SMART FARMING USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

AKEEL HAIDER [Reg No: RA1511008010025]
N. LAVANYA [Reg No: RA1511008010029]
MRINALINI MAJUMDAR [Reg No: RA1511008010093]
TUSHAR YADAV [Reg No: RA1511008010576]

Under the Guidance of

Dr.M. SARAVANAN
(Associate Professor, Department of Information Technology)

In partial fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603203

MAY 2019

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this project report titled "**SMART FARMING USING MACHINE LEARNING"** is the bonafide work of **"AKEEL HAIDER [Reg No: RA1511008010025], N. LAVANYA [Reg No: RA1511008010029], MRINALINI MAJUMDAR [Reg No: RA1511008010093], TUSHAR YADAV [Reg No: RA1511008010576],** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Dr. M.SARAVANAN
GUIDE
Associate Professor
Dept. of Information Technology

Dr. G. VADIVU
HEAD OF THE DEPARTMENT
Dept. of Information Technology

Signature of Internal Examiner

Signature of External Examiner

ABSTRACT

The agriculture industry in India has generated large amounts of data, which is operated by record keeping, compliance and regulatory requirements, govt. of india, world banks and also by the farmers themselves. While most data is stored as hard copies, the current trend is towards the rapid digitization of these large amounts of data. To support a wide range of yield and investment related queries of farmers, large quantities of data (which is called big data), can improve the quality of services necessary and promise to better and profitable yield. Hence the development of agricultural productivity is enhanced based on the crop yield and cost per yield prediction. This paper illustrates classification and regression model to predict yield and cost per yield of various crops, which are influenced by factors such as humidity, type of soil, crop type, rainfall, season, etc., by using a supervised algorithm in machine-learning. This regression algorithm is carried out in previous years of agricultural data to predict future yield and cost per yield of crops. Also, the efficacy of the proposed machine learning algorithm technique has been proved with the best accuracy with entropy calculation, recall, precision, and F1 score. This prediction helps is deriving useful insights to the farmers to pick on the right crop to be sowed for the upcoming year leading to maximum profit.

Keywords - GDP, supervised algorithm, recall, F1 score, regression model

ACKNOWLEDGEMENT

The success of this project and the final outcome of it required guidance and references from various sources. We are exceptionally appreciative to have this chance to make this undertaking. We offer our thanks and gratitude to the Head of the Department of Information Technology, **Dr. G. Vadivu,** for giving all the assistance conceivable.

We would like to enunciate our gratitude to our project guide **Dr. M. Saravanan,** who was absolutely helpful throughout and was always available to help us and provided the required information for our project.

We are very fortunate to get continuous encouragement and support from all the teaching staff of the Department of Information Technology, for successfully finishing our minor project work too.

AKEEL HAIDER (RA1511008010025)
N. LAVANYA (RA1511008010029)
MRINALINI MAJUMDAR (RA1511008010093)
TUSHAR YADAV (RA1511008010576)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ML	Machine Learning
UML	Unified modelling language
ERD	Entity Relationship Diagram
IOT	Internet of Things
GUI	Graphical User Interface
GDP	Gross Domestic Product
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives

CHAPTER 1

INTRODUCTION

In developing countries, farming is considered as the greatest source of revenue for many people. To help farmers and agriculturalists, the government came up with a scheme in 2014 Lok Sabha election, Arun Jaitley as the finance minister, to increase Minimum Support Price (MSP) of certain crops to 50 percent return over cost production and in turn aid in doubling farmers income. At the beginning of sowing season, the government announces the MSPs for certain crops based on the recommendations from the Commission for Agricultural costs and prices (CACP). CACP has three different definitions for production cost which includes A2, A2+FL, and C2. This project uses data from C2 Scheme. C2 cost incorporates every single genuine cost in real money and kind brought about underway by real proprietor and lease paid for rented land and credited estimation of family work in addition to premium paid.

However, over the next few years, many farmers failed to make a profitable yield and the scheme was a failure. The farmers failed to foresee the various factors affecting the cultivation due to the dynamic nature of our economy. Thus ranchers and agriculturalists require unconstrained guidance recommendation in foreseeing future harvesting occurrences to boost crop yield and to take in substantial income out of it.

1.1 DOMAIN OVERVIEW

Machine learning helps in predicting the future from past data. It is a kind of man-made consciousness (AI) that gives the capacity to learn without being expressly modified. AI centers around the improvement of projects that can change when uncovered to new information and basic algorithm to discover patterns that lead to actionable insights in python. Machine learning is generally separated into three categories. There are supervised, unsupervised and support learning. The supervised learning program is both

given information and the parallel marking to learn information must be named by an individual in advance. Unsupervised learning is no marks. It gave to the learning calculation. This algorithm has to figure out the clustering of the input information. This calculation needs to make sense of the clustering of the input information. Lastly, Reinforcement dynamically cooperates with its surrounding and it gets positive or negative input to improve its execution.

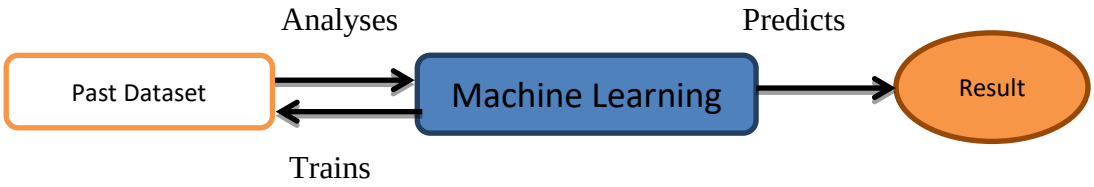

Fig 1.1: Process of Machine learning

In this project supervised learning approach has been put in use. Procedure of preparing and forecast includes the utilization of particular calculations. It sustains the preparation information to a calculation, and the calculation utilizes this preparation information to give expectations on new test information.

1.2 IOT IN SMART FARMING

The IoT can be defined as the embedding internet into hardware and day-to-day items. With Internet available on these devices and connected to some other physical equipment, these ordinary items can be transformed to exchange data over the digital network with other devices likewise, and thus they can be remotely handled and manipulated over internet. In our GUI, the farmer is supposed to choose the temperature, humidity and Soil type to predict the crop yield and cost of production. In an advanced version of our project, through IoT devices, the live temperature, humidity and soil type can be measured. These are small sensors that are placed in the field and are not affected by rainfall or excess heat. IoT devices are placed in different parts of the fields which record the attributes and directly feed the data into the machine. IoT devices are very helpful in a better or advanced version of our project because the measurements will be very accurate. Through this, we have a better clarity and outcome or prediction of the crop

yield and cost of production. As we are concentrating more on predictions and algorithms for it, we are not doing much with IoT as of now.

In the long run, IoT devices are going to be very helpful in the Smart Farming scenario, as the sensors are the ones directly feeding data and the algorithm does all the work. This was Farmers don't have to go through any kind of loss.

1.3 DATASET DESCRIPTION

The dataset used contains historic district-wise rainfall, humidity, season, temperature, cost of production, the yield for all states in India from 2000 to 2014. It covers over 124 crops across 29 states. It contains approximately 2 lakh data for better prediction. This dataset is split into two, one part is used as training data and the remaining is used for testing. Every new detail filled in the GUI interface acts as a test data set. After the process of testing, a model prediction based upon the inference it concludes on the basis of the training data sets.

Table 1.1: Dataset details

Variable	**Description**
State Name	States in India
District Name	District name list of each state
Crop year	From 2000 to 22014
Season	5 major seasons
Crop	124 crops
Area	Total area under cultivation
Rainfall	Water availability of each crop in mm
Average humidity	straightforwardly impacts the water relations of plant and by implication influences yield percentage
Mean Temperature	In degree Celsius
Cost of Cultivation (`/Hectare) C2	Cultivation amount for C2 Scheme
Cost of Production (`/Quintal) C2	Production amount for C2 Scheme
Yield (Quintal/ Hectare)	Yield of crop
Cost Production of per yield crop	Cost of crop yield

	State_Name	District_Name	Crop_Year	Season	Crop	Area	rainfall	Average Humidity	Mean Temp	CC	CP	Y	cost of production per yield
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	0.012360	57	62	23076.74	1941.55	9.83	19085.4365
1	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Arecanut	1254.0	0.084119	56	58	12610.85	1691.66	6.83	11554.0378
2	Andaman and Nicobar Islands	NICOBARS	2002	Whole Year	Arecanut	1258.0	0.080064	58	53	32683.46	3207.35	9.33	29924.5755
3	Andaman and Nicobar Islands	NICOBARS	2003	Whole Year	Arecanut	1261.0	0.181051	57	58	13209.32	2228.97	5.90	13150.9230
4	Andaman and Nicobar Islands	NICOBARS	2004	Whole Year	Arecanut	1264.7	0.035446	63	67	22560.30	1595.56	13.57	21651.7492

Fig 1.2: A look at dataset

1.4 MOTIVATION AND PROBLEM STATEMENT

To support farmers and agriculturalists to take the correct choice in making development Agriculture is the most imperative area that impacts the economy of India. It adds to 18% of India's Gross Domestic Product (GDP) and offers work to half of the number of inhabitants in India. Individuals of India are rehearsing Agriculture for quite a long time yet the outcomes are failing to satisfy because of different components that influence the harvest yield. To satisfy the necessities of around 1.2 billion individuals, it is vital to have a decent yield of harvests. Because of variables like soil type, precipitation, seed quality, absence of specialized offices, and so on the harvest yield is legitimately impacted. To centers around actualizing crop yield forecast framework by utilizing Machine learning systems by doing examination on agribusiness dataset. For assessing execution Accuracy is utilized as one of the variables. The classifiers are additionally contrasted and the estimations of Precision, Recall, and F1score. Lesser the estimation of mistake, increasingly precise the calculation will work. The outcome depends on an examination among the classifiers.

1.5 PROPOSED RESEARCH

To determine the estimated values for a particular crop that can be yielded in the upcoming sowing season and the estimation of cost that has to be invested totally on the crop per square unit of the field. Thereby helping in choosing the right crop, for the season and various other factors. This methodology is exceptionally gainful to ranchers, agriculturalists, nearby self-government, and Tahsildars to assign cost of cultivation for

farming and yield improvisation. This is achieved by comparing machine learning algorithms accuracy, F1 Score, recall, etc. and the best algorithm is implemented enabling a GUI interface through which yield rate and cost per yield are displayed.

1.6 OBJECTIVE

The objective mainly is to help farmers in making the right decision of cultivating the right crop for a particular season along with various factors like the year, rainfall, humidity, state, district etc. Which is achieved by processing the following steps

- Exploration data analysis of variable identification
  - Loading the given dataset
  - Import required libraries packages
  - Analyze the general properties
  - Find duplicate and missing values
  - Checking unique and count values
- Uni-variate data analysis
  - Rename, add data and drop the data
  - To specify data type
- Exploration data analysis of bi-variate and multi-variate
  - Plot diagram of pairplot, heatmap, bar chart and Histogram
- Method of Outlier detection
  - Pre-processing the given dataset
  - Splitting the test and training dataset
  - Comparing the Logistic regression model and random forest
- Comparing algorithm to predict the result
  - Based on the accuracy

1.7 PROJECT OUTLINE

It has to find Accuracy of the training dataset, Accuracy of the testing dataset, Specification, False Positive rate, precision, and recall by comparing algorithm using python code. The following involvement steps are,

- Define a problem
- Preparing data
- Evaluating algorithms
- Improving results
- Predicting results

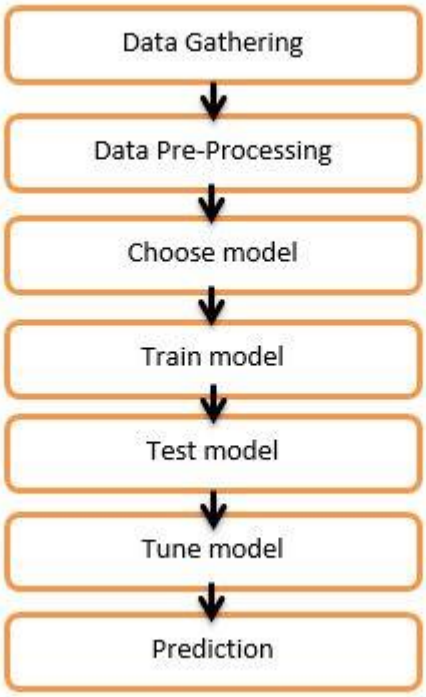

Fig 1.3: Process Flow Diagram

CHAPTER 2

LITERATURE REVIEW

The Project work on Smart Farming System based on Machine learning been carried out based on following literatures reviewed. We will be now be discussing in brief the literatures pertaining to effect of temperature rise, yield variation and predictive analysis based on Machine Learning algorithm.

2.1 EFFECT OF TEMPERATURE RISE

Farming depends on various factors like climate, soil, season, rainfall, humidity, etc. An analysis was made about the surprising ascent of temperature in Pakistan prompted unsettling influence in harvest yield. They studied multitemporal nature of radar signature of one year after another using ERS SAR images of agricultural crops.

2.2 CLIMATIC CHANGES ON FOOD SECURITY AND YIELD

A paper was published on the consequences of drastic climate change in India, where there is a huge dependency on agriculture. It displayed the absence of specialized and budgetary help for adjustment and decrease to environmental change. It likewise demonstrated that something like just rains, or only good temperature is not required, both go hand in hand. If the crop Faced rainfall but a bad temperature, there is no point. It brought into notice that climatic changes have been followed by an increased occurrence of natural calamity, with the confirmation that such occasions can cause a radical downturn in the agrarian yield, exasperating the issues of nourishment weakness and provincial neediness. The present paper looks at changes in climate factors, viz. temperature and precipitation, the results of climatic minor departure from yields of various nourishment crops; recommendations of atmosphere variety for sustenance security.

2.3 EFFECT ON SEASONAL CROPS WITH CLIMATE CHANGE

A research on four vital seasonal Indian crops - cotton, Wheat, Rice and sugarcane for the interval of 2004 to 2013. The main problem the world not only India, is facing is Climate Change. Due to rapid climate change many farmers don't know what crops to grown in their farm. The analysis is about the effect of variation in climate on Agriculture and food assured future over a particular state-level data. Once the Climate variation is known, it is easier to understand the type of farming required. This was performed for seven states intensive in agriculture with varied climatic conditions.

2.4 PREDICTIVE ANALYSIS ON RICE YIELD

The testing of historical data to predict future result to Improve Crop Yield utilizing a NNM (Neural Network model) was distributed, in which a solitary harvest was picked and examination was made dependent on a mix of components including soil properties, atmosphere, height, and water system procedure. The planned hybrid neural system show recognizes ideal mixes of soil factors with which it mixes with the precipitation design in selected district to advance the anticipated harvest result. The spine for prescient examination display as for the precipitation depends on Time-Series approach, which is in in Supervised Learning. The proposed designing gives a computational estimation to update finding out about the yield before the collect sowing period. It's made feasible by an information-driven model. Which is a hybrid model. Since the model plays out a joint desire for both precipitation and soil incorporates on the yield, it is named as a hybrid display.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 PROBLEM DEFINITION

Indian economy is most influenced by Agriculture. It constitutes about 50% of the total employment in India and 18% of the Indian GDP. In India agriculture is being practiced from centuries but the yield is never according to the inputs given which are due to several factors. For the satisfaction of 1.2 billion people, the crop yield should be excellent and profitable for the farmers. Crop yields are directly affected due to factors like soil type, rainfall, quality of seed, lack of technical facilities etc. Analyze agricultural datasets and focus on implementing crop yield prediction system using machine learning techniques. Accuracy is used to evaluate performance as one of the factors. The classifier is compared with the values of Precision, Recall, and F1score. Reduce the value of the error, the algorithm will be more accurate. The result is based on comparison between the classifier.

