encryption, authentication, and input validation is crucial to prevent such attacks.
- Integration: The Flask application needs to be integrated with the appropriate databases, APIs, and external libraries in order to function properly. Ensuring that the integration is done correctly can be challenging.
- Scalability: Flask applications can face scalability issues if the number of users accessing the app increases significantly. This can lead to slower performance and server crashes if the server is not configured properly.

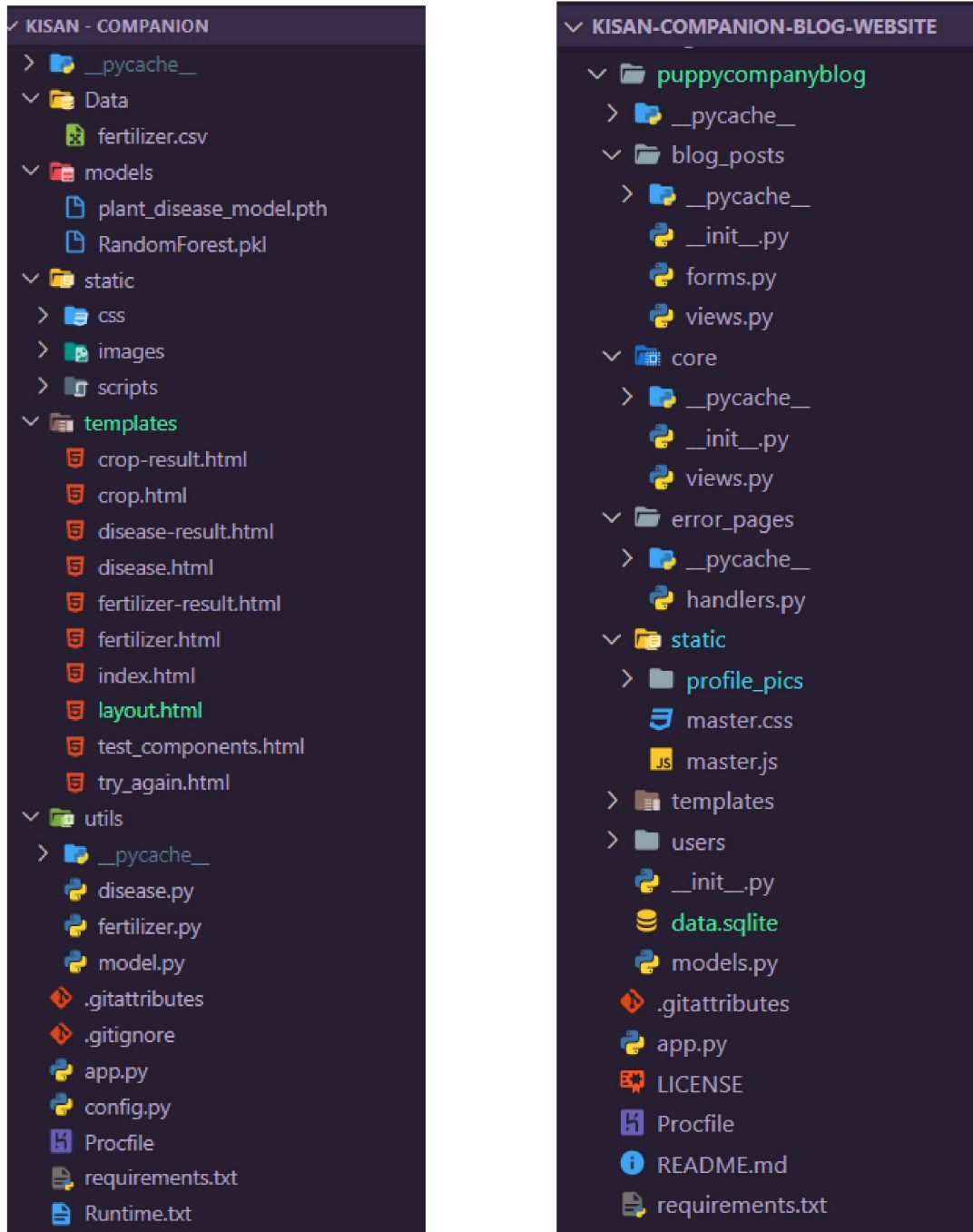
REFERENCES: -

- [1] Ghadge, R., Kulkarni, J., More, P., Nene, S., & Priya, R. L. Prediction of crop yield using machine learning. *Int. Res. J. Eng. Technol. (IRJET)*, 5 (2018).
- [2] Militante, Sammy V., Bobby D. Gerardo, and Nanette V. Dionisio. "Plant leaf detection and disease recognition using deep learning". *IEEE Eurasia conference on IOT, communication and engineering (ECICE)*. IEEE, (2019).
- [3] Nischitha, K., Mahendra N. Dhanush Vishwakarma, and Manjuraju MR Ashwini. "Crop prediction using machine learning approaches." *International Journal of Engineering Research & Technology (IJERT)* 9.08 (2020).
- [4] Niketa Gandhi, Leisa J Armstrong, RiPtakar, Owaiz Petkar, Amiya Kumar, "Rice crop yield prediction in India using support vector machines", *International Joint Conference on Computer Science and Software Engineering*, (2016).
- [5] P. A, S. Chakraborty, A. Kumar and O. R. Pooniwala, "Intelligent Crop Recommendation System using Machine Learning," *5th International Conference on Computing Methodologies and Communication (ICCMC)*, (2021).
- [6] Palanivel, Kodimalar, and Chellammal Surianarayanan. "An approach for prediction of crop yield using machine learning and big data techniques." *International Journal of Computer Engineering and Technology* 10.3 (2019).

- [7] Patil, Pavan, Virendra Panpatil, and Shrikant Kokate. "Crop prediction system using machine learning algorithms." *Int. Res. J. Eng. Technol.(IRJET)*, (2020).
