

INDEX

1. Introduction

- a. **The idea to make the project**
- b. **Introduction of the project**

2. Approach

- a. **Approach to touch end of line**

3. Analysis

- a. **Block Diagram**
- b. **Hardware/Software Requirements**

4. Flowchart

- a. **Control of flow**

5. Model Selection

- a. **Prescriptive process Model**
- b. **MI Model**

6. Result

- a. **pages-view**
- b. **Agrifyzen-view**

7. Advantages and Disadvantages

- a. **Advantages/Disadvantages of Model**
- b. **Advantages of Agrifyzen WebApp**

8. Applications

- a. **Applications of Agrifyzen WebApp**

9. Conclusion

10.Future Scope

11.Bibliography

- a. **Reference**
- b. **Appendix**

1. INTRODUCTION

a. The Idea to make the project

Early detection and management of problems associated with crop, fertilizer, and yield indicators can help increase yield and economic growth. In farmer's life, they face many problems due to the uneven climate changes. They cannot decide which crop to grow on their land and after growing crops they cannot decide which fertilizer is best for their crop. After using proper fertilizer, they may be confused about how much crop they will yield. For this problem we found a solution which improves prediction and recommendation system.

b. Introduction of project

The project is all about a farming website which has two recommenders, a predictor and a AI Chatbot. The two recommenders are Crop recommender and a Fertilizer recommender. Crop recommender analyzes different environmental factors of the land and recommends the best crop that can be grown on this land. Fertilizer recommender recommends the best fertilizer according to the given features of a crop. Then we have a predictor which predicts yield in kilograms per given area. AI chatbot is useful for users to interact with our website. The model of recommenders and predictor are created using Jupyter notebook. For this we have used concepts of Machine learning, data preprocessing, data visualization, etc. For frontend of the website we have used HTML, CSS, JS and Bootstrap whereas for backend, we have used Flask and JS.

