

# Estimate the Crop Yield using Data Analytics

*Submitted by*

**T.PRASANNA REDDY-111719104159**

**V.GREESHMITHA - 111719104173**

**VEMULA KHYATHI – 111719104176**

**P.LAHARI-111719104121**

**T.VYSHNAVI-111719104166**

## **ABSTRACT:**

Among worldwide, agriculture has the major responsibility for improving the economic contribution of the nation. However, still the most agricultural fields are under developed due to the lack of deployment of ecosystem control technologies. Due to these problems, the crop production is not improved which affects the agriculture economy. Hence a development of agricultural productivity is enhanced based on the plant yield prediction. To prevent this problem, Agricultural sectors have to predict the crop from given dataset using machine learning techniques. The analysis of dataset by supervised machine learning technique(SMLT) to capture several information's like, variable identification, uni-variate analysis, bi-variate and multi-variate analysis, missing value treatments etc. A comparative study between machine learning algorithms had been carried out in order to determine which algorithm is the most accurate in predicting the best crop. The results show that the effectiveness of the proposed machine learning algorithm technique can be compared with best accuracy with entropy calculation, precision, Recall, F1 Score, Sensitivity, Specificity.

**Keywords:** dataset, Machine learning-Classification method

## TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	<b>ACKNOWLEDGEMENT</b>	<b>3</b>
	<b>ABSTRACT</b>	<b>4</b>
	<b>LIST OF FIGURES</b>	<b>6</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>7</b>
	1.1 Project Overview	
	1.2 Purpose	
<b>2.</b>	<b>LITERATURE SURVEY</b>	<b>8</b>
	2.1 Existing Problem	
	2.2 References	
	2.3 Problem statement definition	
<b>3.</b>	<b>IDEATION &amp; PROPOSED SOLUTION</b>	<b>12</b>
	3.1 Empathy Map Canvas	
	3.2 Ideation & Brainstorming	
	3.3 Proposed Solution	
	3.4 Problem Solution fit	
<b>4.</b>	<b>REQUIREMENT ANALYSIS</b>	<b>15</b>
	4.1 Functional requirements	
	4.2 Non-Functional requirements	
<b>5.</b>	<b>PROJECT DESIGN</b>	<b>21</b>
	5.1 Data Flow Diagrams	
	5.2 Solution & Technical Architecture	
	5.3 User Stories	
<b>6.</b>	<b>PROJECT PLANNING AND SCHEDULING</b>	<b>36</b>
	6.1 Sprint Planning & Estimation	
	6.2 Sprint Delivery Schedule	

	6.3 Reports from JIRA	
<b>7.</b>	<b>CODING &amp; SOLUTIONING</b>	
	7.1 Feature 1	
	7.2 Feature 2	
	7.3 Database Schema	
<b>8.</b>	<b>TESTING</b>	
	8.1 Test Cases	
	8.2 User Acceptance Testing	
<b>9.</b>	<b>RESULTS</b>	
	9.1 Performance Metrics	
<b>10.</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>	<b>37</b>
<b>11.</b>	<b>CONCLUSION</b>	
<b>12.</b>	<b>FUTURE SCOPE</b>	
<b>13.</b>	<b>APPENDIX</b>	

## **1.Introduction:**

In developing countries, farming is considered as the major source of revenue for many people. In modern years, the agricultural growth is engaged by several innovations, environments, techniques and civilizations. In addition, the utilization of information technology may change the condition of decision making and thus farmers may yield the best way. For decision making process, data mining techniques related to the agriculture are used. Data mining is a process of extracting the most significant and useful information from the huge amount of datasets. Nowadays, we used machine learning approach with developed in

crop or plant yield prediction since agriculture has different data like soil data, crop data, and weather data. Plant growth prediction is proposed for monitoring the plant yield effectively through the machine learning techniques.

It is also applicable for the automated process of farming is the beginning of a new era in Bangladesh that will be suitable for the farmers who seek experts to take suggestion about the appropriate crop on specific location of their land and don't want to forget any step of the cultivation throughout the process. Although, the opinion from experts is the most convenient way, this application is designed to give accurate solution in fastest manner possible. This research's main objective is to bring farming process a step closer to the digital platform.

### 1.1 Project overview:

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

Data scientists use many different kinds of machine learning algorithms to discover patterns in python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they “learn” about data to make predictions: supervised and unsupervised learning. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function from input variables(X) to discrete output variables(y). In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.

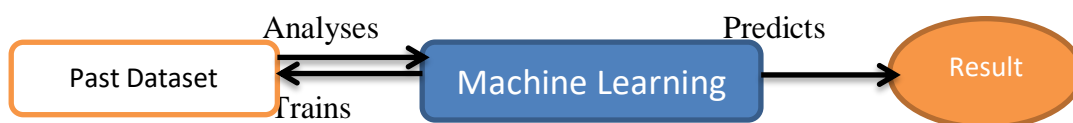


Fig: Process of Machine learning

Supervised Machine Learning is the majority of practical machine learning uses supervised learning. Supervised learning is where have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output is  $y = f(X)$ . The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include logistic regression, multi-class classification, Decision Trees and support vector machines etc. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. Supervised learning problems can be further grouped into Classification problems. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for categorical for classification. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. A classification problem is when the output variable is a category, such as “red” or “blue”.

Agriculture is one of the most important occupations practiced in our country. It is the broadest economic sector and plays an important role in overall development of the country. About 60 % of the land in the country is used for agriculture in order to suffice the needs of 1.2 billion people. Thus, modernization of agriculture is very important and thus will lead the farmers of our country towards profit. Data analytic (DA) is the process of examining data sets in order to draw conclusions about the information they contain, increasingly with the aid of specialized systems and software. Earlier yield prediction was performed by considering the farmer's experience on a particular field and crop. However, as the conditions change day by day very rapidly, farmers are forced to cultivate more and more crops. Being this as the current situation, many of them don't have enough knowledge about the new crops and are not completely aware of the benefits they get while farming them. Also, the farm productivity can be increased by understanding and forecasting crop performance in a variety of environmental conditions. Thus, the proposed system takes the location of the user as an input. From the location, the nutrients of the soil such as Nitrogen, Phosphorous, Potassium is obtained. This static data is the crop production and data related to demands of various crops obtained from various websites. It applies machine learning and prediction algorithm to identify the pattern among data and then process it as per input conditions.

### **Preparing the Dataset:**

The demo dataset is now supplied to machine learning model on the basis of this data set the model is trained. Every new detail filled at the time of application form acts as a test data set. After the operation of testing, model prediction based upon the inference it concludes on the basis of the training data sets. Satellite Imagery (Remote Sensing Data), has been widely used for predicting crop yield. This dataset is collected using the sensors mounted on satellites or planes, which detect the energy (electromagnetic waves), reflected or

diffracted from surface of the earth. Remote sensing data has a lot of energy bands to offer, but mainly only few of them have been used for crop yield prediction. Yet, there are some people who have tried generating relevant features using the bands which are typically ignored, and they have been successful with improving results with that. In case of this dataset, most people rarely explore the high-order moments of the features. Based on these datasets people have used algorithms like Regression models, Random Forest and Nearest Neighbor etc.

Table shows details of the datasets:

Variable	Description
Crop	Crop name
State Name	Indian state name
District Name	District name list of each state
Cost of Cultivation (₹/Hectare) C2	Cultivation amount for C2 Scheme
Cost of Production (₹/Quintal) C2	Production amount for A2+FL Scheme
Yield (Quintal/ Hectare)	Yield of crop
Crop year	Crop year list
District Name	District name for each state
Area	Total area of each place
Rainfall	Water availability of each crop
Average humidity	directly influences the water relations of plant and indirectly affects leaf growth
Mean Temperature	Climate of each crop
Cost Production of per yield crop	Cost of crop yield

## 1.2 Purpose:

Agriculture is the most important sector that influences the economy of India. It contributes to 18% of India's Gross Domestic Product (GDP) and gives employment to 50% of the population of India. People of India are practicing Agriculture for years but the results are never satisfying due to various factors that affect the crop yield. To fulfill the needs of around 1.2 billion people, it is very important to have a good yield of crops. Due to factors like soil type, precipitation, seed quality, lack of technical facilities etc. the crop yield is directly influenced. To focus on implementing crop yield prediction system by using Machine learning techniques by doing analysis on agriculture dataset. For evaluating performance Accuracy is used as one of the factors. The classifiers are further compared with the values of Precision, Recall and F1score. Lesser the value of error, more accurate the algorithm will work. The result is based on comparison among the classifiers.