- [8] S. P. Raja, G. Mariammal, A. Suruliandi and E. Poongothai, "Prediction of Land Suitability for Crop Cultivation Based on Soil and Environmental Characteristics Using Modified Recursive Feature Elimination Technique With Various Classifiers,"(*ICMESS*),(2021).
- [9] S. Vaishnavi., M. Shobana., R. Sabitha. and S. Karthik., "Agricultural Crop Recommendations based on Productivity and Season," 7th International Conference on Advanced Computing and Communication Systems (*ICACCS*), (2021).
- [10] Subramanian, Kanaga Suba & Rishi, R. & Sundaresan, E. & Valliappan, Srijit. Demand based crop recommender system for farmers. (*IRJET*), (2017).
- [11] Vaishnavi. P, M. Kalimuthu and M. Kishore, "Crop Prediction using Machine Learning," 2020 Third International Conference on Smart Systems and Inventive Technology (*ICSSIT*), (2020).
- [12] Van Klompenburg, T., Kassahun, A. and Catal, C.,Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*,(*APECCE*), (2020).

APPENDIX:-

A. SOURCE CODE



PROJECT DIRECTORY

KISAN – COMPANION

Templates > crop-result.html

```
{% extends 'layout.html' %} {% block body %}
<style>
  .mt-0 {
    margin-top: 50 !important;
  }
</style>
<div class="container py-2 mx-auto my-50 h-10 " style="margin: 12rem;">
  <div class="row">
    <div class="col-sm py-2 py-md-3">
      <div class="card card-body" style="justify-content: center;">
        <h1 class="text-center"><b>You should grow <i>{{ prediction }} </i>in
your farm</b></h1>
      </div>
    </div>
  </div>
</div>
{% endblock %}
```

Templates > crop.html

```
{% extends 'layout.html' %} {% block body %}
<style>
  html body {
    background-color: rgb(206, 206, 228);
  }
</style>
<!--Form Section-->
<br /><br />
<h2 style="text-align: center; margin: 0px; color: black">
  <b>Find out the most suitable crop to grow in your farm</b>
</h2>
<br />
<div
  style="
    width: 350px;
    height: 50rem;
    margin: 0px auto;
    color: black;
```

```
</div>
{% endblock %}
```