Numerous seasonal, financial and natural examples impact crop creation. Unusual changes in these examples lead to an extraordinary misfortune to farmers. These risks can be reduced when suitable approaches are employed on data related to temperature, atmospheric pressure, humidity, and region and crop type. Though, yield and climate determining can be anticipated by getting helpful experiences from this rural information that guides farmers to settle on the harvest they might want to plant for the pending year prompting most extreme benefit.

3.2 PROPOSED SYSTEM

In the previous yield, it was taken into consideration that the farmer would be involved in the field and harvest. Regardless, as the conditions change well ordered all around rapidly, the farmer is constrained to build up a consistently expanding number of

yields. Being this as the present circumstance, a basic number of people miss the mark on getting some answers concerning the new yields and are insensible of the advantages of the advancement. Moreover, The harvest yield effectiveness can be expanded through appreciation and furthermore deciding harvest execution in a combination of characteristic conditions. Thusly, the structure made utilizations the customer as a zone of data. This solidified information is the generation of yield and the information is got from various destinations which demonstrate the information identified with various harvests. It utilizes Prediction calculation and ML algorithms to comprehend a pattern with the information and use it as per the info is given.We have to find Accuracy of the training dataset, Accuracy of the testing dataset, Specification, False Positive rate, precision, and recall by comparing algorithm using python code. The following Involvement steps are,

- Outline a problem
- Formulating data
- Assessing algorithms
- Refining outcomes
- Predicting results

A dataset in the form of a table, where the rows are each overview, and columns for each observation represent the characteristics and values of that observation. At the beginning of the machine learning project, a dataset is usually divided into two or three subsets. Minimum subset training and test datasets. Sometimes verification dataset is also made. Once these data submissions are done with a primary dataset, a predictive model or classifier is trained using training data, and then using test data sets the models predictive accuracy. The machine learning system leverages algorithms to model data and discovers patterns simultaneously, usually with the goal of predicting. A large number of statistics and mathematical optimization are the basis of the algorithms. In essence, taking advantage of machine learning algorithms and optimization techniques is about automatically learning a highly accurate prediction or classifier model or finding unknown patterns in data. Importing the library packages with loading given dataset. To analysing the variable identification by data shape, data type and evaluating the missing values, duplicate values.

3.2.1 DATA PROCESSING

The steps involved in Building the data model is depicted below.

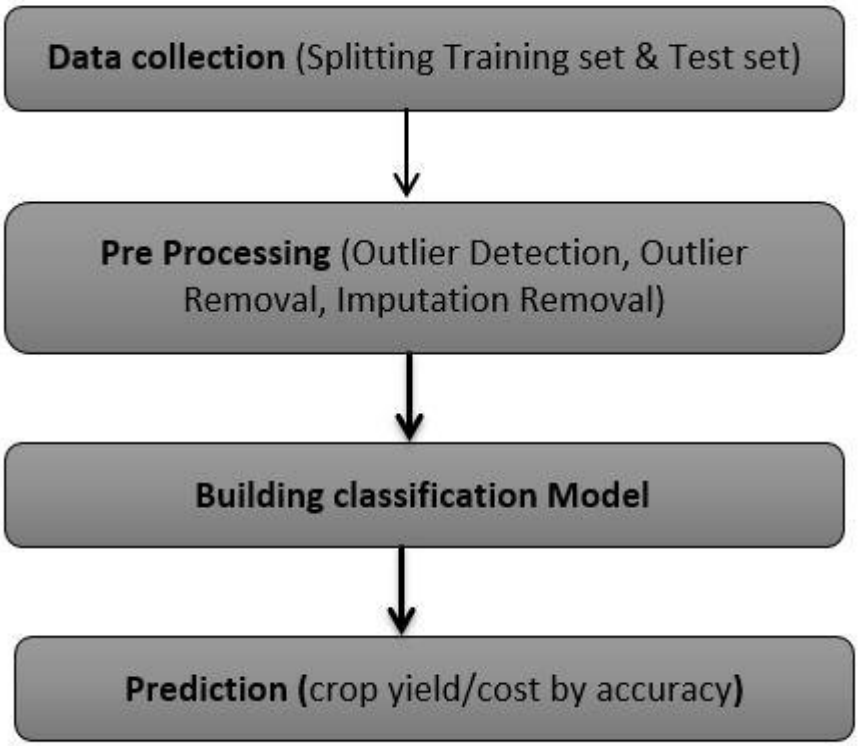

Fig 3.1: Steps of Building Data Model

3.2.1.1 DATA COLLECTION

Collected data sets are divided into training sets and test sets to predict the past farmer's list of yield. Typically, a 7: 3 ratio is applied to divide the training set and test set. The data model, which was created using random forest, logistics, disease tree algorithms, is applied on the training set and based on test results accuracy, the test set is predicted.

3.2.1.2 PREPROCESSING

The information which was gathered may contain missing qualities that may prompt irregularity. To increase better outcomes information should be preprocessed to improve the effectiveness of the calculation. The exceptions must be evacuated and furthermore factor change should be finished. In view of the relationship among qualities it was seen that properties that are huge exclusively incorporate territory, crop year, season, railfall,

which is the most grounded among all. A few factors, for example, candidate pay and co candidate salary are not critical alone, which is bizarre since by instinct it is considered as essential. The relationship among characteristics can be recognized utilizing plot chart. Information pre-processing is the most tedious period of an information mining process. Information cleaning of horticultural information evacuated a few qualities that has no noteworthiness about the conduct of a ranchers. Information incorporation, information decrease and information change are additionally to be material for yield information. For simple investigation, the information is decreased to some base measure of records. At first the Attributes which are basic to make a yield forecast is related to data gain as the quality evaluator and Ranker as the pursuit technique.

3.2.1.3 DATA VALIDATION

Bringing in the library bundles with stacking given dataset. To breaking down the variable recognizable proof by information shape, information type and assessing the missing qualities, copy values. An approval dataset is an example of information kept away from preparing your model that is utilized to give a gauge of model ability while tuning model's and systems that you can use to utilize approval and test datasets while assessing your models.

3.2.1.4 DATA CLEANING AND PREPARING PROCESS

Information cleaning/getting ready by rename the given dataset and drop the segment and so on to investigate the uni-variate, bi-variate and multi-variate process. The means and methods for information cleaning will differ from dataset to dataset. The essential objective of information cleaning is to identify and expel blunders and inconsistencies to build the estimation of information in examination and basic leadership.

3.2.1.5 DATA VISUALIZATION PROCESS

To imagine the given dataset as graphical portrayal like pairplot, heatmap, bar diagram, pi-outline from matplotlib, seaborn library pacakages. Simple to illuminate the agrarian information's.

3.2.2 ADVANTAGES OF PROPOSED SYSTEM

- Our goal is to push for assisting farmers, the government using our predictions. All these publications state they have done better than their competitors but there is no article or public mention of their work being used practically to assist the farmers.
- This system is for those who want to skilfully manage their field by organization, monitoring, recording, tracking and analysing all activities in the fields.
- This system uses climatic factors like rainfall, humidity, and temperature and season.

3.2.3 COMPARING MACHINE LEARNING ALGORITHMS

Prior to that looking at calculation, Building a Machine Learning Model utilizing introduce Scikit-Learn libraries. In this library bundle need to done preprocessing, direct model with strategic relapse strategy, cross approving by KFold technique, troupe with irregular woods strategy and tree with choice tree classifier. Furthermore, part the train set and test set. To foreseeing the outcome by contrasting precision.

3.2.4 PREDICTION RESULT BY ACCURACY

Logistic regression utilizes a direct condition with free indicators to foresee an value. The anticipated value can be anyplace between negative boundlessness to positive unendingness. We need the yield of the calculation to be arranged variable information. Higher exactness foreseeing result is calculated relapse or other model by looking at the best precision.

3.3 REQUIREMENTS

3.3.1 Hardware Tools

Processor : Intel core i3 and above
Hard disk : minimum 10GB of free space

RAM : minimum 4 GB

3.3.2 Software Tools

Operating System : Windows
Tool : Anaconda with Jupyter Notebook

SOFTWARE DESCRIPTION

1. *Anaconda*

It is for scientific computing which is a free and open source software that uses Python and R's distribution. that aims to simplify package management and deployment. Conda is the package management system that manages package versions. Scientific computing can be data science, machine learning applications, large scale data processing, predictive analysis, etc.

2. *The Jupyter Notebook*

The Jupyter Notebook is that enables us to make and share docs that contain code, equations, visualizations, and description text. It is an open source software. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

3.4 ISSUES IN EXISTING SYSTEM

- The amount of agricultural data is increasing gradually, but the prediction accuracy is very poor.
- As we identify certain patterns in words slowly start understanding, keeping in mind the real-time fluctuations in climate and soil conditions, using large and new data inputs.
- It provides an insight about the crop yield, but fails to explain how much money needs to be invested or the profit that is made after the harvest.
- Using the Linear regression algorithm is a statistical model and assumes data is normally distributed in real they are not. Before building model multi co-linearity

should be avoided. Prone to outliers all above are the not real disadvantages they can avoid with suitable treatments.

3.5 NEW METHODOLOGY

Exploratory Data Analysis

In this section of the report, you will load in the data, check for cleanliness, and then trim and clean your dataset for analysis. Make sure that you document your steps carefully and justify your cleaning decisions.

We have to find Accuracy of the training dataset, Accuracy of the testing dataset, Specification, False Positive rate, precision, and recall by comparing algorithm using python code.

The following Involvement steps are,

- Outline a problem
- Formulating data
- Assessing algorithms
- Refining outcomes
- Predicting results

In the previous methodologies, we couldn’t calculate the cost of production as in how much the Farmer is going to earn by planting which crop. This project helps in calculating the yield and the cost too. In this way, the Farmer is aware of the yield.

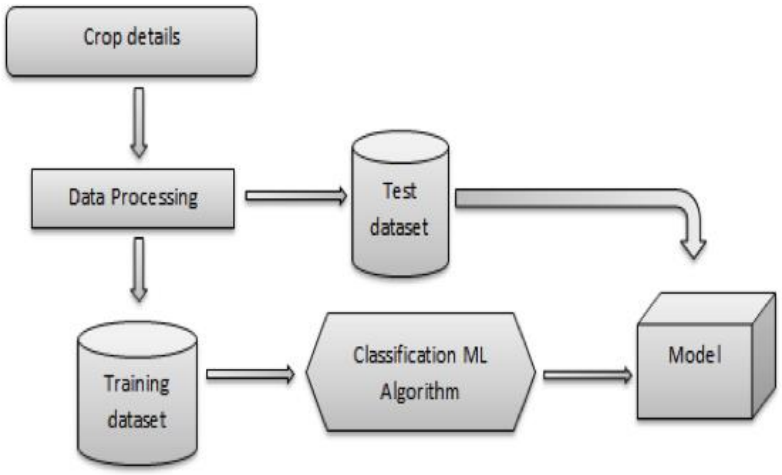

Fig 3.2 Architecture of Proposed model

3.5.1 Examining Data Analysis

In this area of the report, you will stack in the information, check for tidiness, and after that trim and clean your dataset for examination. Ensure that you archive your means cautiously and legitimize your cleaning choices.

Making the Dataset

- The first line imports iris informational index which is as of now predefined in sklearn module. Iris informational collection is fundamentally a table which contains data about different assortments of iris blooms.
- For model, to import any calculation and train_test_split class from sklearn and numpy module for use in this program.
- Then we typify load_data() technique in data_dataset variable. Further we separate the dataset into preparing information and test information utilizing train_test_split strategy. The X prefix in factor signifies the component esteems and y prefix means target esteems.
- This strategy isolates dataset into preparing and test information arbitrarily in proportion of 67:33. At that point we epitomize any calculation.
- In the following line, we fit our preparation information into this calculation with the goal that PC can get prepared utilizing this information. Presently the preparation part is finished.

Testing the Dataset

- Now we have measurements of another blossom in a numpy exhibit called 'n' and we need to foresee the types of this bloom. We do this utilizing the anticipate strategy which accepts this cluster as information and releases anticipated target an incentive as yield.
- So the anticipated target esteem turns out to be 0. At long last we discover the test score which is the proportion of no. of forecasts discovered right and absolute expectations made. We do this utilizing the score strategy which essentially thinks about the genuine estimations of the test set with the anticipated qualities.

General Properties:

Make cells unreservedly to investigate your information and you ought not perform an excessive number of activities in every cell. One alternative that you can take with this venture is to complete a ton of investigations in an underlying journal. These don't need to be sorted out, yet ensure you utilize enough remarks to comprehend the reason for each code cell. At that point, after you're finished with your examination, make a copy scratch pad where you will trim the overabundance and arrange your means so you have a streaming, strong report and ensure that you keep your peruser educated on the means that you are taking in your examination. Pursue each code cell, or each arrangement of related code cells, with a markdown cell to depict to the peruser what was found in the first cell. Endeavor to influence it so the peruser to would then be able to comprehend what they will find in the accompanying cell.

Precision figuring:

Precision = (TP + TN)/(TP + TN + FP + FN)

Where,

TP - True Positives TN - True Negatives FP - False Positives FN-False Negatives

Calculation Explanation: Used Python Packages:

sklearn :

- In python, sklearn is an AI bundle which incorporate a ton of ML calculations.
- Here, we are utilizing a portion of its modules like train_test_split, DecisionTreeClassifier or Logistic Regression and accuracy_score.

NumPy :

- It is a numeric python module which gives quick maths capacities to counts.
- It is utilized to peruse information in numpy clusters and for control reason.

Pandas :

- Used to peruse and compose diverse records.
- Data control should be possible effectively with dataframes.

CHAPTER 4

SYSTEM DESIGN

4.1 ALGORITHM

4.1.1 LOGISTIC REGRESSION

Logistic Regression is also called the Logistic Model. It is a statistical Model. In its simple form, it makes use of a logistic function and makes a binary variable which is dependent, though many more difficult extensions exist in this particular system. In regression analysis, it forms a binomial regression. So the prediction is in 1 or 0.

For example: To predict whether a obese or not, we predict using 0 or 1. So, if the person is obese, the outcome is 1 and if the person is not obese the outcome is 0.

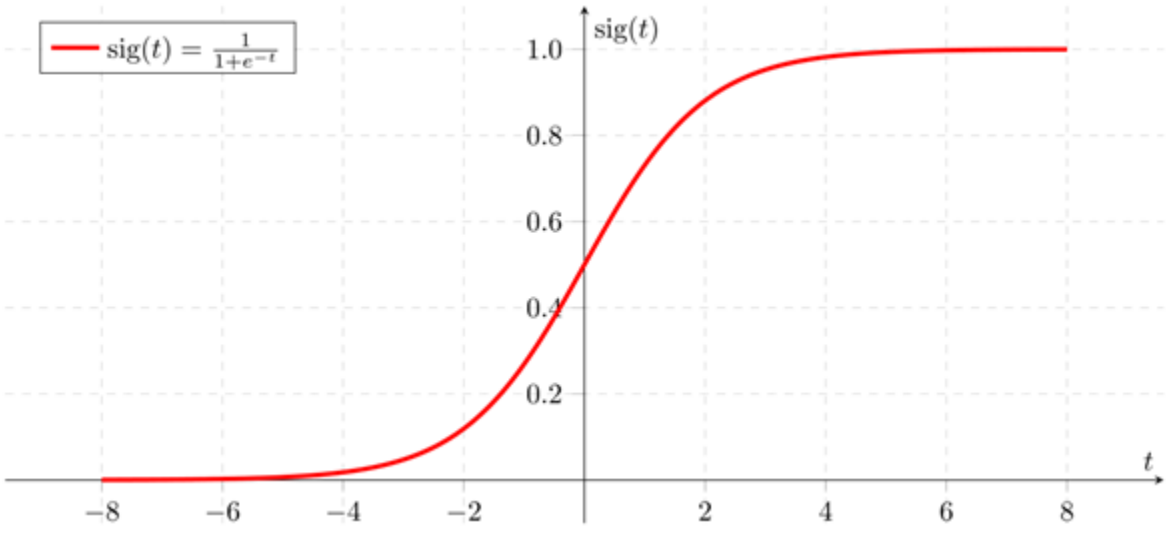

Fig 4.1: Obesity prediction graph

prototype:

Result => 0 or 1

Theorem => Z = PX + Q

$h\Theta(x) \Rightarrow$ sigmoid (Z)

Analysis:

The estimated Probability is derived as below. This concludes as to how accurate the value is when input is X.