## Scope:

The scope of this project is to investigate a dataset of crop records for agricultural sector using machine learning technique. To identifying crop predicting by farmer is more difficult. We try to reduce this risk factor behind selection of the crop.

**Objectives:**

- Data validation
- Data Cleaning/ Preparing
- Data Visualization
- Using more algorithm with comparing to predict more accuracy (Like random forest, Decision tree Logistic classification algorithm)

## **2. LITERATURE SURVEY**

**General:**

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources

and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them.

### **Review of Literature Survey:**

**Title:** Estimation of Organic Matter Content in Coastal Soil Using Reflectance Spectroscopy Research

**Author:** ZHENG Guanghui<sup>1</sup>, Dongryeol RYU<sup>2,\*</sup>, JIAO Caixia<sup>1</sup> and HONG Changqiao<sup>1</sup>



**Year:** 2015

**Description:**

Rapid determination of soil organic matter (SOM) using regression models based on soil reflectance spectral data serves an important function in precision agriculture. “Deviation of arch” (DOA)-based regression and partial least squares regression (PLSR) are two modeling approaches to predict SOM. However, few studies have explored the accuracy of the DOA-based regression and PLSR models. Therefore, the DOA-based regression and PLSR were applied to the visible near-infrared (VNIR) spectra to estimate SOM content in the case of various dataset divisions. A two-fold cross-validation scheme was adopted and repeated 10000 times for rigorous evaluation of the DOA-based models in comparison with the widely used PLSR model. Soil samples were collected for SOM analysis in the coastal area of northern Jiangsu Province, China. The results indicated that both modelling methods provided reasonable estimation of SOM, with PLSR outperforming DOA-based regression in general. However, the performance of PLSR for the validation dataset decreased more noticeably. Among the four DOA-based regression models, a linear model provided the best estimation of SOM and a cutoff of SOM content ( $19.76 \text{ g kg}^{-1}$ ), and the performance for calibration and validation datasets was consistent. As the SOM content exceeded  $19.76 \text{ g kg}^{-1}$ , SOM became more effective in masking the spectral features of other soil properties to a certain extent. This work confirmed that reflectance spectroscopy combined with PLSR could serve as a non-destructive and cost-efficient way for rapid determination of SOM when hyper spectral data were available. The DOA-based model, which requires only 3 bands in the visible spectra, also provided SOM estimation with acceptable accuracy.

**Title:** Preliminary Study of Soil Available Nutrient Simulation Using a Modified WOFOST Model and Time-Series Remote Sensing Observations

**Author:** Zhiqiang Cheng 1,2 ID , Jihua Meng 1,\*, Yanyou Qiao 1, Yiming Wang 1,2, Wenquan Dong 1 and Yanxin Han 1,2

**Year:** 2017

**Description:**

The approach of using multispectral remote sensing (RS) to estimate soil available nutrients (SANs) has been recently developed and shows promising results. This method overcomes the limitations of commonly used methods by building a statistical model that connects RS-based crop growth and nutrient content. However, the stability and accuracy of this model require improvement. In this article, we replaced the statistical model by integrating the World Food Studies (WOFOST) model and time series of remote sensing (T-RS) observations to ensure stability and accuracy. Time series of HJ-1 A/B data was assimilated into the WOFOST model to extrapolate crop growth simulations from a single point to a large area using a specific assimilation method. Because nutrient-limited growth within the growing season is required and the SAN parameters can only be used at the end of the growing season in the original model, the WOFOST model was modified. Notably, the calculation order was changed, and new soil nutrient uptake algorithms were implemented in the model for nutrient-limited growth estimation. Finally, experiments were conducted in the spring maize plots of Hongxing Farm to analyze the effects of nutrient stress on crop growth and the SAN simulation accuracy. The results confirm the differences in crop growth status caused by a lack of soil nutrients. The new approach can take advantage of these differences to provide better SAN estimates. In general, the new approach can overcome the limitations of existing methods and simulate the SAN status with reliable accuracy.

**Title:** Distinguishing Heavy-Metal Stress Levels in Rice Using Synthetic Spectral Index Responses to Physiological Function Variations

**Author:** Ming Jin, Xiangnan Liu, Ling Wu, and Meiling Liu

**Year:** 2016

**Description:**

Accurately assessing the heavy-metal contamination in crops is crucial to food security. This study provides a method to distinguish heavy-metal stress levels in rice using the variations of two physiological functions as discrimination indices, which are obtained by assimilation of remotely sensed data with a crop growth model. Two stress indices, which correspond to  $\text{daily total CO}_2$  assimilation and dry-matter conversion coefficient, were incorporated into the World Food Study (WOFOST) crop growth model and calculated by assimilating the model with leaf area index (LAI), which was derived from time-series HJ1-CCD data. The stress levels are not constant with rice growth; thus, to improve the reliability, the two stress indices were obtained at both the first and the latter half periods of rice growth. To compare the stress indices of different stress levels, a synthetic stress index was established by combining the two indices; then, three types of stress index discriminant spaces based on the synthetic index of different growth periods were constructed, in which the two-dimensional discriminant space based on two growth periods showed the highest accuracy, with a misjudgment rate of 4.5%. When the discrimination rules were applied at a regional scale, the average correct discrimination rate was 95.0%.

**2.1 Existing System:**

It presents a crop/weeds classification approach based on a three-steps procedure. The first step is a robust pixel-wise segmentation (i.e., soil/plant) and image patches containing plants are extracted in the second step. The third step, a deep CNN for crop/weed classification is used. The extracted blobs in the masked image containing plants information

are fed to a CNN classifier based on a fine-tuned model of VGG-16 exploiting the ability of deep CNN in object classification and to reduce the limitations of CNNs in generalizing when a limited amount of data is available. The classification step can then be specialized to the types of plants needed by the application scenario. It evaluated the complete pipeline, including the first background removal phase and the subsequent classification stage. Experimental results demonstrate that can achieve good classification results on challenging data.

Precision agriculture is gaining increasing attention because of the possible reduction of agricultural inputs (e.g., fertilizers and pesticides) that can be obtained by using high-tech equipment, including robots. To focus on an agricultural robotics system that addresses the weeding problem by means of selective spraying or mechanical removal of the detected weeds. To describe a deep learning based method to allow a robot to perform an accurate weed/crop classification using a sequence of two Convolutional Neural Networks (CNNs) applied to RGB images. The first network, based on encoder-decoder segmentation architecture, performs a pixel wise, plant-type agnostic, segmentation between vegetation and soil that enables to extract a set of connected blobs representing plant instances.

#### **Drawbacks:**

- It can't determine to improve the classification accuracy of our pipeline.
- Connecting the bridge manually and some corruption are happened.
- Private sectors domination high, profit low and credits not getting concern farmer.

## **2.2 REFERENCES**

- P.Priya, U.Muthaiah M.Balamurugan . Predicting yield of the crop using machine learning algorithm. International Journal of Engineering Science Research Technology.
- J.Jeong, J.Resop , N.Mueller and team . Random forests for global and regional crop yield prediction.PLoS ONE Journal.
- Narayanan Balkrishnan and Dr. Govindarajan Muthukumarasamy . Crop production Ensemble Machine Learning model for prediction. International Journal of Computer Science and Software Engineering (IJSSE).
- S.Veenadhari , Dr. Bharat Misra , Dr. CD Singh. Machine learning approach for forecasting crop yield based on climatic parameters. International Conference on Computer Communication and Informatics (ICCCI).
- Shweta K Shahane , Prajakta V Tawale . Prediction On Crop Cultivation. International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE) Volume 5, Issue 10, October 2016.
- D Ramesh ,B Vishnu Vardhan. Analysis Of Crop Yield Prediction Using Data Mining

Techniques. IJRET: International Journal of Research in Engineering and Technology.

