Technical Answer to Real World Problem

CSE3999

Grain Market Trade benefit to farmer via forecast estimation

A Project Report

Submitted by

Aniket Kumar (18BCE0101)

Sapan Kumar Shrivastava (18BCE0779)

Yogek Garg (18BCE0810)

Anand Kumar (18BCE2225)

in partial fulfilment for the award of the degree of

B.Tech.

In

Computer Science & Engineering

Under the Guidance of

Prof. Sendhil Kumar K S

School of Computer Sciences

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

May, 2021

DECLARATION

We hereby declare that the report entitled "**Grain Market Trade Benefit to Farmers via Forecast Estimation**" submitted by us, for J component of CSE3999 Technical Answers to Real World Problem to VIT is a record of bonafide work carried out by us under the guidance & supervision of Prof. Sendhil Kumar K S.

We further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for any other courses in this institute or any other institute or university.

Signature of the Candidates

Aniket Kumar

Place: **Vellore** **Sapan Kumar Shrivastva**

Date: **8th May, 2021** **Yogek Garg**

Anand Kumar

Certificate

This is to certify that the thesis entitled "**Grain Market Trade Benefit To farmer via Forecast Estimation**" submitted by Aniket Kumar (18BCE0101), Sapan Kumar Shrivastava (18BCE0779), Yogek Garg (18BCE0810), Anand Kumar (18BCE2225) VIT, for the award of the degree of Bachelor of Technology in Computer Science & Engineering, is a record of bonafide work carried out by them under my supervision during the period, 02. 02. 2021 to 31.05.2021, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Signature of the Candidates

Aniket Kumar

Place: **Vellore** **Sapan Kumar Shrivastva**

Date: **8th May, 2021** **Yogek Garg**

Anand Kumar

Acknowledgement

We thank the Almighty God for his blessings for granting us the opportunity to gain the knowledge and right aptitude to successfully complete our project work.

First of all, we express our gratitude & thanks to our guide and supervisor **Prof. Sendhil Kumar K S** for all is guidance and instruction to complete this project and this course.

We wish to express our sincere thanks to **Honorable Chancellor, Dr. G. Viswanathan**; esteemed Vice-Presidents, Shri. Sankar Viswanathan; Shri. Sekar Viswanathan and Shri. G. V. Selvam; respected Vice Chancellor, Dr. Anand A. Samuel; respected Pro-Vice Chancellor Dr. S. Narayanan of this prestigious VIT University, Vellore for providing us an excellent world class academic environment and facilities for pursuing our B.Tech. Programme.

My sincere gratitude lies **Dr. S Vairamuthu**, Head of the Department of Software Systems for providing us an opportunity to do our project work in the VIT University.

My special thanks to my friends for their timely help and suggestions rendered for the successful completion of this internship course.

This acknowledgement would be incomplete without expressing our whole hearted thanks to my parents for their continuous support and guidance in all walks of our life.

Signature of the Candidates

Aniket Kumar

Place: **Vellore** **Sapan Kumar Shrivastva**

Date: **8th May, 2021** **Yogek Garg**

Anand Kumar

EXECUTIVE SUMMARY

The impact of climate changes, the crops that are used to be grown by the farmers in a procedural way over the whole year has changed, now if the procedure are followed the crops yields are not up to the mark, which further puts the farmers in loss. Major problem which is currently Perpetuating in Indian Market is the value Prediction for varieties of crops in market. Since with the impact of climate changes, the crops that are used to be grown by the farmers in a procedural way over the whole year has changed, now if the procedure are followed the crops yields are not up to the mark, which further puts the farmers in loss. In order to tackle with such problems, we will build a prediction web app, which will suggest what to Grow and when to Grow. The farmer shall be provided with some tools comprising of Ph, moisture etc. sensors which will be connected to the Mobile App and the data shall be fetched from cloud database, next we shall be performing the data cleaning followed by machine learning techniques, will focus on the optimizing the Decision Tree Regressor using Scikit tools, Scikit-Optimize library is an open-source Python library that provides an implementation of Bayesian Optimization that can be used to tune the hyper parameters of machine learning models from the scikit-Learn Python library. We will be using 80% data for training and remaining 20% data for testing. Other technologies such as chart.js, NumPy, JavaScript, Bootstrap etc. shall be used according to the requirement. Finally, the result shall be presented to the farmer in the GUI Interface.

Table of Contents

LIST OF FIGURES

1. INTRODUCTION

1.1 Theoretical Background

The effect of environment changes, the harvests that are utilized to be developed by the ranchers in a procedural manner over the entire year has changed, presently if the system are followed the yields are not sufficient, which further places the ranchers in misfortune. To handle with such issues, we will fabricate an expectation web application, which will propose what to Grow and when to Grow. A significant issue which is right now Perpetuating in Indian Market is the worth Prediction for assortments of harvests in market. Since with the effect of environment changes, the harvests that are utilized to be developed by the ranchers in a procedural manner over the entire year has changed, presently if the system are followed the yields are not sufficient, which further places the ranchers in misfortune. To handle with such issues, we will fabricate an expectation web application, which will propose what to Grow and when to Grow. In the present period of Science and Technology, it's significant for every one of the various fields to have an advancement with the continuous advances, in later occasions, analysts learning at different schools and colleges commit their lives to imaginative work of developing things and practices. Ranchers have benefitted and added to the reliably creating investigation of cultivating. In this undertaking we target assisting the rancher with having an exchange advantage at Grain Market by means of conjecture data. The majority of the occasions the rancher is dealing with issues of picking the sort of harvest to be filled to have most elevated benefit, however the issue can't just be restricted to this as consider a case in some cases regardless of whether the yield developed has high market interest, yet the quality isn't acceptable which further prompts the rancher in the red, so a mix of the two things i.e., picking the best harvest as per the current soil pH, temperature, dampness and precipitation help in picking the best harvest for development. Further the rancher will likewise approach the anticipated cost for the yield in coming months, so the rancher can utilize the data consequently gave choosing the best harvest to development. We will give the finished result a Mobile Web App since portable application is not difficult to work in any event, for an individual having less specialized information on working the Smartphone. The rancher will be given a few devices involving Ph, dampness and so on sensors which will be associated with the Mobile App and the information will be gotten from cloud data set, next we will play out the information cleaning followed by AI strategies, will zero in on the upgrading the Decision Tree Regressor utilizing Scikit apparatuses, Scikit-Optimize library is an open-source Python library that gives an execution of Bayesian Optimization that can be utilized to tune the hyperparameters of AI models from the scikit-Learn Python library. We will utilize 80% information for preparing and staying 20% information for testing. Different advancements, for example, chart.js, NumPy,

JavaScript, Bootstrap and so on will be utilized by the prerequisite. At long last, the outcome will be introduced to the rancher in the GUI Interface.

1.2 Motivation

In today's era of Science and Technology, its important for all the different fields to have a progress with the ongoing technologies, in more recent times, researchers learning at various colleges and universities dedicate their lives to innovative work of cultivating items and practices. Farmers have profited and added to the consistently developing study of farming. In this project we aim at helping the farmer have a trade benefit at Grain Market via forecast information. Most of the times the farmer is facing problems of choosing the type of crop to be grown in order to have highest profit, but the problem can't only be limited to this as consider a case sometimes even if the crop grown has high market demand, but the quality is not good which further leads to the farmer in debt, so a combination of the two things i.e., choosing the best crop according to the current soil pH, temperature, moisture and rainfall help in choosing the best crop for cultivation. Further the farmer will also have access to the predicted price for the crop in coming months, so the farmer can use the information thus provided deciding the best crop for cultivation. We will be presenting the end product with a Mobile Web App since mobile app is easy to operate even for a person having less technical knowledge on operating the Smartphone. The farmer shall be provided with some tools comprising of Ph, moisture etc. sensors which will be connected to the Mobile App and the data shall be fetched from cloud database, next we shall be performing the data cleaning followed by machine learning techniques, will focus on the optimizing the Decision Tree Regressor using Scikit tools, Scikit-Optimize library is an open-source Python library that provides an implementation of Bayesian Optimization that can be used to tune the hyperparameters of machine learning models from the scikit-Learn Python library. We will be using 80% data for training and remaining 20% data for testing. Other technologies such as chart.js, NumPy, JavaScript, Bootstrap etc. shall be used according to the requirement. Finally, the result shall be presented to the farmer in the GUI Interface.

The environment aspect of the project is also put into consideration while developing the project. Since the crop shall be predicted beforehand which suits the best for the current

Environmental conditions, the farmer not has to un-necessarily spend money on buying the fertilizers and putting on the soil, which usually degrades the quality of the soil in long run. Also, in case previously the farmer grew the crop according to his mind, which lead to sometimes less quality product and not being able to be sold in the market, which further required packaging material as well sometimes in case if crops left for long time in storage areas also were degraded by the environmental forces acting. So, in all this will also help in development to the environmental aspect as well.

This project mainly aims to prevent the economic loss of its user, as we know that due to the continuous changes in the climate of our country, the crops that are grown by the farmers in the procedural way over

the whole year has faced a drastic change. This suggest us that if our farmers continue to follow the fashion, the crop yields will not be up to that mark. Therefore Our product of forecasting the market cost of grains in advance will let the farmers to decide the things more to marketing and the type of crop to produce and store beforehand.

The product will try to collect the data of moisture, temperature and other things of the field through the sensors and Arduino circuit and will create dataset along with the previous price datasets to calculate which crop to grow for the farmer in the most minimal possible cost.

1.3 Aim of Proposed Work

As we can Observe that , a major problem which is currently Perpetuating in Indian Market is the value Prediction for varieties of crops in market. Since with the impact of climate changes, the crops that are used to be grown by the farmers in a procedural way over the whole year has changed, now if the procedure are followed the crops yields are not up to the mark, which further puts the farmers in loss. We are aiming to tackle with such problems, by building a prediction web app, which will suggest what to Grow and when to Grow.

1.4 Objective Of Proposed Work

To Build a Best Crop to grow Predictive web app in order to forecast , which crop to grow for the Ultimate Benefit of Farmer. We are having objective to Achieve 93 to 94 precent success rate in training our model. Apart from that we are also trying to build a user friendly interface to make it access to each and every person who uses it.

2. Literature Survey

2.1 Survey of existing work/ Models

[2.1.1]

In the paper entitled "**Environment Monitoring System for Agricultural Application using IoT and Predicting Crop Yield using Various Data Mining Techniques**" Authors (***Gautam Gupta, Rashi Setia, Archana Meena, Bhawnesh Jaint***) have tried to demonstrate their work which they have carried out by using IOT devices, & Data Mining techniques to predict the crop yield. Mainly they are acquiring data through their sensing devices and processing and delivering it to cloud with the help of IOT devices e.g., microcontrollers and Arduinos. After consolidating these data, they are feeding to a user generated site and getting the estimation. While Our Project will be auto fetching the processed data and gathering some real-time data like, weather, humidity, rain prediction etc. which are available at meteorological department and then processing these data through ML

algorithm and then produce a prediction in both yield and local market rate. Their solution is constrained to a limited location as well as limited geographical areas while our solution is not constraint to a particular location or a geographical area. Moreover, it would be a Global Solution.