Mathematically:

$h_\Theta(x) = P(Y=1 | X; theta)$

Probability that Y=1 given X which is parameterized by 'theta'.

$P(Y=1 | X; theta) + P(Y=0 | X; theta) = 1$

$P(Y=0 | X; theta) = 1 - P(Y=1 | X; theta)$

The types of regression are:

- Multinomial Logistic Regression
- Binomial Regression
- Ordinal Logistic Regression

We will be using Binomial Regression in our project. In this, the outcome is either 1 or 0. Different Columns with this outcome are arranged and we find out the relationship between them.

Simplified Cost Function:

$Cost(h_\Theta(x), y) = -y \log(h_\Theta(x)) - (1-y) \log(1- h_\Theta(x))$

If y = 1, (1-y) term will become zero, therefore $-\log(h_\Theta(x))$ alone will be present

If y = 0, (y) term will become zero, therefore $-\log(1- h_\Theta(x))$ alone will be present

4.1.2 RANDOM FOREST

Random Forest Classifier (RFC) is an ensemble algorithm. In this we will combine two or more algorithms to classify data accordingly. Running prediction with “Naive Bayes”, “Support Vector Machine” and “Decision Tree”, at that point taking a vote in favor of conclusive comprehension of class for the test object.

RFC makes group from randomly chosen subsets of training sets of the different decision trees. Then it makes a group of the votes from several different outputs of decision trees to choose the concluding class of the given test entity.

Suppose the training data set is given as : [X1, X2, X3, X4] with labels mapped to the following as [B1, B2, B3, B4], random forest classifier may make three or four decision trees taking input of subset for example as given,

1. [X1, X2, X3]
2. [X1, X2, X4]
3. [X2, X3, X4]

The code used for Random Forest Classifier is similar to previous classifiers.

1. Import library
2. Create model
3. Train
4. Predict

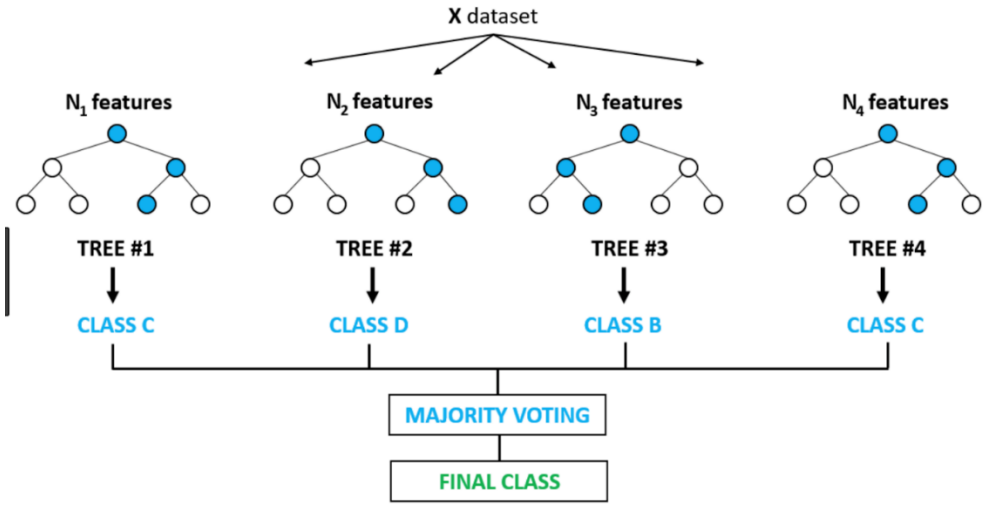

Fig 4.2: Describing Random Forest

4.2 MODULES AND FUNCTIONALITIES

4.2.1 SYSTEM ARCHITECTURE

The design makes a representation, provides details about the software data structure, architecture, interfaces, components that are particularly required to used in a given system.

The crop details are stored in a dataset. These have different attributes which are used for comparing and building relationships in between the attributes. Once the behavior is compared, Data Transformation takes place. As the dataset is huge, we build a training data set using for example only the first thousand values of the dataset. So we have a Model Training data set. Using a particular algorithm, we predict the results with maximum accuracy.

This system basically helps a farmer predict as to what crop to sow and also predicts the cost so that the farmer knows how much profit he would make approximately.

System Architecture

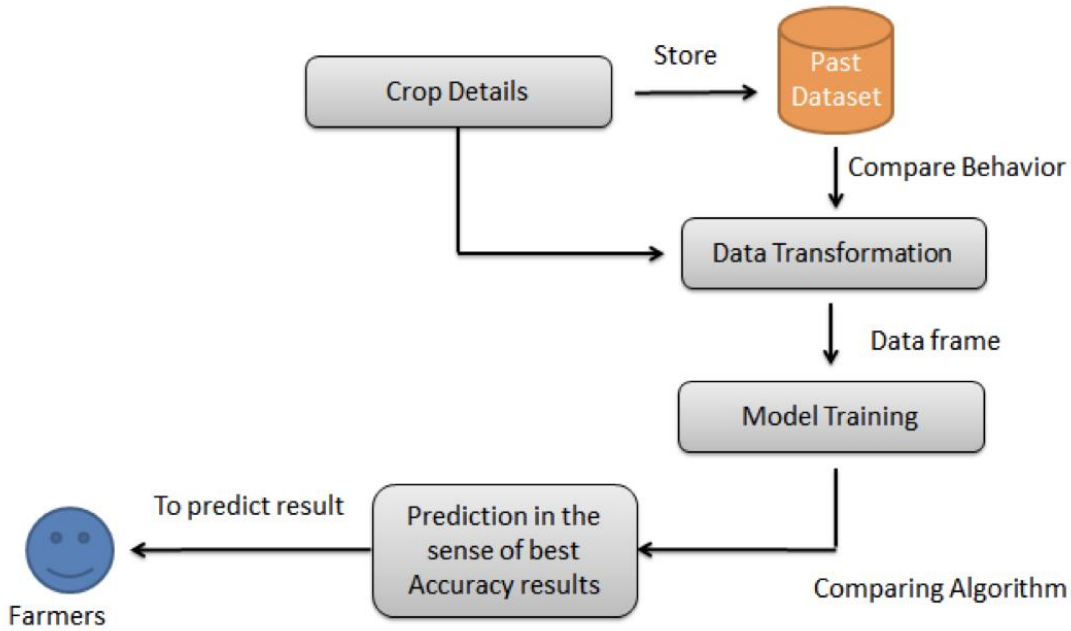

Fig 4.3: System Architecture

4.2.2 DATA FLOW DIAGRAM

The Source Data is the actual dataset. It is processed and cleaned as in all the redundant and dirty data is removed in this process. The dataset is then trained using the for example the first 1000 data. The rest of the dataset is tested using this training data set.

We use different algorithms for predicting the outcome using the dataset that we have. According to the accuracy of each model we find out which model is most accurate to use.

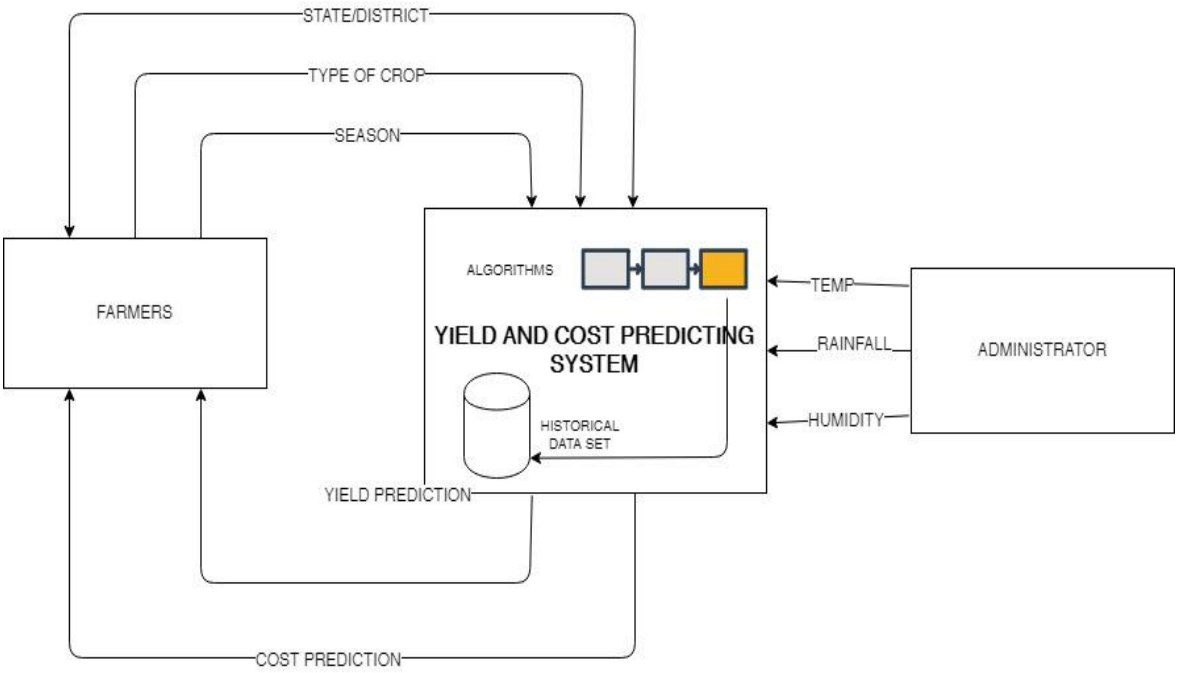

Fig 4.4: Work Flow Diagram

4.2.3 USE CASE DIAGRAM

A use case, diagrammatically shows the variables involved in the whole system and processes that are working internally. This diagram shows the role and how the system work from getting the details of the crops from farmers to making a prediction of the crop yield. The agricultural officer here can be any source that can supply details elated to harvest like season, temperature, rainfall , soil details etc.

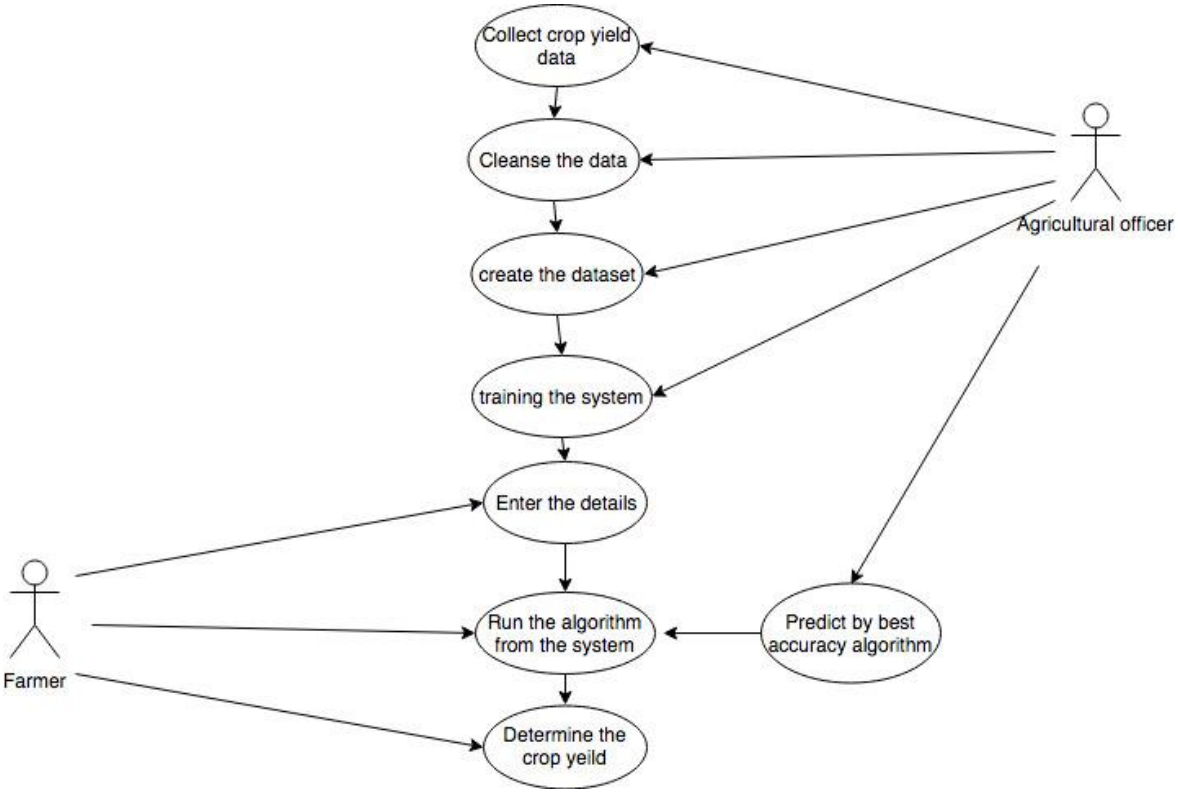

Fig 4.5: Use case Diagram

4.2.4 CLASS DIAGRAM

Class diagram, a depiction of the framework for a system involving different classes or real life entities and several functions or processes related among those entities. A single class depiction contains values of different kinds and functions involving those values. Each class is related to atleast one class and one can be a parent of the other class. In this diagram from taking the crop details and area details to training the dataset and pre processing it to splitting the dataset and then applying algorithms to get a prediction result, every step is been divided into several classes. This helps in decrease in complexity and

also a well-structured system is created.

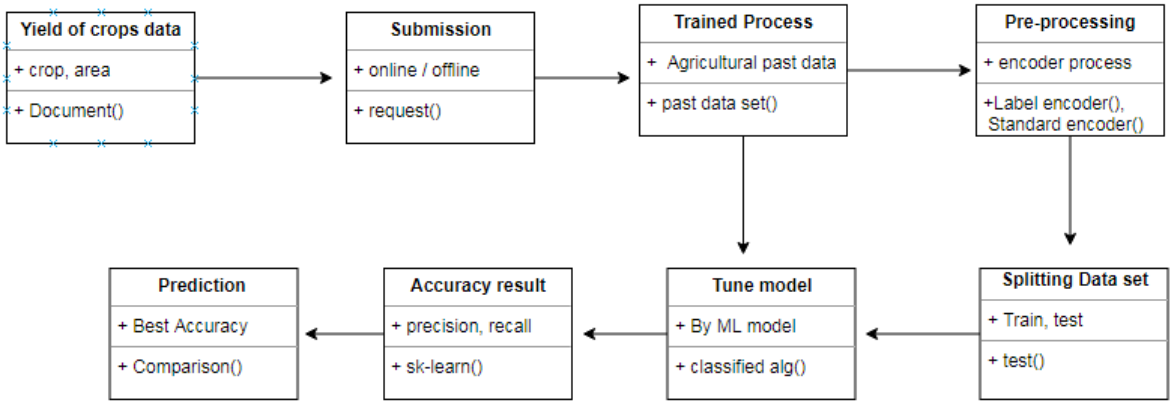

Fig 4.6 Class Diagram

4.2.5 ACTIVITY DIAGRAM

Activity diagram is mainly used to picture nature of a framework and figuring out procedures. The uniqueness of activity diagram is, it doesn't contain the message and resembles stream diagram.