## 2.3 Problem Statement Definition

Machine Learning based on prior crop prediction, soil quality analysis to achieve high crop yield through out technology solution. The main objectives of this project is to predict crop-yield which can be extremely useful to farmers in planning for harvest and sale of grain harvest.

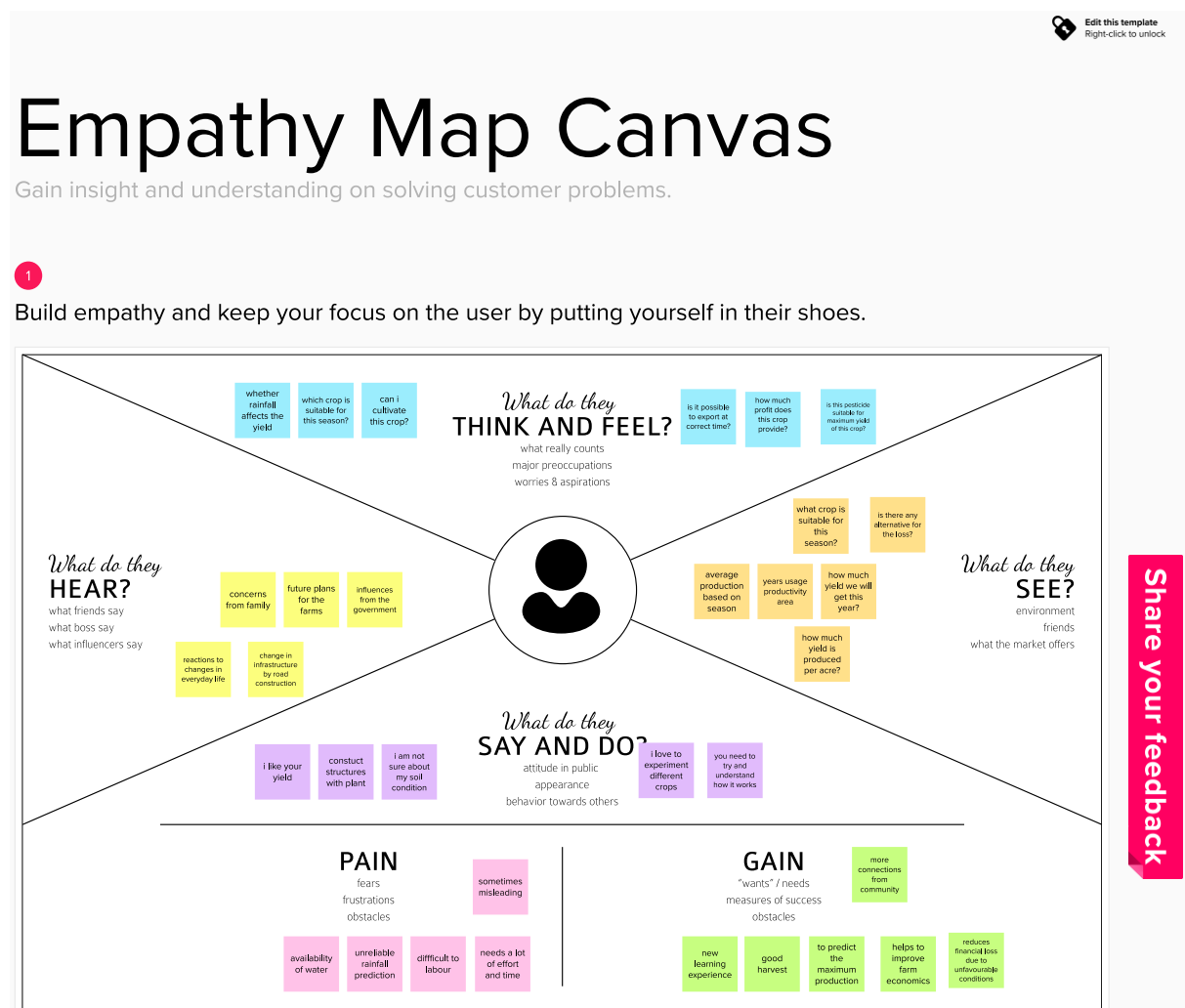
## 3. IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas

### 3.2 Ideation & Brainstorming

### 3.3 Proposed Solution

### 3.4 Problem Solution fit




Template




## Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

 10 minutes to prepare

 1 hour to collaborate

 2-8 people recommended

Share template feedback

➔

### Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

🕒 10 minutes

A

#### Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B

#### Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

C

#### Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

Open article ➔

1

### Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

PROBLEM

How accurately we can determine the flight delay from our model?




#### Key rules of brainstorming

To run a smooth and productive session


 Stay in topic.

 Encourage wild ideas.

 Defer judgment.

 Listen to others.

 Go for volume.

 If possible, be visual.

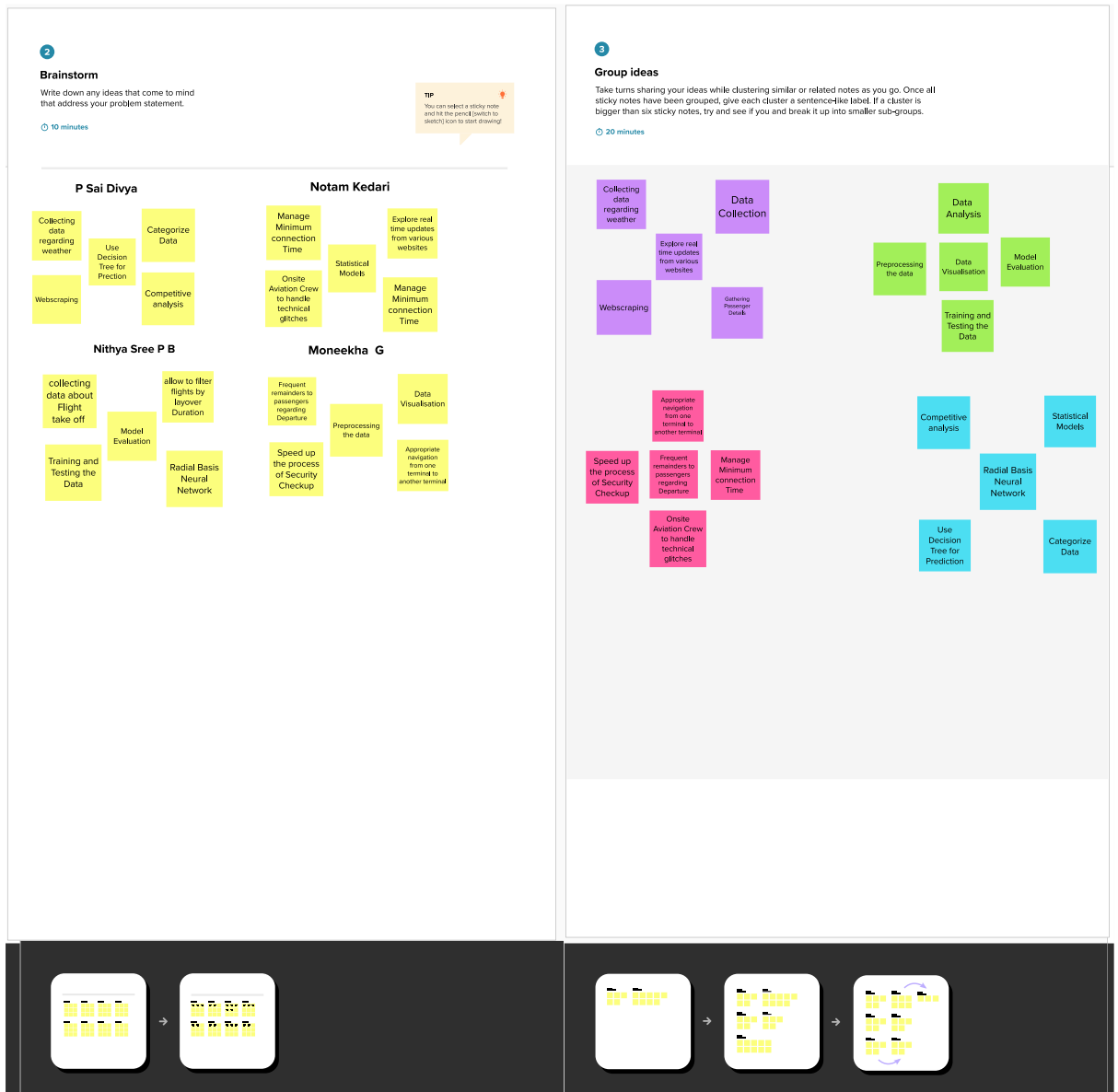


#### Need some inspiration?

See a finished version of this template to kickstart your work.

