

CROP YIELD PREDICTION USING MACHINE LEARNING ALGORITHM

Submitted in partial fulfillment of the requirements for the
award of Bachelor of Engineering Degree in Computer
Science and Engineering

By

RITHWIK REDDY NATHI

(38110462)

VEMIREDDY JASWANT SAI

(38110629)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI – 600119, TAMIL NADU**

April 2022



SATHYABAMA

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

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

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Rithwik Reddy Nathi (38110462)** and **Vemireddy Jaswant Sai (38110629)** who carried out the project entitled "**CROP YIELD PREDICTION USING MACHINE LEARNING TECHNIQUES**" under my supervision from November 2021 to April 2022.

Internal Guide

Dr. K. ASHOK KUMAR

Head of the Department

Dr. L. LAKSHMANAN M.E., Ph.D.

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

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I Rithwik Reddy Nathi (38110462) & Vemireddy Jaswant Sai (38110629)

hereby declare that the Project Report entitled "**CROP YIELD PREDICTION USING MACHINE LEARNING TECHNIQUES**" done by me under the guidance of **Dr. K. Ashok kumar**(Internal) at **Sathyabama Institute of Science and Technology, CHENNAI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

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

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. K. Ashok kumar** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

No.	Title	Page No.
1	Abstract	6
2	Introduction	7
2.1	Objective	8
3	Literature Survey	9-15
3.1	Existing System	16
3.2	Proposed System	17
4	Aim and Scope of present investigation	18
5	Experimental or materials and methods algorithms used	19-38
6	Results and Discussions	39
6.1	Screenshots	39-41
7	Conclusion	42
8	References	43
9	Source Code	44

1. ABSTRACT

Crop yield production value updation has a positive practical significance for guiding agricultural production and for notifying the change in market rate of crop to the farmer. The concept of this paper is to implement the crop selection method so that this method helps in solving many agriculture and farmers problems. This improves our Indian economy by maximizing the yield rate of crop production. Different types of land condition. So the quality of the crops are identified using ranking process. By this process the rate of the low quality and high quality crop is also notified. The usage of ensemble of classifiers paves a path way to make a better decision on predictions due to the usage of multiple classifiers. Further, a ranking process is applied for decision making in order to select the classifiers results. This system is used to predict the cost of crop which is yielded for further.

2. INTRODUCTION

In the world of developing technologies, the success of sharing information will help the agriculturists in realizing and developing their potential. The information sharing is that the valuable and timely information is being shared between agriculturists, either formally or informally. The willingness of information sharing refers to the open attitude among agriculturists. This open attitude determines the degree and scope of information sharing. Using web-technologies like html and css we build the web application, We create dataset by gathering data from multiple resources and place them in place which is used to predict the price of the crop and results are subjected to non-linear test later priorities are set and rankings are given to the list of crops. Place information in our application and share that information to agriculturists whose data is collected and stored in the mysql server. we software to automatically send the updated information to the agriculturists in the form of text message.so that agriculturists no need to go to near by towns and cities to know the updated information. We will be machine learning algorithms to predict the price of the crop for the next two months. For prediction purpose we will be using Support vector machine(SVM), Naïve Baye's (NB) and K-Nearest Neighbour(KNN) algorithms to predict the cost of the crop production. Further, a ranking process is applied for decision making in order to select the classifiers results.

2.1 OBJECTIVE

- Data set collection from various sources.
- Data parsing and cleansing technique is applied to make the raw data into processing data.
- The data collected is subject to machine learning system along with run time analysis makes an efficient crop value updation system.
- Usage of Ensemble of classifiers makes the model more robust and efficient.
- Ranking technique used in the project helps us to make efficient decisions.
- Creating a web application for user registrations and collection of data.

3. LITERATURE SURVEY

[1]Title : A Review on Data Mining Techniques for Fertilizer Recommendation,2018

Authors : Jignasha M. Jethva, Nikhil Gondaliya, Vinita Shah

To keep up nutrition levels in the soil in case of deficiency, fertilizers are added to soil. The standard issue existing among the Indian agriculturists choose approximate amount of fertilizers and add them manually. Excess or deficient extension of fertilizers can harm the plants life and reduce the yield. This paper gives overview of various data mining frameworks used on cultivating soil dataset for fertilizer recommendation.

[2]Title : A Survey on Data Mining Techniques in Agriculture,2015

Authors : M.C.S.Geetha

Agriculture is the most critical application area especially in the developing nations like India .Use of information technology in agriculture can change the situation of decision making and farmers can yield in better way. Data mining plays a important role in decision making on several aspects with agriculture field. It examines about role of data mining in the farming field and their related work by a few authors in related to agriculture domain. It additionally talks about on various data mining applications in taking care of the several agriculture problems. This paper integrates the work of several authors in a single place so it is valuable for specialists to get data of current situation of data mining systems and applications in context to farming field.

[3]Title : AgroNutri Android Application,2016

Authors : S. Srija, R. Geetha Chanda, S.Lavanya, Dr. M. Kalpana Ph.D

This paper communicates the idea regarding the making of AgroNutri an android application that helps in conveying the harvest particular fertilizer amount to be applied. The idea is to calculate the measure of NPK composts to be applied depend on the blanked proposal of the crop of interest. This application works depend on the product chosen by the farmer and that is taken as input, thus providing the farmers. The future scope of the AgroNutri is that GPRS can be included so that according to location nutrients are suggested. Further this application would be incorporated as a piece of the accuracy agriculture wherein sensors can be utilized to discover the measure of NPK present in the dirt and that sum can be deducted from the suggestion and giving us the exact measure of supplements to be added.

[4]Title : Machine Learning: Applications in Indian Agriculture,2016

Authors : Karandeep Kaur

Agriculture is a field that has been lacking from adaption of technologies and their advancements. Indian agriculturists should be up to the mark with the universal procedures. Machine learning is a native concept that can be applied to every field on all inputs and outputs. It has effectively settled its ability over ordinary calculations of software engineering and measurements. Machine learning calculations have improved the exactness of artificial intelligence machines including sensor based frameworks utilized in accuracy farming. This paper has evaluated the different uses of machine learning in the farming area. It additionally gives a knowledge into the inconveniences looked by Indian farmers and how they can be resolved using these procedures.