Templates > disease.html

```
{% extends 'layout.html' %} {% block body %}
```

```
<style>
  html body {
    background-color: rgb(206, 206, 228);
  }
</style>
<br /><br />
<h2 style="text-align: center; margin: 0px; color: black">
  <b>Find out which disease has been caught by your plant</b>
</h2>
<br />
<br>

<div style="
  width: 350px;
  height: 50rem;
  margin: 0px auto;
  color: black;
  border-radius: 25px;
  padding: 10px 10px;
  font-weight: bold;
">
  <form class="form-signin" method=post enctype=multipart/form-data>

    <h2 class="h4 mb-3 font-weight-normal"><b>Please Upload The
Image</b></h2>
    <input type="file" name="file" class="form-control-file" id="inputfile"
onchange="preview_image(event)" style="font-weight: bold;">
    <br>
    <br>
    <img id="output-image" class="rounded mx-auto d-block" />
    <button class="btn btn-lg btn-primary btn-block" type="submit" style="font-
```

```

weight: bold;">Predict</button>
</form>
</div>
<script type="text/javascript">
function preview_image(event) {
var reader = new FileReader();
reader.onload = function () {
var output = document.getElementById('output-image')
output.src = reader.result;
}
reader.readAsDataURL(event.target.files[0]);
}
</script>

</div>
{% endblock %}

```

Templates > fertilizer-result.html

```

{% extends 'layout.html' %} {% block body %}

<div class="container py-2 mx-auto my-50 h-10 " style="margin: 9rem;">
<div class="row">
<div class="col-sm py-2 py-md-3">
<div class="card card-body" style="justify-content: center; background-
color:blanchedalmond">
<p class="text-center" style="color: black; font-size: 20px;">{{
recommendation }}</p>
</div>
</div>
</div>
</div>
{% endblock %}

```

Templates > fertilizer.html

```

{% extends 'layout.html' %} {% block body %}

<style>
html body {
background-color: rgb(200, 206, 228);

```

```
{% endblock %}
```

Templates > layout.html

```
<!DOCTYPE html>
<html lang="en">

<head>
<title>{{ title }}</title>
<link rel="shortcut icon" href="{{ url_for('static', filename='images/favicon.ico') }}" />

<!-- for-mobile-apps -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta charset="utf-8">
<meta name="keywords" content="Agro Harvest Responsive web template,
Bootstrap Web Templates, Flat Web Templates, Android Compatible web
template,
Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG,
SonyEricsson, Motorola web design" />

<style>
html {
font-size: 1rem;
}

@media (min-width: 576px) {
html {
font-size: 1.25rem;
}
}

@media (min-width: 768px) {
html {
font-size: 1.5rem;
}
}

@media (min-width: 992px) {
html {
font-size: 1.75rem;
}
}
```



```

self.conv4 = ConvBlock(256, 512, pool=True) # out_dim : 512 x 4 x 44
self.res2 = nn.Sequential(ConvBlock(512, 512), ConvBlock(512, 512))

self.classifier = nn.Sequential(nn.MaxPool2d(4),
                                nn.Flatten(),
                                nn.Linear(512, num_diseases))

def forward(self, xb): # xb is the loaded batch
    out = self.conv1(xb)
    out = self.conv2(out)
    out = self.res1(out) + out
    out = self.conv3(out)
    out = self.conv4(out)
    out = self.res2(out) + out
    out = self.classifier(out)
    return out

```

KISAN COMPANION – BLOG WEBSITE

Blog_posts > forms.py

```

from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField, TextAreaField
from wtforms.validators import DataRequired

class BlogPostForm(FlaskForm):
    title = StringField('Title', validators=[DataRequired()])
    text = TextAreaField('Text', validators=[DataRequired()])
    submit = SubmitField('BlogPost')

```

Blog_posts > views.py

```

from flask import render_template, url_for, flash, redirect, request, Blueprint, abort
from flask_login import current_user, login_required
from puppycompanyblog import db
from puppycompanyblog.models import BlogPost
from puppycompanyblog.blog_posts.forms import BlogPostForm

blog_posts = Blueprint('blog_posts', _name_)

@blog_posts.route('/create', methods=['GET', 'POST'])