2. APPROACH

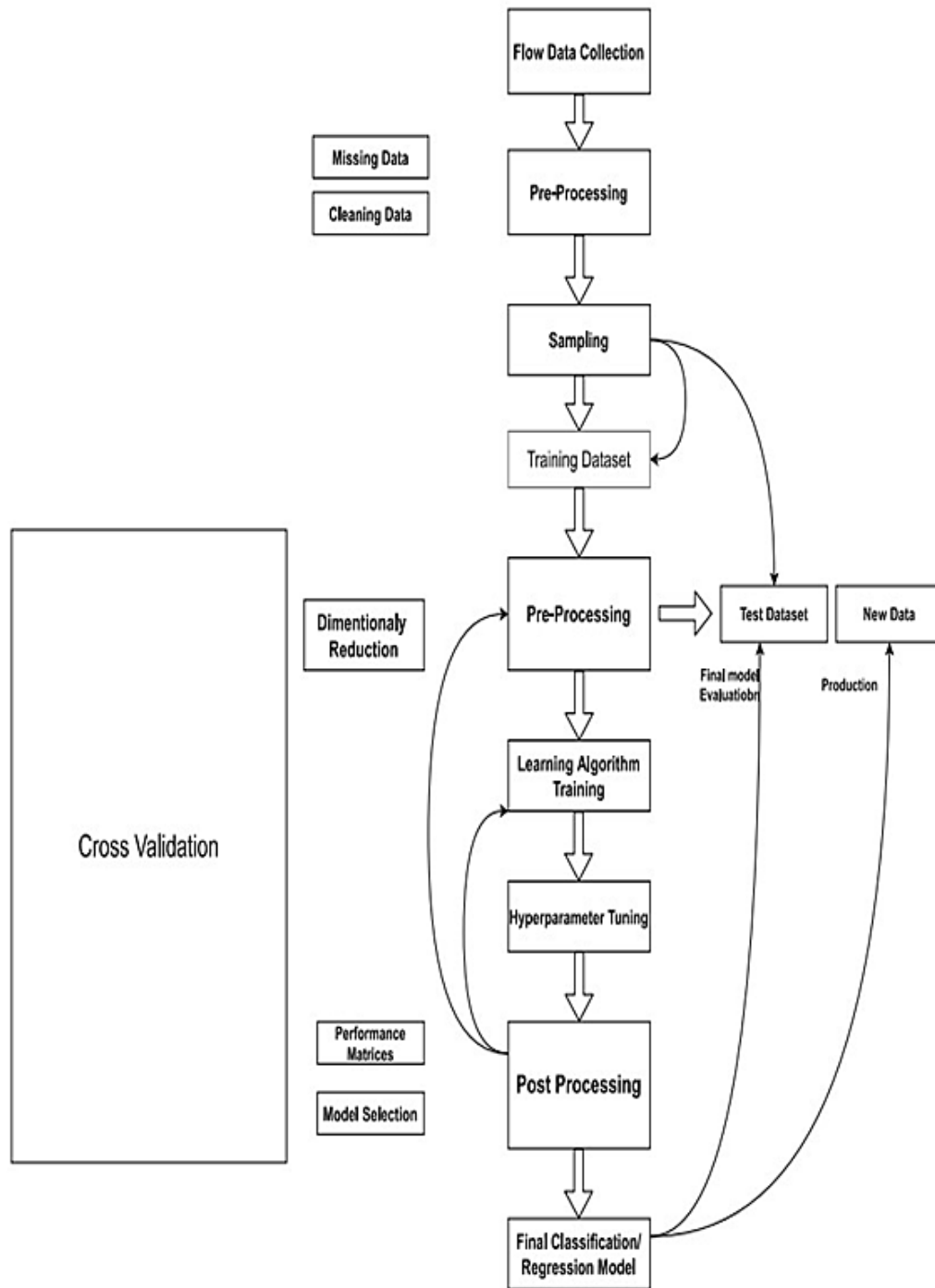
a. Approach to touch end of line

For making the predictor and recommenders, first we searched for the proper dataset according to requirements. And then we cleaned our data according to our requirements. Then we preprocessed our data. And then we applied the different type of regressor and classifiers on the training data. And then we choose the model which was giving highest accuracy (Decision Tree and Random forest). After making model using Jupyter Notebook , we designed a website which has frontend made of HTML and CSS. HTML and CSS are used to design a form which takes features from user. This form is connected to Flask Server (Backend) which is then connected to our predictor and recommender. In this way, website will give predictions and recommendation. We also integrated our chatbot on the website. This chatbot is designed using Watson AI tech.

3. **ANALYSIS**

a. **Block Diagram**

- 1) Dataset collection
- 2) Data visualization
- 3) Data pre-processing in the form of data cleaning and feature extraction
- 4) Data splitting into train and test sets
- 5) Fitting the algorithm a Parameter tuning
- 6) Testing the accuracy of the model
- 7) Data post-processing in the form of performance metrics



b. **Hardware/software requirements**

Hardware Requirements

- **Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional:**
 - Processor: 800MHz Intel Pentium III or equivalent,i3
 - Memory: 512 MB
 - **DISK SPACE: 750 MB OF FREE DISK SPACE**
- **UBUNTU 9.10:**
 - **PROCESSOR: 800MHZ INTEL PENTIUM III OR EQUIVALENT**
 - **MEMORY: 512 MB**
 - **DISK SPACE: 650 MB OF FREE DISK SPACE**

SOFTWARE REQUIREMENTS

OPERATING SYSTEM: Windows 7/ XP/8 and above.

FRONT END: Html, CSS, java script,bootstrap.