[5]Title : Impacts of population growth, economic development, and technical change on global food production and consumption,2011

Author: Uwe A. Schneider a,†, Petr Havlik b, Erwin Schmid c, Hugo Valin b, Aline Mosnier b,c, Michael Obersteiner b, Hannes Bottcher b, Rastislav Skalsky´ d, Juraj Balkovicˇ d, Timm Sauer a, Steffen Fritz b

Throughout the following decades humanity will request more food from less land and water assets. This investigation evaluates the food production effects of four elective advancement situations from the Millennium Ecosystem Assessment and the Special Report on Emission Scenarios. partially and jointly considered are land and water supply impacts from population development, and specialized change, and forests and agriculture demand request shifts from population development and economic improvement. The income impacts on nourishment request are registered with dynamic flexibilities. Worldwide farming area increments by up to 14% somewhere in the range of 2010 and 2030. Deforestation restrictions strongly impact the price of land and water resources but have little consequences for the global level of food production and food prices. While projected income changes have the highest partial impact on per capita food consumption levels, population growth leads to the highest increase in total food production. The impact of technical change is amplified or mitigated by adaptations of land management intensities

[6]Title : Brief history of agricultural systems modelling,2016

Author: James W. Jones a,*, John M. Antle b, Bruno O. Basso c, Kenneth J. Boote a, Richard T. Conant d, Ian Foster e, H. Charles J. Godfray f, Mario Herrero g, Richard E. Howitt h, Sander Jansseni, Brian A. Keating g, Rafael Munoz-Carpena a, Cheryl H. Porter a, Cynthia Rosenzweig j, Tim R.Wheeler k

Rural frameworks science creates information that enables analysts to consider complex issues or take educated farming choices. The rich history of this science represents the decent variety of frameworks and scales over which they work and have been contemplated. Demonstrating, a basic apparatus in agrarian

frameworks science, has been expert by researchers from an extensive variety of controls, who have contributed ideas and instruments over six decades. As agrarian researchers currently consider the "people to come" models, information, and learning items expected to meet the inexorably mind boggling frameworks issues looked by society, it is vital to check out this history and its exercises to guarantee that we stay away from re-innovation and endeavor to think about all elements of related difficulties. To this end, we outline here the historical backdrop of rural frameworks demonstrating and distinguish exercises discovered that can help control the structure and advancement of up and coming age of farming framework apparatuses and techniques. Various past occasions joined with generally innovative advancement in different fields have unequivocally added to the development of farming framework demonstrating, including improvement of process-based bio-physical models of yields and domesticated animals, factual models dependent on verifiable perceptions, and financial streamlining and reproduction models at family unit and local to worldwide scales. Attributes of rural frameworks models have changed broadly relying upon the frameworks included, their scales, and the extensive variety of purposes that spurred their advancement and use by specialists in various controls. Late patterns in more extensive joint effort crosswise over establishments, crosswise over orders, and between people in general and private segments recommend that the stage is set for the significant advances in rural frameworks science that are required for the up and coming age of models, databases, learning items and choice emotionally supportive networks. The exercises from history ought to be considered to help stay away from barricades and entanglements as the network builds up this up and coming age of horticultural frameworks models.

[7]Title : A Smart Agricultural Model by Integrating IoT, Mobile and Cloud-based Big Data Analytics,2017

Authors : S.Rajeswari, K.Suthendran, K.Rajkumar.

In the cultivating field, the system models play a significant role to the enhancement of the agro-normal and money related conditions. In the proportions of benefits of the field and farm examinations to give the information and to

recognize fitting and fruitful organization practices. It can recognize the organization to arrive managers and transversely over reality as long as the required soil, the board, environment, and money related information. Decision Support Systems (DSSs) use to make the information for the vermin the board, develop the officials. These systems are not using the impelled strategies to process the data. Thusly, use the adroit system thoughts to take the decisions for the issue. It expect a crucial activity in the comprehension of agronomic results, and their use as decision sincerely steady systems for farmers is extending.

[8]Title : An Overview of Internet of Things and Data Analytics in Agriculture: Benefits and Challenges,2018

Authors : Olakunle Elijah, Tharek Abdul Rahman, Igbafe Orikumhi, Chee Yen Leow, Nour Hindia.

A blueprint of lot and DA in agriculture has been shown in this paper. A couple of zones related to the association of lot in agribusiness have been discussed in detail. The investigation of composing exhibits that there are clusters of work advancing being produced of lot development that can be used to increase operational efficiency and gainfulness of plant and creatures. The benefits of lot and DA, and open troubles have been identified and inspected in this paper. lot is depended upon to offer a couple of benefits to the agribusiness division. Regardless, there are up 'til now different issues to be steered to make it moderate for close to nothing and medium-scale farmers. The key issues are security and cost. It is typical that as contention increases in the cultivating part

[9]Title : Circulation Mode Selection Based on Cost Analysis,2017

Authors: Xiurong Sun*, Jingshan Zhang, Chenglin Wang, Tao Zhang

If every farmer and each average production base will join their optimal conditions in making cooperatives, it will accomplish economies of scale. Furthermore, producers will have an all the more favourable position in the plans with downstream firms (shipper or retailer).Second, the main customers of wholesale market are not inhabitants nearby who buy small quantities products but lower distributors or retailers. More redesigned transportation mode respects intensive attempt of new agrarian things, which prompts bolster the movement of new chain

joint logistics and strengthen resource utilize and made logistics advantage quality. Refresh everything considered agrarian things spread. By then, regard the examination of gigantic worth control of standard things and achieve the mind blowing control to stream process.

[10]Title: Support Vector Machine-based Fuzzy Self-learning Control for Induction Machines,2010

Authors : Zongkai Shao

Using support vector machine (SVM) is to realize the self learning of fuzzy inference system (FIS), based on a fast modified varying metric method (MDFP) and a support vector machine identifier (SVMI), a SVM-FIS self-learning controller for the threephase induction machine adjustable speed system has been designed. The proposed controller is not only of the advantages that FIS does not depend on the plant model, strong robustness, and adaptive self-learning ability, but also learning ability and generalization performance of SVM. The designed processes of SVM-FIS, MDFP, and SVMI algorithms have been described in details. Simulation results show the feasibility, correctness and effectiveness of the proposed control strategy, such as the excellent static and dynamic performances, and strong anti-interference ability.