[2.1.2]

In the paper entitled "**Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data**" Authors (***Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, Stefano Ermon***) have tried to give a solution by using ML, Deep Learning & Image Processing, and Neural Networks. They are trying to get images as input data to train as well as analyze their model from different federal agencies who are working in the field of agriculture, surveys, and meteorology. Well I find this solution more of theoretical nature rather than practical results. As a model just cannot analyze pixels of the images which are taken from satellite and estimate the yield. While taking this utopian approach we have chosen a very practical and feasible approach of analyzing Realtime data and then conclude a result

[2.1.3]

In the paper entitled "**Crop yield estimation model for Iowa using remote sensing and surface parameters**" Authors (***Anup K. Prasad, Lim Chai, Ramesh P. Singh, Menas Kafatos***) have performed a location centric study to develop a index using other indices named Normalized difference vegetation index (NDVI) & Vegetation condition index (VCI). They are acquiring data from 8 kmX8 km NDVI datasets from very advance technologies like Advanced very High- Resolution radiometer (AVHRR) National Oceanic and Atmospheric Administration (NOAA). Well these are very advanced technologies being used here and results are not very precise everything is based on statistics and probabilistic approach. While we are processing the results with much precision and our results are most concluding and easier to be interpreted by humans.

[2.14]

A review paper entitled as "**Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review**" by Authors (***Anna Chlingaryana, Salah Sukkarieha, Brett Whelan***) is presented to notice the issue and find a solution of it by using various machine learning algorithms like, K-NN, SNN, CNN & c-means and k-means algorithms by taking Nitrogen and its management as key element. They are using management zones and clustering data explicitly for unsupervised learnings. Also, they are sensing level of nitrogen in soil as well as in

plants and their roots and accepting this data as in-situ remote sensed. Well this is a novel approach we have found in this paper and we will include it in our future scopes but focusing on only one components of nutrition or micronutrition is not a good idea rather analyzing all the components of nutrition would be a good Idea. We shall incorporate this point in our project for more precision.

[2.1.5]

[K.gosh,R.Balasubramaniam,2014] Under this paper crop yield forecast are prepared at District ,state(West Bengal) and National level under ongoing Project "**Forecasting Agricultural output using Space, Agro meteorology and Land based observations (FASAL)**",operational at Ministry of Agriculture, Govt. of INDIA. Models Suggested by Fisher (1924) and Hendrick and Scholl (1943), which has further been modified at Indian Agricultural Statistics Research Institute (IASRI), where partial crop season data considering different weather variables simultaneously to develop a forecast model. The result showed that 92% of yield of kharif Rice have been mostly contributed by Midnapur(west) and the performance of the model in predicting yields at district level for various different crops across the country is quite satisfactory. This Project differs from us as we are aiming to get the datasets of a particular land across any part of the country whereas this has covered a scenario and climatic situations of West Bengal Only.

[2.1.6]

[Ananta vashisth,R.singh,2014] Under this Paper forecast of crop yield is done on the variability of weather. Model based on the weather parameters can be used for reliable forecast in advance for crop yield. This research paper has used stepwise regression analysis. In this Weather variables are used as independent variables which are related to crop responses such as yield and to account for the technological changes, function of time is used as independent variables. Results from the simulated studies showed that the statistical models based upon the weather indices can successfully simulate pre harvest yield forecast of wheat under semi-arid region. The percentage deviation between observed and simulated yield was ranged from 5 to 11 and the correlation coefficient was 0.93 to 0.99. Proposed advantages of this model are that it is simple, no sophisticated tools required and can be used for district agro climatic zone and state level forecast. This project mainly aims to forecast through weather conditions of a particular region where we will consider factors like earlier costing datasets, moisture , temperature etc through our IOT devices.

[2.1.7]

[Ranjana Agarwal and S.C. Mehta,2004] In this paper various approaches for forecasting crop yield and forewarning issued which enable the farmers to optimize plant protection measures. Various models such as Yield forecast models, models with composite weather variable,

discriminant function analysis, water balance technique, ordinal logistic model, deviation method and artificial neural network technique. This paper also focuses on forewarning of pests and diseases using qualitative data model. This model can be used even in the lack of availability of detailed and exact datasets. Apart from all of the models proposed
, the ANN model is considered best for the forewarning of crop yield. To sum up, different types of models were developed at IASRI for forecasting crop yield and forewarning pests and diseases. The performance found to be good. The methodologies were used successfully by various other workers and organizations. This paper has forecast the outcome through a variety of models , where as we are implementing a subset models and algorithm of it to predict the outcome.

[2.1.8]

Randall J. Donohuea, Roger A. Lawes, Gonzalo Mata, David Gobbett and Jackie Ouzmanc [1] (2018) recently contributed to remote sensing domain for predicting crop yield of different crops in the Australian crop farming areas. They devised a new technique called C-Crop for crop yield production, making use of crop type, foliage length cover, air temperature etc, some of the inputs are fed via remote sensing like air temperature and other locally such as foliage length cover etc, the data which they used for processing is yield data as that of canola and wheat of respectively 31 field years and 160 field years, which is used for further prediction of the crop yield i.e., for forecasting. The improvements in relation to our study being we aim to focus on a no. of important factors i.e., soil pH, humidity, rainfall etc important factors, also no analysis has been given to crop prices.

[2.1.9]

Bruno Basso, Lin Liu [2] (2019) focus on crop yield forecast comprising of different methods and accuracies. The author discusses about various statical methods, agrometeorological techniques for predicting the crop yield, the author also discusses that the predictions usually have some error, which the author suggest to have a combination of statical and agrometeorological both to have a better prediction. Accuracy is totally defined on the quality of the data received. In relation to the conclusion, the data quality as received from our model, from different IOT devices in Realtime would lead to high accuracy calculations

[2.1.10]

Rohit Kumar Rajak, Ankit Pawar, Mitalee Pendke, Pooja Shinde, Suresh Rathod and Avinash

Devare [3] (2017) discusses about different machine learning techniques to have a maximum output for the crop yields and help the farmers. The data used I this is collected from Polyset Laboratories testing lab, collected from different areas. Then this data is used for further analysis using different machine leaning techniques such as Support Vector Machine, Naïve Bayes, Random Forest etc. The limitations to the paper being that it mostly focuses on soil, other parameters like temperature, humidity etc are not put into consideration, which form important parameters while calculation, the same are developed as improvements to our research.

[2.1.11]

Mohit Kumar Saini, Rakesh Kumar Saini, (Nov. 2020) This paper has helped fill the gap between quality, production, and quantity. Data entered by collecting and importing data from multiple real-time applications or cloud storage in the database to ensure fast action. With seamless end-to-end operation and high-end business process, the manufacturer speeds up the process and reaches the supermarkets in a timely manner and makes the proposed system fully operational. In this paper, firstly collect data on the cloud and calculated automatically. The sensors that is used for this purpose like soil moisture, air pressure, rain detection and humidity sensors etc. Wireless data communication: The data is collected from sensors node and forwarded to the server through wireless transmission. Sensor data attainment: The sensors are boundary with Adriano Uno board such as temperature sensors, Rain detection sensor, humidity and Soil moisture etc. Data analysis & Decision making: The data analysis is the process of analysing the data collected from different sensors from the agriculture field. Automated irrigation system: In the automated irrigation system once, the control established from the mobile application or web application. Mobile Application: The mobile phone application is providing help for monitoring and controlling the agriculture filed from any place. Web application: - The web application will be designed to observe the ground and crops from everywhere using an internet connection. This paper focus on to collect data from various sensors and analyse it to increase crop production where's but in our project, we are trying to utilise iot devices in more effective ways in respective to cost as well as crop production.

3. Overview Of Proposed System

3.1 Introduction and Related Concepts

The impact of climate changes, the crops that are used to be grown by the farmers in a procedural way over the whole year has changed, now if the procedure are followed the crops yields are not up to the mark, which further puts the farmers in loss. In order to tackle with such problems, we will build a prediction web app, which will suggest what to Grow and when to Grow. A major problem which is currently Perpetuating in Indian Market is the value Prediction for varieties of crops in market. Since with the impact of climate

changes, the crops that are used to be grown by the farmers in a procedural way over the whole year has changed, now if the procedure are followed the crops yields are not up to the mark, which further puts the farmers in loss. In order to tackle with such problems, we will build a prediction web app, which will suggest what to Grow and when to Grow. In today's era of Science and Technology, it's important for all the different fields to have a progress with the ongoing technologies, in more recent times, researchers learning at various colleges and universities dedicate their lives to innovative work of cultivating items and practices. Farmers have profited and added to the consistently developing study of farming. In this project we aim at helping the farmer have a trade benefit at Grain Market via forecast information. Most of the times the farmer is facing problems of choosing the type of crop to be grown in order to have highest profit, but the problem can't only be limited to this as consider a case sometimes even if the crop grown has high market demand, but the quality is not good which further leads to the farmer in debt, so a combination of the two things i.e., choosing the best crop according to the current soil pH, temperature, moisture and rainfall help in choosing the best crop for cultivation. Further the farmer will also have access to the predicted price for the crop in coming months, so the farmer can use the information thus provided deciding the best crop for cultivation. We will be presenting the end product with a Mobile Web App since mobile app is easy to operate even for a person having less technical knowledge on operating the Smartphone. The farmer shall be provided with some tools comprising of Ph, moisture etc. sensors which will be connected to the Mobile App and the data shall be fetched from cloud database, next we shall be performing the data cleaning followed by machine learning techniques, will focus on the optimizing the Decision Tree Regressor using Scikit tools, Scikit-Optimize library is an open-source Python library that provides an implementation of Bayesian Optimization that can be used to tune the hyper parameters of machine learning models from the sickie-Learn Python library. We will be using 80% data for training and remaining 20% data for testing. Other technologies such as chart.js, NumPy, JavaScript, Bootstrap etc. shall be used according to the requirement. Finally, the result shall be presented to the farmer in the GUI Interface.