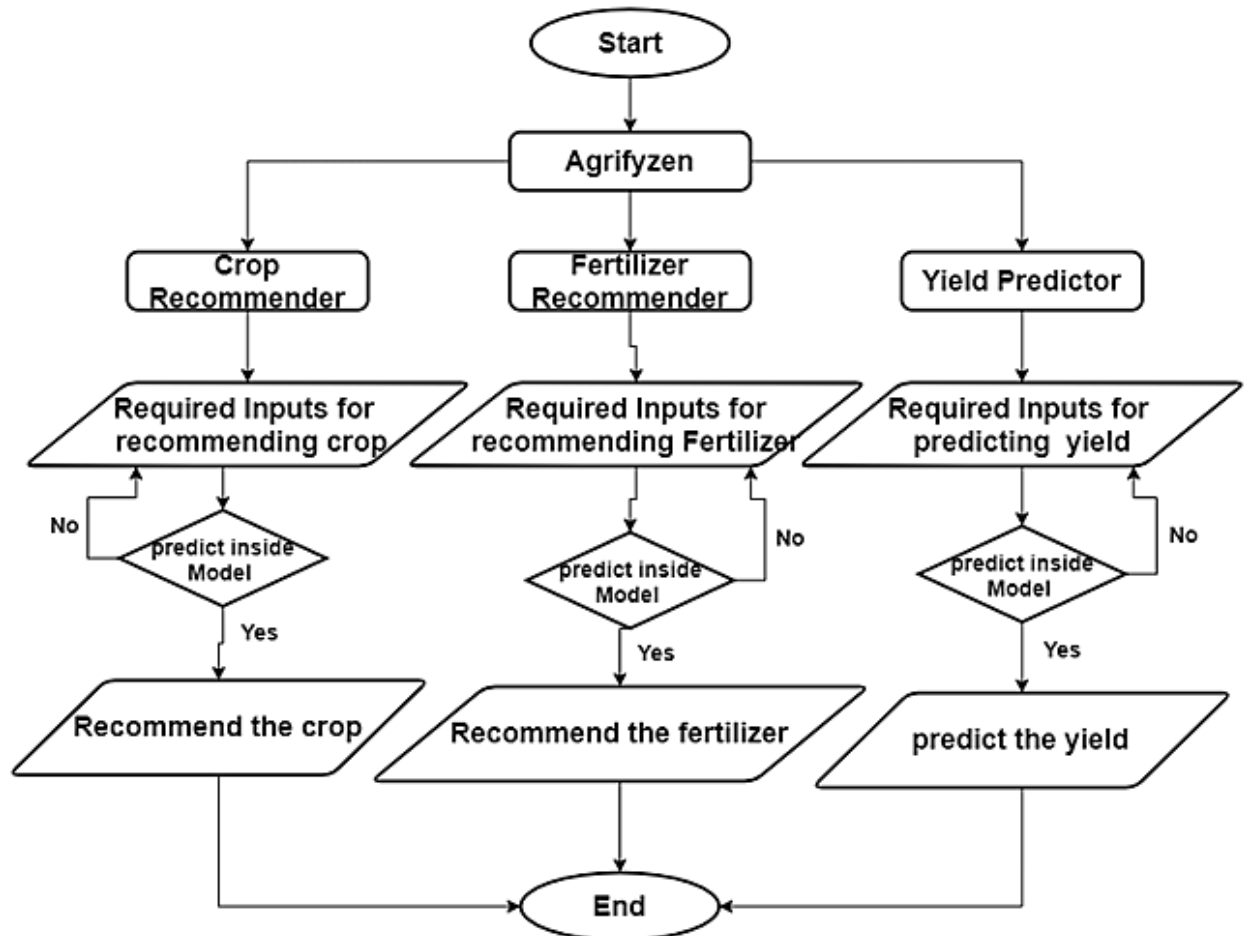
BACKEND: Flask

PREDICTOR AND RECOMMENDAR: Python, Matplotkib, Numpy, Pandas, Seaborne, Anaconda-jupyter notebook or colab