[11]Title : Machine Learning Facilitated Rice Prediction in Bangladesh,2015

Authors: Mohammad Motiur Rahman, Naheena Haq, Rashedur M Rahman

In this examination, self organising map (SOM) was utilized to group the information relationship between the information factors. After that chi-square test strategy was utilized to set up the level of reliance between the related variable qualities. It was discovered that the day by day outrageous climate conditions, for example, most extreme and least fluctuation in temperature, precipitation, dampness and wind speed were the principle drivers of product development, yield and wine quality

[12]Title : Support Vector Machine-Based Classification Scheme of Maize Crop,2017

Authors : Suhas S Athani, CH Tejeshwar

This paper says about, advancement of a mechanized framework to distinguish and group weeds from the products would be of extraordinary help and we have proposed a set-up that decreases labour. We have considered pictures of maize edits as the informational index. Separating surface highlights of the picture and applying SVM (support vector machine) to arrange whether the given picture is a weed or a yield brought about a precision of 82%. In this way, picture preparing is a proficient method to group the given picture about whether it is a weed or a yield. A similar classifier can be connected to recognize number of harvests like groundnut, paddy from weeds. The proposed framework gives a chance to investigate more about element extraction methods. Further research and usage may incorporate building up a mechanized equipment framework which could help in removal of weeds in the fields.

[13] WITH MACHINE LEARNING ALGORITHMS FOR ESTIMATING WINTER WHEAT AREAS,2017

Authors : Y.Z. Pan²

we utilize different kernel functions in the CPPI models to depict the connection between fractional winter wheat area and MODIS EVI time series data. We tried three straight and non-direct kernel functions, including linear regression, artificial neural system, and support vector machine. The differences when utilizing various kernel functions are minor for areas with basic planting structure. For areas like DT where multiple crop types have comparative phenology cycles, ANN-CPPI is found to play out the best. The two crop types to be specific winter wheat and rapeseed, can be separated well. These tests give elective answers for the uses of CPPI in mixed areas.

3.1 EXISTING SYSTEM

The computational and data demands of structural price forecasting generally far exceed than what is routinely available in developing countries. Consequently, researchers often rely on parsimonious representations of price processes for their forecasting needs. Contemporary parsimonious form of price forecasting relies heavily on time series modelling. In time series modelling, past observations of the same variable are collected and analyzed to develop a model describing the underlying relationship. During the past few decades, much effort has been devoted to the development and improvement of time series forecasting models. Time series modelling requires less onerous data input for regular and up-to date price forecasting. Hence there is a need for better classification which would be an ensemble or hybrid classification model.

DISADVANTAGES OF EXISTING SYSTEM

- Efficiency is low.
- More number of repeated work.

3.2 PROPOSED SYSTEM

In proposed system, the data analysis technology is used to update the crop yield rate change. The concept of this paper is to implement the crop selection method so that this method helps in solving many agriculture and farmers problems. This improves our Indian economy by maximizing the yield rate of crop production. Different types of land condition. So the quality of the crops are identified using ranking process. By this process the rate of the low quality and high quality crop is also intimated. The usage of ensemble of classifiers paves a path way to make a better decision on predictions due to the usage of multiple classifiers. Further, a ranking process is applied for decision making in order to select the classifiers results. This system is used to predict the cost of the fertilizers for further. This project uses Ensemble of classifiers such as SVM, NAÏVE BAYES, KNN or hybrid classifier. In addition, this project uses Ranking technique.

ADVANTAGES OF PROPOSED SYSTEM

- Useful to people far away from towns/cities.
- Better time efficiency.
- Reduction of repeated work.

4. AIM AND SCOPE OF THE PRESENT INVESTIGATION

4.1 Aim

Our Aim from the project is to make a ML model which takes student data trains itself using various data mining techniques and Algorithms(KNN ,Naïve Bytes and SVM) and predict the price of crops in virtual environment by considering the overall factors that contribute in his overall price.

Secondly, to learn the required techstacks and use it to make model with an python application and lastly to execute it get output about price of cropla.

4.2 Scope:

This Project can be used to get the student performance with more accuracy than any other model published earlier and we can also make some mobile or web application based on the model.

5.EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

5.1 K-NEAREST NEIGHBOR(KNN) CLASSIFICATION METHOD

K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms. The neighbors are taken from a set of objects for which the class (for k -NN classification) or the object property value (for k -NN regression) is known.

STEP 1: BEGIN

STEP 2: Input: $D = \{(x_1, c_1), \dots, (x_N, c_N)\}$

STEP 3: $x = (x_1, \dots, x_n)$ new instance to be classified

STEP 4: FOR each labelled instance (x_i, c_i) calculate d

(x_i, x) STEP 5: Order $d(x_i, x)$ from lowest to highest, $(i = 1, \dots$

$.N)$ STEP 6: Select the K nearest instances to x : D_x^k

STEP 7: Assign to x the most frequent

class STEP 8: END

5.2 NAIVE BAYES ALGORITHM

- It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.
- Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

How Naive Bayes algorithm works?

- Step 1: Convert the data set into a frequency table
- Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.
- Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

5.3 SVM (support vector machine):

- For a dataset consisting of features set and labels set, an SVM classifier builds a model to predict classes for new examples. It assigns new example/data points to one of the classes. If there are only 2 classes then it can be called as a Binary SVM Classifier.
- There are 2 kinds of SVM classifiers:
- Linear SVM Classifier
- Non-Linear SVM Classifier

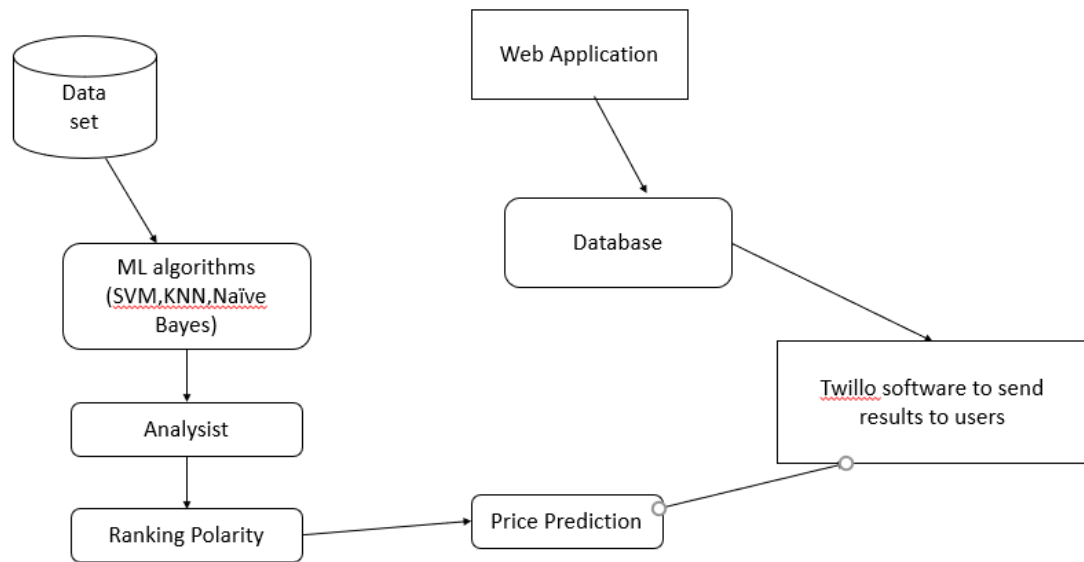
SVM Linear Classifier:

- In the linear classifier model, we assumed that training examples plotted in space. These data points are expected to be separated by an apparent gap. It predicts a straight hyperplane dividing 2 classes. The primary focus while drawing the hyperplane is on maximizing the distance from hyperplane to the nearest data point of either class. The drawn hyperplane called as a maximum-margin hyperplane.