Open example ➔

14



## Proposed System:

## 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.

## Training the Dataset:

- The first line imports iris data set which is already predefined in sklearn module. Iris data set is basically a table which contains information about various varieties of iris flowers.
- For example, to import any algorithm and train\_test\_split class from sklearn and numpy module for use in this program.
- Then we encapsulate load\_data() method in data\_dataset variable. Further we divide the dataset into training data and test data using train\_test\_split method. The X prefix in variable denotes the feature values and y prefix denotes target values.
- This method divides dataset into training and test data randomly in ratio of 67:33. Then we encapsulate any algorithm.
- In the next line, we fit our training data into this algorithm so that computer can get trained using this data. Now the training part is complete.

#### Testing the Dataset:

- Now we have dimensions of a new flower in a numpy array called 'n' and we want to predict the species of this flower. We do this using the predict method which takes this array as input and spits out predicted target value as output.
- So the predicted target value comes out to be 0. Finally we find the test score which is the ratio of no. of predictions found correct and total predictions made. We do this using the score method which basically compares the actual values of the test set with the predicted values.

#### Advantages:

- Our goal is push for assisting farmers, 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. If there are some genuine problems in rolling out that work to next stage, then identify those problems and try solving them.
- It is targeted to those farmers who wish to professionally manage their farm by planning, monitoring and analyzing all farming activities.

#### Application:

- It is an integrated farm management application using mobile app.
- Agricultural sector to automate to identify the crop prediction process (real time world) and predicting by desktop application / web application.



Define CS, fit into CC

<b>1. CUSTOMER SEGMENT(S)</b> <b>CS</b> 1.customer who is unable to estimate the yield of the crop. 2.customer find it difficult because it requires more statistical data to be Analyzed 3.customer is the person who involved in agricultural sector.	<b>6. CUSTOMER CONSTRAINTS</b> <b>CC</b> 1.Too much costs of pesticides 2.No proper system for efficient storage of natural resources 3.Too much of data to be analyzed which is hard to prepare and support.	<b>5. AVAILABLE SOLUTIONS</b> <b>AS</b> 1.Previously there was no proper tool for estimation farmers estimate on there own by estimating the crop yield by using grain weight and some crop models 2..This method do not provide much profit and this is not systematic manner.
--	--	---

Focus on J&P, tap into BE, understand RC

<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <b>J&amp;P</b> 1.Using minimum resources and increasing productivity 2.Advising the customer about marketing,harvesting and crop Rotation 3 Promoting less use of pesticides and improving organic farming	<b>9. PROBLEM ROOT CAUSE</b> <b>RC</b> 1.For an normal individual it is hard to estimate crop yield because it takes long time must data need to be verified and studied which is practical not possible 2.No proper system for efficient storage of natural resources 3.Too much cost of pesticides and other agriculture products	<b>7. BEHAVIOUR</b> <b>BE</b> 1.Suggesting the crop to be planted in the coming season can be done by this model by evaluating various criterias 2.This application provides a way for crop rotation
--	--	--

Focus on J&P, tap into BE, understand RC

<b>3. TRIGGERS</b> <b>TR</b> It provokes the customer when they get to know about benefits and features by various communication methods	<b>10. YOUR SOLUTION</b> <b>SL</b> To focus on implementing crop yield prediction system by using machine learning techniques by doing analysis on agricultural datasets analyzing various parameters and calculating the maximum crop yield by processing datasets according to the areas of cultivation By using data analytics techniques the problems will be solved and helps in predicting the productivity of crop, such predictions will be help in business logistics	<b>8.CHANNELS of BEHAVIOUR</b> <b>CH</b> <b>8.1 ONLINE</b> This application will run online and all the data will be stored in online platform <b>8.2 OFFLINE</b> There is no offline platform for this model
---	---	---

Identify strong TR & EM

<b>4. EMOTIONS: BEFORE / AFTER</b> <b>EM</b> 1 Before using this approach, customer feels complicated, confused because they are too many factors like climatic conditions and prices for better seeds, low demand for the market and very low crop yield which is unmanageable 2 After using this application the customer can easily predict crop yield and estimate the profit which improve economic stability
--

Identify strong TR & EM

## 4. REQUIREMENT ANALYSIS

### General:

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

1. Functional requirements
2. Non-Functional requirements
3. Environment requirements
  - A. Hardware requirements
  - B. software requirements

### 4.1 Functional requirements:

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

requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

## 4.2 Non-Functional Requirements:

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

## Environmental Requirements:

### 1. Software Requirements:

Operating System	: Windows
Tool	: Anaconda with Jupyter Notebook

### 2. Hardware requirements:

Processor	: Pentium IV/III
Hard disk	: minimum 80 GB
RAM	: minimum 2 GB

## Software Description:

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system “Conda”. The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS. So, Anaconda distribution comes with more than 1,400 packages as well as the Conda package and virtual environment manager called Anaconda Navigator and it eliminates the need to learn to install each library independently. The open source packages can be individually installed from the Anaconda repository with the `conda install` command or using the `pip install` command that is installed with Anaconda. Pip packages provide many of the features of conda packages and in most cases they can work together. Custom packages can be made using the `conda build` command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda.

### Anaconda Navigator:

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glueviz
- Orange
- Rstudio
- Visual Studio Code

### Conda:

Conda is an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multi-languages. The Conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

### The Jupyter Notebook:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

### Notebook document:

Notebook documents (or “notebooks”, all lower case) are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc...). Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc.) as well as executable documents which can be run to perform data analysis.

### Jupyter Notebook App:

The Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet. In addition to displaying/editing/running notebook documents, the Jupyter Notebook App has a “Dashboard” (Notebook Dashboard), a “control panel” showing local files and allowing to open notebook documents or shutting down their kernels.

### kernel:

A notebook kernel is a “computational engine” that executes the code contained in a Notebook document. The ipython kernel, referenced in this guide, executes python code. Kernels for many other languages exist (official kernels). When you open a Notebook document, the associated kernel is automatically launched. When the notebook is executed (either cell-by-cell or with menu Cell -> Run All), the kernel performs the computation and produces the results. Depending on the type of computations, the kernel may consume significant CPU and RAM. Note that the RAM is not released until the kernel is shut-down.

### Notebook Dashboard:

The Notebook Dashboard is the component which is shown first when you launch Jupyter Notebook App. The Notebook Dashboard is mainly used to open notebook documents, and to manage the running kernels (visualize and shutdown). The Notebook Dashboard has other features similar to a file manager, namely navigating folders and renaming/deleting files.

## 5.PROJECT DESIGN

### 5.1 Data flow Diagram

#### Overview of the system:

This helps all others department to carried out other formalities. It 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,

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

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

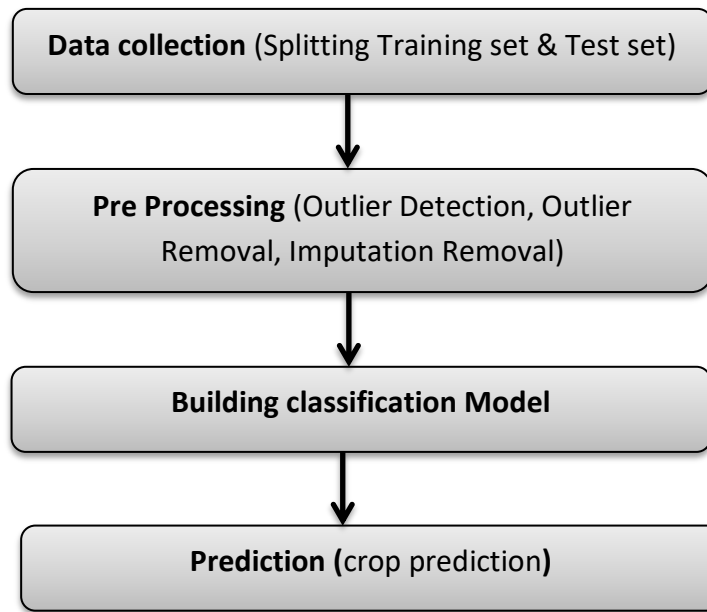


Fig: data flow diagram for Machine learning model

#### Project Goals:

- 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 with feature engineering
  - Pre-processing the given dataset
  - Splitting the test and training dataset
  - Predicting on the accuracy

#### Data collection:

The data set collected for predicting past farmer list of yield is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data

Model which was created using Random Forest , logistic , Decision tree algorithms are applied on the Training set and based on the test result accuracy, Test set prediction is done.