TOOLS AND TECHNOLOGY: Vs code, Spyder, jupyter notebook, heroku, Github

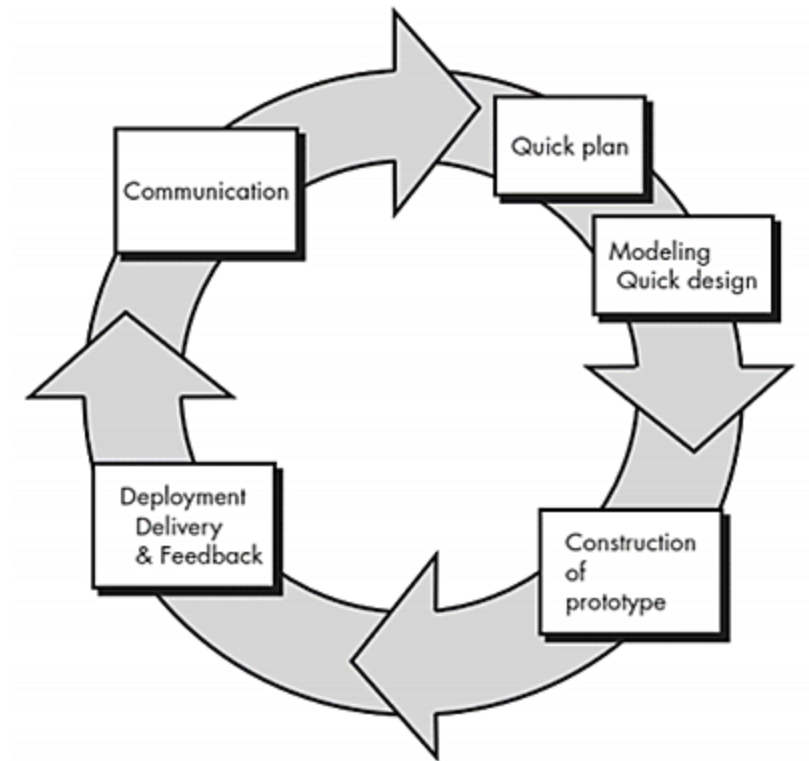
4. FLOWCHART

a. **Control of Flow**



5. MODEL SELECTION

a. **Prescriptive process model**



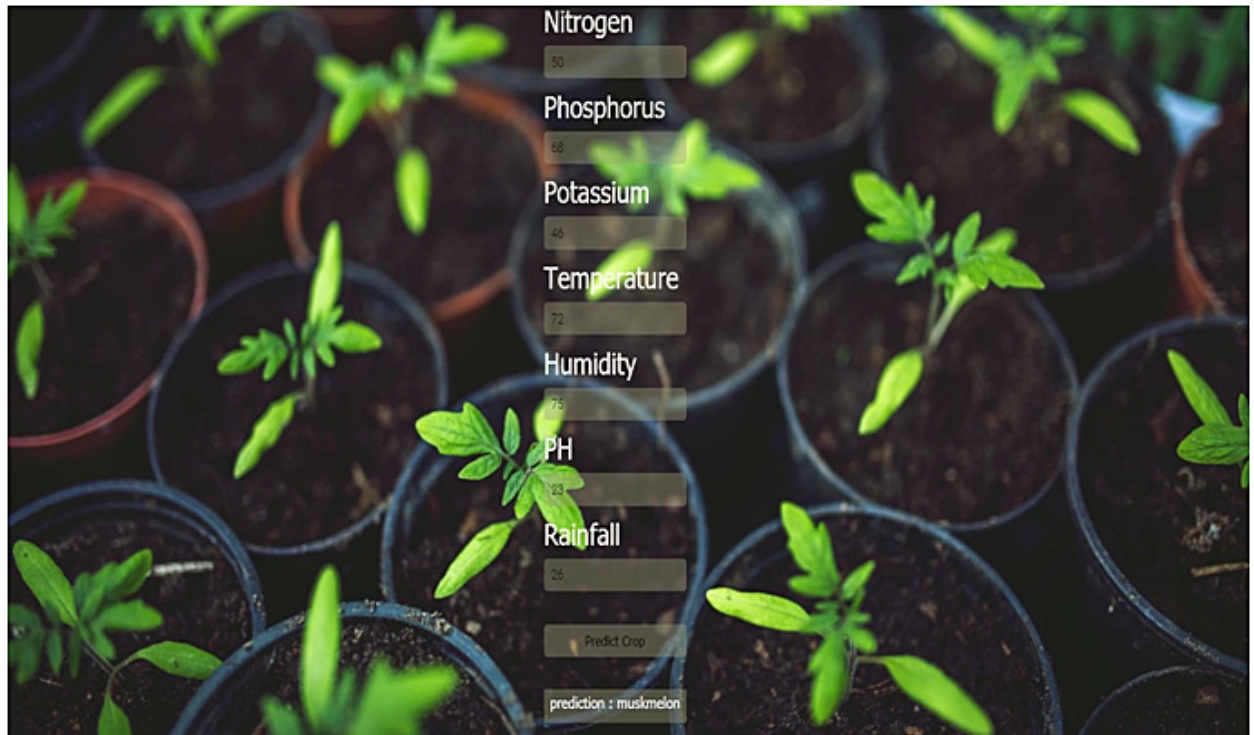
b. **ML model**

- Random Forest is used in Crop recommendation system as well as fertilizer recommendation system
- Decision tree is used in yield prediction system.

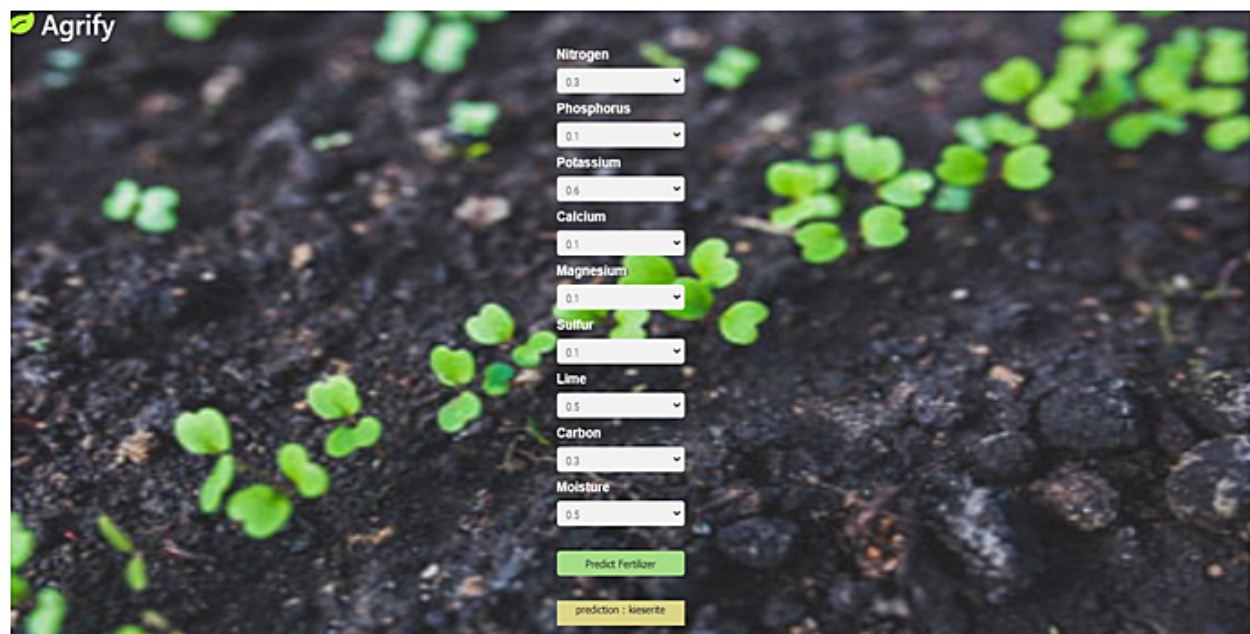
6. **RESULT**

a. **Pages view**

i. **Crop page**



i. **Fertilizer page**



The image shows the 'Fertilizer Prediction' interface of the Agrify application. The background is a close-up photograph of dark, rich soil with several small, bright green seedlings sprouting from it. On the right side, there is a vertical stack of input fields, each with a dropdown arrow. Below these fields is a green button labeled 'Predict Fertilizer'. At the bottom, a yellow box displays the prediction result.