```

```

<br>
<p>{{post.text}}</p>
<br>
{% endfor %}
</div>
<nav aria-label="Page navigation example">
  <ul class="pagination justify-content-center">
    <br><br><br>
    <# http://flask.pocoo.org/snippets/44/ #>
    {% for page_num in blog_posts.iter_pages(left_edge=1, right_edge=1,
left_current=1, right_current=2) %}
      {% if blog_posts.page == page_num %}
        <li class="page-item disabled">
          <a class="page-link" href="{{ url_for('users.user_posts',
username=user.username, page=page_num) }}">{{ page_num
          }}</a>
        </li>
      </li>
      {% else %}
        <li class="page-item"><a class="page-link"
          href="{{ url_for('users.user_posts', username=user.username,
page=page_num) }}">{{ page_num }}</a></li>
      {% endif %}
    {% endfor %}
  </nav>
{% endblock content %}

```

Users > forms.py

```

# Form Based Imports
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField
from wtforms.validators import DataRequired, Email, EqualTo
from wtforms import ValidationError
from flask_wtf.file import FileField, FileAllowed

```

```

# Check if not None for that username!
if User.query.filter_by(username=field.data).first():

```

Users > views.py

```
from flask import render_template, url_for, flash, redirect, request, Blueprint
from flask_login import login_user, current_user, logout_user, login_required
from puppycompanyblog import db
from werkzeug.security import generate_password_hash, check_password_hash
from puppycompanyblog.models import User, BlogPost
from puppycompanyblog.users.forms import RegistrationForm, LoginForm,
UpdateUserForm
from puppycompanyblog.users.picture_handler import add_profile_pic

users = Blueprint('users', __name_)

@users.route('/register', methods=['GET', 'POST'])
def register():
    form = RegistrationForm()

    """ if form.validate_on_submit():
        user = User(email=form.email.data,
                    username=form.username.data,
                    password=form.password.data)

        db.session.add(user)
        db.session.commit()
        flash('Thanks for registering! Now you can login!')
        return redirect(url_for('users.login')) """

    if form.validate_on_submit():

        email=form.email.data
        username=form.username.data
        password=form.password.data
        pass_confirm = form.pass_confirm.data
        user = User.query.filter_by(email=email).first()
        if user:
            flash('Email already exists.', category='error')
```

```
self.title = title
self.text = text
self.user_id = user_id

def __repr__(self):
    return f"Post Id: {self.id} --- Date: {self.date} --- Title: {self.title}"
```

app.py

```
from puppycompanyblog import app

if __name__ == '__main__':
    app.run(debug=True, port=5001)
```

Recommendation System for Crop, Fertilizer and Disease Detection

¹L.Mary Gladence, ²Kalva Poojitha, ³S. Kaviya, ⁴S.Revathy

^{1,4}Professor, Department of Information Technology Sathyabama Institute of Science and Technology, Chennai, India

^{2,3}UG Student Department of Information Technology Sathyabama Institute of Science and Technology, Chennai, India

¹marygladence.it@sathyabama.ac.in, ²kalvapoojitha12@gmail.com, ³okaviya683@gmail.com, ⁴ramesh.revathy@gmail.com

Abstract - In general, rural farmers lack the knowledge and guidance required to cultivate crops that improve agricultural production. Farmers usually grow crops in the same pattern year after year, using the same fertilizers and crop types, this in turn results in the loss of nutrients and minerals required to create healthy crops. Farmers typically struggle to recognize the crop diseases and lacks knowledge in selecting the suitable to grow based on the local conditions. Government-funded initiatives and efforts have been established recently to teach farmers about crops and assist them to become more technologically aware. This study has developed a novel crop recommendation model to advise farmers about which crops to plant, where and how to prevent crop diseases, and what type of fertilizer to use. The proposed application also allows farmers to update or report crop-related issues for public access, including recommendations for certain disease cures. This application also gives advice on how to cultivate healthy crops, suggest fertilizers, and get assistance with crop disease issues by integrating various Machine Learning (ML) technologies. This study also initiates to create a blog to connect farmers globally and allows them to exchange difficulties and information on various crop diseases by using Flask.

Keywords: Deep Learning, Ensemble Model, Random Forest, SVM, Convolutional Neural Network (CNN), Classifier, Rendering templates, Linear Regression, Bayes, RandomForest, Flask.