3.2 Framework, Architecture or Module for the Proposed System (with explanation)

ARCHITECTURE DIAGRAM

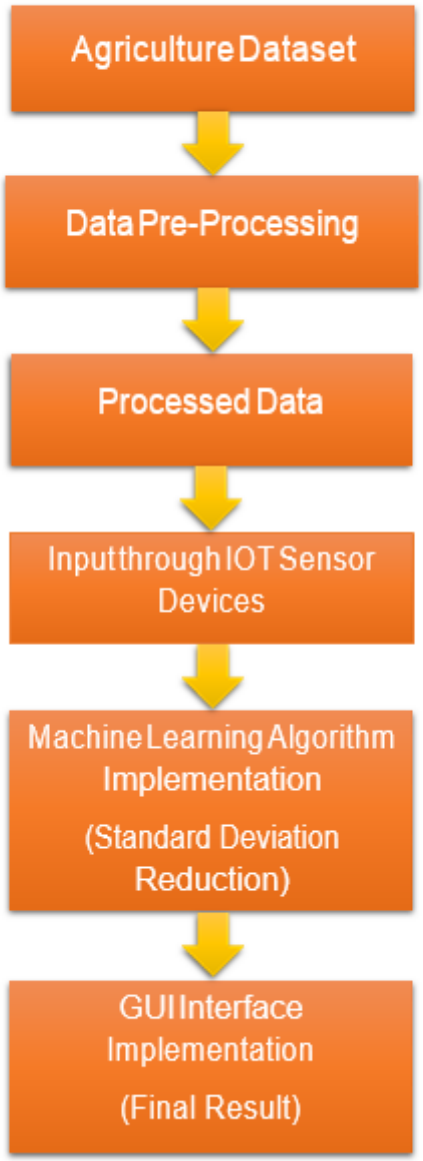

Figure 1. Architectural Diagram

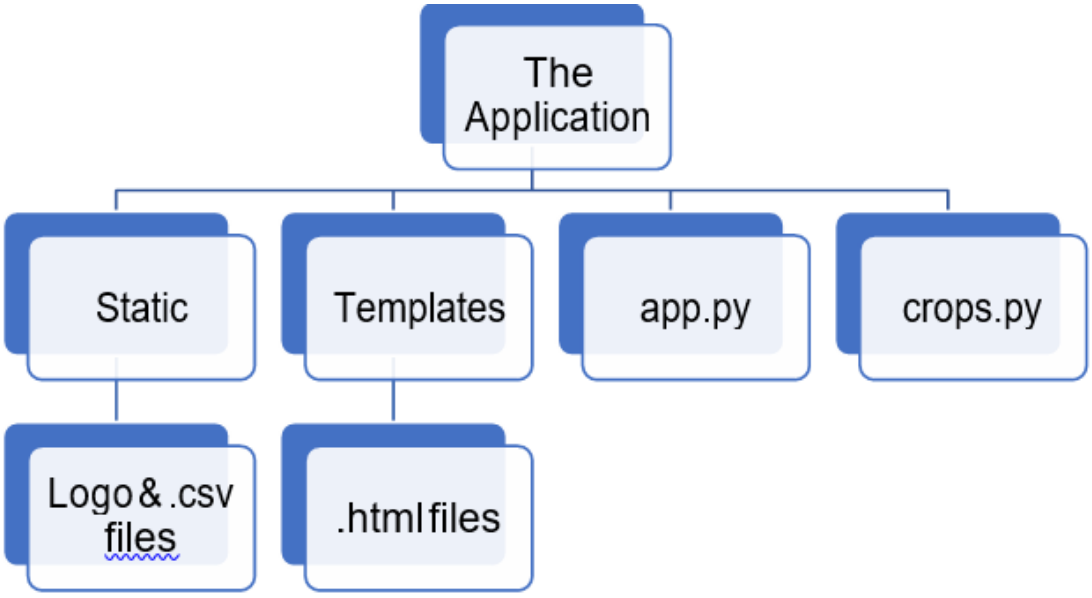

Figure 2. Structure of Software

Basic Info of the Software.

A web application developed as the flexibility of API and strongly bootstrapped so that it can be used as website as well as android application. The application is being developed using python web development framework "Flask" with the help of HTML & CSS.
As the application is powered with machine learning and artificial intelligence algorithms, so the python libraries like numpy, pandas, Scipy, Flask-CORS, and a very important library sklearn is used in order develop this application.

What is being done?

We are collecting sets of data of different factors which exhibits key role in the growth of the crops and processing these data to conclude a decision. There are various checks and standard on which our model is trained. Basically, sets of data are kept into csv files from where our model is fetching values and processing it.

Algorithms & Machine learning approaches used:

Right now, our model is incorporated with decision tree regression algorithm. For each crop our model is analyzing dependencies of the value and concluding through decision tree. For the splitting of the frequencies standard deviation reduction is being used

Descriptions of module and functions:

app.py

This is the file in which all the functions are defined and declared the framework. Different

getPredictedValue(self, value)

This function gives the predicted value of the crop.

index()

This function renders the html page basically, it gives renders home page.

TopFiveWinners()

This function gives the top five winner crops.

TopFiveLosers()

This function gives the top five looser crops.

SixMonthsForecast()

This function forecasts the values for the next six months.

SixMonthsForecastHelper(name)

This function helps with forecasts for the next six months.

TwelveMonthsForecast(name)

This function forecasts the values for the next twelve months.

TwelveMonthPrevious(name)

This function forecasts the values for the next twelve months.

crops.py

This file carries the dictionary of the crops

4. Proposed System Analysis and Design.

4.1 Introduction

The effect of environment changes, the harvests that are utilized to be developed by the ranchers in a procedural manner over the entire year has changed, presently if the system are followed the yields are not sufficient, which further places the ranchers in misfortune. To handle with such issues, we will fabricate an expectation web application, which will propose what to Grow and when to Grow. A significant issue which is right now Perpetuating in Indian Market is the worth Prediction for assortments of harvests in market. Since with the effect of environment changes, the harvests that are utilized to be developed by the ranchers in a procedural manner over the entire year has changed, presently if the system are followed the yields are not sufficient, which further places the ranchers in misfortune. To handle with such issues, we will fabricate an expectation web application, which will propose what to Grow and when to Grow. In the present period of Science and Technology, it's significant for every one of the various fields to have an advancement with the continuous advances, in later occasions, analysts learning at different schools and colleges commit their lives to imaginative work of developing things and practices. Ranchers have benefitted and added to the reliably creating investigation of cultivating. In this undertaking we target assisting the rancher with having an exchange advantage at Grain Market by means of conjecture data. The majority of the occasions the rancher is dealing with issues of picking the sort of harvest to be filled to have most elevated benefit, however the issue can't just be restricted to this as consider a case in some cases regardless of whether the yield developed has high market interest, yet the quality isn't acceptable which further prompts the rancher in the red, so a mix of the two things i.e., picking the best harvest as per the current soil pH, temperature, dampness and precipitation help in picking the best harvest for development. Further the rancher will likewise approach the anticipated cost for the yield in coming months, so the rancher can utilize the data consequently gave choosing the best harvest to development. We will give the finished result a Mobile Web App since portable application is not difficult to work in any event, for an individual having less specialized information on working the Smartphone. The rancher will be given a few devices involving Ph, dampness and so on sensors which will be associated with the Mobile App and the information will be gotten from cloud data set, next we will play out the information cleaning followed by AI strategies, will zero in on the upgrading the Decision Tree Regressor utilizing Scikit apparatuses, Scikit-Optimize library is an open-source Python library that gives an execution of Bayesian Optimization that can be utilized to tune the hyperparameters of AI models from the scikit-Learn Python library. We will utilize 80% information for preparing and staying 20% information for testing. Different advancements, for example, chart.js, NumPy, JavaScript, Bootstrap and so on will be utilized by the prerequisite. At long last, the outcome will be introduced to the rancher in the GUI Interface.

4.2 Requirement Analysis

4.2.1 Functional Requirement

4.2.1.1 Product Perspective

Our Product aims to create a simple interface for our beloved country farmers so that they don't have to face any difficulty in deciding which crop to grow on their land , so that it could be optimum in economic way to them.

4.2.1.2 Product Features

Product calculates different parameters like moisture, temperature, humidity of the land , earlier prices of a particular crop and then gives result whether it is optimum to grow that crop or not for the farmer of a particular region.

4.2.1.3 User Characteristics

User are mostly related to agriculture Field, Like Farmers, who would use the Product for

Getting an idea , of which crop to grow in a particular season.

4.2.1.4 Assumptions & Dependencies

The IOT devices and the algorithms which we are using , do not show 100% Results, but are made to work Precisely. Therefore the Final Output of the Product will depend on the collection of Data's for humidity, temperature and Earlier Prices.

4.2.1.5 Domain Requirements

4.2.1.5.1 Arduino System

4.2.1.5.2 Web Development (HTML, CSS,JAVASCRIPT)

4.2.1.5.3 Backend Development(on Flask)

4.2.1.5.4 Python Programming Language.

4.2.1.6 User Requirement

User will require a mobile Device(Android) and Pre designed IOT device , in order to calculate carryout the forecast estimation on his/her land.

4.2.2 Non Functional Requirement

4.2.2.1 Product Requirement

The product Requires hardware's such as Arduino Uno R3, Bread Board, Wifi Module etc and software web apps to provide easy and minimalistic interface to the user, in order to carry out the prediction on the user

land. Product is made in such a way that it will take minimal specifications on user device and will be able to generate an output , using its precise algorithms.

4.2.2.1.1 Efficiency (in terms of time and space)

Since Product is linking different modules like web app and Arduino module each other to calculate the prediction , it shows a little delay in fetching the data from the land, but once data collected , it generates a precise output , with collected specification.

4.2.2.1.2 Reliability

Since it uses Data tree machine Learning algorithm with accuracy greater than 90%, this product , in most cases give reliable answers to its users.

4.2.2.1.3 Portability

The product uses a user's mobile device for opening web app and an IOT device to measures different parameters of the land. Therefore the Overall product is Quite handy.

4.2.2.1.4 Usability

The Product is generally made for agriculture Domain, to get the land humidity, temperature, moisture like data's and pass it through the applied datasets and algorithm to predict the crop to grow.

4.2.2.2 Organisational Requirements

4.2.2.2.1 Implementation Requirement

For the implementation requirements as required by the organization the system requires the Flask Technology as the backend technology, rest the frontend of the system is designed in HTML, CSS, JavaScript which can be well handled by the client browser window or any other platform user using, so the main load that comes into place is the server side, we here deployed the system using the digital ocean droplet with the basic configuration i.e., 1 GB / 1 CPU. 25 GB SSD Disk, if the system needs to be extended to worldwide the same can be increased according to the requirements of the organization, the application works well with the deployed configuration.