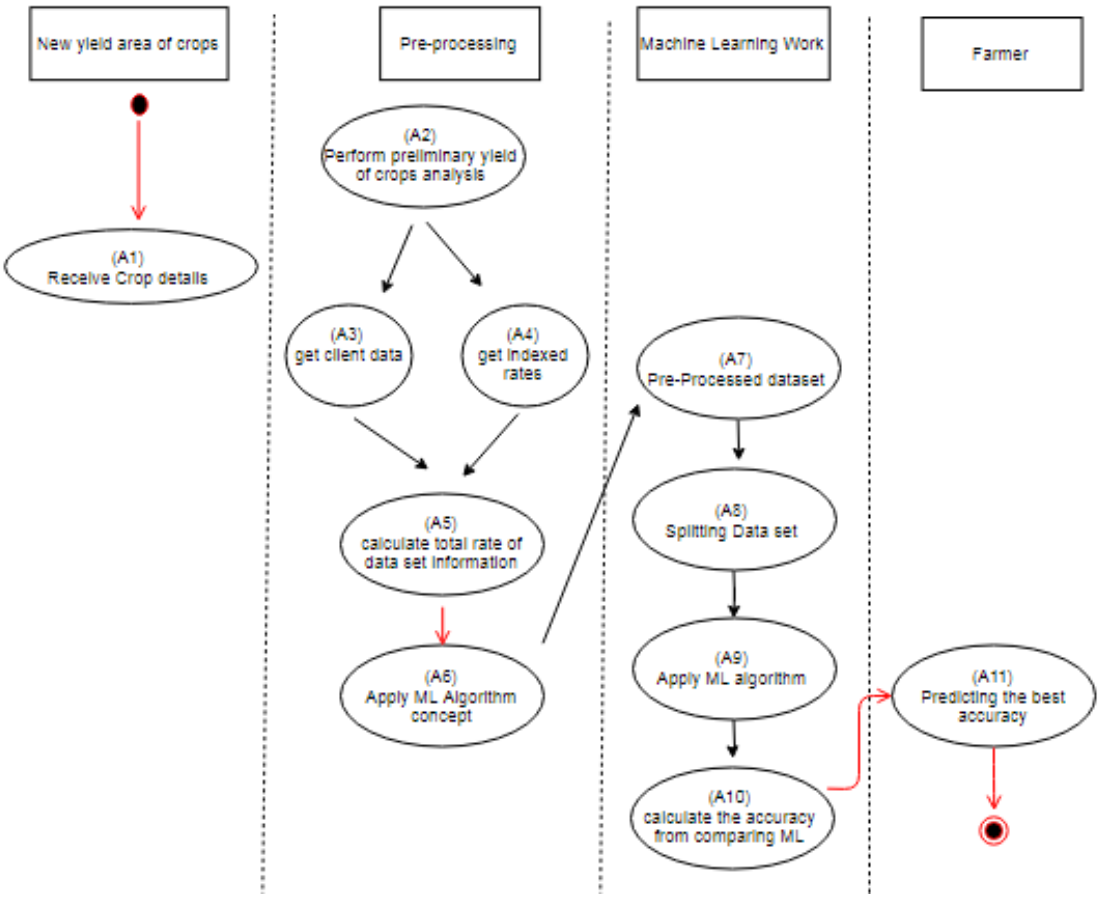

Fig 4.7: Activity Diagram

4.2.6 SEQUENCE DIAGRAM

A Sequence Diagram displays the progression of the inside of the system in a figurative way, making, to both, approve your system and your record, and are mainly, used for examination and arrangement purposes.

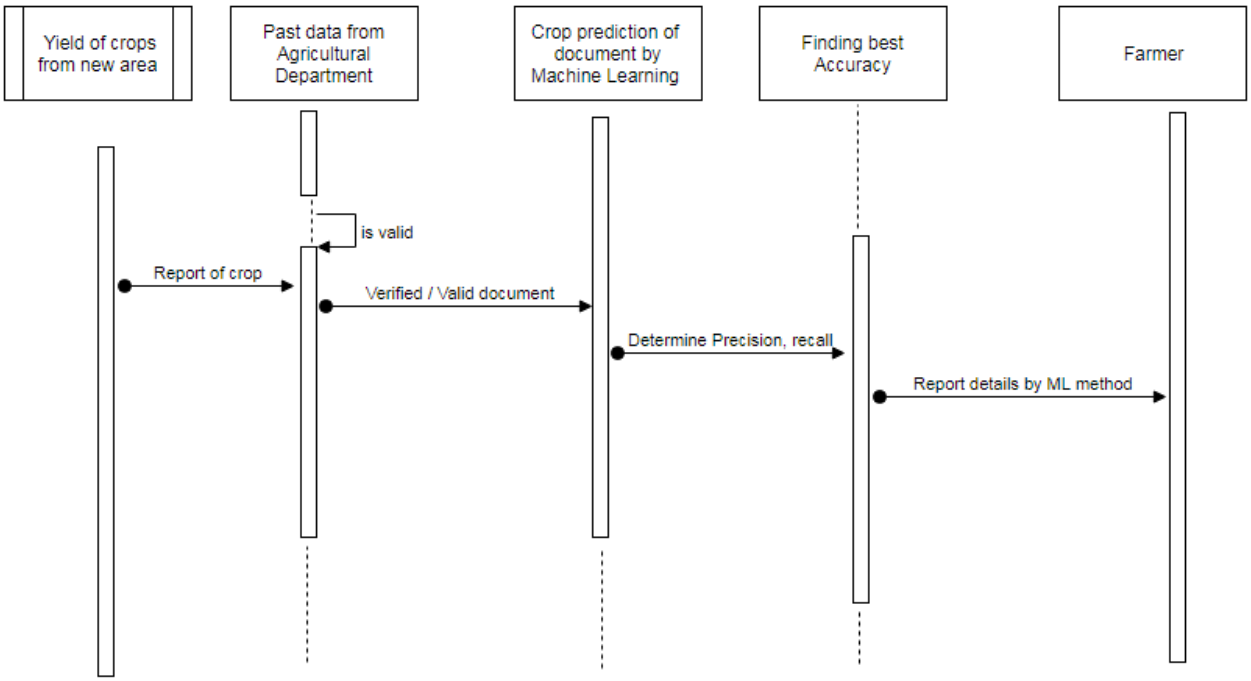

Fig 4.8: Sequence Diagram

4.2.7 ENTITY RELATIONSHIP DIAGRAM

ERD can be described as data representation technique that helps in mapping out the entities to provide a foundation for database. Component relationship graphs gives a way for to easily visualize the database structure. It is needed to select the structure of information which is important that ca be deduced from relationship.

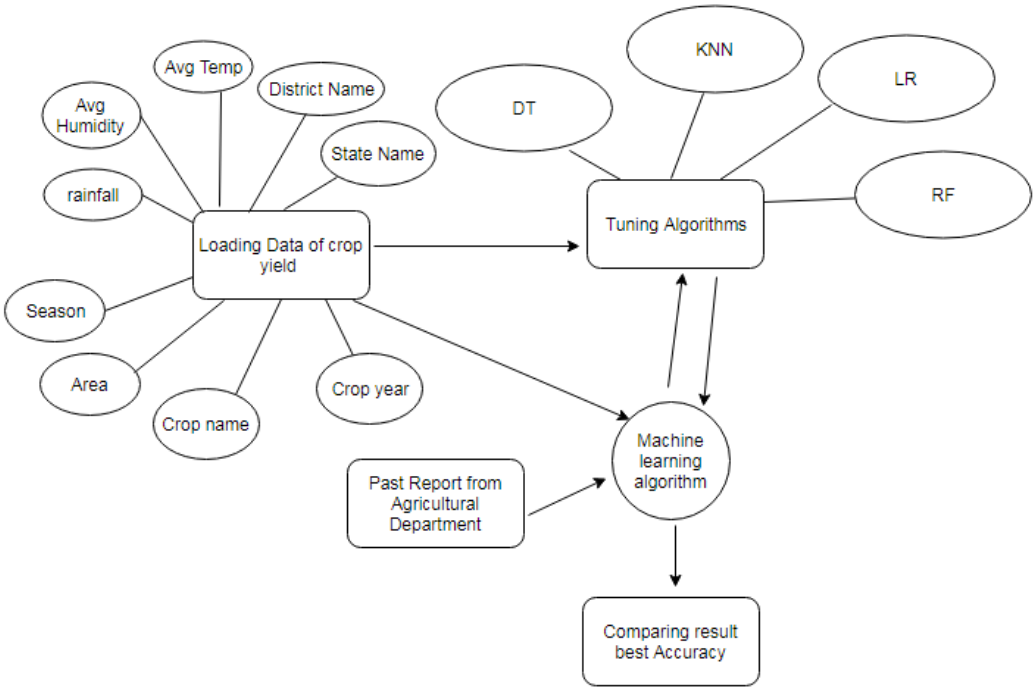

Fig 4.9 Entity Relationship Diagram

CHAPTER 5

RESULTS AND ANALYSIS

5.1 TESTING

1. Predicting the crop yield for a farmer named Ramesh of district Kanchipuram for the crop Rice in Rabi season.

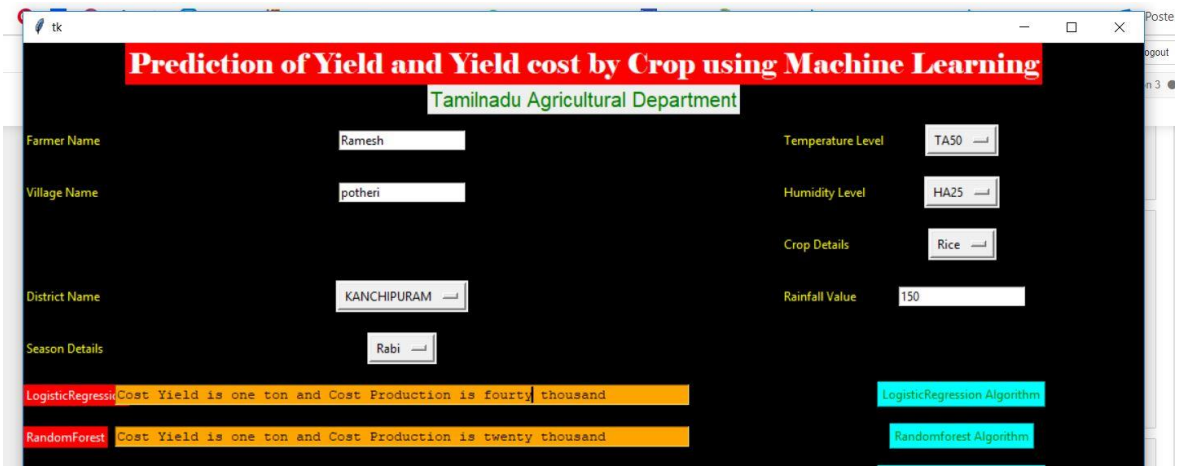

Fig 5.1: Testing System with test values(I)

2. Predicting the crop yield for a farmer named Velu of district Salem for the crop Banana in Autumn season.

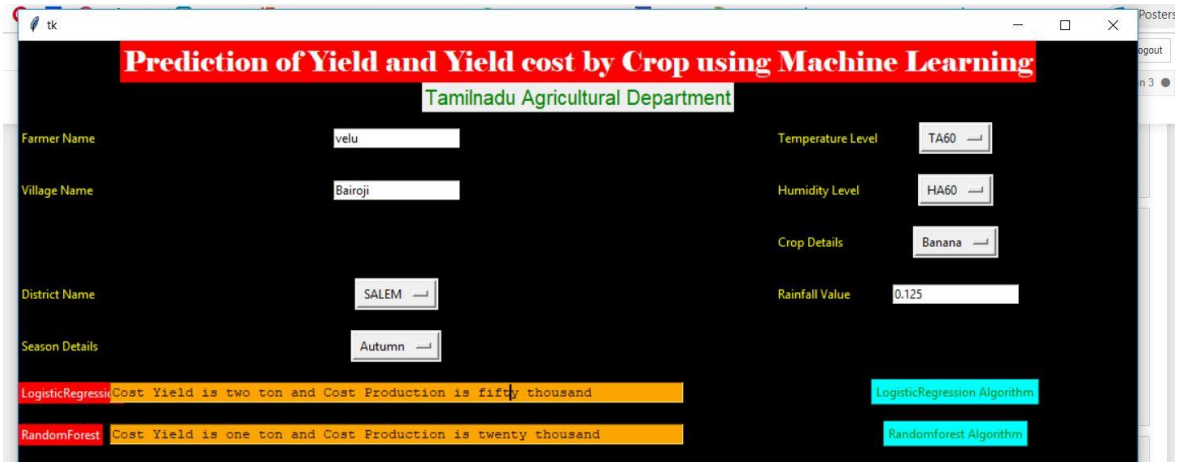

Fig 5.2: Testing System with test values(II)