Agrify

Nitrogen
0.3

Phosphorus
0.1

Potassium
0.6

Calcium
0.1

Magnesium
0.1

Sulfur
0.1

Lime
0.5

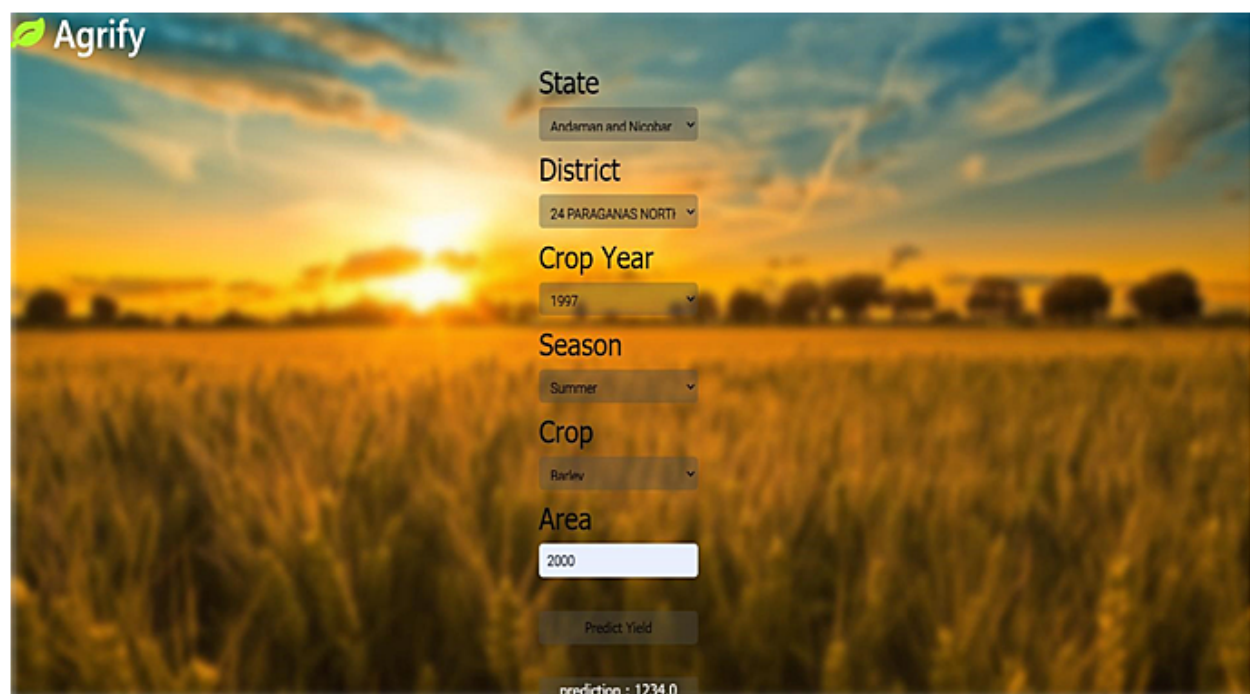
Carbon
0.3

Moisture
0.5

Predict Fertilizer

prediction : kieserite

ii. Yield page



The image shows the 'Yield Prediction' interface of the Agrify application. The background is a photograph of a vast, golden-brown field of crops under a bright, hazy sky with a low sun. On the right side, there is a vertical stack of input fields, each with a dropdown arrow. Below these fields is a grey button labeled 'Predict Yield'. At the bottom, a yellow box displays the prediction result.

Agrify

State
Andaman and Nicobar

District
24 PARAGANAS NORTH

Crop Year
1997

Season
Summer

Crop
Barley

Area
2000

Predict Yield

prediction : 1234.0

b. **Agrifyzen-view**

- Click this link to view full WebApp.