4.2.2.2.2 Engineering standard Requirements

The system designed takes care of most of the engineering standard requirements, all the user data is well protected when being transferred as a header information, we are also using one of the most secure servers as offered by Digital Ocean, so the system is secure in terms of cyber security, the system will

handle data breaches if any. Also, the system is deployed with the most light and fast technology for the server-side development i.e., we are using flask for deployment. Also, the hardware as designed is designed to occupy the lowest space as possible and is designed with the components in a way to reduce the cost. So, in a conclusion the system is complies with the engineering standard requirements.

4.2.2.3 Operational Requirement(applicability of work w.r.t following operation)

- Economic

  The system designed is economic in terms of deployment as for deployment we only require the basic servers for configuration, the system is lightweight as the system is being used with flask. Also coming to the economic trend towards hardware, the system is designed with the components and techniques to lower the cost as far as possible, further the system can be more economical if implemented further with the government agencies.

- Environmental

  From the environmental concept, the system is environment friendly since we are using e-technology on mobiles, and other devices, also for hardware in case if the system needs to dumped it can be well recycled today so no problem to this system from environmental aspect.

- Social

  The system is well socially accepted as will help the farmer to make good and valuable decisions before itself, thus helping farmer to be a smart farmer.

- Ethical

  All the ethical requirements as from the industry requiring customer trust, reliability, customer relationship, behavior, mortality and etc. are well complied with the system.

- Health and Safety

  Health and safety are taken well care, the hardware is lightweight, does not have any issues since no chance of electric shock or any fire hazards.

- Sustainability

  The hardware system designed is sustainable, can withstand long time if handled and used correctly by the farmer.

- Legality

  The system is legally accepted since all engineering standard requirements were met also the system designed has no issues in legality issues when implementing.

- Inspectability

  The system is well inspected and tested before the release, further since the system is open-source the developers shall update the required updates with compliance to the customer requirements.

4.2.3 System Requirement

4.2.3.1 H/W Requirements

As our project consists of both of the components. We will be analyzing our requirements on two aspects, Developer End and the User End. To make the Humidity & Temperature sensing Device we need following component.

- Arduino Uno R3
- Bread Board
- 16x2 LCD Display
- Wi-fi Module (ESP 8266)
- Humidity Sensor (DHT 22)
- Temperature Sensor (TMP 36)
- Registers
- Jumper Wires

4.2.3.2 S/W Requirements

SOFTWARE REQUIREMENT to make use of the Humidity & Temperature sensing Device at Developer End

- Thingspeak API
- Python Script to decode Json Data From the feed of API
- IDE to write the Code

Software Requirement to make the flask app which runs all the machine learning modules to predict and show results to users.

- IDE for Python to code

- Python Libraries
    - Flask
    - Flask-Cors
    - Requests
    - Numpy
    - Pandas
- Apache 2 Server
- My SQL
- PHP My Admin
- Microsoft Excel
- HTML Codes to make front end Structures
- CSS to design Web pages
- JavaScript Codes
- Bootstrap

5.Results and Discussion

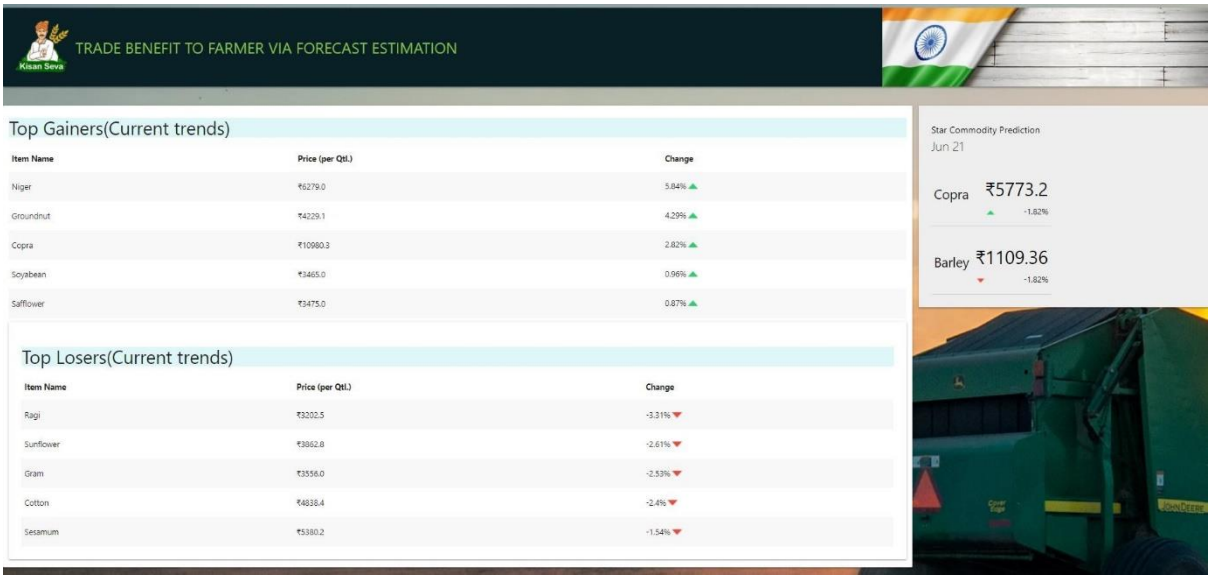

Figure 3. Homepage

The Starting of the web app contains 2 different sections

a) Top Gainer sections shows the top data of top 5 crops that can show possibly good results for a particular land using datasets of humidity, temperatures, prices of previous years, moisture. The list shows the item name,

price (per quintal) and change percentage of the crop.

b) Similarly, the section below the top gainers contain the “The Top Losers Section”, which shows the 5 crops list that is about to show poor yield results on the applied datasets.

Apart from these two sections on Right side of the web app we have provided a section/ separate window which shows the percentage change of both top gainer and top loser crop on a range from January to December.

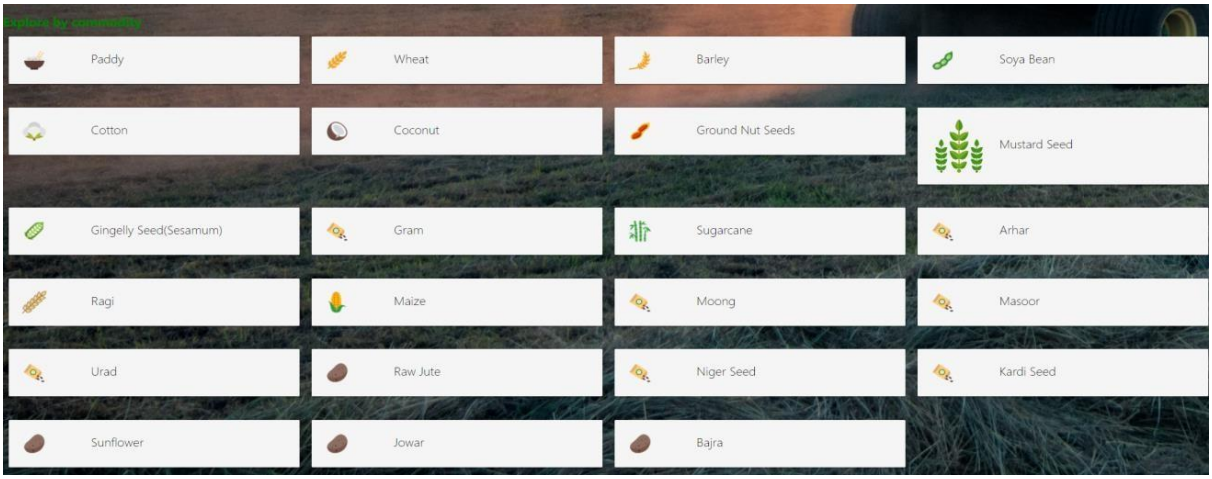

Figure 4. Explore by Commodity

In Explore by commodity section We have provided a single window system to navigate inside any desired crop section to view its full statistics like current price, prime location, crop type, exporting countries etc.

paddy

Current Price	₹ 1436.0615 / ql
Prime Location	W.B., U.P., Andhra Pradesh, Punjab, T.N.
Crop Type	kharif
Export	Bangladesh, Saudi Arabia, Iran

Brief Forecast

Min. crop price time	Oct 21	₹1362.58
Max. crop price time	Dec 21	₹1543.17

Figure 5. Commodity Section

Under commodity section of any crop chosen, different details will be provided like current price, prime location, crop type and right-side section show the minimum and maximum crop price time including their month and price.

Forecast Trends

Month	Price (per Qtl.)	Change
May 21	₹1443.53	0.52% ▲
Jun 21	₹1409.91	-1.82% ▼
Jul 21	₹1381.67	-3.79% ▼
Aug 21	₹1381.67	-3.79% ▼
Sep 21	₹1381.67	-3.79% ▼
Oct 21	₹1362.58	-5.12% ▼
Nov 21	₹1419.87	-1.13% ▼
Dec 21	₹1543.17	7.46% ▲
Jan 22	₹1431.08	-0.35% ▼
Feb 22	₹1451.01	1.04% ▲

Figure 6. Graph and Table of Commodity Section

This section also contains a list of price change list where all the price change percentage is listed according to the months in a year. Apart from all these the page shows the Graphs related to the

previous year and calculated upcoming year price of a particular chosen crop.

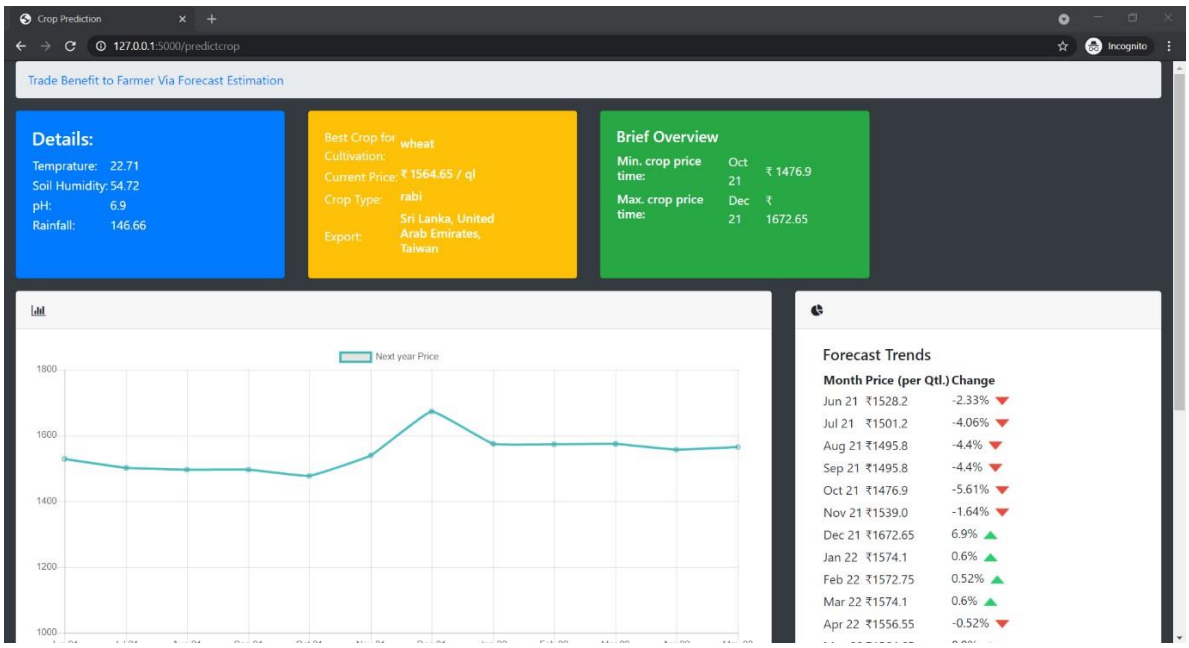

Figure 7. Crop Prediction based on suited environmental conditions

Figure 8. Graphs and Tables under prediction Page.