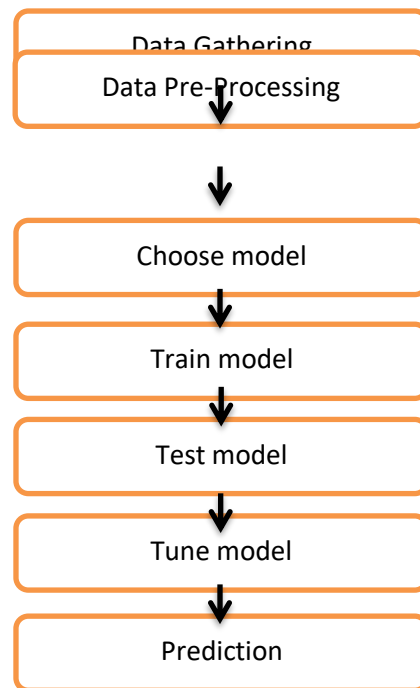
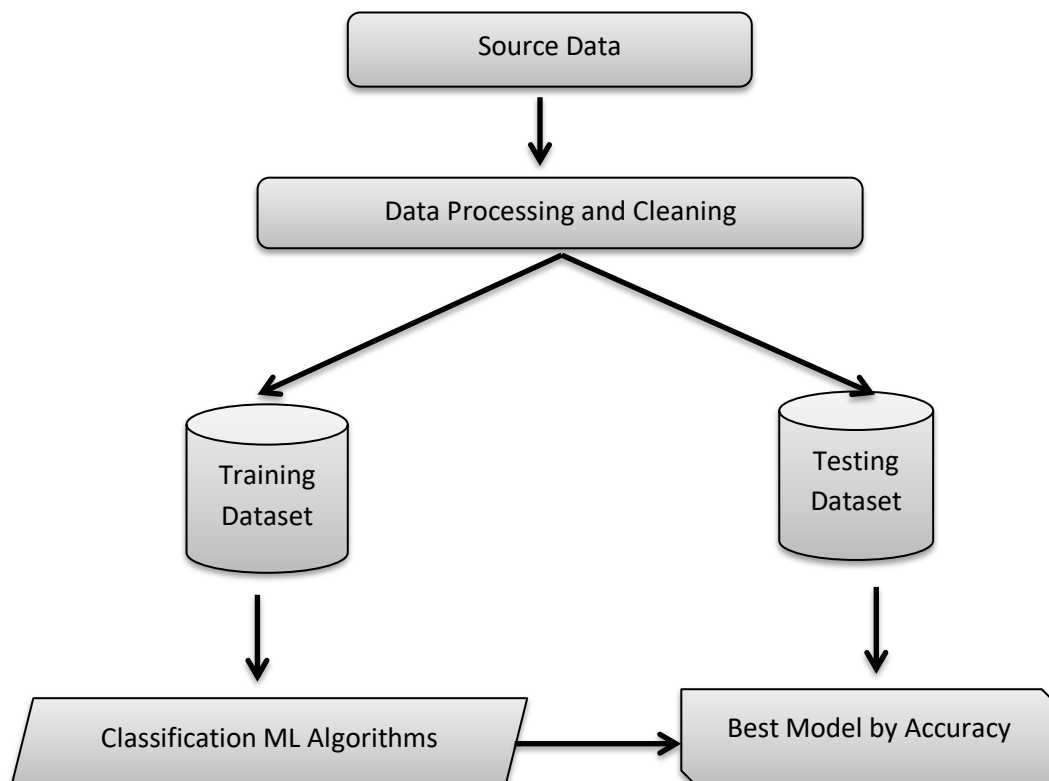
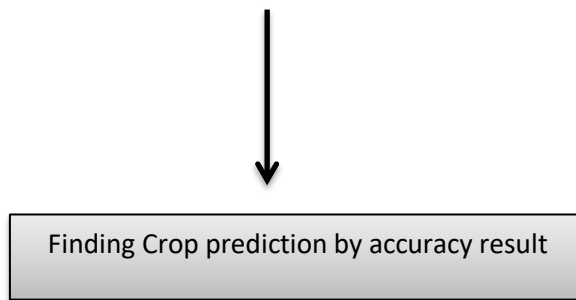


Fig: Process of dataflow diagram

### Work flow diagram

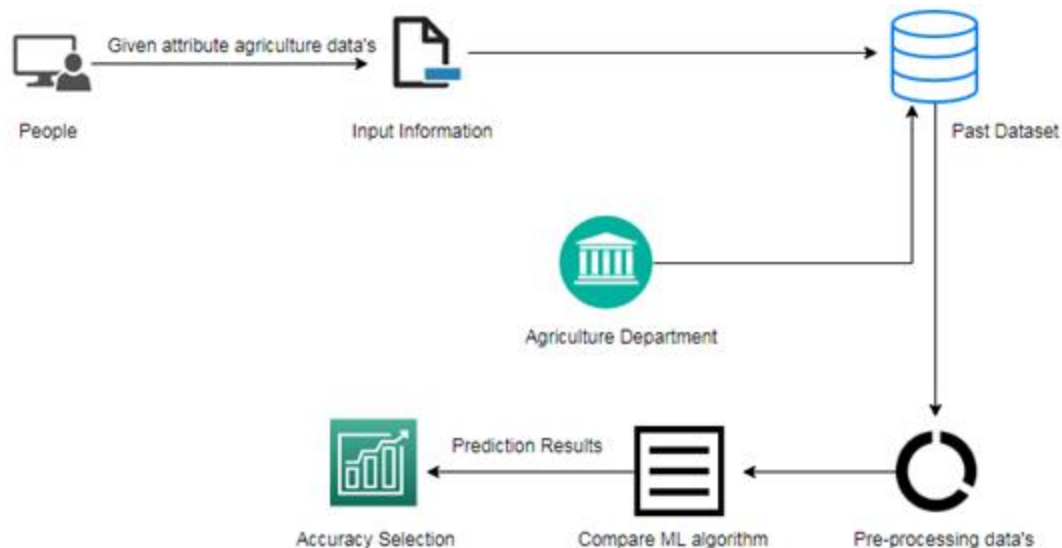




## 5.2 Solution & Technical Architecture

Create cells freely to explore your data and you should not perform too many operations in each cell. One option that you can take with this project is to do a lot of explorations in an initial notebook. These don't have to be organized, but make sure you use enough comments to understand the purpose of each code cell. Then, after you're done with your analysis, create a duplicate notebook where you will trim the excess and organize your steps so that you have a flowing, cohesive report and make sure that you keep your reader informed on the steps that you are taking in your investigation. Follow every code cell, or every set of related code cells, with a markdown cell to describe to the reader what was found in the preceding cell. Try to make it so that the reader can then understand what they will be seeing in the following cell.