SVM Non-Linear Classifier:

- In the real world, our dataset is generally dispersed up to some extent. To solve this problem separation of data into different classes on the basis of a straight linear hyperplane can't be considered a good choice. For this Vapnik suggested creating Non-Linear Classifiers by applying the kernel trick to maximum-margin hyperplanes. In Non-Linear SVM Classification, data points plotted in a higher dimensional space.

5.4 SYSTEM ARCHITECTURE



5.5 SYSTEM REQUIREMENTS

5.5.1 HARDWARE REQUIREMENTS:

- System - Pentium-IV
- Speed - 2.4GHZ
- Hard disk - 40GB
- Monitor - 15VGA color
- RAM - 512MB

4.5.2 SOFTWARE REQUIREMENTS:

- Operating System - Windows XP
- Coding language - Python
- IDE - Flask web app

5.6 SOFTWARE ENVIRONMENT

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.

- It supports automatic garbage collection.

5.7 MODULES

- Data Collection
- Data Preparation
- Analyze and Prediction
- Accuracy on test set

MODULES DESCRIPTION

Data Collection:

This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform.

There are several techniques to collect the data, like web scraping, manual interventions and etc. The dataset used in this crop yield prediction in India taken from some other source

Data Preparation:

Wrangle data and prepare it for training. Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.)

Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data. Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis. Split into training and evaluation sets

Analyze and Prediction

Predict crop value

In this module the crop value is predicted by applying machine learning algorithms to the collected and train data. So that we can know the crop min max value of the crop at any particular area i.e based on the input.

Accuracy on test set:

We got a accuracy of 96.7% on test set.

5.8 UML DIAGRAMS

UML is simply another graphical representation of a common semantic model. UML provides a comprehensive notation for the full lifecycle of object-oriented development.

ADVANTAGES

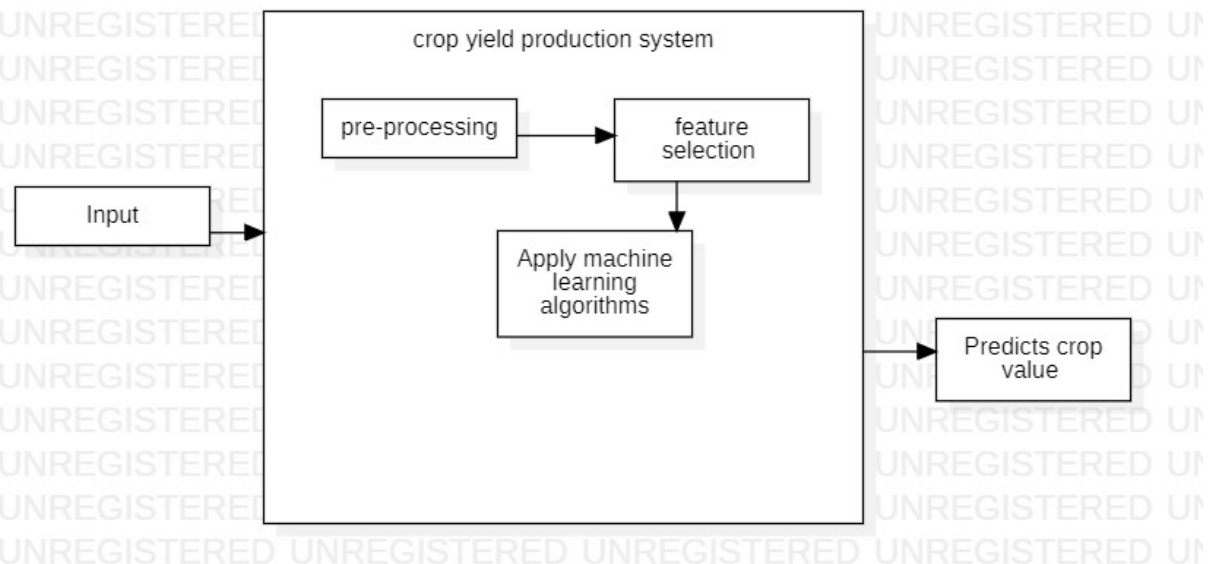
- To represent complete systems (instead of only the software portion) using object oriented concepts
- To establish an explicit coupling between concepts and executable code
- To take into account the scaling factors that are inherent to complex and critical systems
- To creating a modeling language usable by both humans and machines

UML defines several models for representing systems

- The class model captures the static structure
- The state model expresses the dynamic behavior of objects
- The use case model describes the requirements of the user
- The interaction model represents the scenarios and messages flows
- The implementation model shows the work units
- The deployment model provides details that pertain to process allocation

5.8.1 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



5.8.2 UML DIAGRAM

UML is simply another graphical representation of a common semantic model. UML provides a comprehensive notation for the full lifecycle of object-oriented development.