This module basically focuses on the crop prediction of the best crop to be grown, as we can see in the blue box, we are having values for different parameters of the farmers farm i.e., Temperature, Soil Humidity, Ph and rainfall, all the above values are fetched from the Arduino module as discussed. In the next box mentioned i.e., yellow box we have the best crop prediction for the suited environmental conditions, as we can see from the above observation, we have wheat as the best suited crop for the above conditions mentioned. Also, the data also infers about the crop type

and export to different countries. Coming to the next box gives information to the brief overview guiding to the minimum crop price and maximum crop price with the respective months of the year, so it becomes easy for the farmer when he/she should see the crops. All the above implementations are done via Machine Learning Algorithm i.e., Decision Tree with 80% + accuracy.Coming to the next phase of the implementation we have implemented two graphs placed side by side guiding about the future forecast of the crop predicted with each and every month, so the farmer can get an idea for the hype or lower of the crop price, the same is compared with the graph as shown to the below of this for the previous year statistics. Next implementation to this module is average variation change to the crops every month, the same is shown in the forecast trends card, showing what variation in price is there for the particular crop, red colour alerts for the price gone down and green colour denotes the increase in the price of the commodity predicted. So above implementation helps the farmer to get the best predicted crop along with the prices of the commodity for the upcoming months will certainly help the farmers to reduce debts and start farming.

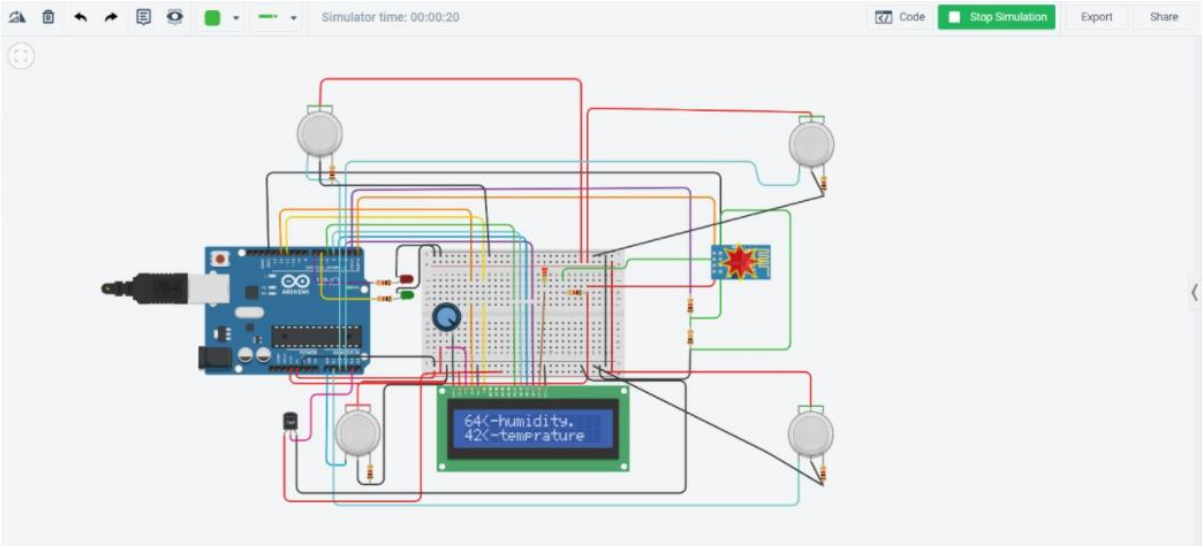

Figure 9. Arduino Circuit showing Data

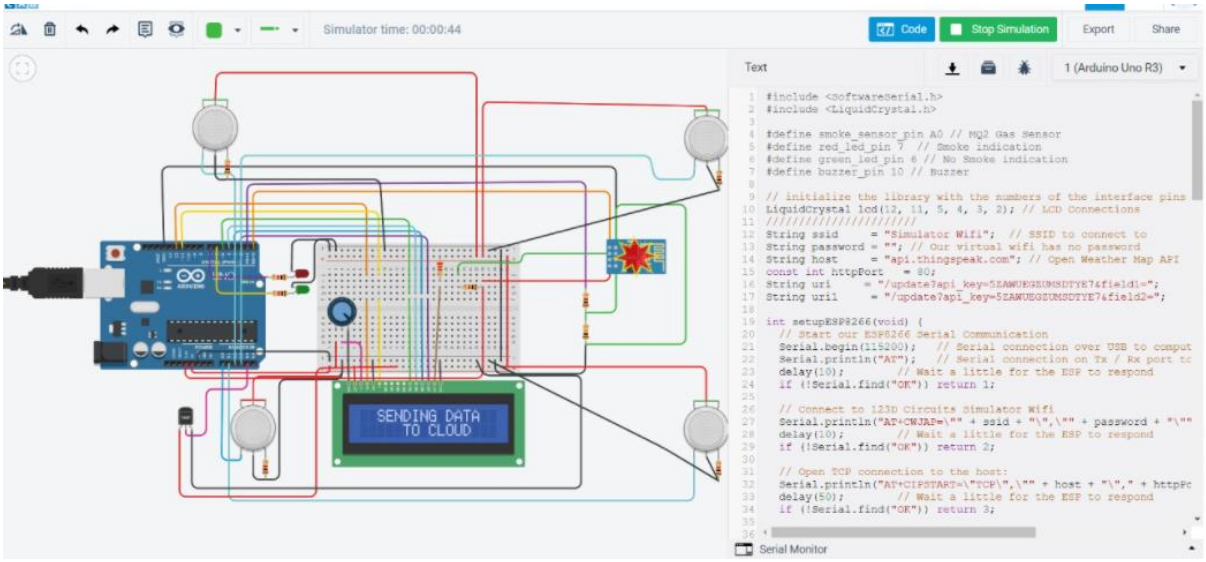

Figure 10. Circuit sending data to cloud

5.2 Analysis

The model proposed has attained an accuracy of 80%, the model suits well in most of the situations. The works accounts for a completion of 80-85%. The work-done shall certainly help the farmer to solve a major problem which is currently perpetuating in Indian Market that is the value prediction for varieties of crops in market and help reduce the losses and debts occurred. It's important for the farmers to adopt such smart practices and be a smart farmer since this shall help the nation in supporting the economic growth.

6. References

[1]Randall J. Donohuea, Roger A. Lawes, Gonzalo Mata, David Gobbett and Jackie Ouzmanc (2018) Towards a national, remote-sensing-based model for predicting field-scale crop yield, Elsevier Pages 79-90

[2]Bruno Basso, Lin Liu (2019) Seasonal crop yield forecast: Methods, applications, and accuracies,

Elsevier Pages 201-255

[3]Rohit Kumar Rajak, Ankit Pawar, Mitalee Pendke, Pooja Shinde, Suresh Rathod and Avinash Devare (2017) Crop Recommendation System to Maximize Crop Yield using Machine Learning Technique, IRJET

[4]R.Harine Rajashree, C.Kavya, T.Kiruthika and J.Nisha (2017) Crop Recommendation System for Precision Agriculture, IEEE

[5]Gupta, G., Setia, R., Meena, A., & Jaint, B. (2020). Environment Monitoring System for Agricultural Application using IoT and Predicting Crop Yield using Various Data Mining Techniques. 2020 5th International Conference on Communication and Electronics Systems (ICCES). doi:10.1109/icces48766.2020.9138032

[6]Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data Jiaxuan You,

Xiaocheng Li, Melvin Low, David Lobell, Stefano Ermon 2017

[7]Prasad, A. K., Chai, L., Singh, R. P., & Kafatos, M. (2006). Crop yield estimation model for Iowa using remote sensing and surface parameters. International Journal of Applied Earth Observation and Geoinformation, 8(1), 26–33. doi:10.1016/j.jag.2005.06.002

[8]Chlingaryan, A., Sukkarieh, S., & Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. Computers and Electronics in Agriculture, 151, 61–69. doi:10.1016/j.compag.2018.05.012

[9]Development of crop yield forecast models under FASAL- a case study of kharif rice in West Bengal

K. GHOSH, R. BALASUBRAMANIAN, S. BANDOPADHYAY2 , N. CHATTOPADHYAY, K.K. SINGH1 and L.S. RATHORE1 ,2013

[10]Crop Yield Forecast at Different Growth Stage of Wheat Crop using Statistical Model under Semi Arid Region, Ananta vashisth , R.singh

[11]LowerThanExpected Crop Yield Stimulation with Rising CO2 Concentrations, Stephen P.long,

Donald R.ort

[12]Weather Based Forecasting of Crop Yields, Pests and Diseases - IASRI Models, Ranjana Agarwal and S.C. Mehta

[13]System for Precision Agriculture, 2017 T.Kiruthika and J.Nisha

[14]Agriculture monitoring and prediction using Internet of Things (IoT), Mohit Kumar Saini, Rakesh Kumar Saini

[15]Cunha, R. L. F., Silva, B., & Netto, M. A. S. (2018). A Scalable Machine Learning System for Pre- Season Agriculture Yield Forecast. 2018 IEEE 14th International Conference on e-Science (e-

Science). doi:10.1109/escience.2018.00131

[16]Liu, Q., Jin, D., Shen, J., Fu, Z., & Linge, N. (2016). A WSN-based prediction model of microclimate in a greenhouse using extreme learning approaches. 2016 18th International Conference on Advanced Communication Technology (ICACT). doi:10.1109/icact.2016.7423609