### Business diagram/system architecture: - Phase II



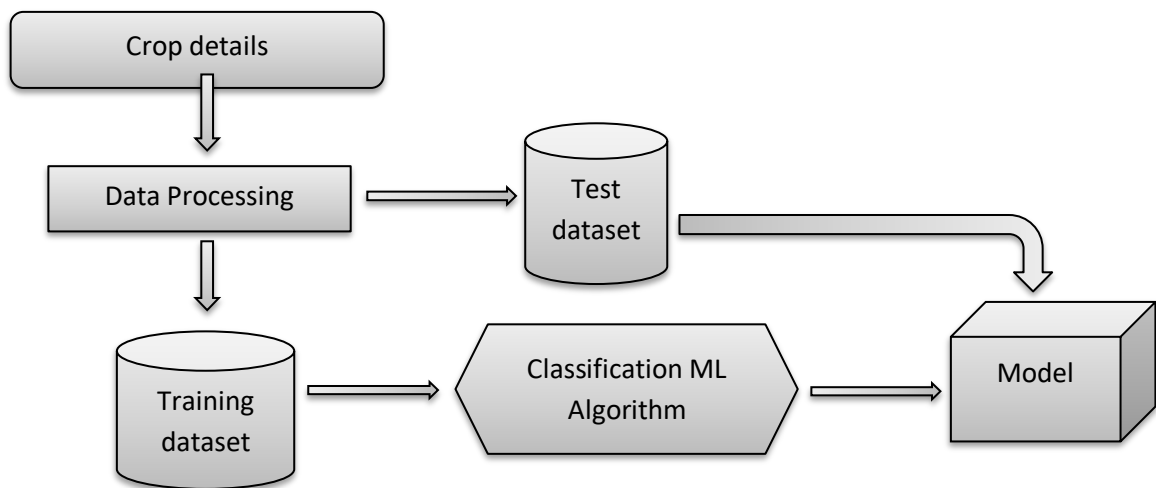
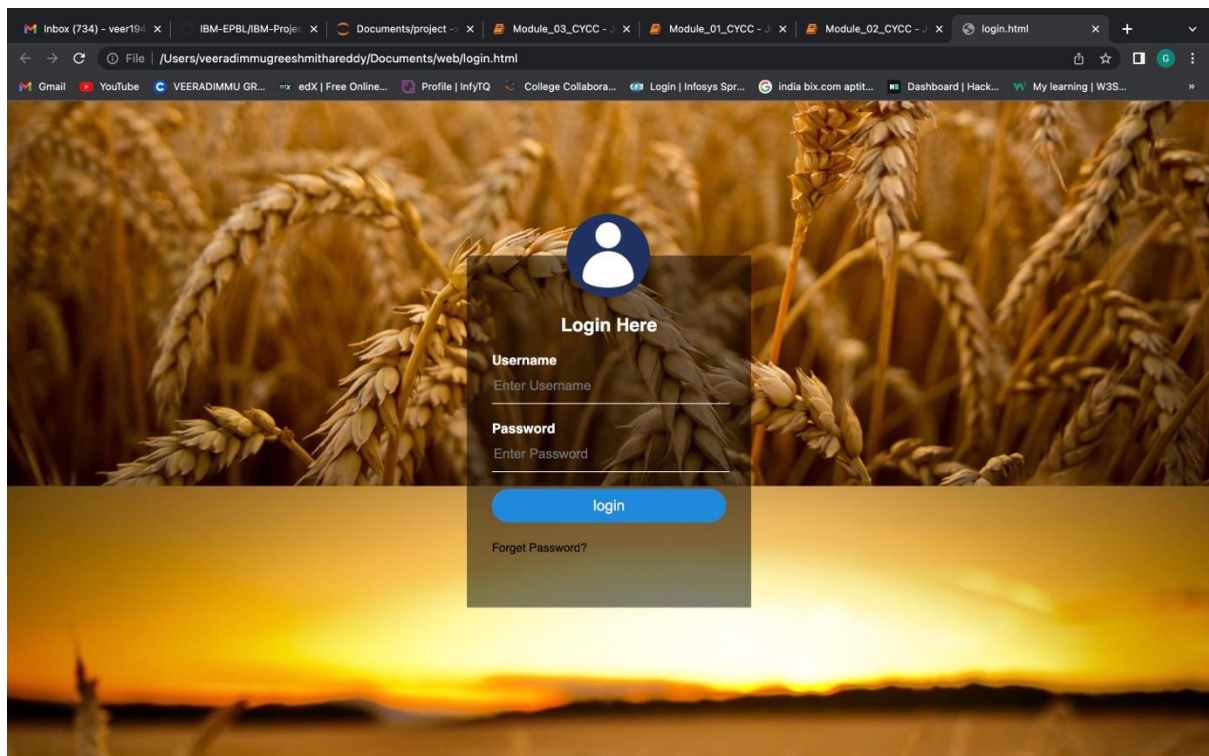


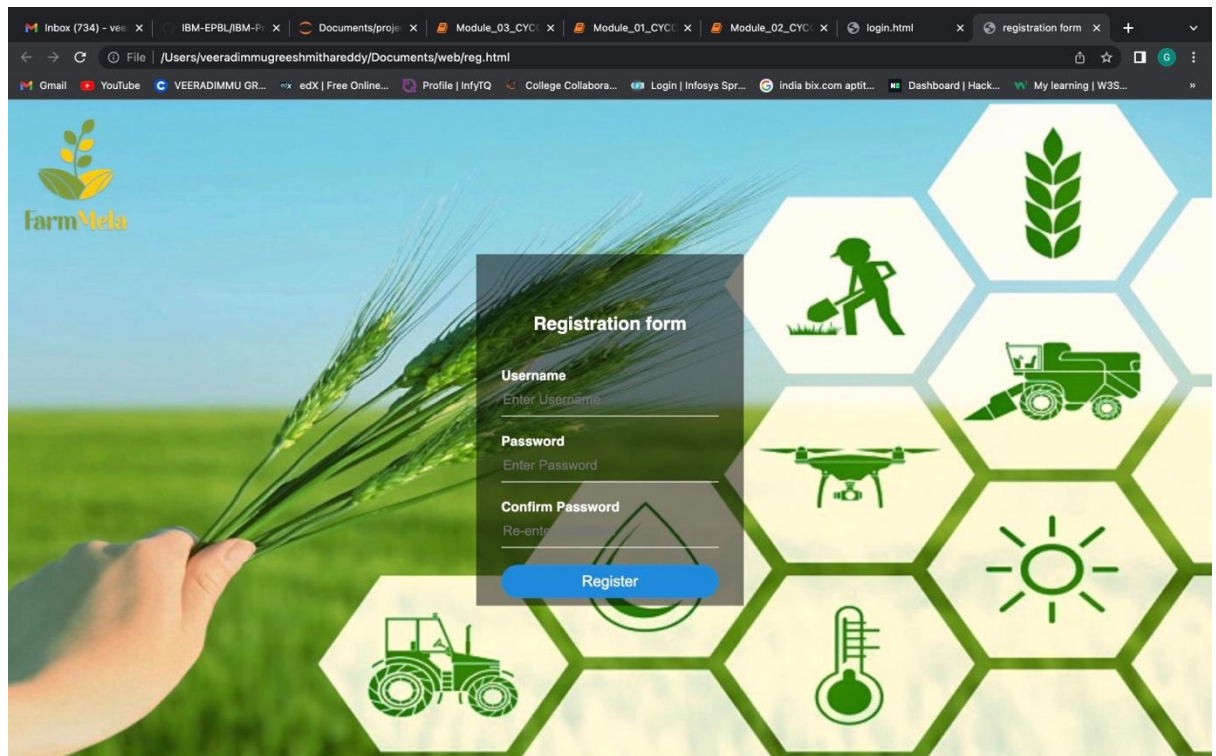
Fig: Architecture of Proposed mode

### 5.3 User Stories

#### USN-1: Login Page







## 6 PROJECT PLANNING & SCHEDULING

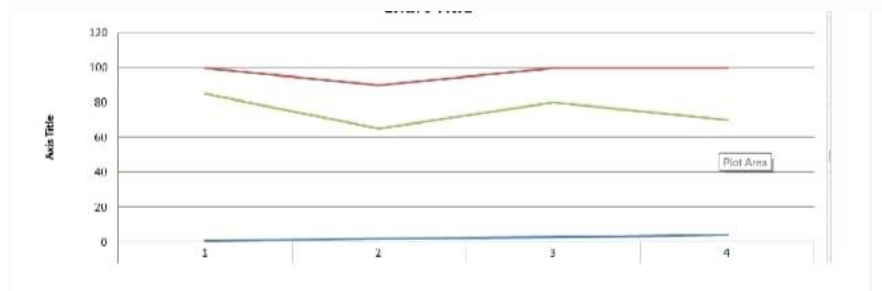
### 6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Khyathi Greeshmitha
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	2	High	Lahari
Sprint-1		USN-3	As a user, I can register for the application through Facebook	1	Low	Prasanna Vyshnavi
Sprint-1		USN-4	As a user, I can register for the application through Gmail	1	Medium	Greeshmitha Khyathi
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	2	High	Greeshmitha Khyathi Prasanna
Sprint-2	Dashboard	USN-6	As a user, I can visualize the data	2	High	Khyathi Greeshmitha
Sprint-3	Code and test cases	USN-7	Source code(python) is integrated into jupyter notebook(anaconda navigator)	2	High	Greeshmitha Khyathi
Sprint-4	Final project story	USN-8	Overview of the entire project (including source code)	1	Medium	Prasanna Lahari Vyshnavi