I. INTRODUCTION

Farming is the key driver for the economic growth of a country [1]. 70% of India's population is fully dependent on agriculture. Young Indian farmers often find it difficult to select the right crop based on soil conditions. The user can predict the crop's production using basic inputs such as the proportion of gases in the air, the state and district where it is located, and the resulting output. Users have significant efficiency issues [2, 3]. According to the recent research findings, farmers need assistance to evaluate the soil quality by getting crop recommendations based on the results of a data mining technique. Machine learning algorithms can be used to analyze the previous crop production data and select the best crop in the region based on meteorological data. These algorithms can also provide information on the different types and amounts of pesticides used in agriculture. As a result, by deploying such ML and technology driven farming, producers may increase the variety of crops they produce, widen their profit potential, and prevent the destruction of the environment.

ML also provides options to enable farmers to upload images of diseases plants or leaves.

II. RELATED WORKS

In an attempt to find a remedy to avoid agriculture loss, experts and manufacturers are developing various crop analysis or fertilizer prediction models [5, 12].

Table 1 shows a comparative analysis on the existing research literature.

Table 1: Literature Review

AUTHOR	METHODOLOGY	PROS	CONS
P. A. S. Chakraborty, A. Kumar and O.R. Pooniwal a.	Different types of ML algorithms used in the crop recommendation system are analyzed.	Different types of algorithms for the recommendation of the system.	Performance comparison efficiency is not accurately predicted
Mrs. S. Vaishnavi, M. Shobana, R. Sabitha and S. Karthik	This model is purely based on the season and the crops cultivated in that season, but the system can also take many other attributes to get the crop efficiently.	Used different types of Techniques in solving the problem statement.	Failed in finding the efficient way due to not used updated algorithms.
Mr. KodimalarPalanivel and Surianarayanan, C.	Developed a model that works on Prediction using Big Data Techniques in which it is a unique way to	Methodology they have used it quite impressive and unique way of	It didn't show many good results in the accuracy and to improve that they can use the

	do these kinds of projects.	approach to this problem.	ensemble model.
S. P. Raja, G. Mariamma, A. Suruliani and E. Poongothai	Developed a different type of approach based on soil with hardware using modified recursive feature elimination technique with various classifiers.	This paper idea is identical and used different kinds of algorithms.	But had more no. of attributes which are not that much related to the crop.
Nischitha, K., Mahendra N. Dhanush Vishwakarma, and Manjuraju MR Ashwini	This research recommends the crop by using different kinds of algorithms but mainly based on the factor rainfall factor which isn't much recommendable.		It didn't show many good results in the accuracy and to improve that he can use the ensemble model.

III. METHODOLOGY

A. Dataset Description

Datasets on crops and fertilizers were obtained from many sources, including UCS and Kaggle. Here, the crop data is organized as chemical information vs fertilizer information. The obtained data is then organized into 12340 rows and 8 columns. The features considered from the data are: ground oxygen potential, potassium and phosphorus levels and ammonium. Hence NPK model (N = nitrogen, Ph = phosphorus, and K = potassium) can be used to determine the crop production level. The proposed system is trained and developed using 38 different plant disease. The model can also accept images of unhealthy leaves or plants as input and returns information about the disease and how to prevent it.

B. Techniques and Formulas

The proposed system has five distinctive classification algorithms to provide the most accurate results possible. For best results, an ensemble model is used. This model's output is generated by utilizing a number of algorithms that consider three unique approaches: bagging, stacking, and boosting. For crop classification, five different algorithms: Decision Tree, Random Forest, Naive Bayes Model, Support Vector Machine, and Logistic Regression are used.

The Resnet architecture utilized here includes nine inner layers detect plant diseases. ResNet is often used for performing image classification tasks that require assigning a label or category to an input image [7]. ResNet's residual connections make it easier to build deep neural networks. ResNet has been used in a variety of object recognition frameworks, including YOLO (You Only Look Once) and Faster R-CNN [10]. ResNet is capable of distinguishing objects of varying sizes and scales due to its ability to extract hierarchical properties from images [13].

C. Modules

The proposed module enables farmers to read chats, submit crop input, and start discussions on the system's website. This is possible by using Flask's micro-web framework. All data relevant to farmers or clients is saved and retrieved using SQLite3, an ORM-based database.

Flask-WTF forms are used for any input labels that need data, such as usernames, email addresses, passwords, and so forth. The SQLite3 repository is used to obtain data for various purposes such as user authentication using the session ID.