Appendix A

Arduino Code

```
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>

#define smoke_sensor_pin A0 // MQ2 Gas Sensor
#define red_led_pin 7  // Smoke indication
#define green_led_pin 6 // No Smoke indication
#define buzzer_pin 10 // Buzzer

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // LCD Connections
//////////////////////
String ssid     = "Simulator Wifi";  // SSID to connect to
```

```
String password = ""; // Our virtual wifi has no password
String host     = "api.thingspeak.com"; // Open Weather Map API
const int httpPort   = 80;
String uri     = "/update?api_key=5ZAWUEGZUMSDTYE7&field1=";
String uri1     = "/update?api_key=5ZAWUEGZUMSDTYE7&field2=";

int setupESP8266(void) {
  // Start our ESP8266 Serial Communication
  Serial.begin(115200);   // Serial connection over USB to computer
  Serial.println("AT");   // Serial connection on Tx / Rx port to ESP8266
  delay(10);        // Wait a little for the ESP to respond
  if (!Serial.find("OK")) return 1;

  // Connect to 123D Circuits Simulator Wifi
  Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
  delay(10);        // Wait a little for the ESP to respond
  if (!Serial.find("OK")) return 2;

  // Open TCP connection to the host:
  Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);
  delay(50);        // Wait a little for the ESP to respond
  if (!Serial.find("OK")) return 3;

  return 0;
}

void anydata(int &intensity) {

  // Construct our HTTP call
  String httpPacket = "GET " + uri + String(intensity) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
  int length = httpPacket.length();

  // Send our message length
  Serial.print("AT+CIPSEND=");
```

```
  Serial.println(length);
  delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

  // Send our http request
  Serial.print(httpPacket);
  delay(10); // Wait a little for the ESP to respond
  if (!Serial.find("SEND OK\r\n")) return;

}
void anydata1(int &intensity) {

  // Construct our HTTP call
  String httpPacket = "GET " + uri1 + String(intensity) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
  int length = httpPacket.length();

  // Send our message length
  Serial.print("AT+CIPSEND=");
  Serial.println(length);
  delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

  // Send our http request
  Serial.print(httpPacket);
  delay(10); // Wait a little for the ESP to respond
  if (!Serial.find("SEND OK\r\n")) return;

}

//////////////////////////////

void update(int&, int&, int&, int&, int&);

void setup(){
  pinMode(red_led_pin, OUTPUT);
  pinMode(green_led_pin, OUTPUT);
```

```
  pinMode(buzzer_pin, OUTPUT);
  pinMode(smoke_sensor_pin, INPUT);
  Serial.begin(9600); //  serial data transmission at Baudrate of 9600
  lcd.begin(16, 2);  // to intialize LCD
  lcd.setCursor(0,0);
  lcd.print("Welcome");
  delay(300);
  lcd.setCursor(0,0);
  lcd.print("Loading...");
  delay(300);
  setupESP8266();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("     WIFI");
  lcd.setCursor(0,1);
  lcd.print("  CONNECTED");

}



void loop(){
  int data1 = 0;
  int data2 = 0;
  int data3 = 0;
  int data4 = 0;
  int temp  = 0;
  static int input_count = 16;
  update(data1, data2, data3, data4, temp);
  int avg = (data1+data2+data3+data4)/4;
  if (input_count>=16){
          input_count %= 16;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("  SENDING DATA");
```

```
    lcd.setCursor(0,1);
    lcd.print("     TO CLOUD");
    anydata(avg);
    delay(16000);
    anydata1(temp);
  }
  input_count++;
}

void update(int &data1, int &data2, int &data3, int &data4, int &temp){
  delay(250);
  data1 = map(analogRead(A0),10,350,0,100);
  data2 = map(analogRead(A1),10,350,0,100);
  data3 = map(analogRead(A2),10,350,0,100);
  data4 = map(analogRead(A3),10,350,0,100);
  temp  = map(analogRead(A4),10,350,0,100);
  int avg = (data1+data2+data3+data4)/4;
  Serial.println();
  lcd.clear();
  lcd.setCursor (0, 0);
  lcd.print (avg);
  lcd.print ("<-humidity.");
  lcd.setCursor (0,1);
  lcd.print(temp);
  lcd.print("<-temprature");
}
```