### 6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05-Nov-2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12-Nov-2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19-Nov-2022

### 6.3 Reports from JIRA



## 7. CODING & SOLUTIONING

### Model Selection:

This is the most exciting phase in Applying Machine Learning to any Dataset. It is also known as Algorithm selection for Predicting the best results. Usually Data Scientists use different kinds of Machine Learning algorithms to the large data sets. But, at high level all those different algorithms can be classified in two groups : supervised learning and unsupervised learning. Supervised learning : Supervised learning is a type of system in which both input and desired output data are provided. Input and output data are labeled for classification to provide a learning basis for future data processing. Supervised learning problems can be further grouped into **Regression** and **Classification** problems.

A **regression** problem is when the output variable is a real or continuous value, such as “salary” or “weight”. A **classification** problem is when the output variable is a category like filtering emails “spam” or “not spam”

Unsupervised Learning : Unsupervised learning is the algorithm using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. In our dataset we have the outcome variable or Dependent variable i.e Y having only two set of values, either M (Malign) or B(Benign). So we will use Classification algorithm of supervised learning.

### Modules:

- Data validation and pre-processing technique (Module-01)
- Exploration data analysis of visualization and training a model by given attributes (Module-02)
- Performance measurements of logistic regression and decision tree algorithms (Module-03)

### 7.1 Module-01:

## Variable Identification Process / data validation process:

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers uses this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model. For example, time series data can be analyzed by regression algorithms; classification algorithms can be used to analyze discrete data. (For example to show the data type format of given dataset)

	State_Name	District_Name	Crop_Year	Season	Crop	Area	rainfall	Average Humidity	Mean Temp	Cost of Cultivation ('/Hectare) C2	Cost of Production ('/Quintal) C2	Yield (Quintal/ Hectare)	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: Given data frame

## Data Validation/ Cleaning/Preparing Process:

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

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

#We'll use a test size of 30%. We also stratify
from sklearn.model_selection import train_test
X_train, X_test, y_train, y_test = train_test_
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))

Number of training dataset: 163704
Number of testing dataset: 70160
Total number of dataset: 233864
```

Fig: Splitting the given dataset

#### Data Pre-processing:

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format; for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set.

	State_Name	District_Name	Crop_Year	Season	Crop	Area	R	H	T	CC	CP	Y	CPPY
0	0	410	3	1	2	2025	33	45	10	21	30	13	126
1	0	410	4	1	2	2025	121	44	6	3	26	7	72
2	0	410	5	4	2	2030	118	46	1	33	45	11	200
3	0	410	6	4	2	2033	172	45	6	4	36	4	78
4	0	410	7	4	2	2037	75	51	15	20	24	23	144

Fig: After preprocessing given data frame

## 7.2 Module-02:

Exploration data analysis of visualization:

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- How to chart time series data with line plots and categorical quantities with bar charts.
- How to summarize data distributions with histograms and box plots.
- How to summarize the relationship between variables with scatter plots.

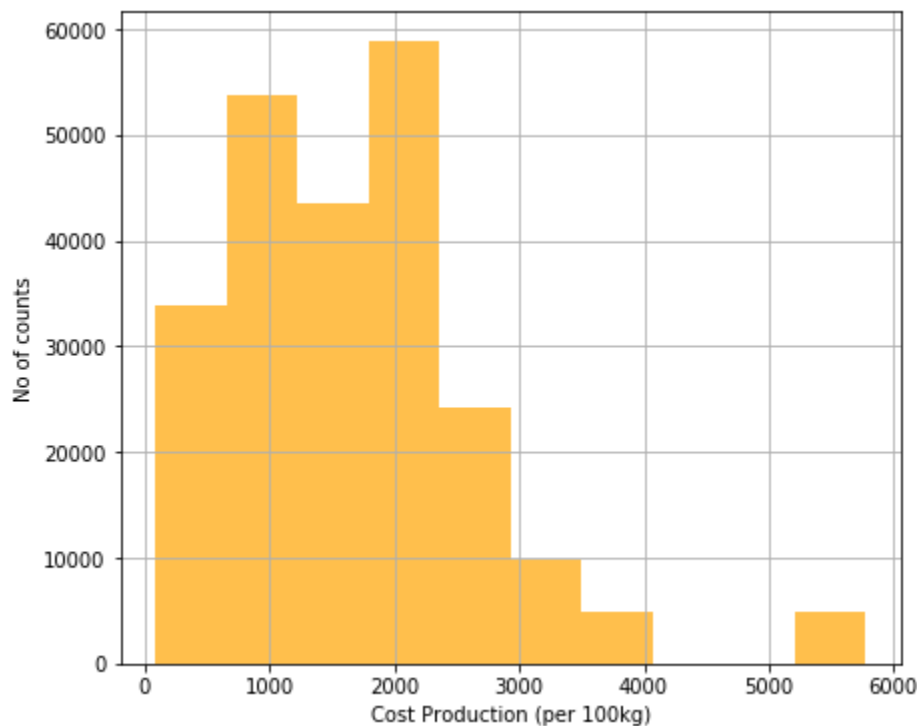


Fig: Cost production per 100kg by counts

Many machine learning algorithms are sensitive to the range and distribution of attribute values in the input data. Outliers in input data can skew and mislead the training process of machine learning algorithms resulting in longer training times, less accurate models and ultimately poorer results.

Even before predictive models are prepared on training data, outliers can result in misleading representations and in turn misleading interpretations of collected data. Outliers can skew the summary distribution of attribute values in descriptive statistics like mean and standard deviation and in plots such as histograms and scatterplots, compressing the body of the data. Finally, outliers can represent examples of data instances that are relevant to the problem such as anomalies in the case of fraud detection and computer security.

It couldn't fit the model on the training data and can't say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from the data, and it is not getting up too much noise. Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

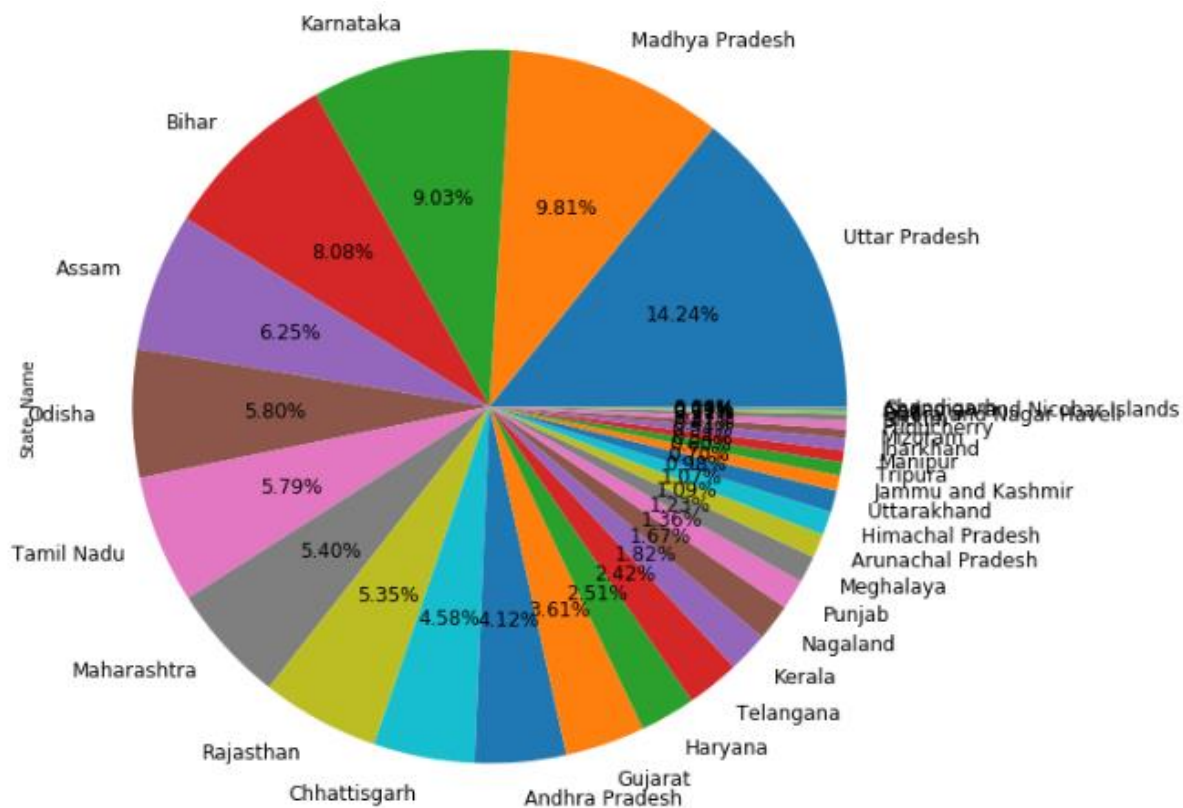


Fig: Percentage level of crop yield production by state



### 7.3 Module-03:

#### Logistic Regression:

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