5.2 ANALYSIS

1. Crop vs number of occurrences

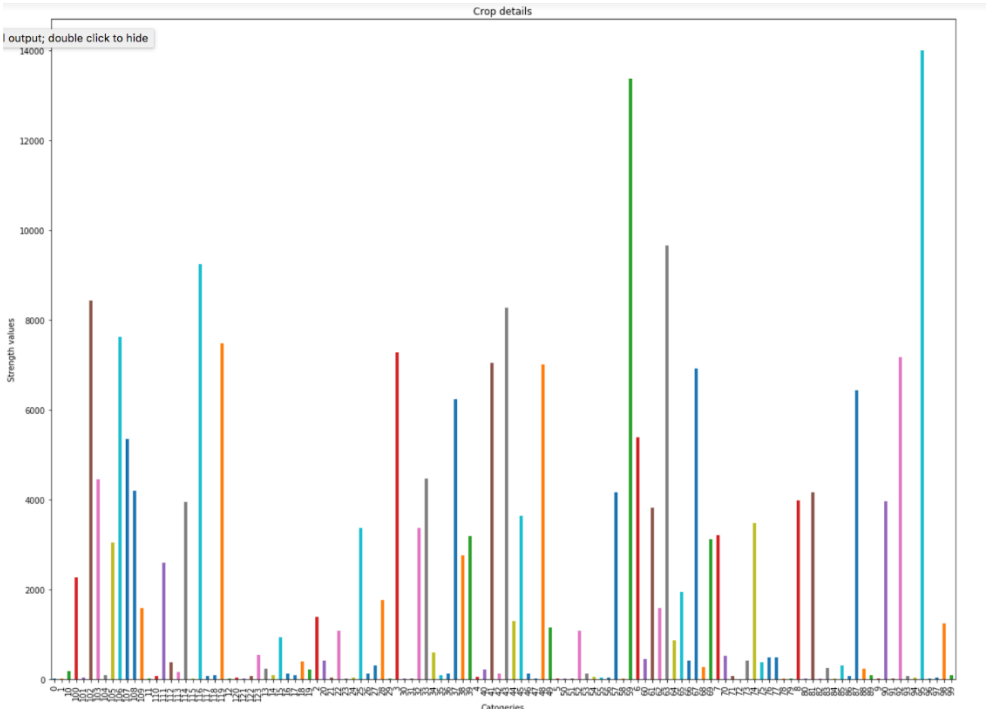

Fig 5.3 Crop vs Number of Occurrence

2. Percentage of crops sown in the last 17 years

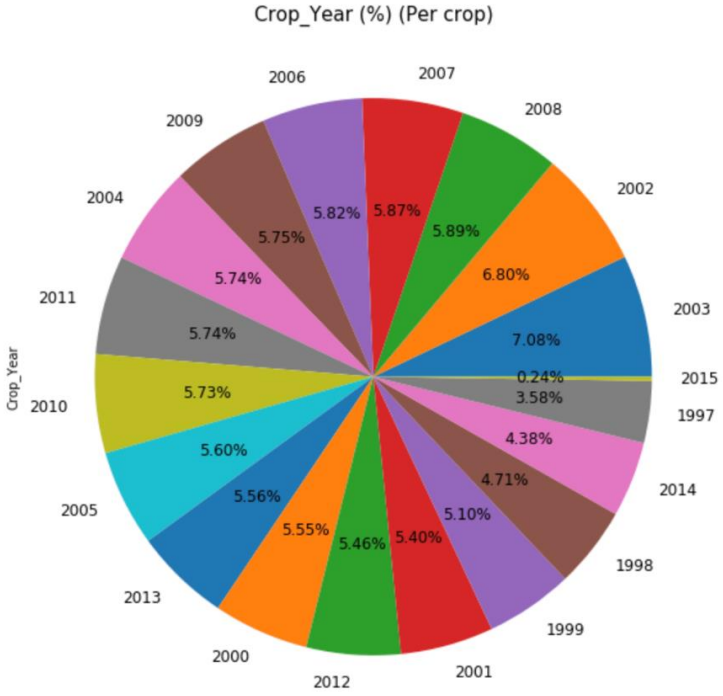

Fig 5.4 Percentage of crops sown in the last 17 years

3. State vs number of occurrences

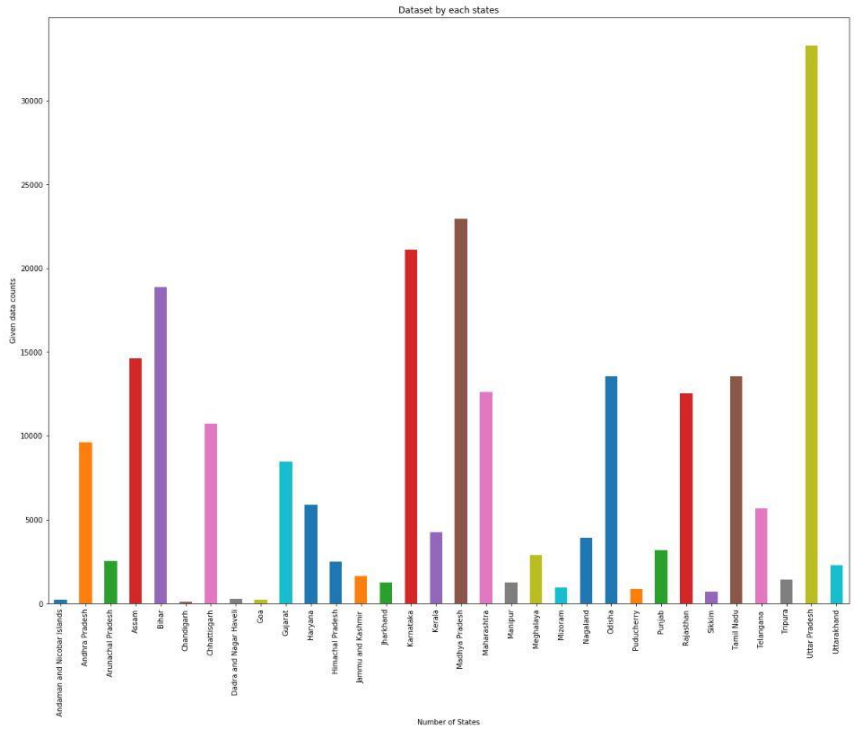

Fig 5.5 State vs number of occurrences

4. Percentage of crops sown in every state

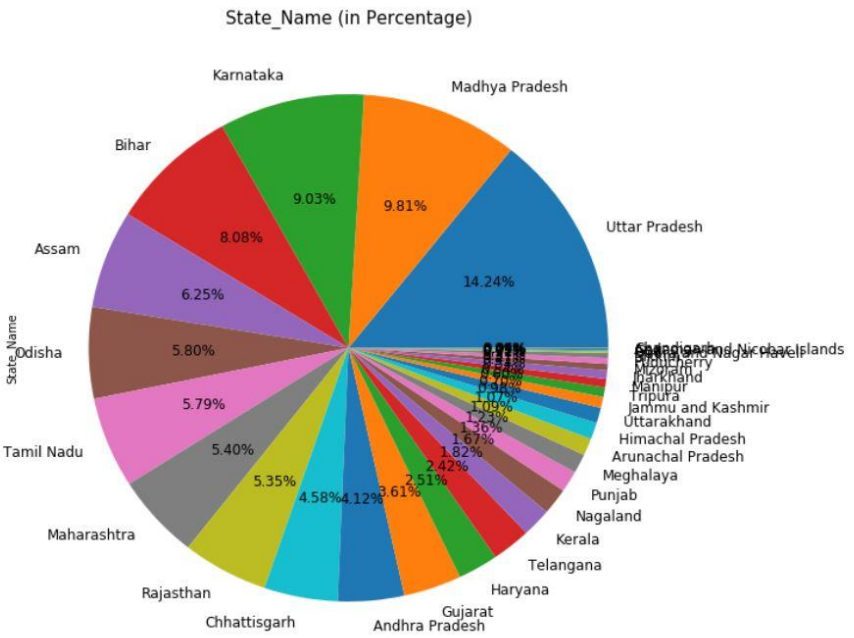

Fig 5.6 Percentage of crops sown in every state

5. Percentage of total crops sown

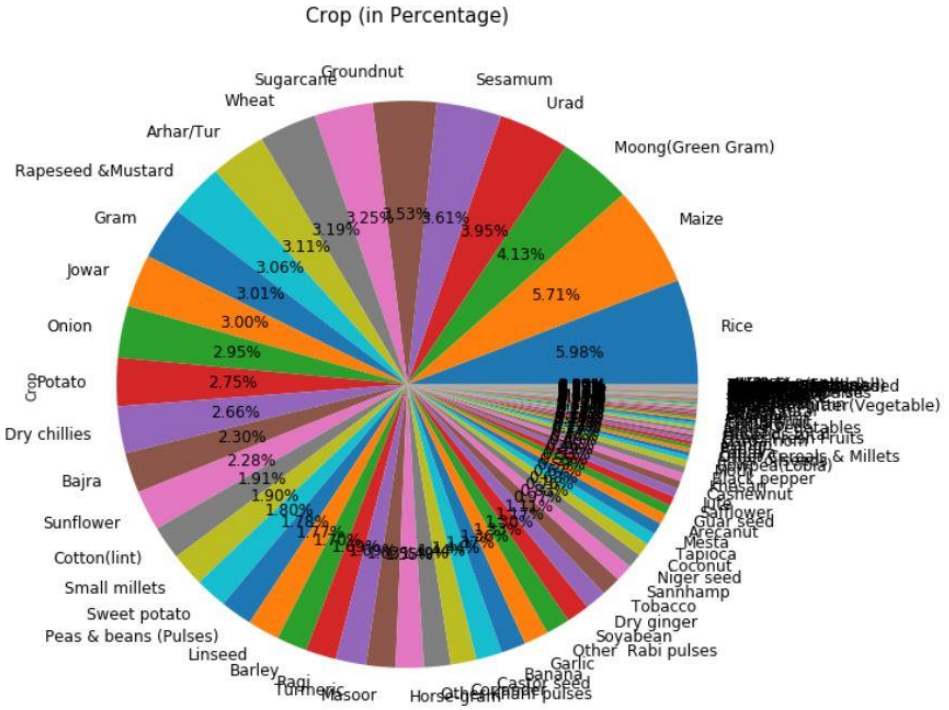

Fig 5.7 Percentage of total crops sown

6. Prediction results expected from farmers by yield of crops

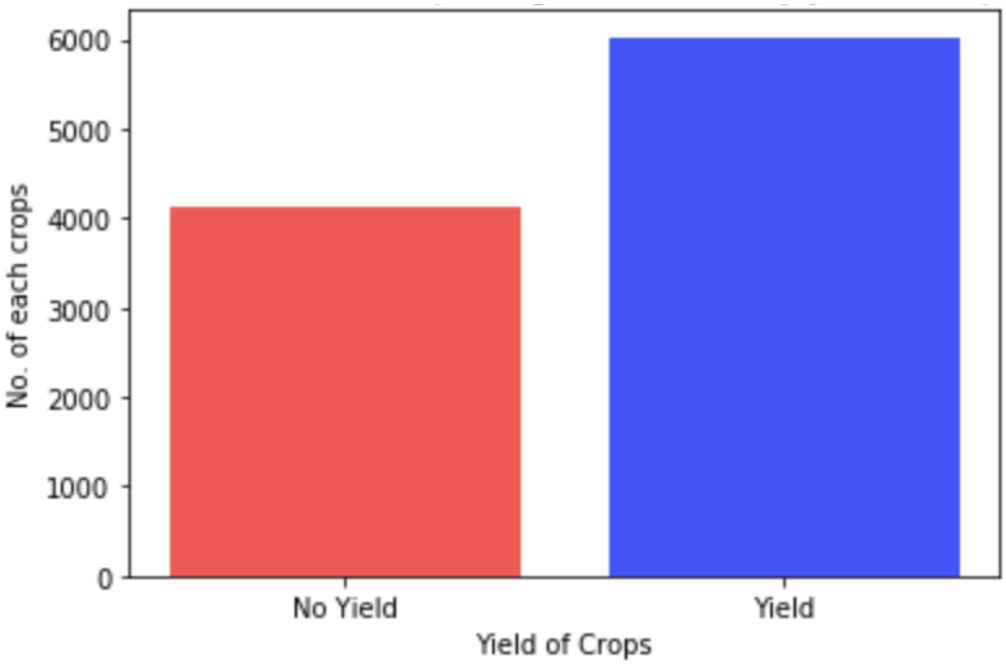

Fig 5.8 Prediction results expected from farmers by yield of crops

7. Heat map for yield

Fig 5.9 Heat map for yield

8. Yield vs cost production per 100kg

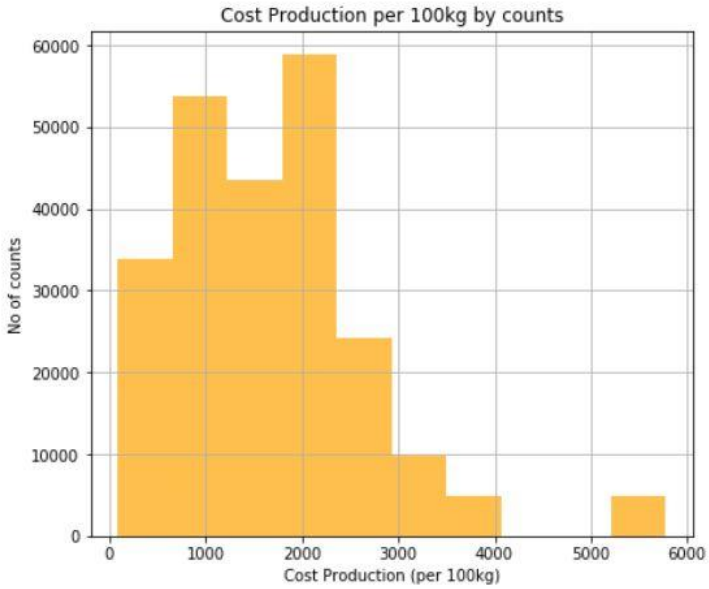

Fig 5.10 Yield vs cost production per 100kg

9. Density graphs for all variables

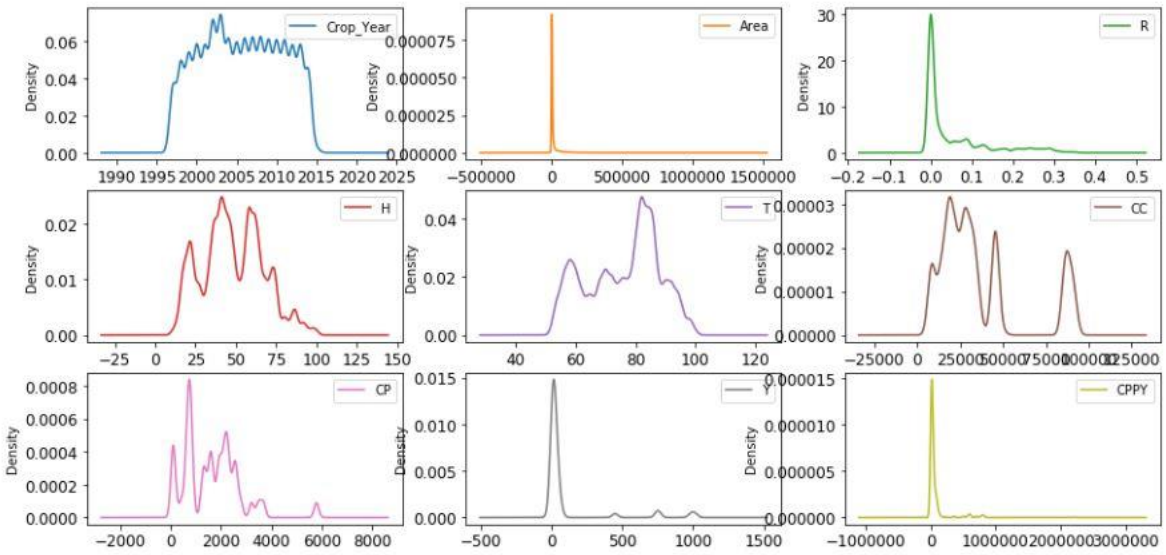

Fig 5.11 Density graphs for all variables

5.3 RESULTS

The main result or the output of the system is:

1. Crop yield per hectare.
2. Cost of the crop yield to be produced.

The output of the system is achieved by giving necessary input in the GUI and clicking the appropriate algorithm that needs to be applied. The input fields to be given are:

1. Name
2. State
3. District
4. Temperature
5. Humidity
6. Rainfall
7. Crop Type
8. Season

Algorithm	Precision	Recall	F1-Score	Support	Accuracy (100%)
Logistic Regression	0.95	0.95	0.95	70160	94.64
Random Forest	1	1	1	70160	100

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on the public test set is higher accuracy score of DT/RF Machine learning method from calculating cross validation checking, Precision, recall, and F1 score. This brings some of the following insights about crop prediction:

- Makes a list of all the crops, it helps in decision making for the farmer and helps the farmer to decide which crop to sow.

- Likewise, the given framework makes sure that the past generation of information that can enable the farmer to comprehend the understanding of the different expense of harvests and their interest in the market.

- This system helps the farmer in prediction the yield of any crop in a particular district based on various environmental and ecological conditions like Temperature, Humidity, and amount of rainfall, etc.

6.2 FUTURE ENHANCEMENTS

- The agricultural department wants to automate the detecting the yield crops from eligibility process (real time).

- To automate this process by showing the prediction result in a web application or desktop application.

- To optimize the work to implement in Artificial Intelligence environment.

- Include sensors to detect the temperature and humidity levels at any center where the system will be set up.

- Predicting environmental conditions like Temperature and Humidity at a particular place to make it easier for farmers to predict yields for upcoming seasons.