App.Py

```
# -*- coding: utf-8 -*-
"""
Created on Sat Mar  2 21:46:27 2019
pip
"""
```

```
from flask import Flask, render_template
from flask_cors import CORS, cross_origin
import numpy as np
import pandas as pd
from datetime import datetime
import crops
import random

# import matplotlib.pyplot as plt

app = Flask(__name__)
app.config['CORS_HEADERS'] = 'Content-Type'

cors = CORS(app, resources={r"/ticker": {"origins": "http://localhost:port"}})

commodity_dict = {
    "arhar": "static/Arhar.csv",
    "bajra": "static/Bajra.csv",
    "barley": "static/Barley.csv",
    "copra": "static/Copra.csv",
    "cotton": "static/Cotton.csv",
    "sesamum": "static/Sesamum.csv",
    "gram": "static/Gram.csv",
    "groundnut": "static/Groundnut.csv",
    "jowar": "static/Jowar.csv",
    "maize": "static/Maize.csv",
    "masoor": "static/Masoor.csv",
    "moong": "static/Moong.csv",
    "niger": "static/Niger.csv",
    "paddy": "static/Paddy.csv",
    "ragi": "static/Ragi.csv",
    "rape": "static/Rape.csv",
    "jute": "static/Jute.csv",
    "safflower": "static/Safflower.csv",
    "soyabean": "static/Soyabean.csv",
    "sugarcane": "static/Sugarcane.csv",
    "sunflower": "static/Sunflower.csv",
    "urad": "static/Urad.csv",
    "wheat": "static/Wheat.csv"
}

annual_rainfall = [29, 21, 37.5, 30.7, 52.6, 150, 299, 251.7, 179.2, 70.5, 39.8, 10.9]
base = {
    "Paddy": 1245.5,
    "Arhar": 3200,
    "Bajra": 1175,
    "Barley": 980,
    "Copra": 5100,
    "Cotton": 3600,
    "Sesamum": 4200,
    "Gram": 2800,
```

```
    "Groundnut": 3700,
    "Jowar": 1520,
    "Maize": 1175,
    "Masoor": 2800,
    "Moong": 3500,
    "Niger": 3500,
    "Ragi": 1500,
    "Rape": 2500,
    "Jute": 1675,
    "Safflower": 2500,
    "Soyabean": 2200,
    "Sugarcane": 2250,
    "Sunflower": 3700,
    "Urad": 4300,
    "Wheat": 1350

}
commodity_list = []


class Commodity:

    def __init__(self, csv_name):
        self.name = csv_name
        dataset = pd.read_csv(csv_name)
        self.X = dataset.iloc[:, :-1].values
        self.Y = dataset.iloc[:, 3].values

        #from sklearn.model_selection import train_test_split
        #X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1, random_s
tate=0)

        # Fitting decision tree regression to dataset
        from sklearn.tree import DecisionTreeRegressor
        depth = random.randrange(7,18)
        self.regressor = DecisionTreeRegressor(max_depth=depth)
        self.regressor.fit(self.X, self.Y)
        #y_pred_tree = self.regressor.predict(X_test)
        # fsa=np.array([float(1),2019,45]).reshape(1,3)
        # fask=regressor_tree.predict(fsa)

    def getPredictedValue(self, value):
        if value[1]>=2019:
            fsa = np.array(value).reshape(1, 3)
            #print(" ",self.regressor.predict(fsa)[0])
            return self.regressor.predict(fsa)[0]
        else:
            c=self.X[:,0:2]
            x=[]
            for i in c:
                x.append(i.tolist())
            fsa = [value[0], value[1]]
```

```
            ind = 0
            for i in range(0,len(x)):
                if x[i]==fsa:
                    ind=i
                    break
            #print(index, " ",ind)
            #print(x[ind])
            #print(self.Y[i])
            return self.Y[i]

    def getCropName(self):
        a = self.name.split('.')
        return a[0]


@app.route('/')
def index():
    context = {
        "top5": TopFiveWinners(),
        "bottom5": TopFiveLosers(),
        "sixmonths": SixMonthsForecast()
    }
    return render_template('index.html', context=context)


@app.route('/commodity/<name>')
def crop_profile(name):
    max_crop, min_crop, forecast_crop_values = TwelveMonthsForecast(name)
    prev_crop_values = TwelveMonthPrevious(name)
    forecast_x = [i[0] for i in forecast_crop_values]
    forecast_y = [i[1] for i in forecast_crop_values]
    previous_x = [i[0] for i in prev_crop_values]
    previous_y = [i[1] for i in prev_crop_values]
    current_price = CurrentMonth(name)
    #print(max_crop)
    #print(min_crop)
    #print(forecast_crop_values)
    #print(prev_crop_values)
    #print(str(forecast_x))
    crop_data = crops.crop(name)
    context = {
        "name":name,
        "max_crop": max_crop,
        "min_crop": min_crop,
        "forecast_values": forecast_crop_values,
        "forecast_x": str(forecast_x),
        "forecast_y":forecast_y,
        "previous_values": prev_crop_values,
        "previous_x":previous_x,
        "previous_y":previous_y,
        "current_price": current_price,
        "image_url":crop_data[0],
```

```
        "prime_loc":crop_data[1],
        "type_c":crop_data[2],
        "export":crop_data[3]
    }
    return render_template('commodity.html', context=context)

@app.route('/ticker/<item>/<number>')
@cross_origin(origin='localhost',headers=['Content- Type','Authorization'])
def ticker(item, number):
    n = int(number)
    i = int(item)
    data = SixMonthsForecast()
    context = str(data[n][i])

    if i == 2 or i == 5:
        context = '₹' + context
    elif i == 3 or i == 6:

        context = context + '%'

    #print('context: ', context)
    return context


def TopFiveWinners():
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    prev_month = current_month - 1
    prev_rainfall = annual_rainfall[prev_month - 1]
    current_month_prediction = []
    prev_month_prediction = []
    change = []

    for i in commodity_list:
        current_predict = i.getPredictedValue([float(current_month), current_year, current
_rainfall])
        current_month_prediction.append(current_predict)
        prev_predict = i.getPredictedValue([float(prev_month), current_year, prev_rainfall
])
        prev_month_prediction.append(prev_predict)
        change.append((((current_predict - prev_predict) * 100 / prev_predict), commodity_
list.index(i)))
    sorted_change = change
    sorted_change.sort(reverse=True)
    # print(sorted_change)
    to_send = []
    for j in range(0, 5):
        perc, i = sorted_change[j]
        name = commodity_list[i].getCropName().split('/')[1]
        to_send.append([name, round((current_month_prediction[i] * base[name]) / 100, 2),
round(perc, 2)])
```

```
    #print(to_send)
    return to_send


def TopFiveLosers():
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    prev_month = current_month - 1
    prev_rainfall = annual_rainfall[prev_month - 1]
    current_month_prediction = []
    prev_month_prediction = []
    change = []

    for i in commodity_list:
        current_predict = i.getPredictedValue([float(current_month), current_year, current
_rainfall])
        current_month_prediction.append(current_predict)
        prev_predict = i.getPredictedValue([float(prev_month), current_year, prev_rainfall
])
        prev_month_prediction.append(prev_predict)
        change.append((((current_predict - prev_predict) * 100 / prev_predict), commodity_
list.index(i)))
    sorted_change = change
    sorted_change.sort()
    to_send = []
    for j in range(0, 5):
        perc, i = sorted_change[j]
        name = commodity_list[i].getCropName().split('/')[1]
        to_send.append([name, round((current_month_prediction[i] * base[name]) / 100, 2),
round(perc, 2)])
   # print(to_send)
    return to_send



def SixMonthsForecast():
    month1=[]
    month2=[]
    month3=[]
    month4=[]
    month5=[]
    month6=[]
    for i in commodity_list:
        crop=SixMonthsForecastHelper(i.getCropName())
        k=0
        for j in crop:
            time = j[0]
            price = j[1]
            change = j[2]
            if k==0:
                month1.append((price,change,i.getCropName().split("/")[1],time))
```

```
            elif k==1:
                month2.append((price,change,i.getCropName().split("/")[1],time))
            elif k==2:
                month3.append((price,change,i.getCropName().split("/")[1],time))
            elif k==3:
                month4.append((price,change,i.getCropName().split("/")[1],time))
            elif k==4:
                month5.append((price,change,i.getCropName().split("/")[1],time))
            elif k==5:
                month6.append((price,change,i.getCropName().split("/")[1],time))
            k+=1
    month1.sort()
    month2.sort()
    month3.sort()
    month4.sort()
    month5.sort()
    month6.sort()
    crop_month_wise=[]
    crop_month_wise.append([month1[0][3],month1[len(month1)-1][2],month1[len(month1)-
1][0],month1[len(month1)-1][1],month1[0][2],month1[0][0],month1[0][1]])
    crop_month_wise.append([month2[0][3],month2[len(month2)-1][2],month2[len(month2)-
1][0],month2[len(month2)-1][1],month2[0][2],month2[0][0],month2[0][1]])
    crop_month_wise.append([month3[0][3],month3[len(month3)-1][2],month3[len(month3)-
1][0],month3[len(month3)-1][1],month3[0][2],month3[0][0],month3[0][1]])
    crop_month_wise.append([month4[0][3],month4[len(month4)-1][2],month4[len(month4)-
1][0],month4[len(month4)-1][1],month4[0][2],month4[0][0],month4[0][1]])
    crop_month_wise.append([month5[0][3],month5[len(month5)-1][2],month5[len(month5)-
1][0],month5[len(month5)-1][1],month5[0][2],month5[0][0],month5[0][1]])
    crop_month_wise.append([month6[0][3],month6[len(month6)-1][2],month6[len(month6)-
1][0],month6[len(month6)-1][1],month6[0][2],month6[0][0],month6[0][1]])

   # print(crop_month_wise)
    return crop_month_wise

def SixMonthsForecastHelper(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.split("/")[1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 7):
        if current_month + i <= 12:
            month_with_year.append((current_month + i, current_year, annual_rainfall[curre
nt_month + i - 1]))
        else:
```

```
            month_with_year.append((current_month + i - 12, current_year + 1, annual_rainf
all[current_month + i - 13]))
    wpis = []
    current_wpi = commodity.getPredictedValue([float(current_month), current_year, current
_rainfall])
    change = []

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), y, r])
        wpis.append(current_predict)
        change.append(((current_predict - current_wpi) * 100) / current_wpi)

    crop_price = []
    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y, m, 1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2) , round(c
hange[i], 2)])

   # print("Crop_Price: ", crop_price)
    return crop_price

def CurrentMonth(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    current_wpi = commodity.getPredictedValue([float(current_month), current_year, current
_rainfall])
    current_price = (base[name.capitalize()]*current_wpi)/100
    return current_price

def TwelveMonthsForecast(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 13):
        if current_month + i <= 12:
```

```
            month_with_year.append((current_month + i, current_year, annual_rainfall[curre
nt_month + i - 1]))
        else:
            month_with_year.append((current_month + i - 12, current_year + 1, annual_rainf
all[current_month + i - 13]))
    max_index = 0
    min_index = 0
    max_value = 0
    min_value = 9999
    wpis = []
    current_wpi = commodity.getPredictedValue([float(current_month), current_year, current
_rainfall])
    change = []

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), y, r])
        if current_predict > max_value:
            max_value = current_predict
            max_index = month_with_year.index((m, y, r))
        if current_predict < min_value:
            min_value = current_predict
            min_index = month_with_year.index((m, y, r))
        wpis.append(current_predict)
        change.append(((current_predict - current_wpi) * 100) / current_wpi)

    max_month, max_year, r1 = month_with_year[max_index]
    min_month, min_year, r2 = month_with_year[min_index]
    min_value = min_value * base[name.capitalize()] / 100
    max_value = max_value * base[name.capitalize()] / 100
    crop_price = []
    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y, m, 1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2) , round(c
hange[i], 2)])
   # print("forecasr", wpis)
    x = datetime(max_year,max_month,1)
    x = x.strftime("%b %y")
    max_crop = [x, round(max_value,2)]
    x = datetime(min_year, min_month, 1)
    x = x.strftime("%b %y")
    min_crop = [x, round(min_value,2)]

    return max_crop, min_crop, crop_price


def TwelveMonthPrevious(name):
    name = name.lower()
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
```

```
    commodity = commodity_list[0]
    wpis = []
    crop_price = []
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 13):
        if current_month - i >= 1:
            month_with_year.append((current_month - i, current_year, annual_rainfall[curre
nt_month - i - 1]))
        else:
            month_with_year.append((current_month - i + 12, current_year - 1, annual_rainf
all[current_month - i + 11]))

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), 2013, r])
        wpis.append(current_predict)

    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y,m,1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2)])
   # print("previous ", wpis)
    new_crop_price =[]
    for i in range(len(crop_price)-1,-1,-1):
        new_crop_price.append(crop_price[i])
    return new_crop_price


if __name__ == "__main__":
    arhar = Commodity(commodity_dict["arhar"])
    commodity_list.append(arhar)
    bajra = Commodity(commodity_dict["bajra"])
    commodity_list.append(bajra)
    barley = Commodity(commodity_dict["barley"])
    commodity_list.append(barley)
    copra = Commodity(commodity_dict["copra"])
    commodity_list.append(copra)
    cotton = Commodity(commodity_dict["cotton"])
    commodity_list.append(cotton)
    sesamum = Commodity(commodity_dict["sesamum"])
    commodity_list.append(sesamum)
    gram = Commodity(commodity_dict["gram"])
    commodity_list.append(gram)
    groundnut = Commodity(commodity_dict["groundnut"])
    commodity_list.append(groundnut)
    jowar = Commodity(commodity_dict["jowar"])
    commodity_list.append(jowar)
    maize = Commodity(commodity_dict["maize"])
```

```
    commodity_list.append(maize)
    masoor = Commodity(commodity_dict["masoor"])
    commodity_list.append(masoor)
    moong = Commodity(commodity_dict["moong"])
    commodity_list.append(moong)
    niger = Commodity(commodity_dict["niger"])
    commodity_list.append(niger)
    paddy = Commodity(commodity_dict["paddy"])
    commodity_list.append(paddy)
    ragi = Commodity(commodity_dict["ragi"])
    commodity_list.append(ragi)
    rape = Commodity(commodity_dict["rape"])
    commodity_list.append(rape)
    jute = Commodity(commodity_dict["jute"])
    commodity_list.append(jute)
    safflower = Commodity(commodity_dict["safflower"])
    commodity_list.append(safflower)
    soyabean = Commodity(commodity_dict["soyabean"])
    commodity_list.append(soyabean)
    sugarcane = Commodity(commodity_dict["sugarcane"])
    commodity_list.append(sugarcane)
    sunflower = Commodity(commodity_dict["sunflower"])
    commodity_list.append(sunflower)
    urad = Commodity(commodity_dict["urad"])
    commodity_list.append(urad)
    wheat = Commodity(commodity_dict["wheat"])
    commodity_list.append(wheat)

    app.run(debug=True)
```