ADVANTAGES

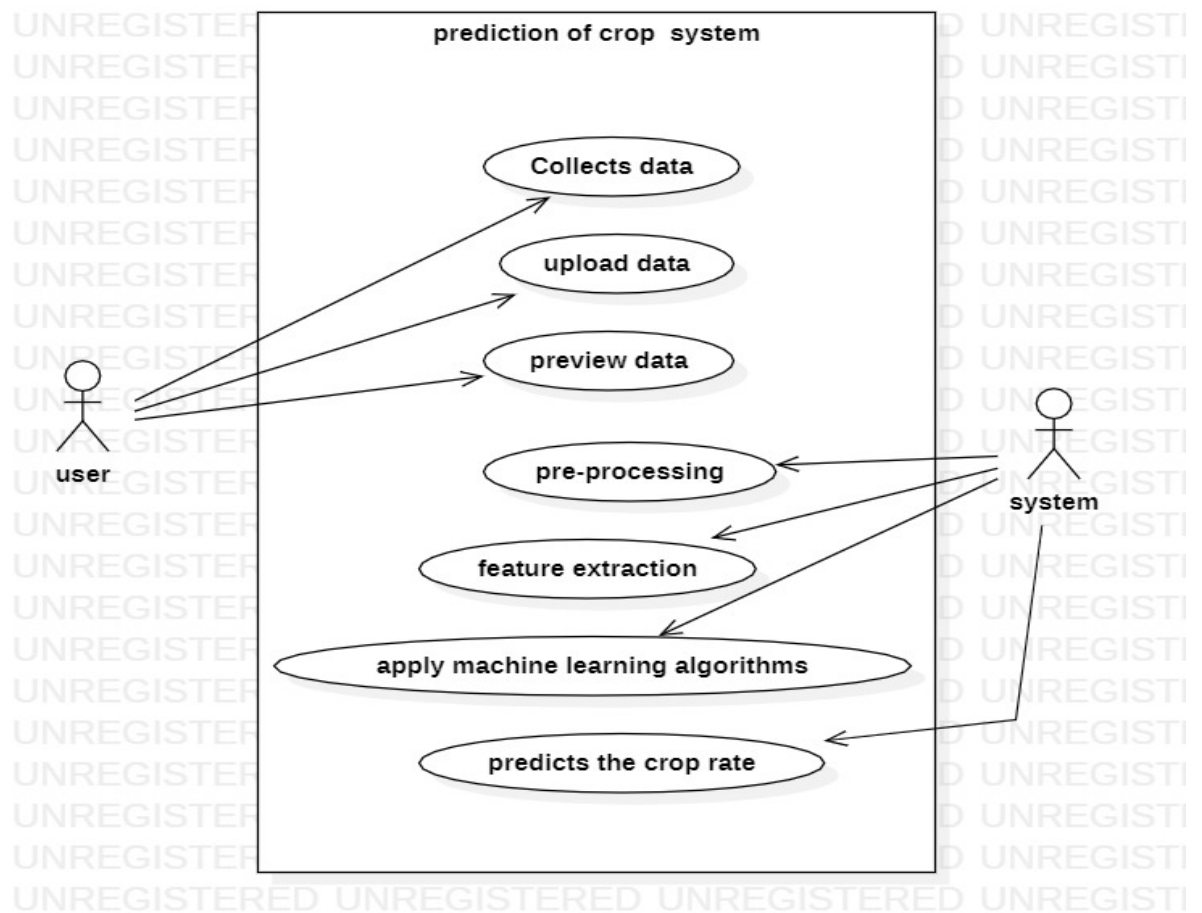
- To represent complete systems (instead of only the software portion) using object oriented concepts
- To establish an explicit coupling between concepts and executable code
- To take into account the scaling factors that are inherent to complex and critical systems
- To creating a modeling language usable by both humans and machines

UML defines several models for representing systems

- The class model captures the static structure
- The state model expresses the dynamic behavior of objects
- The use case model describes the requirements of the user
- The interaction model represents the scenarios and messages flows
- The implementation model shows the work units
- The deployment model provides details that pertain to process allocation

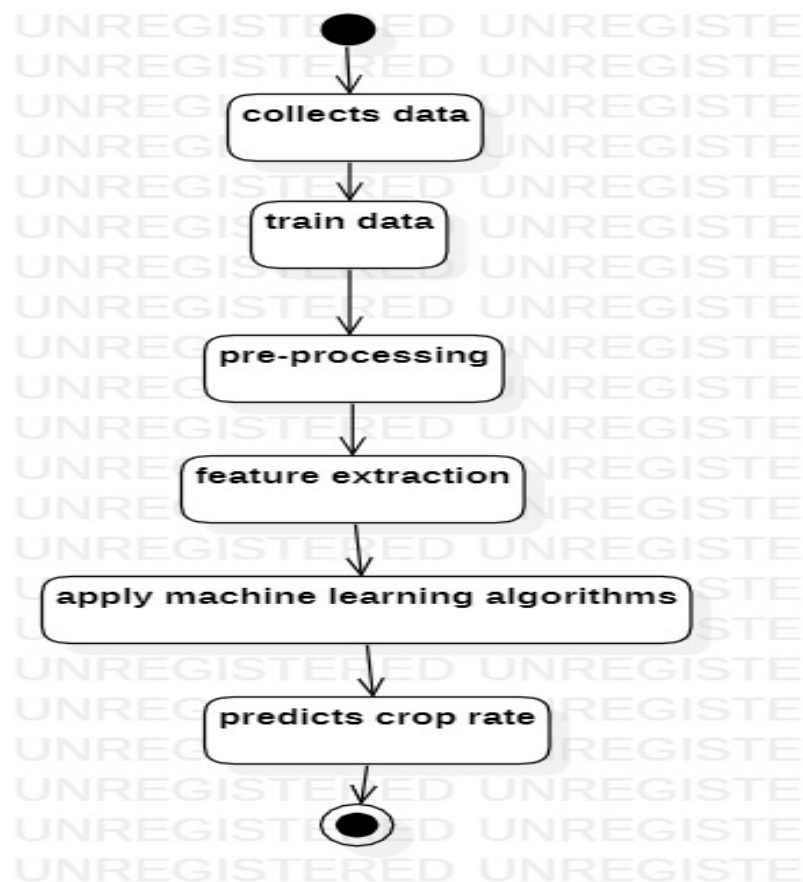
5.8.3 USECASE DIAGRAM

Use case diagrams overview the usage requirement for system. They are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of action that provides something of measurable value to an action and is drawn as a horizontal ellipse.



5.8.4 ACTIVITY DIAGRAM

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between.



5.9 INPUT DESIGN AND OUTPUT DESIGN

5.9.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that

the user will not be in a maze of instant. Thus the objective of input design is to create an input layout that is easy to follow

5.9.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displayed for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analyzing design computer output, they should identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

5.10 SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion

of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business

process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.

- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

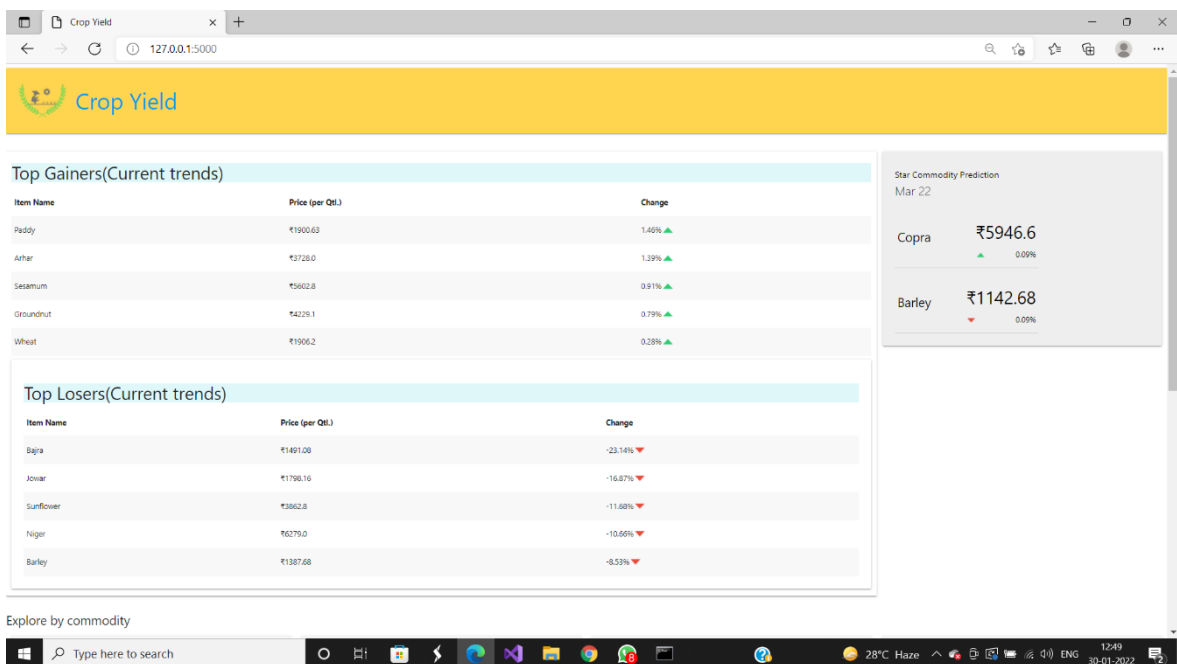
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6 RESULTS AND DISCUSSIONS

In this project the data analysis technology is used to update the rate change through notification. Further, a ranking process is applied for decision making in order to select the classifiers results. This system is used to predict the cost of the crop yield.

6.1 SCREENSHOTS



Crop Yield

127.0.0.1-5000

Sunflower ₹1362.8 -11.80%

Niger ₹6279.0 -10.66%

Barley ₹1387.68 -8.53%

Explore by commodity

Paddy

Wheat

Barley

Soya Bean

Cotton

Coconut

Ground Nut Seeds

Mustard Seed

Gingelly Seed(Sesamum)

Gram

Sugarcane

Arhar

Ragi

Maize

Moong

Masoor

Urad

Raw Jute

Niger Seed

Kardi Seed

Sunflower

Jowar

Bajra

Type here to search
28°C Haze
12:50 30-01-2022

Crop Yield

localhost:5000/commodity/wheat

Crop Yield

wheat



Current Price ₹ 1572.75 / qtl

Prime Location U.P, Punjab, Haryana, Rajasthan, M.P, bihar

Crop Type rabi

Export Sri Lanka, United Arab Emirates, Taiwan

Brief Forecast

Min. crop price time Oct 22 ₹1476.9

Max. crop price time Dec 22 ₹1672.65

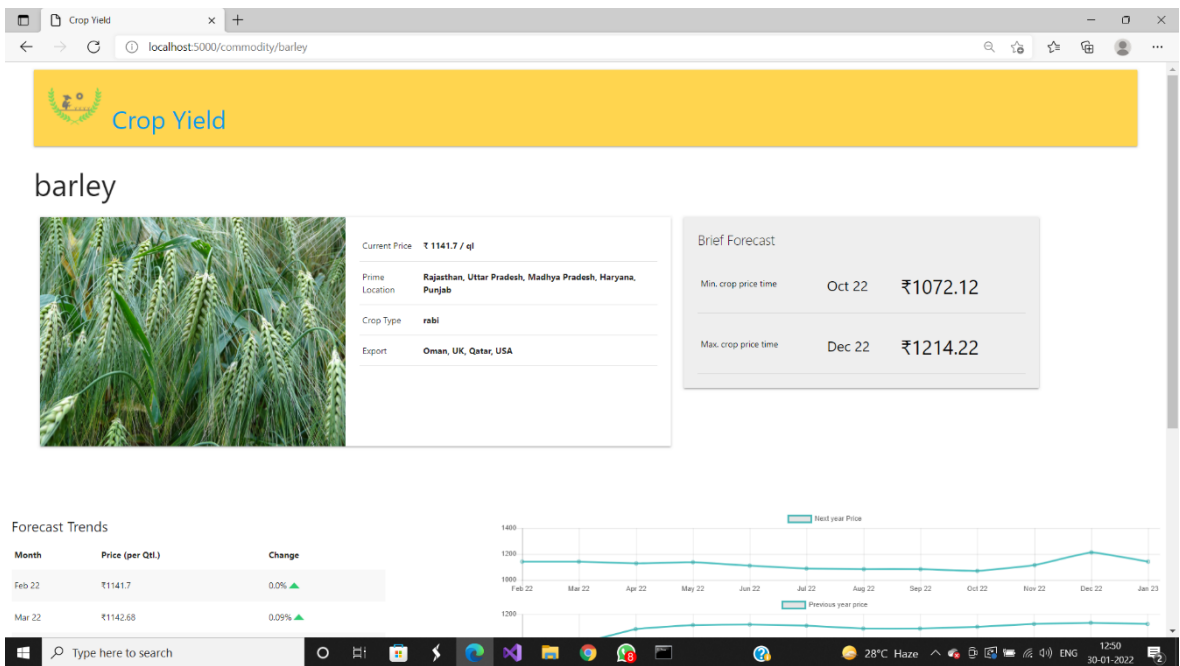
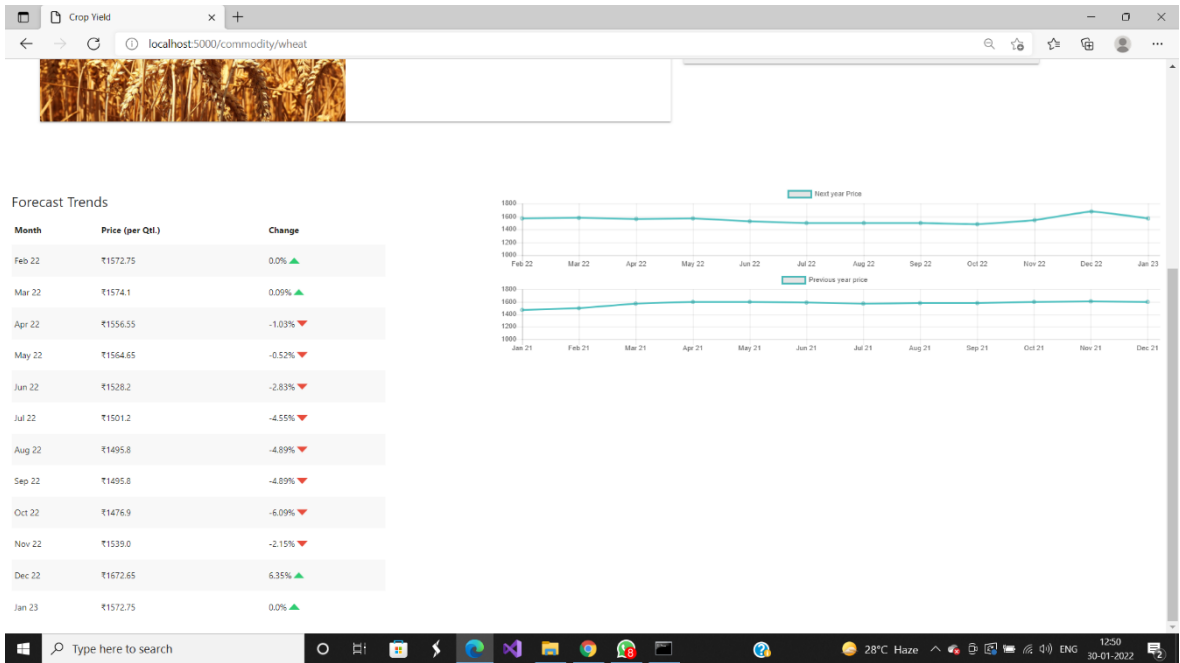
Forecast Trends