REFERENCES

[1] Rasul G, Q. Z. Chaudhry, A. Mahmood and K. W. Hyder, “Effect of. 40Temperature Rise on Crop Growth & Productivity”, Pakistan Journal of Meteorology, Volume 8, Issue 15, 2011, pp. 7-8

[2] Japneet Kaur, “Impact of Climate Change on Agricultural Productivity and Food Security Resulting in Poverty in India”, Università Ca' Foscari Venezia, 2017, pp. 16-18, 23

[3] Pratap S. Birthal, Md. Tajuddin Khan, Digvijay S. Negi and Shaily Agarwal, “Impact of Climate Change on Yields of Major Food Crops in India:
Implications for Food Security”, Agricultural Economics Research Review, Volume 27 (No. 2), pp. 145-155, July-December 2014.

[4] Mr. Dhawal Hirani, Dr. Nitin Mishra, “A Survey on Rainfall Prediction Techniques”, International Journal of Computer Application (2250-1797), Volume 6- No.2, March-April 2016, pp. 28-40.

[5] Basso B, Bodson B, V. Leemans, B. Bodson, J-P Destain, M-F Destain, “A comparison of within season yield predictions algorithm based on crop model behaviour analysis”, Agricultural and Forest Meteorology, Volume 204, pp. 10-21, May 2015.

[6] Stanley A Changnon. “Prediction of corn and soya bean yields using weather data”, CHIAA Research Report No. 22, Crop-Hail Insurance Actuarial Association, February 1965, pp. 6-10.

[7] Betty. J, Shem G Juma, Everline. O, “On the Use of Regression Models to Predict Tea Crop Yield Responses to Climate Change: A Case of Nandi East, Sub-County of Nandi County, Kenya”, Assesing the Value of Systematic Cycling in a Polluted Urban Environment, Climate, Volume 5, Issue 3, July 2017, pp. 5.

[8] Christian Baron and Mathieu Vrac, Oettli. P, Sultan. B, “Are regional climate models relevant for crop yield prediction in West Africa?”, Environmental Research Letters, Volume 6, 2011, pp. 2-6.

[9] https://www.ksndmc.org/ReportHomePage.aspx

[10] http://drdpat.bih.nic.in/PA-Table-10-Karnataka.htm

[11]https://www.kaggle.com/srinivas1/agricuture-crops-production-in-india/home

[12]http://eaindustry.nic.in/download_data_0405.asp

[13] https://data.gov.in/catalog/wholesale-price-index-0

[14] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5584412/

[15] http://dspace.unive.it/bitstream/handle/10579/10586/855733-1205794.pdf?sequence=2

[16] http://ageconsearch.umn.edu/bitstream/196659/2/1-PS-Birthal.pdf

[17]https://www.researchgate.net/publication/281103807_Detection_and_identification_of_bacterial_speck_of_tomato_Pseudomonas_syringae_pv_tomato_by_PCR

APPENDIX

MODULE 1 - Data Validation Process

```
#import libraries for access and functional purpose
import pandas as p
import numpy as n
import matplotlib.pyplot as plt
import seaborn as s

#read the given dataset
df = p.read_csv("df.csv")

df.head()

listcrops = p.Categorical(df['Crop'])
listcrops

df['Crop'].value_counts()

df['Crop'].nunique()

df['Crop'].unique()

df.shape

#To describe the dataframe
df.describe()

#Checking datatype and information about dataset
df.info()

df[df.dtypes[df.dtypes == 'float64'].index].describe()

p.Categorical(df['State_Name']).describe()

p.Categorical(df['District_Name']).describe()

p.Categorical(df['Mean Temp']).describe()

p.Categorical(df['Season']).describe()

p.Categorical(df['Average Humidity']).describe()

p.Categorical(df['rainfall']).describe()

p.Categorical(df['Crop']).describe()

#Checking for duplicate data
df.duplicated()

#find sum of duplicate data
sum(df.duplicated())
```

```
df.nunique()

#Correlation
df.corr()

#Checking minimum or maximum yields (100kg/2.47 acre)
print("Minimum yield of crops is  (100kg/2.47 acre):", df["Yield (Quintal/ Hectare) "].min())
print("Maximun yield of crops is  (100kg/2.47 acre):", df["Yield (Quintal/ Hectare) "].max())

#Checking minimum or maximum cost production for c2 scheme (per 2.47 acre)
print("Minimum cost production for c2 scheme(per 2.47 acre):", df["Cost of Production (`/Quintal) C2"].min())
print("Maximun cost production for c2 scheme(per 2.47 acre):", df["Cost of Production (`/Quintal) C2"].max())

#Rename the data
df.rename(columns={'Cost of Cultivation (`/Hectare) C2':'CC'}, inplace=True)
df.rename(columns={'Cost of Production (`/Quintal) C2':'CP'}, inplace=True)
df.rename(columns={'Yield (Quintal/ Hectare) ':'Y'}, inplace=True)
#show the dataframe
df.head()
```

MODULE 2 - Data Pre-Processing with unvariate, bivariate, multivariate analysis

```
p.crosstab(df.State_Name,df.Crop)

df.rename(columns={'Mean Temp':'T'}, inplace=True)
df.rename(columns={'Average Humidity':'H'}, inplace=True)
df.rename(columns={'rainfall':'R'}, inplace=True)

#Rename the data
df.rename(columns={'Cost of Cultivation (`/Hectare) C2':'CC'}, inplace=True)
df.rename(columns={'Cost of Production (`/Quintal) C2':'CP'}, inplace=True)
df.rename(columns={'Yield (Quintal/ Hectare) ':'Y'}, inplace=True)

p.crosstab(df.CC,df.CP)

df.rename(columns={'cost of production per yield':'CPPY'}, inplace=True)

p.crosstab(df.CC,df.CPPY)

p.crosstab(df.Crop,df.H)

df.dropna()

df['Y'].unique()

df.columns

from sklearn.preprocessing import LabelEncoder
var_mod = ['Y']
le = LabelEncoder()
for i in var_mod:
    df[i] = le.fit_transform(df[i]).astype(str)
```

```
df['Y'].unique()

df['YPr']= df.Y.map({'13':0, '7':0, '11':0, '4':0, '23':0, '39':1, '10':0, '18':0, '36':1, '47':1, '8':0, '3':0,
      '38':1, '46':1, '5':0, '9':0, '2':0, '44':1, '17':0, '41':1, '6':0, '16':0, '35':1, '19':0,
      '0':0, '43':1, '12':0, '45':1, '25':0, '33':1, '29':0, '37':1, '32':1, '21':0, '42':1,
      '48':1, '30':1, '34':1, '26':0, '20':0, '31':1, '24':0, '27':0, '22':0, '40':1, '28':0,
      '1':0, '14':0, '15':0})


df['CPPY'].unique()

from sklearn.preprocessing import LabelEncoder
var_mod = ['CPPY']
le = LabelEncoder()
for i in var_mod:
   df[i] = le.fit_transform(df[i]).astype(str)


df['CPPY'].unique()

df['CPPYPr']= df.CPPY.map({'126':0, '72':0, '200':0, '78':0, '144':0, '212':0, '149':0, '151':0, '178':0,
'312':0,
      '187':0, '69':0, '205':0, '276':1, '76':0, '134':0, '66':0, '254':1, '150':0, '291':1,
      '155':0, '106':0, '87':0, '119':0, '183':0, '170':0, '28':0, '265':1, '180':0, '36':0,
      '293':1, '326':1, '148':0, '234':1, '3':0, '124':0, '111':0, '182':0, '166':0, '169':0,
      '270':1, '243':1, '98':0, '230':0, '282':1, '334':1, '188':0, '65':0, '34':0, '328':1,
      '160':0, '115':0, '89':0, '152':0, '240':1, '141':0, '233':1, '32':0, '281':1, '335':1,
      '109':0, '47':0, '136':0, '112':0, '261':1, '229':0, '42':0, '325':1, '213':0, '175':0,
      '125':0, '196':0, '292':1, '82':0, '235':1, '79':0, '105':0, '123':0, '210':0, '39':0,
      '146':0, '185':0, '201':0, '41':0, '164':0, '184':0, '222':0, '250':0, '301':1,
      '218':0, '217':0, '168':0, '264':1, '46':0, '103':0, '284':1, '198':0, '56':0, '171':0,
      '331':1, '256':1, '99':0, '58':0, '247':1, '286':1, '67':0, '133':0, '143':0, '204':0,
      '227':0, '194':0, '6':0, '280':1, '225':0, '48':0, '258':1, '94':0, '244':1, '294':1,
      '215':0, '132':0, '277':1, '71':0, '219':0, '315':1, '97':0, '91':0, '156':0, '289':1,
      '100':0, '295':1, '147':0, '214':0, '19':0, '223':0, '90':0, '269':1, '300':1, '114':0,
      '135':0, '173':0, '55':0, '113':0, '127':0, '73':0, '177':0, '274':1, '118':0, '191':0,
      '145':0, '307':1, '162':0, '228':0, '50':0, '5':0, '248':1, '203':0, '61':0, '129':0,
      '192':0, '83':0, '158':0, '206':0, '242':1, '267':1, '137':0, '92':0, '176':0, '298':1,
      '25':0, '296':1, '186':0, '239':1, '193':0, '165':0, '310':1, '20':0, '287':1, '75':0,
      '107':0, '271':1, '138':0, '121':0, '241':1, '74':0, '43':0, '232':0, '154':0, '181':0,
      '195':0, '102':0, '237':1, '110':0, '268':1, '49':0, '104':0, '64':0, '318':1, '13':0,
      '128':0, '153':0, '31':0, '93':0, '309':1, '257':0, '52':0, '262':1, '202':0, '174':0,
      '101':0, '224':0, '86':0, '45':0, '263':1, '167':0, '17':0, '29':0, '30':0, '54':0,
      '303':1, '163':0, '251':1, '142':0, '273':1, '96':0, '35':0, '327':1, '53':0, '15':0,
      '190':0, '308':1, '226':0, '139':0, '62':0, '2':0, '1':0, '27':0, '84':0, '285':1,
      '246':1, '23':0, '189':0, '299':1, '231':0, '24':0, '16':0, '333':1, '306':1, '18':0,
      '288':1, '199':0, '260':1, '323':1, '37':0, '278':1, '324':1, '140':0, '4':0, '245':1,
      '322':1, '329':1, '159':0, '80':0, '197':0, '275':1, '11':0, '40':0, '33':0, '0':0,
      '320':1, '290':1, '68':0, '220':0, '21':0, '330':1, '12':0, '157':0, '302':1, '44':0,
      '221':0, '216':0, '279':1, '63':0, '313':1, '311':1, '332':1, '266':1, '238':1,
      '211':0, '172':0, '14':0, '117':0, '161':0, '122':0, '60':0, '321':1, '95':0, '208':0,
      '272':1, '131':0, '207':0, '81':0, '314':1, '51':0, '283':1, '120':0, '38':0, '9':0,
      '130':0, '70':0, '179':0, '7':0, '10':0, '108':0, '255':1, '77':0, '88':0, '22':0,
      '57':0, '85':0, '304':1, '253':1, '249':0, '297':1, '116':0, '305':1, '317':1, '252':1,
      '236':1, '209':0, '259':1, '26':0, '319':1, '316':1, '59':0, '336':1, '8':0})

df.head()
```

```
df['YPr'].unique()

df['CPPYPr'].unique()

# Spliting Train/Test:

#preprocessing, split test and dataset, split response variable
X = df.drop(labels='CPPY', axis=1)
#Response variable
y = df.loc[:,'CPPY']

#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important
to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)
print("Number of training dataset: ", len(X_train))
print("Number of testing dataset: ", len(X_test))
print("Total number of dataset: ", len(X_train)+len(X_test))

count_classes = p.value_counts(df['Crop'], sort = True).sort_index()
count_classes.plot(kind = 'bar', figsize=(20,15))
plt.title("Crop details")
plt.xlabel("Catogeries")
plt.ylabel("Strength values")
no=sum(df['CPPYPr']==0)
yes=sum(df['CPPYPr']==1)
colors=['orange','black']
locations=[1,2]
heights=[no,yes]
labels=['Unexpected Cost Production','Expected Cost Production']
plt.bar(locations,heights,color=colors,tick_label=labels,alpha=0.7)
plt.xlabel('Yield of Crost Production')
plt.ylabel('No. of each crop')
plt.title('Prediction results expecting from farmer by yield of crost production amount')
no=sum(df['YPr']==0)
yes=sum(df['YPr']==1)
colors=['orange','black']
locations=[1,2]
heights=[no,yes]
labels=['No Yield','Yield']
plt.bar(locations,heights,color=colors,tick_label=labels,alpha=0.7)
plt.xlabel('Yield of Crop')
plt.ylabel('No. of each crop')
plt.title('Prediction results expecting from farmer by yield of crop')
```

MODULE 3 - Data Visualization Process

```
#import libraries for access and functional purpose
import pandas as p
import numpy as n
import matplotlib.pyplot as plt
import seaborn as s

#read the given dataset
```

```
df = p.read_csv("df.csv")

df.rename(columns={'Mean Temp':'T'}, inplace=True)
df.rename(columns={'Average Humidity':'H'}, inplace=True)
df.rename(columns={'rainfall':'R'}, inplace=True)
df.rename(columns={'Cost of Cultivation (`/Hectare) C2':'CC'}, inplace=True)
df.rename(columns={'Cost of Production (`/Quintal) C2':'CP'}, inplace=True)
df.rename(columns={'Yield (Quintal/ Hectare) ':'Y'}, inplace=True)

df.rename(columns={'cost of production per yield':'CPPY'}, inplace=True)

df['CP'].hist(figsize=(7,6), color='orange', alpha=0.7)
plt.xlabel('Cost Production (per 100kg)')
plt.ylabel('No of counts')
plt.title('Cost Production per 100kg by counts')

df.columns

df['CPPY'].hist(figsize=(7,6), color='black', alpha=0.7)
plt.xlabel('Cost Production of crop')
plt.ylabel('No of counts')
plt.title('Cost Production of crop by counts')

# Heatmap plot diagram
fig, ax = plt.subplots(figsize=(15,10))
s.heatmap(df.corr(), ax=ax, annot=True)

df.columns

df.boxplot(column="CP", by="Season", figsize=(15,10))

df.boxplot(column="CPPY", by="Season", figsize=(15,10))

#Propagation by variable
def PropByVar(df, variable):
    dataframe_pie = df[variable].value_counts()
    ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize = 12)
    ax.set_title(variable + ' (in Percentage)', fontsize = 15)
    return n.round(dataframe_pie/df.shape[0]*100,2)

PropByVar(df, 'Crop')

#Propagation by variable
def PropByVar(df, variable):
    dataframe_pie = df[variable].value_counts()
    ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize = 12)
    ax.set_title(variable + ' (in Percentage)', fontsize = 15)
    return n.round(dataframe_pie/df.shape[0]*100,2)


PropByVar(df, 'State_Name')

#Propagation by variable
def PropByVar(df, variable):
    dataframe_pie = df[variable].value_counts()
    ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize = 12)
    ax.set_title(variable + ' (in Percentage)', fontsize = 15)
    return n.round(dataframe_pie/df.shape[0]*100,2)
```

```
PropByVar(df, 'CPPY')

#Propagation by variable
def PropByVar(df, variable):
   dataframe_pie = df[variable].value_counts()
   ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize = 12)
   ax.set_title(variable + ' (in Percentage)', fontsize = 15)
   return n.round(dataframe_pie/df.shape[0]*100,2)


PropByVar(df, 'Y')

count_classes = p.value_counts(df['State_Name'], sort = True).sort_index()
count_classes.plot(kind = 'bar', figsize=(20,15))
plt.title("Dataset by each states")
plt.xlabel("Number of States")
plt.ylabel("Given data counts")

#Density Plots
plt = df.plot(kind= 'density', subplots=True, layout=(4,3), sharex=False,
                sharey=False,fontsize=12, figsize=(15,10))
```