index.html

```
<!DOCTYPE html>
<html>
<head>

  <title>Apna Anaaj</title>
  <!-- Compiled and minified CSS -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/
css/materialize.min.css">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">


    <!-- Compiled and minified JavaScript -->
    <style>
        div.main {
            padding: 5px 50px 75px 50px;
```

```
        }

        span,img {
            vertical-align: middle;
         }

        .brand-logo > span,.brand-logo > img {
            vertical-align: middle;
        }

        .amber.lighten-2 {
    background-color: #092126 !important;
}


    </style>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></scrip
t>

</head>
<body class="main" style="background-image: url('/static/bg1.jpg')" >
  <div>



  <div class="nav-wrapper"  >

    <h3 class="card-panel #092126 amber lighten-2 center valign-
wrapper center" style="background-image: url('https://img.freepik.com/free-photo/india-
flag-old-white-wall-horizontal-panoramic-banner_118047-
4766.jpg?size=626&ext=jpg');background-repeat: no-repeat;background-
position: right;"><a class="brand-logo  #ffd54f-text text-lighten-
4" href="#"><img alt="" src="/static/logo.png" height="100px" width="100px" class="respons
ive-img" />
          <span style="color:#8cc050;font-
size: 1.92rem;">TRADE BENEFIT TO FARMER VIA FORECAST ESTIMATION</span></a>
          </a></h3><h3>

  </div>

    <!--
  <h3 class="card-panel #ffd54f amber lighten-2 center valign-
wrapper center"><img src="/static/ApnaAnaajLogo.png
" height="70px" width="70px">Apna Anaaj</h3>
    -->

    <div class="row">
    <div class="card-panel col s9">
      <h4 class=" #e0f7fa cyan lighten-5">Top Gainers(Current trends)</h4>
      <table class="striped">
        <thead>
          <tr>
```

```
            <th>Item Name</th>
            <th>Price (per Qtl.)</th>
            <th>Change</th>
          </tr>
        </thead>

        <tbody>
        {% for item in context['top5'] %}
          <tr>
            <td>{{item[0]}}</td>
            <td>₹{{item[1]}}</td>
            <td class="valign-wrapper">{{item[2]}}% <img src="../static/gain-
icon.png" height="25" width="25"></td>
          </tr>
        {% endfor %}
        </tbody>
      </table>
        <div class="card-panel">
      <h4 class="#e0f7fa cyan lighten-5">Top Losers(Current trends)</h4>
      <table class="striped">
        <thead>
          <tr>
              <th>Item Name</th>
              <th>Price (per Qtl.)</th>
              <th>Change</th>
          </tr>
        </thead>

        <tbody>
        {% for item in context['bottom5'] %}
          <tr>
            <td>{{item[0]}}</td>
            <td>₹{{item[1]}}</td>
            <td class="valign-wrapper">{{item[2]}}% <img src="../static/loss-
icon.png" height="25" width="25"></td>
          </tr>
        {% endfor %}
        </tbody>
      </table>
            </div>
    </div>
    <div class="col s4">
      <div class="card grey lighten-3">
        <div class="card-content black-text">
          <h6>Star Commodity Prediction</h6>
           <span class="card-title" id="time">{{context.sixmonths[0][0]}}</span>
          <table class="valign-wrapper">
            <tr>
              <td><h5 id="crop1">{{context.sixmonths[0][1]}}</h5></td>

              <td class="right">
                  <h4 id="price1">₹{{context.sixmonths[0][2]}}</h4>
```

```
              <p id="pos-change" class="valign-
wrapper right">{{context.sixmonths[0][3]}}%</p><img id="change1" src="../static/gain-
icon.png" height="20" width="20" style="{margin-left: 5;}"></td>

            </tr>
            <tr>
              <td><h5 id="crop2">{{context.sixmonths[0][4]}}</h5></td>

              <td class="right"><h4 id="price2">₹{{context.sixmonths[0][5]}}</h4>
              <p id="neg-change" class="valign-
wrapper right">{{context.sixmonths[0][6]}}%</p><img id="change2" src="../static/loss-
icon.png" height="20" width="20" style="{margin-left: 5;}">
                </td>
            </tr>
          </table>
          <script>
            function sleep(ms) {
              return new Promise(resolve => setTimeout(resolve, ms));
             }
            function updateTime(i){
              $('#time').load('http://localhost:5000/ticker/'+0+ '/' + i);
            }

            function updateCrop1(i){
              $('#crop1').load('http://localhost:5000/ticker/'+ 1 + '/' + i);
            }

            function updatePrice1(i){
              $('#price1').load('http://localhost:5000/ticker/'+ 2 + '/' + i);
            }

            function updateChange1(i){
              $("#change1").attr("src","../static/gain-icon.png");
              $('#pos-change').load('http://localhost:5000/ticker/'+ 3 + '/'  + i);

            }

            function updateCrop2(i){
              $('#crop2').load('http://localhost:5000/ticker/'+ 4 + '/'  + i);
            }

            function updatePrice2(i){
              $('#price2').load('http://localhost:5000/ticker/'+ 5 + '/'  + i);
            }

            function updateChange2(i){
              $("#change2").attr("src","../static/loss-icon.png");
              $('#neg-change').load('http://localhost:5000/ticker/'+ 6 + '/'  + i);

            }
          </script>
```

```
        <script>
                var i = 0
                setInterval(async function(){
                i = (i+1)%5;
                updateTime(i);

                updateCrop1(i);
                await sleep(200);

                updatePrice1(i);
                await sleep(200);

                updateChange1(i);
                await sleep(200);

                updateCrop2(i);
                await sleep(200);

                updatePrice2(i);
                await sleep(200);

                updateChange2(i);
                await sleep(200);

                /*
                setTimeout(updateCrop1(i), 100);
                setTimeout(updatePrice1(i), 300);
                setTimeout(updateChange1(i), 500);
                setTimeout(updateCrop2(i), 700);
                setTimeout(updatePrice2(i), 800);
                setTimeout(updateChange2(i), 900);
                */

                }, 4000);

        </script>

      </div>
    </div>

  </div>
</div>
<h5 style="color:green"><b>Explore by commodity</b></h5>
<div class="row">
  <div class="col s3">
    <a href="http://localhost:5000/commodity/paddy" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/rice-bowl.png">
      </div>
      <div class="col s9">
```

```
        <span class="card-title">Paddy</span>
      </div>
      </div>
    </div>
  </a>
  </div>
   <div class="col s3">
    <a href="http://localhost:5000/commodity/wheat" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/wheat.png">
      </div>
      <div class="col s9">
        <span class="card-title">Wheat</span>
      </div>
      </div>
    </div>
  </a>
  </div> <div class="col s3">
    <a href="http://localhost:5000/commodity/barley" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/barley.png">
      </div>
      <div class="col s9">
        <span class="card-title">Barley</span>
      </div>
      </div>
    </div>
  </a>
  </div> <div class="col s3">
    <a href="http://localhost:5000/commodity/soyabean" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/soy.png">
      </div>
      <div class="col s9">
        <span class="card-title">Soya Bean</span>
      </div>
      </div>
    </div>
  </a>
  </div>
</div>
  <div class="row">
  <div class="col s3">
    <a href="http://localhost:5000/commodity/cotton" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
```

```
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/cotton.png">
      </div>
      <div class="col s9">
        <span class="card-title">Cotton</span>
      </div>
      </div>
    </div>
  </a>
  </div>
   <div class="col s3">
    <a href="http://localhost:5000/commodity/copra" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/coconut.png">
      </div>
      <div class="col s9">
        <span class="card-title">Coconut</span>
      </div>
      </div>
    </div>
  </a>
  </div> <div class="col s3">
    <a href="http://localhost:5000/commodity/groundnut" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/peanuts.png">
      </div>
      <div class="col s9">
        <span class="card-title">Ground Nut Seeds</span>
      </div>
      </div>
    </div>
  </a>
  </div> <div class="col s3">
    <a href="http://localhost:5000/commodity/rape" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://image.flaticon.com/icons/svg/188/188317.svg">
      </div>
      <div class="col s9">
        <span class="card-title">Mustard Seed</span>
      </div>
      </div>
    </div>
  </a>
  </div>
</div>
  <div class="row">
```

```
<div class="col s3">
  <a href="http://localhost:5000/commodity/sesamum" style="color: #000000">
  <div class="card grey lighten-4">
    <div class="card-content row valign-wrapper">
      <div class="col s3">
      <img src="https://img.icons8.com/color/48/000000/sesame.png">
    </div>
    <div class="col s9">
      <span class="card-title">Gingelly Seed(Sesamum)</span>
    </div>
    </div>
  </div>
</a>
</div>
 <div class="col s3">
  <a href="http://localhost:5000/commodity/gram" style="color: #000000">
  <div class="card grey lighten-4">
    <div class="card-content row valign-wrapper">
      <div class="col s3">
      <img src="https://img.icons8.com/color/48/000000/paper-bag-with-seeds.png">
    </div>
    <div class="col s9">
      <span class="card-title">Gram</span>
    </div>
    </div>
  </div>
</a>
</div> <div class="col s3">
  <a href="http://localhost:5000/commodity/sugarcane" style="color: #000000">
  <div class="card grey lighten-4">
    <div class="card-content row valign-wrapper">
      <div class="col s3">
      <img src="https://img.icons8.com/color/48/000000/bamboo.png">
    </div>
    <div class="col s9">
      <span class="card-title">Sugarcane</span>
    </div>
    </div>
  </div>
</a>
</div> <div class="col s3">
  <a href="http://localhost:5000/commodity/arhar" style="color: #000000">
  <div class="card grey lighten-4">
    <div class="card-content row valign-wrapper">
      <div class="col s3">
      <img src="https://img.icons8.com/color/48/000000/paper-bag-with-seeds.png">
    </div>
    <div class="col s9">
      <span class="card-title">Arhar</span>
    </div>
    </div>
  </div>
```

```
  </a>
  </div>
</div>
  <div class="row">
  <div class="col s3">
    <a href="http://localhost:5000/commodity/ragi" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/office/48/000000/wheat.png">
      </div>
      <div class="col s9">
        <span class="card-title">Ragi</span>
      </div>
      </div>
    </div>
  </a>
  </div>
   <div class="col s3">
    <a href="http://localhost:5000/commodity/maize" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/corn.png">
      </div>
      <div class="col s9">
        <span class="card-title">Maize</span>
      </div>
      </div>
    </div>
  </a>
  </div> <div class="col s3">
    <a href="http://localhost:5000/commodity/moong" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/paper-bag-with-seeds.png">
      </div>
      <div class="col s9">
        <span class="card-title">Moong</span>
      </div>
      </div>
    </div>
  </a>
  </div> <div class="col s3">
    <a href="http://localhost:5000/commodity/masoor" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/paper-bag-with-seeds.png">
      </div>
      <div class="col s9">
```

```
        <span class="card-title">Masoor</span>
      </div>
      </div>
    </div>
  </a>
  </div>
</div>
  <div class="row">
  <div class="col s3">
    <a href="http://localhost:5000/commodity/urad" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/paper-bag-with-seeds.png">
      </div>
      <div class="col s9">
        <span class="card-title">Urad</span>
      </div>
      </div>
    </div>
  </a>
  </div>
   <div class="col s3">
    <a href="http://localhost:5000/commodity/jute" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/potato.png">
      </div>
      <div class="col s9">
        <span class="card-title">Raw Jute</span>
      </div>
      </div>
    </div>
  </a>
  </div> <div class="col s3">
    <a href="http://localhost:5000/commodity/niger" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/paper-bag-with-seeds.png">
      </div>
      <div class="col s9">
        <span class="card-title">Niger Seed</span>
      </div>
      </div>
    </div>
  </a>
  </div> <div class="col s3">
    <a href="http://localhost:5000/commodity/safflower" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
```

```
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/paper-bag-with-seeds.png">
      </div>
      <div class="col s9">
        <span class="card-title">Kardi Seed</span>
      </div>
      </div>
    </div>
  </a>
  </div>
</div>
  <div class="row">
  <div class="col s3">
    <a href="http://localhost:5000/commodity/sunflower" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/potato.png">
      </div>
      <div class="col s9">
        <span class="card-title">Sunflower</span>
      </div>
      </div>
    </div>
  </a>
  </div>
   <div class="col s3">
    <a href="http://localhost:5000/commodity/jowar" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/potato.png">
      </div>
      <div class="col s9">
        <span class="card-title">Jowar</span>
      </div>
      </div>
    </div>
  </a>
  </div> <div class="col s3">
    <a href="http://localhost:5000/commodity/bajra" style="color: #000000">
    <div class="card grey lighten-4">
      <div class="card-content row valign-wrapper">
        <div class="col s3">
        <img src="https://img.icons8.com/color/48/000000/potato.png">
      </div>
      <div class="col s9">
        <span class="card-title">Bajra</span>
      </div>
      </div>
    </div>
  </a>
```

```
    </div>
  </div>

<script type="text/javascript" charset="utf-8">
  $(document).ready(function(){
    $('.slider').slider({full_width :true});
  });
</script>
  </div>
</body>
<script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.j
s"></script>

</html>
```