<https://agrifyzen.herokuapp.com/>

7. **Advantages and Disadvantages**

a. **Advantages and Disadvantages of ML model**

- **Advantages of decision tree algorithm**

1. Easy
2. Transparent process
3. Handle both numerical and categorical data
4. Larger the data, the better the result
5. Speed

- **Advantages of random forest algorithm**

1. Power and highly accurate
2. No need to normalizing
3. Can handle several features at once

4. Run tree in parallel ways

- **Disadvantages of decision tree algorithm**

1. Many over-fit

2. Pruning Process Large

3. Optimization unguaranteed

4. Complex calculations

5. Deflection high

- **Disadvantages of random forest algorithm**

1. They are biased to certain features sometimes

2. Slow

3. Cannot be used for linear methods

4. Worse for high dimensional data

b. **Advantages of Agrifyzen WebApp**

- **Advantages of Agrifyzen**

1. Recommendation about the type of crops to be cultivated which is best suited for the respective conditions.

2. Recommendation about the type of fertilizer best suited for the particular soil and the recommended crop.

3. Yield Recommendation based on the area and recommended crop and season.

8. **APPLICATIONS**

a. **Applications of Agrifyzen WebApp**

- No agricultural plots be wasted by growing less efficient crop by adopt this model.
- This can feed the interest of both researchers and entrepreneurs.
- This will let farmers in one country to know the prospect of farming in another part of world.
- Implement ML for crop prediction and viability on this, also this we desire to take this model into the mobile phone platform.
- Machine learning can help farmers by recommending the different crops based on weather conditions.
- Machine learning makes these activities smarter over time.

9. **CONCLUSION**

We finally connected all the domains (frontend, backend, recommender, predictor and chatbot) and created a website for Farmers where one can find predictor and recommenders at one place. Crop Recommender takes features

like Nitrogen, Phosphors, Potassium, Temperature, Humidity, Ph and rainfall and recommends suitable Crop. Fertilizer Recommender takes features like Nitrogen, Phosphors, Potassium, Calcium, Magnesium, Sulfur, Lime, Carbon and Moisture and recommends best Fertilizer. Yield predictor takes features like State, District, Crop, Season, Crop year and area and predicts the yield. Chatbot is useful for users to interact with our website.

10. **FUTURE SCOPE**

- We can update the data by adding new data to increase the accuracy of our model.
- We can add a feature of login/sign in to our website so that only authenticated users can enter our website.

11. **BIBLIOGRAPHY**

a. **Reference**

- HTML/CSS/JS- <https://developer.mozilla.org/en-US/docs/Web/HTML>
- CSS- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Java Script-

<https://developer.mozilla.org/en-US/docs/Web/JAVASCRIPT>

- Flask Server- <https://flask.palletsprojects.com/en/2.0.x/>
- Bootstrap- <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
- Python- <https://docs.python.org/3/tutorial/>
- ML model- <https://scikit-learn.org/stable/>
- Data cleaning- <https://pandas.pydata.org/docs/> , <https://numpy.org/doc/>
- Data visualization- <https://matplotlib.org/stable/tutorials/index.html> , <https://seaborn.pydata.org/tutorial.html>
- Dataset- <https://www.kaggle.com/atharvaingle/crop-recommendation-dataset> , <https://www.kaggle.com/ritik619/ffdfdfd?select=FertPredictDataset.csv> , <https://www.kaggle.com/c/fyp/data>
- <https://blog.gramener.com/crop-yield-prediction/>
- Boot camps- <https://youtu.be/qcf6a9Qzkhl> , <https://youtu.be/1VXO8p8yX9Y> , <https://youtu.be/WRxa1JczBeQ> , https://youtu.be/nlK0t_9V8kE

b. **Appendix**

Git Hub link: - <https://github.com/Keyurchaniyara/Agrifyzen>

Representation: - <https://youtu.be/Lfd6nBk9izg>.