MODULE 4 – Outlier detection process

```
#import library packages
import pandas as p
import matplotlib.pyplot as plt
import seaborn as s
import numpy as n
#read the given dataset
df = p.read_csv("df.csv")

df.rename(columns={'Mean Temp':'T'}, inplace=True)
df.rename(columns={'Average Humidity':'H'}, inplace=True)
df.rename(columns={'rainfall':'R'}, inplace=True)
df.rename(columns={'Cost of Cultivation (`/Hectare) C2':'CC'}, inplace=True)
df.rename(columns={'Cost of Production (`/Quintal) C2':'CP'}, inplace=True)
df.rename(columns={'Yield (Quintal/ Hectare) ':'Y'}, inplace=True)
df.rename(columns={'cost of production per yield':'CPPY'}, inplace=True)

from sklearn.preprocessing import LabelEncoder
var_mod = ['Y','CPPY']
le = LabelEncoder()
for i in var_mod:
   df[i] = le.fit_transform(df[i]).astype(str)


df['YPr']= df.Y.map({'13':0, '7':0, '11':0, '4':0, '23':0, '39':1, '10':0, '18':0, '36':1, '47':1, '8':0, '3':0,
      '38':1, '46':1, '5':0, '9':0, '2':0, '44':1, '17':0, '41':1, '6':0, '16':0, '35':1, '19':0,
      '0':0, '43':1, '12':0, '45':1, '25':0, '33':1, '29':0, '37':1, '32':1, '21':0, '42':1,
      '48':1, '30':1, '34':1, '26':0, '20':0, '31':1, '24':0, '27':0, '22':0, '40':1, '28':0,
      '1':0, '14':0, '15':0})
```

```
df['CPPYPr']= df.CPPY.map({'126':0, '72':0, '200':0, '78':0, '144':0, '212':0, '149':0, '151':0, '178':0, '312':0,
    '187':0, '69':0, '205':0, '276':1, '76':0, '134':0, '66':0, '254':1, '150':0, '291':1,
    '155':0, '106':0, '87':0, '119':0, '183':0, '170':0, '28':0, '265':1, '180':0, '36':0,
    '293':1, '326':1, '148':0, '234':1, '3':0, '124':0, '111':0, '182':0, '166':0, '169':0,
    '270':1, '243':1, '98':0, '230':0, '282':1, '334':1, '188':0, '65':0, '34':0, '328':1,
    '160':0, '115':0, '89':0, '152':0, '240':1, '141':0, '233':1, '32':0, '281':1, '335':1,
    '109':0, '47':0, '136':0, '112':0, '261':1, '229':0, '42':0, '325':1, '213':0, '175':0,
    '125':0, '196':0, '292':1, '82':0, '235':1, '79':0, '105':0, '123':0, '210':0, '39':0,
    '146':0, '185':0, '201':0, '41':0, '164':0, '184':0, '222':0, '250':0, '301':1,
    '218':0, '217':0, '168':0, '264':1, '46':0, '103':0, '284':1, '198':0, '56':0, '171':0,
    '331':1, '256':1, '99':0, '58':0, '247':1, '286':1, '67':0, '133':0, '143':0, '204':0,
    '227':0, '194':0, '6':0, '280':1, '225':0, '48':0, '258':1, '94':0, '244':1, '294':1,
    '215':0, '132':0, '277':1, '71':0, '219':0, '315':1, '97':0, '91':0, '156':0, '289':1,
    '100':0, '295':1, '147':0, '214':0, '19':0, '223':0, '90':0, '269':1, '300':1, '114':0,
    '135':0, '173':0, '55':0, '113':0, '127':0, '73':0, '177':0, '274':1, '118':0, '191':0,
    '145':0, '307':1, '162':0, '228':0, '50':0, '5':0, '248':1, '203':0, '61':0, '129':0,
    '192':0, '83':0, '158':0, '206':0, '242':1, '267':1, '137':0, '92':0, '176':0, '298':1,
    '25':0, '296':1, '186':0, '239':1, '193':0, '165':0, '310':1, '20':0, '287':1, '75':0,
    '107':0, '271':1, '138':0, '121':0, '241':1, '74':0, '43':0, '232':0, '154':0, '181':0,
    '195':0, '102':0, '237':1, '110':0, '268':1, '49':0, '104':0, '64':0, '318':1, '13':0,
    '128':0, '153':0, '31':0, '93':0, '309':1, '257':0, '52':0, '262':1, '202':0, '174':0,
    '101':0, '224':0, '86':0, '45':0, '263':1, '167':0, '17':0, '29':0, '30':0, '54':0,
    '303':1, '163':0, '251':1, '142':0, '273':1, '96':0, '35':0, '327':1, '53':0, '15':0,
    '190':0, '308':1, '226':0, '139':0, '62':0, '2':0, '1':0, '27':0, '84':0, '285':1,
    '246':1, '23':0, '189':0, '299':1, '231':0, '24':0, '16':0, '333':1, '306':1, '18':0,
    '288':1, '199':0, '260':1, '323':1, '37':0, '278':1, '324':1, '140':0, '4':0, '245':1,
    '322':1, '329':1, '159':0, '80':0, '197':0, '275':1, '11':0, '40':0, '33':0, '0':0,
    '320':1, '290':1, '68':0, '220':0, '21':0, '330':1, '12':0, '157':0, '302':1, '44':0,
    '221':0, '216':0, '279':1, '63':0, '313':1, '311':1, '332':1, '266':1, '238':1,
    '211':0, '172':0, '14':0, '117':0, '161':0, '122':0, '60':0, '321':1, '95':0, '208':0,
    '272':1, '131':0, '207':0, '81':0, '314':1, '51':0, '283':1, '120':0, '38':0, '9':0,
    '130':0, '70':0, '179':0, '7':0, '10':0, '108':0, '255':1, '77':0, '88':0, '22':0,
    '57':0, '85':0, '304':1, '253':1, '249':0, '297':1, '116':0, '305':1, '317':1, '252':1,
    '236':1, '209':0, '259':1, '26':0, '319':1, '316':1, '59':0, '336':1, '8':0})

df.columns

df.pivot_table(values='YPr',index = ['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area',
    'R', 'H', 'T', 'CC', 'CP', 'Y'])

df.pivot_table(values='CPPYPr',index = ['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area',
    'R', 'H', 'T', 'CC', 'CP', 'Y'])

df['State_Name'].unique()

TN = df[df.State_Name.str.contains('Tamil Nadu')]
TN.Crop.nunique()

TN.Crop.unique()

TN.pivot_table(values='CPPYPr',index = ['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area',
    'R', 'H', 'T', 'CC', 'CP', 'Y'])

TN.pivot_table(values='YPr',index = ['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area',
```

```
      'R', 'H', 'T', 'CC', 'CP', 'Y'])

def y_No_y_bar_plot(df, bygroup):
    dataframe_by_Group = p.crosstab(df[bygroup], columns=df["YPr"], normalize = 'index')
    dataframe_by_Group = n.round((dataframe_by_Group * 100), decimals=2)
    ax = dataframe_by_Group.plot.bar(figsize=(10,5));
    vals = ax.get_yticks()
    ax.set_yticklabels(['{:3.0f}%'.format(x) for x in vals]);
    ax.set_xticklabels(dataframe_by_Group.index,rotation = 0, fontsize = 15);
    ax.set_title('Crop Yield Prediction Vs No Crop Yield Prediction (%) (by ' +
dataframe_by_Group.index.name + ')\n', fontsize = 15)
    ax.set_xlabel(dataframe_by_Group.index.name, fontsize = 12)
    ax.set_ylabel('(%)', fontsize = 12)
    ax.legend(loc = 'upper left',bbox_to_anchor=(1.0,1.0), fontsize= 12)
    rects = ax.patches

    # Add Data Labels

    for rect in rects:
        height = rect.get_height()
        ax.text(rect.get_x() + rect.get_width()/2,
                height + 2,
                str(height)+'%',
                ha='center',
                va='bottom',
                fontsize = 12)
    return dataframe_by_Group


y_No_y_bar_plot(df, 'Season')


def c_No_y_bar_plot(df, bygroup):
    dataframe_by_Group = p.crosstab(df[bygroup], columns=df["CPPYPr"], normalize = 'index')
    dataframe_by_Group = n.round((dataframe_by_Group * 100), decimals=2)
    ax = dataframe_by_Group.plot.bar(figsize=(10,5));
    vals = ax.get_yticks()
    ax.set_yticklabels(['{:3.0f}%'.format(x) for x in vals]);
    ax.set_xticklabels(dataframe_by_Group.index,rotation = 0, fontsize = 15);
    ax.set_title('Crop Yield Production cost Vs No Crop Yield Production cost (%) (by ' +
dataframe_by_Group.index.name + ')\n', fontsize = 15)
    ax.set_xlabel(dataframe_by_Group.index.name, fontsize = 12)
    ax.set_ylabel('(%)', fontsize = 12)
    ax.legend(loc = 'upper left',bbox_to_anchor=(1.0,1.0), fontsize= 12)
    rects = ax.patches

    # Add Data Labels

    for rect in rects:
        height = rect.get_height()
        ax.text(rect.get_x() + rect.get_width()/2,
                height + 2,
                str(height)+'%',
                ha='center',
                va='bottom',
                fontsize = 12)
    return dataframe_by_Group
```

```
c_No_y_bar_plot(df, 'Season')

df.columns
c_No_y_bar_plot(df, 'District_Name')

y_No_y_bar_plot(df, 'District_Name')
```

MODULE 5 - Comparing best accuracy and entropy

```
#import library packages
import pandas as p
import matplotlib.pyplot as plt
import seaborn as s
import numpy as n
#read the given dataset
df = p.read_csv("df.csv")

df.rename(columns={'Mean Temp':'T'}, inplace=True)
df.rename(columns={'Average Humidity':'H'}, inplace=True)
df.rename(columns={'rainfall':'R'}, inplace=True)
df.rename(columns={'Cost of Cultivation (`/Hectare) C2':'CC'}, inplace=True)
df.rename(columns={'Cost of Production (`/Quintal) C2':'CP'}, inplace=True)
df.rename(columns={'Yield (Quintal/ Hectare) ':'Y'}, inplace=True)
df.rename(columns={'cost of production per yield':'CPPY'}, inplace=True)

from sklearn.preprocessing import LabelEncoder
var_mod = ['Y']
le = LabelEncoder()
for i in var_mod:
   df[i] = le.fit_transform(df[i]).astype(str)
df['YPr']= df.Y.map({'13':0, '7':0, '11':0, '4':0, '23':0, '39':1, '10':0, '18':0, '36':1, '47':1, '8':0, '3':0,
      '38':1, '46':1, '5':0, '9':0, '2':0, '44':1, '17':0, '41':1, '6':0, '16':0, '35':1, '19':0,
      '0':0, '43':1, '12':0, '45':1, '25':0, '33':1, '29':0, '37':1, '32':1, '21':0, '42':1,
      '48':1, '30':1, '34':1, '26':0, '20':0, '31':1, '24':0, '27':0, '22':0, '40':1, '28':0,
      '1':0, '14':0, '15':0})



df.columns

from sklearn.preprocessing import LabelEncoder
var_mod = ['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area',
      'R', 'H', 'T', 'CC', 'CP', 'Y','CPPY']
le = LabelEncoder()
for i in var_mod:
   df[i] = le.fit_transform(df[i]).astype(str)
df.head()

#According to the cross-validated MCC scores, the random forest is the best-performing model, so now
let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report, matthews_corrcoef,
cohen_kappa_score, accuracy_score, average_precision_score, roc_auc_score

# Prediction of Crop by yield
```

```
X = df.drop(labels='YPr', axis=1)
#Response variable
y = df.loc[:,'YPr']

del df
#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)

#for our convienient we delete X,y variable for differentiate confusion
del X, y

Logistic Regression

from sklearn.linear_model import LogisticRegression
logR= LogisticRegression()

logR.fit(X_train,y_train)

predictR = logR.predict(X_test)
print(classification_report(y_test,predictR))
x = (accuracy_score(y_test,predictR)*100)


print('Accuracy result is', x)
print("")

print("")
print(confusion_matrix(y_test,predictR))

Decision Tree Classifier

from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()

dtree.fit(X_train, y_train)

predictDT = dtree.predict(X_test)
print(classification_report(y_test,predictDT))
x = (accuracy_score(y_test,predictDT)*100)



print('Accuracy result is', x)
print("")
print(confusion_matrix(y_test,predictDT))


RandomForest Classifier

from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()

rf.fit(X_train, y_train)

predictrf = rf.predict(X_test)
```

```
print(classification_report(y_test,predictrf
                            ))
x = (accuracy_score(y_test,predictrf)*100)


print('Accuracy result is', x)

print("")
print(confusion_matrix(y_test,predictrf))
```

K-Neighbors Classifier

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()

knn.fit(X_train, y_train)

predictknn = knn.predict(X_test)

print(classification_report(y_test,predictknn
                            ))
x = (accuracy_score(y_test,predictknn)*100)


print('Accuracy result is', x)

print("")

print(confusion_matrix(y_test,predictknn))
```

Support Vector Classifier

```
from sklearn.svm import SVC
s = SVC()

s.fit(X_train, y_train)

predictSV = s.predict(X_test)

print(classification_report(y_test,predictSV
                            ))
x = (accuracy_score(y_test,predictSV)*100)


print('Accuracy result is', x)



print("")
print(confusion_matrix(y_test,predictSV))