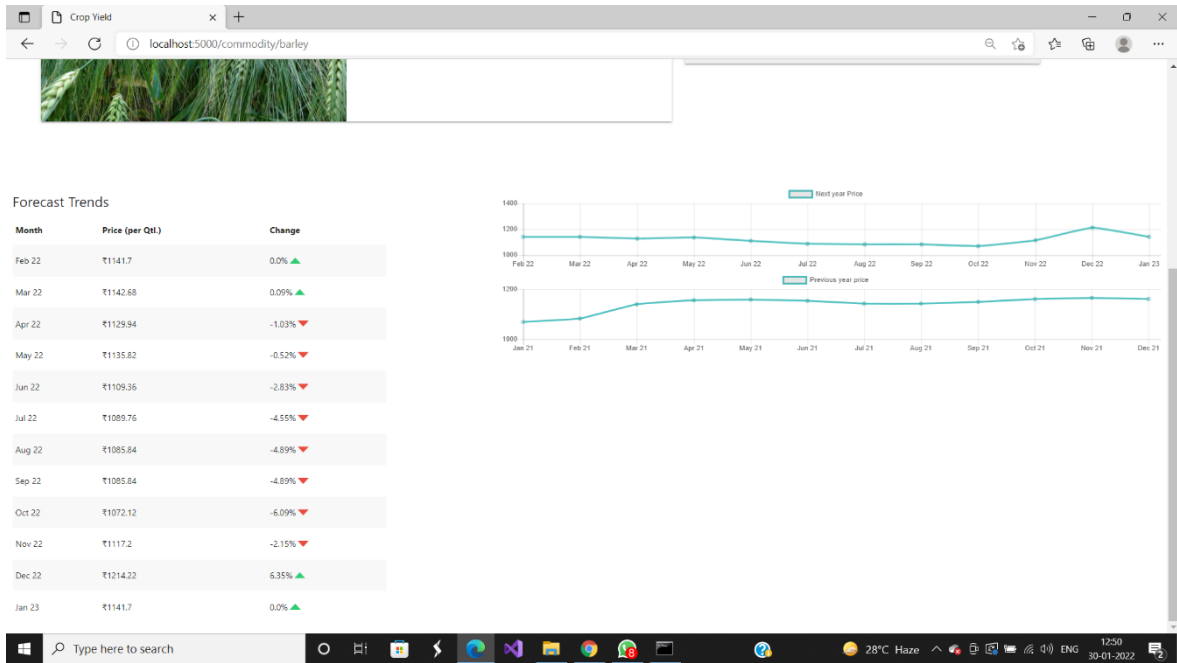
Month	Price (per Qtl.)	Change
Feb 22	₹1572.75	0.0% ▲
Mar 22	₹1574.1	0.09% ▲

Next year Price

Previous year price

Type here to search
28°C Haze
12:50 30-01-2022





7. CONCLUSION

This open attitude determines the degree and scope of information sharing. Big data analysis technology can effectively improve the crop yield production is updatation. This project proposes a novel intelligent system for agricultural crop price prediction. The key idea is to use ensemble of classifiers for prediction. The usage of ensemble of classifiers paves a path way to make a better decision on predictions due to the usage of multiple classifiers. Further, a ranking process is applied for decision making in order to select the classifiers results. This system is used to predict the cost of the crop rate for further.

8. REFERENCES

- [1] Manpreet Kaur, Heena Gulati, Harish Kundra, "Data Mining in Agriculture on Crop Price Prediction: Techniques and Applications", *International Journal of Computer Applications*, Volume 99– No.12, August 2014.
- [2] J. Meng, "Research on the cost of agricultural products circulation and its control under the new normal economic development," *Commercial Times*, no. 23, pp. 145147, 2016.
- [3] A. Kaloxylos et al., "Farm management systems and the future Internet era," *Comput. Electron. Agricult.*, vol. 89, pp. 130–144, Nov. 2012.
- [4] N. N. Li, T. S. Li, Z. S. Yu, Y. Rui, Y. Y. Miao, and Y. S. Li, "Factors influencing farmers' adoption of new technology based on Logistic-ISM model-a case study of

potato planting technology in Dingxi City, Gansu Province," Progress in Geography, vol. 33, no. 4, pp. 542-551, 2014.