IV. PROPOSED SYSTEM

The proposed method improves yields by utilizing classification algorithms. Customers get easy access to the proposed framework. It works with a variety of models, including decision trees, random forest classifiers, SVM, Naive Bayes, and logistic regression. Since it produces the most accurate results of the methods used, an ensemble model is created from it. When accuracy is critical, the ensemble model is used, and depending on the machine learning process type, it consistently produces better, or average, outcomes than other methods.

The proposed method analyzes data sent by users or farmers. It predicts the best crops for that type of soil based on the user's selection of some of the most important characteristics, such as potassium, nitrogen levels, and rainfall. In addition to the aforementioned and fertilizer, a prediction can be made based on the projected crop production and another factor, soil erosion. Technology enables farmers to increase their produce and expand their commercial activities. Interesting features include the ability to detect practically any plant disease based on a picture of a leaf, as well as repetitive fertilizer treatments and advice on how to keep the plant healthy in the future. All of these also benefit from a Flask-developed webpage that is less complex for end users and simpler for clients to use when this strategy is put into practice.

Flask can help to construct a farmer-focused blog, allowing farmers to engage and discuss different agriculture topics. This website uses multiple module types and authentications and is totally Flask-designed. Using internet tools to produce crops can help farmers to make better decisions, enhance efficiency, output, cost- effectiveness, accessibility, and seek professional assistance. Farmers may easily use the system via the online application, which is a responsive website. Farmers will also be able to submit feedback on the system's forecasts and recommendations via an integrated feedback mechanism. Farmers could readily access all the data they want if a single web platform includes all of the tools required for crop, fertilizer, and disease identification.

V. RESULTS AND DISCUSSION

When the farmer provided information about soil type, climate, city, and state, the recommendation algorithm will recommend crops while identifying pests and cropping methods. The proposed system's fertilizer suggestions have shown to be effective based on the land's inputs and the chemical components present in the substrate. The disease detection component provides effective treatment options by accurately identifying common crop diseases. Farmers may communicate with one another via the interactive platform by sharing their experiences and learning from one other.

The accuracy of the results produced by the five different types of ML techniques is as follows.

Table 5.1 Accuracy of prediction models.

ALGORITHM	ACCURACY (%)
Naive Bayes	96%
Random Forest (RF)	99%
Decision Tree (DT)	86%
SVM	94%
Logistic Regression	91%

Ensemble model's Accuracy is: 0.9840909090909091				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.76	1.00	0.86	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	0.96	1.00	0.98	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.84	0.96	0.90	28
kidneybeans	0.93	1.00	0.97	14
lentil	1.00	0.91	0.95	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	0.84	0.91	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	1.00	0.94	0.97	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.69	0.81	16
watermelon	1.00	1.00	1.00	15
accuracy			0.97	440
macro avg	0.98	0.97	0.97	440
weighted avg	0.98	0.97	0.97	440

Fig.5.1 Accuracy of ensemble model.

The Random Forest Algorithm obtained the highest accuracy when compared to all other models examined. While using an ensemble model might help us get zero error precision results and is also a superior option when working with sensible data, using an individual model can also offer high accuracy yields.

```
%%time
history = fit_OneCycle(epochs, max_lr, model, train_dl, valid_dl,
                      grad_clip=grad_clip,
                      weight_decay=1e-4,
                      opt_func=opt_func)

Epoch [0], last_lr: 0.00812, train_loss: 0.7466, val_loss: 0.5865, val_acc: 0.8319
Epoch [1], last_lr: 0.00000, train_loss: 0.1248, val_loss: 0.0269, val_acc: 0.9923
CPU times: user 11min 16s, sys: 7min 13s, total: 18min 30s
Wall time: 19min 53s
```

Fig.5.2 Accuracy of disease model.

Thus, 99.2% accuracy was achieved. which, after going through several training and modeling stages, is quite accurate. When certain parameters are adjusted and methods such as gradient clipping, scheduling learning rate, and weight decay are used, ResNets perform well in all image classification tasks. All of the test sets' images can be predicted by the proposed model.