# Prediction of Crop by Cost production
```

```
#import library packages
import pandas as p
import matplotlib.pyplot as plt
import seaborn as s
import numpy as n
#read the given dataset
df = p.read_csv("df.csv")

df.rename(columns={'Mean Temp':'T'}, inplace=True)
df.rename(columns={'Average Humidity':'H'}, inplace=True)
df.rename(columns={'rainfall':'R'}, inplace=True)
df.rename(columns={'Cost of Cultivation (`/Hectare) C2':'CC'}, inplace=True)
df.rename(columns={'Cost of Production (`/Quintal) C2':'CP'}, inplace=True)
df.rename(columns={'Yield (Quintal/ Hectare) ':'Y'}, inplace=True)
df.rename(columns={'cost of production per yield':'CPPY'}, inplace=True)

from sklearn.preprocessing import LabelEncoder
var_mod = ['CPPY']
le = LabelEncoder()
for i in var_mod:
   df[i] = le.fit_transform(df[i]).astype(str)
df['CPPYPr']= df.CPPY.map({'126':0, '72':0, '200':0, '78':0, '144':0, '212':0, '149':0, '151':0, '178':0,
'312':0,
      '187':0, '69':0, '205':0, '276':1, '76':0, '134':0, '66':0, '254':1, '150':0, '291':1,
      '155':0, '106':0, '87':0, '119':0, '183':0, '170':0, '28':0, '265':1, '180':0, '36':0,
      '293':1, '326':1, '148':0, '234':1, '3':0, '124':0, '111':0, '182':0, '166':0, '169':0,
      '270':1, '243':1, '98':0, '230':0, '282':1, '334':1, '188':0, '65':0, '34':0, '328':1,
      '160':0, '115':0, '89':0, '152':0, '240':1, '141':0, '233':1, '32':0, '281':1, '335':1,
      '109':0, '47':0, '136':0, '112':0, '261':1, '229':0, '42':0, '325':1, '213':0, '175':0,
      '125':0, '196':0, '292':1, '82':0, '235':1, '79':0, '105':0, '123':0, '210':0, '39':0,
      '146':0, '185':0, '201':0, '41':0, '164':0, '184':0, '222':0, '250':0, '301':1,
      '218':0, '217':0, '168':0, '264':1, '46':0, '103':0, '284':1, '198':0, '56':0, '171':0,
      '331':1, '256':1, '99':0, '58':0, '247':1, '286':1, '67':0, '133':0, '143':0, '204':0,
      '227':0, '194':0, '6':0, '280':1, '225':0, '48':0, '258':1, '94':0, '244':1, '294':1,
      '215':0, '132':0, '277':1, '71':0, '219':0, '315':1, '97':0, '91':0, '156':0, '289':1,
      '100':0, '295':1, '147':0, '214':0, '19':0, '223':0, '90':0, '269':1, '300':1, '114':0,
      '135':0, '173':0, '55':0, '113':0, '127':0, '73':0, '177':0, '274':1, '118':0, '191':0,
      '145':0, '307':1, '162':0, '228':0, '50':0, '5':0, '248':1, '203':0, '61':0, '129':0,
      '192':0, '83':0, '158':0, '206':0, '242':1, '267':1, '137':0, '92':0, '176':0, '298':1,
      '25':0, '296':1, '186':0, '239':1, '193':0, '165':0, '310':1, '20':0, '287':1, '75':0,
      '107':0, '271':1, '138':0, '121':0, '241':1, '74':0, '43':0, '232':0, '154':0, '181':0,
      '195':0, '102':0, '237':1, '110':0, '268':1, '49':0, '104':0, '64':0, '318':1, '13':0,
      '128':0, '153':0, '31':0, '93':0, '309':1, '257':0, '52':0, '262':1, '202':0, '174':0,
      '101':0, '224':0, '86':0, '45':0, '263':1, '167':0, '17':0, '29':0, '30':0, '54':0,
      '303':1, '163':0, '251':1, '142':0, '273':1, '96':0, '35':0, '327':1, '53':0, '15':0,
      '190':0, '308':1, '226':0, '139':0, '62':0, '2':0, '1':0, '27':0, '84':0, '285':1,
      '246':1, '23':0, '189':0, '299':1, '231':0, '24':0, '16':0, '333':1, '306':1, '18':0,
      '288':1, '199':0, '260':1, '323':1, '37':0, '278':1, '324':1, '140':0, '4':0, '245':1,
      '322':1, '329':1, '159':0, '80':0, '197':0, '275':1, '11':0, '40':0, '33':0, '0':0,
      '320':1, '290':1, '68':0, '220':0, '21':0, '330':1, '12':0, '157':0, '302':1, '44':0,
      '221':0, '216':0, '279':1, '63':0, '313':1, '311':1, '332':1, '266':1, '238':1,
      '211':0, '172':0, '14':0, '117':0, '161':0, '122':0, '60':0, '321':1, '95':0, '208':0,
      '272':1, '131':0, '207':0, '81':0, '314':1, '51':0, '283':1, '120':0, '38':0, '9':0,
      '130':0, '70':0, '179':0, '7':0, '10':0, '108':0, '255':1, '77':0, '88':0, '22':0,
      '57':0, '85':0, '304':1, '253':1, '249':0, '297':1, '116':0, '305':1, '317':1, '252':1,
      '236':1, '209':0, '259':1, '26':0, '319':1, '316':1, '59':0, '336':1, '8':0})
```

```
from sklearn.preprocessing import LabelEncoder
var_mod = ['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area',
      'R', 'H', 'T', 'CC', 'CP', 'Y','CPPY']
le = LabelEncoder()
for i in var_mod:
   df[i] = le.fit_transform(df[i]).astype(str)
df.head()

#According to the cross-validated MCC scores, the random forest is the best-performing model, so now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report, matthews_corrcoef, cohen_kappa_score, accuracy_score, average_precision_score, roc_auc_score

X = df.drop(labels='CPPYPr', axis=1)
#Response variable
y = df.loc[:,'CPPYPr']

del df
#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)

#for our convienient we delete X,y variable for differentiate confusion
del X, y

Logistic Regression

from sklearn.linear_model import LogisticRegression
logR= LogisticRegression()

logR.fit(X_train,y_train)

predictR = logR.predict(X_test)
print(classification_report(y_test,predictR))
x = (accuracy_score(y_test,predictR)*100)


print('Accuracy result is', x)
print("")

print("")
print(confusion_matrix(y_test,predictR))

Decision Tree Classifier

from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()

dtree.fit(X_train, y_train)

predictDT = dtree.predict(X_test)
print(classification_report(y_test,predictDT))
x = (accuracy_score(y_test,predictDT)*100)
```

```
print('Accuracy result is', x)
print("")
print(confusion_matrix(y_test,predictDT))
```

RandomForest Classifier

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()

rf.fit(X_train, y_train)

predictrf = rf.predict(X_test)


print(classification_report(y_test,predictrf
                            ))
x = (accuracy_score(y_test,predictrf)*100)



print('Accuracy result is', x)

print("")
print(confusion_matrix(y_test,predictrf))
```

K-Neighbors Classifier

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()

knn.fit(X_train, y_train)

predictknn = knn.predict(X_test)

print(classification_report(y_test,predictknn
                            ))
x = (accuracy_score(y_test,predictknn)*100)



print('Accuracy result is', x)

print("")

print(confusion_matrix(y_test,predictknn))
```

Support Vector Classifier

```
from sklearn.svm import SVC
s = SVC()

s.fit(X_train, y_train)

predictSV = s.predict(X_test)

print(classification_report(y_test,predictSV
```

```
                    ))
x = (accuracy_score(y_test,predictSV)*100)


print('Accuracy result is', x)



print("")
print(confusion_matrix(y_test,predictSV))
```

MODULE – GUI

```
from tkinter import *
import numpy as np
import pandas as pd

df = pd.read_csv("re.csv")

df

df.shape

df.columns



l1=['Sugercane', 'Rice', 'Tobacco', 'Wheat', 'Coconut', 'Yam',
    'Sweetpotato', 'Tapioca', 'Turmeric', 'WaterMelon', 'Urad', 'Varagu',
    'Banana']

l2=['DINDIGUL', 'THE NILGIRIS', 'KARUR', 'KRISHNAGIRI', 'MADURAI',
    'NAGAPATTINAM', 'NAMAKKAL', 'PERAMBALUR', 'PUDUKKOTTAI', 'SALEM',
    'THANJAVUR', 'THENI', 'THIRUVARUR', 'TIRUCHIRAPPALLI', 'TIRUNELVELI',
    'TIRUPPUR', 'TIRUVANNAMALAI', 'VELLORE', 'VILLUPURAM', 'VIRUDHUNAGAR',
    'COIMBATORE', 'CUDDALORE', 'DHARMAPURI', 'ERODE', 'KANNIYAKUMARI',
    'KANCHIPURAM', 'ARIYALUR', 'RAMANATHAPURAM', 'SIVAGANGA', 'THIRUVALLUR',
    'TUTICORIN']



l4=[ 'Summer', 'Winter', 'Kharif', 'Rabi', 'Whole Year',
    'Autumn']

l5=['TA50', 'TA60', 'TA80']

l6=['HB25', 'HA25', 'HA60', 'HA80']

df['Class'].unique()

cropyc=['Cost Yield is ten tons and Cost Production is ten lakshs',
    'Cost Yield is one ton and Cost Production is twenty thousand',
    'Cost Yield is twenty tons and Cost Production is fourty lakshs',
    'Cost Yield is two tons and Cost Production is fourty thousand',
    'Cost Yield is three tons and Cost Production is sixty thousand',
```

```
        'Cost Yield is six tons and Cost Production is six lakshs ',
        'Cost Yield is two tons and Cost Production is fourty  thousand',
        'Cost Yield is five tons and Cost Production is one lakshs',
        'Cost Yield is ten tons and Cost Production is two lakshs']

l7=['Sugercane', 'Rice', 'Tobacco', 'Wheat', 'Coconut', 'Yam',
        'Sweetpotato', 'Tapioca', 'Turmeric', 'WaterMelon', 'Urad', 'Varagu',
        'Banana','DINDIGUL', 'THE NILGIRIS', 'KARUR', 'KRISHNAGIRI', 'MADURAI',
        'NAGAPATTINAM', 'NAMAKKAL', 'PERAMBALUR', 'PUDUKKOTTAI', 'SALEM',
        'THANJAVUR', 'THENI', 'THIRUVARUR', 'TIRUCHIRAPPALLI', 'TIRUNELVELI',
        'TIRUPPUR', 'TIRUVANNAMALAI', 'VELLORE', 'VILLUPURAM', 'VIRUDHUNAGAR',
        'COIMBATORE', 'CUDDALORE', 'DHARMAPURI', 'ERODE', 'KANNIYAKUMARI',
        'KANCHIPURAM', 'ARIYALUR', 'RAMANATHAPURAM', 'SIVAGANGA', 'THIRUVALLUR',
        'TUTICORIN','Summer', 'Winter', 'Kharif', 'Rabi', 'Whole Year',
        'Autumn','TA50', 'TA60', 'TA80','HB25', 'HA25', 'HA60', 'HA80']

l8=[]
for x in range(0,len(l7)):
    l8.append(0)


df.replace({'Class':{'Cost Yield is ten tons and Cost Production is ten lakshs':0,
        'Cost Yield is one ton and Cost Production is twenty thousand':1,
        'Cost Yield is twenty tons and Cost Production is fourty lakshs':2,
        'Cost Yield is two tons and Cost Production is fourty thousand':3,
        'Cost Yield is three tons and Cost Production is sixty thousand':4,
        'Cost Yield is six tons and Cost Production is six lakshs ':5,
        'Cost Yield is two tons and Cost Production is fourty  thousand':6,
        'Cost Yield is five tons and Cost Production is one lakshs':7,
        'Cost Yield is ten tons and Cost Production is two lakshs':8}},inplace=True)

X= df[l7]


y = df[["Class"]]
np.ravel(y)

# TRAINING DATA tr ----------------------------------------------------------------------------
tr=pd.read_csv("tere.csv")

X_test= tr[l7]
y_test = tr[["Class"]]
np.ravel(y_test)

def randomforest():
    from sklearn.ensemble import RandomForestClassifier
    clf4 = RandomForestClassifier()
    clf4 = clf4.fit(X,np.ravel(y))

    # calculating accuracy-------------------------------------------------------------------
    from sklearn.metrics import accuracy_score
    y_pred=clf4.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred,normalize=False))
    # ---------------------------------------------------

    terms = [Dist.get(),Season.get(),Tl.get(),Hl.get(),Crop.get()]
```

```
    for k in range(0,len(l7)):
        for z in terms:
            if(z==l7[k]):
                l8[k]=1

    inputtest = [l8]
    predict = clf4.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(cropyc)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t2.delete("1.0", END)
        t2.insert(END, cropyc[a])
    else:
        t2.delete("1.0", END)
        t2.insert(END, "Not Found")


def LogisticRegression():
    from sklearn.linear_model import LogisticRegression
    gnb= LogisticRegression()

    gnb=gnb.fit(X,np.ravel(y))

    # calculating accuracy---------------------------------------------------------------------
    from sklearn.metrics import accuracy_score
    y_pred=gnb.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred,normalize=False))
    # ---------------------------------------------------

    terms = [Dist.get(),Season.get(),Tl.get(),Hl.get(),Crop.get()]
    for k in range(0,len(l7)):
        for z in terms:
            if(z==l7[k]):
                l8[k]=1

    inputtest = [l8]
    predict = gnb.predict(inputtest)
    predicted=predict[0]

    h='no'
    for a in range(0,len(cropyc)):
        if(predicted == a):
            h='yes'
            break

    if (h=='yes'):
        t3.delete("1.0", END)
        t3.insert(END, cropyc[a])
    else:
        t3.delete("1.0", END)
        t3.insert(END, "Not Found")
```

```
root = Tk()
root.configure(background='black')

# gui_stuff----------------------------------------------------------------------------

# entry variables

Dist = StringVar()
Dist.set(None)

Season = StringVar()
Season.set(None)

Crop =StringVar()
Crop.set(None)

Tl = StringVar()
Tl.set(None)

Hl = StringVar()
Hl.set(None)

Yl = StringVar()
Yl.set(None)



fn= StringVar()
vn= StringVar()

rl= IntVar()

# Heading
w2 = Label(root, justify=LEFT, text="Prediction of Yield and Yield cost by Crop using Machine Learning", fg="white", bg="red")
w2.config(font=("Elephant", 20))
w2.grid(row=1, column=0, columnspan=2, padx=100)
w2 = Label(root, justify=LEFT, text="Tamilnadu Agricultural Department", fg="green")
w2.config(font=("Times Roman", 15))
w2.grid(row=2, column=0, columnspan=2, padx=100)

# labels
FNLb = Label(root, text="Farmer Name",fg="yellow", bg="black")
FNLb.grid(row=5, column=0, pady=15, sticky=W)

VNLb = Label(root, text="Village Name",fg="yellow", bg="black")
VNLb.grid(row=6, column=0, pady=15, sticky=W)




RFLb = Label(root, text="Rainfall Value",fg="yellow", bg="black")
RFLb.grid(row=8, column=1, pady=15, sticky=W)
```

```
# labels
dtLb = Label(root, text="District Name",fg="yellow", bg="black")
dtLb.grid(row=8, column=0, pady=15, sticky=W)

seLb = Label(root, text="Season Details",fg="yellow", bg="black")
seLb.grid(row=9, column=0, pady=15, sticky=W)

tlLb = Label(root, text="Temperature Level",fg="yellow", bg="black")
tlLb.grid(row=5, column=1, pady=15, sticky=W)

hlLb = Label(root, text="Humidity Level",fg="yellow", bg="black")
hlLb.grid(row=6, column=1, pady=15, sticky=W)


crLb = Label(root, text="Crop Details",fg="yellow", bg="black")
crLb.grid(row=7, column=1, pady=15, sticky=W)

destreeLb = Label(root, text="LogisticRegression", fg="white", bg="red")
destreeLb.grid(row=14, column=0, pady=10,sticky=W)

ranfLb = Label(root, text="RandomForest", fg="white", bg="red")
ranfLb.grid(row=15, column=0, pady=10, sticky=W)


# entries
OPTIONS1 = sorted(l1)
OPTIONS2 = sorted(l2)

OPTIONS4 = sorted(l4)
OPTIONS5 = sorted(l5)
OPTIONS6 = sorted(l6)




FNEn = Entry(root, textvariable=fn)
FNEn.grid(row=5, column=0)

VNEn = Entry(root, textvariable=vn)
VNEn.grid(row=6, column=0)

RFEn = Entry(root, textvariable=rl)
RFEn.grid(row=8, column=1)

crEn = OptionMenu(root, Crop,*OPTIONS1)
crEn.grid(row=7, column=1)


dtEn = OptionMenu(root, Dist,*OPTIONS2)
dtEn.grid(row=8, column=0)

seEn = OptionMenu(root, Season,*OPTIONS4)
seEn.grid(row=9, column=0)
```

```
tlEn = OptionMenu(root, Tl,*OPTIONS5)
tlEn.grid(row=5, column=1)

hlEn = OptionMenu(root, Hl,*OPTIONS6)
hlEn.grid(row=6, column=1)

dst = Button(root, text="LogisticRegression Algorithm",
command=DecisionTree,bg="cyan",fg="green")
dst.grid(row=14, column=1,padx=10)

rnf = Button(root, text="Randomforest Algorithm", command=randomforest,bg="cyan",fg="green")
rnf.grid(row=15, column=1,padx=10)


#textfileds
t1 = Text(root, height=1, width=70,bg="orange",fg="black")
t1.grid(row=14, column=0, padx=10)

t2 = Text(root, height=1, width=70,bg="orange",fg="black")
t2.grid(row=15, column=0 , padx=10)

root.mainloop()
```

PAPER PUBLICATION STATUS

Publication process not yet started