[5] Y. Wang, "A neural network adaptive control based on rapid learning method and its application," Advances In Modeling and Analysis, Vol. 46(3), pp. 27-34,1994.

9. SOURCE CODE

APP.PY

```
# -*- coding: utf-8 -*-
```

```
"""
```

Created on Sat Mar 2 21:46:27 2019

@author: PRATYUSH, Rahul, Somya, Abhay

"""

```
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_state=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[1  
en(month1)-1][0],month1[len(month1)-1][1],month1[0][2],month1[0][0],mon  
th1[0][1]])
```

```
crop_month_wise.append([month2[0][3],month2[len(month2)-1][2],month2[1
```

```
en(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[current_month + i - 1]))

    else:

        month_with_year.append((current_month + i - 12, current_year + 1,
annual_rainfall[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(change[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[current_month + i - 1]))

```

else:

```
        month_with_year.append((current_month + i - 12, current_year + 1,  
annual_rainfall[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(change[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[current_month - i - 1]))
```

```
        else:
```

```
            month_with_year.append((current_month - i + 12, current_year - 1,
annual_rainfall[current_month - i + 11]))
```

```

for m, y, r in month_with_year:

    current_predict = commodity.getPredictedValue([float(m), 2013, r])

    wpris.append(current_predict)


for i in range(0, len(wpris)):

    m, y, r = month_with_year[i]

    x = datetime(y,m,1)

    x = x.strftime("%b %y")

    crop_price.append([x, round((wpris[i]* base[name.capitalize()]) / 100,
2)])

# print("previous ", wpris)

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()
```

CROPS.PY

```
def crop(crop_name):
```

```
    crop_data = {
```

```
        "wheat":["/static/images/wheat.jpg", "U.P., Punjab, Haryana, Rajasthan,  
M.P., bihar", "rabi","Sri Lanka, United Arab Emirates, Taiwan"],
```

```
        "paddy":["/static/images/paddy.jpg", "W.B., U.P., Andhra Pradesh, Punjab,  
T.N.", "kharif","Bangladesh, Saudi Arabia, Iran"],
```

```
        "barley":["/static/images/barley.jpg", "Rajasthan, Uttar Pradesh, Madhya  
Pradesh, Haryana, Punjab", "rabi","Oman, UK, Qatar, USA"],
```

```
        "maize":["/static/images/maize.jpg", "Karnataka, Andhra Pradesh, Tamil  
Nadu, Rajasthan, Maharashtra", "kharif", "Hong Kong, United Arab Emirates,  
France"],
```

```
        "bajra":["/static/images/bajra.jpg", "Rajasthan, Maharashtra, Haryana, Uttar  
Pradesh and Gujarat", "kharif", "Oman, Saudi Arabia, Israel, Japan"],
```

```
        "copra":["/static/images/copra.jpg", "Kerala, Tamil Nadu, Karnataka,  
Andhra Pradesh, Orissa, West Bengal","rabi", "Veitnam, Bangladesh, Iran,  
Malaysia"],
```

```
        "cotton":["/static/images/cotton.jpg", "Punjab, Haryana, Maharashtra,  
Tamil Nadu, Madhya Pradesh, Gujarat", " China, Bangladesh, Egypt"],
```

```
        "masoor":["/static/images/masoor.jpg", "Uttar Pradesh, Madhya Pradesh,  
Bihar, West Bengal, Rajasthan", "rabi", "Pakistan, Cyprus,United Arab  
Emirates"],
```

"gram":["/static/images/gram.jpg", "Madhya Pradesh, Maharashtra, Rajasthan, Uttar Pradesh, Andhra Pradesh & Karnataka", "rabi", "Veitnam, Spain, Myanmar"],

"groundnut":["/static/images/groundnut.jpg", "Andhra Pradesh, Gujarat, Tamil Nadu, Karnataka, and Maharashtra", "kharif", "Indonesia, Jordan, Iraq"],

"arhar":["/static/images/arhar.jpg", "Maharashtra, Karnataka, Madhya Pradesh and Andhra Pradesh", "kharif", "United Arab Emirates, USA, Chicago"],

"sesamum":["/static/images/sesamum.jpg", "Maharashtra, Rajasthan, West Bengal, Andhra Pradesh, Gujarat", "rabi", "Iraq, South Africa, USA, Netherlands"],

"jowar":["/static/images/jowar.jpg", "Maharashtra, Karnataka, Andhra Pradesh, Madhya Pradesh, Gujarat", "kharif", "Torronto, Sydney, New York"],

"moong":["/static/images/moong.jpg", "Rajasthan, Maharashtra, Andhra Pradesh", "rabi", "Qatar, United States, Canada"],

"niger":["/static/images/niger.jpg", "Andha Pradesh, Assam, Chattisgarh, Gujarat, Jharkhand", "kharif", "United States of American,Argenyina, Belgium"],

"rape":["/static/images/rape.jpg", "Rajasthan, Uttar Pradesh, Haryana, Madhya Pradesh, and Gujarat", "rabi", "Veitnam, Malaysia, Taiwan"],

"jute":["/static/images/jute.jpg", " West Bengal , Assam , Orissa , Bihar , Uttar Pradesh", "kharif", "Jordan, United Arab Emirates, Taiwan"],

"safflower":["/static/images/safflower.jpg", "Maharashtra, Karnataka, Andhra Pradesh, Madhya Pradesh, Orissa", "kharif", " Philippines, Taiwan, Portugal"],

```
"soyabean":["/static/images/soyabean.jpg", "Madhya Pradesh, Maharashtra, Rajasthan, Madhya Pradesh and Maharashtra", "kharif", "Spain, Thailand, Singapore"],
```

```
"urad":["/static/images/urad.jpg", "Andhra Pradesh, Maharashtra, Madhya Pradesh, Tamil Nadu", "rabi", "United States, Canada, United Arab Emirates"],
```

```
"ragi":["/static/images/ragi.jpg", "Maharashtra, Tamil Nadu and Uttarakhand", "kharif", "United Arab Emirates, New Zealand, Bahrain"],
```

```
"sunflower":["sunflower.jpg", "Karnataka, Andhra Pradesh, Maharashtra, Bihar, Orissa", "rabi", "Phillippines, United States, Bangladesh"],
```

```
"sugarcane":["sugarcane.jpg", "Uttar Pradesh, Maharashtra, Tamil Nadu, Karnataka, Andhra Pradesh" , "kharif", "Kenya, United Arab Emirates, United Kingdom"]
```

```
}
```

```
return crop_data[crop_name
```

```
η
```