VI. CONCLUSION AND FUTURE WORK

The proposed model has delivered personalized recommendations on crops, fertilizer products, and disease detection. The system's utilization on image recognition and machine learning algorithms improved the accuracy of the recommendations, allowing farmers to enhance their quality of life and improve agricultural production. Furthermore,

the interactive platform enables farmers to connect and interact with one another. The proposed system may be improved further by including more data sources, such as satellite images, which might offer more precise information on crop growth and soil conditions.

REFERENCES

- [1] Ghadge, R., Kulkarni, J., More, P., Nene, S., & Priya, R. L. (2018). Pre-diction of crop yield using machine learning. *Int. Res. J. Eng. Tech-nol. (IRJET)*, 5.
- [2] Guest Editorial - Deep Learning In Biomedical And Healthcare: Emerging Trends, Applications And Research Challenges Chen, M.-Y., Gladence, M. *Journal of Mechanics in Medicine and Biology* 2023, 23(4),2302002
- [3] Saumya Nath, Sneha Sinha, L. Mary Gladence, Y. Bevis Jinila, V. Rajalakshmi, "Health Analysis of Bicycle Rider and Security of Bicycle using IoT in international conference on Communication and Signal Processing (ICCSP-2017) 6th to 8th April 2017 in Adiparasakthi Engineering College. DOI. 10.1109/ICCSP.2017.8286475
- [4] Jinila, Y. Bevis, V. Rajalakshmi, L. Mary Gladence, and V. Maria Anu. "Food Consumption Monitoring and Tracking in Household Using Smart Container." In *Proceedings of the Third International Conference on Computational Intelligence and Informatics*, pp. 693-700. Springer, Singapore, 2020.
- [5] Militante, Sammy V., Bobby D. Gerardo, and Nanette V. Dionisio. "Plant leaf detection and disease recognition using deep learning." 2019 IEEE Eurasia conference on IOT, communication and engineering (ECICE). IEEE, 2019.
- [6] Nischitha, K., Mahendra N. Dhanush Vishwakarma, and Manjuraju MR Ashwini. "Crop prediction using machine learning approaches." *International Journal of Engineering Research & Technology (IJERT)* 9.08 (2020).
- [7] Niketa Gandhi, Leisa J Armstrong, RiPtka, Owaiz Petkar, Amiya Kumar, "Rice crop yield prediction in India using support vector machines", *Inter-national Joint Conference on Computer Science and Software Engineering*, 2016, IEEEExplore.
- [8] P. A, S. Chakraborty, A. Kumar and O. R. Pooniwalla, "Intelligent Crop Recommendation System using Machine Learning," 2021 5th International Conference on Computing Methodologies and Communication (IC-CMC), 2021, pp.843-848, doi: 10.1109/ICCMC51019.2021.9418375.
- [9] Palanivel, Kodimalar, and Chellammal Surianarayanan. "An approach for prediction of crop yield using machine learning and big data techniques." *International Journal of Computer Engineering and Technology* 10.3 (2019): 110-118.
- [10] Patil, Pavan, Virendra Panpatil, and Shrikant Kokate. "Crop prediction system using machine learning algorithms." *Int. Res. J. Eng. Tech-nol. (IRJET)* 7.02 (2020).
- [11] S. P. Raja, G. Mariammal, A. Suruliandi and E. Poongothai, "Prediction of Land Suitability for Crop Cultivation Based on Soil and Environmental Characteristics Using Modified Recursive Feature Elimination Technique With Various Classifiers," in *IEEE Transactions on Computational Social Systems*, vol. 8, no. 5, pp. 1132-1142, Oct, 2021, doi: 10.1109/TCSS.2021.3074534.
- [12] S. Vaishnavi., M. Shobana., R. Sabitha. and S. Karthik., "Agricultural Crop Recommendations based on Productivity and Season," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp.883-886, doi:10.1109/ICACCS51430.2021.9441736.
- [13] Subramanian, Kanaga Suba & Rishi, R. & Sundaresan, E. & Valliappan, Srijit. (2017). Demand based crop recommender system for farmers. 194-199. 10.1109/TIAR.2017.8273714.
- [14] Vaishnavi, P, M. Kalimuthu and M. Kishore, "Crop Prediction using Machine Learning," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 926-932, doi: 10.1109/ICSSIT48917.2020.9214
- [15] Van Klompenburg, T., Kassahun, A. and Catal, C., 2020. Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*, 177, p.105709