```
Classification report of Logistic Regression Results:
```

	precision	recall	f1-score	support
0	0.96	0.96	0.96	1776
1	0.94	0.94	0.94	1087
accuracy			0.95	2863
macro avg	0.95	0.95	0.95	2863
weighted avg	0.95	0.95	0.95	2863

```
Accuracy result of Logistic Regression is: 95.2497380370241
```

```
Confusion Matrix result of Logistic Regression is:
```

```
[[1708  68]
 [  68 1019]]
```

```
Sensitivity : 0.9617117117117117
```

```
Specificity : 0.937442502299908
```

In other words, the logistic regression model predicts  $P(Y=1)$  as a function of  $X$ .  
Logistic regression Assumptions:

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
- Only the meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little.

- The independent variables are linearly related to the log odds.
- Logistic regression requires quite large sample sizes.

---

```
True Positive : 1019
True Negative : 1708
False Positive : 68
False Negative : 68
```

```
True Positive Rate : 0.937442502299908
True Negative Rate : 0.9617117117117117
False Positive Rate : 0.038288288288288286
False Negative Rate : 0.062557497700092
```

```
Positive Predictive Value : 0.937442502299908
Negative predictive value : 0.9617117117117117
```

### Decision Tree:

It is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. Assumptions of Decision tree:

Classification report of Decision Tree Classifier Results:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1776
1	1.00	1.00	1.00	1087
accuracy			1.00	2863
macro avg	1.00	1.00	1.00	2863
weighted avg	1.00	1.00	1.00	2863

Accuracy result of Decision Tree Classifier is 100.0

Confusion Matrix result of Decision Tree Classifier is:

```
[[1776  0]
 [  0 1087]]
```

Sensitivity : 1.0

Specificity : 1.0

- At the beginning, we consider the whole training set as the root.
- Attributes are assumed to be categorical for information gain, attributes are assumed to be continuous.
- On the basis of attribute values records are distributed recursively.



- We use statistical methods for ordering attributes as root or internal node.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed.

```
True Positive : 1087
True Negative : 1776
False Positive : 0
False Negative : 0

True Positive Rate : 1.0
True Negative Rate : 1.0
False Positive Rate : 0.0
False Negative Rate : 0.0

Positive Predictive Value : 1.0
Negative predictive value : 1.0
```

This process is continued on the training set until meeting a termination condition. It is constructed in a top-down recursive divide-and-conquer manner. All the attributes should be categorical. Otherwise, they should be discretized in advance. Attributes in the top of the tree have more impact towards in the classification and they are identified using the information gain concept. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers.

## 8. TESTING

### 8.1 Test Cases

#### Testing Levels:-

All major activities of various testing level are described below.

1. Unit Testing
2. Integration Testing
3. Functional Testing
4. System Testing
5. White box Testing

## **6. Black Box Testing**

### **1. 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 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.

### **2. 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 satisfactory, 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.

### **3. Functional Testing:-**

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

### **4. System Testing:**

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.

### **5. 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 purpose. It is used to test areas that cannot be reached from a black box level.

### **8.2 User 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.

## **9. RESULTS**

### **9.1 Performance Metrics**

**Comparing Algorithm with prediction in the form of best accuracy result:**

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 2 different algorithms are compared:

- Logistic Regression
- Random Forest
- dimensions of new features in a numpy array called 'n' and it want to predict the species of this features and to do using the predict method which takes this array as input and spits out predicted target value as output.
- So, the predicted target value comes out to be 0. Finally to find the test score which is the ratio of no. of predictions found correct and total predictions made and finding accuracy score method which basically compares the actual values of the test set with the predicted values.

Sensitivity:

Sensitivity is a measure of the proportion of actual positive cases that got predicted as positive (or true positive). Sensitivity is also termed as Recall. This implies that there will be another proportion of actual positive cases, which would get predicted incorrectly as negative (and, thus, could also be termed as the false negative). This can also be represented in the form of a false negative rate. The sum of sensitivity and false negative rate would be 1. Let's try and understand this with the model used for predicting whether a person is suffering from

the disease. Sensitivity is a measure of the proportion of people suffering from the disease who got predicted correctly as the ones suffering from the disease. In other words, the person who is unhealthy actually got predicted as unhealthy.

Mathematically, sensitivity can be calculated as the following:

$$\text{Sensitivity} = (\text{True Positive}) / (\text{True Positive} + \text{False Negative})$$

The following is the details in relation to True Positive and False Negative used in the above equation.

- True Positive = Persons predicted as suffering from the disease (or unhealthy) are actually suffering from the disease (unhealthy); In other words, the true positive represents the number of persons who are unhealthy and are predicted as unhealthy.
- False Negative = Persons who are actually suffering from the disease (or unhealthy) are actually predicted to be not suffering from the disease (healthy). In other words, the false negative represents the number of persons who are unhealthy and got predicted as healthy. Ideally, we would seek the model to have low false negatives as it might prove to be life-threatening or business threatening.

The higher value of sensitivity would mean higher value of true positive and lower value of false negative. The lower value of sensitivity would mean lower value of true positive and higher value of false negative. For healthcare and financial domain, models with high sensitivity will be desired.

Specificity:

Specificity is defined as the proportion of actual negatives, which got predicted as the negative (or true negative). This implies that there will be another proportion of actual negative, which got predicted as positive and could be termed as false positives. This proportion could also be called a false positive rate. The sum of specificity and false positive rate would always be 1. Let's try and understand this with the model used for predicting whether a person is suffering from the disease. Specificity is a measure of the proportion of people not suffering from the disease who got predicted correctly as the ones who are not suffering from the disease. In other words, the person who is healthy actually got predicted as healthy is specificity.

Mathematically, specificity can be calculated as the following:

$$\text{Specificity} = (\text{True Negative}) / (\text{True Negative} + \text{False Positive})$$

The following is the details in relation to True Negative and False Positive used in the above equation.

- True Negative = Persons predicted as not suffering from the disease (or healthy) are actually found to be not suffering from the disease (healthy); In other words, the true

negative represents the number of persons who are healthy and are predicted as healthy.

- False Positive = Persons predicted as suffering from the disease (or unhealthy) are actually found to be not suffering from the disease (healthy). In other words, the false positive represents the number of persons who are healthy and got predicted as unhealthy.

The higher value of specificity would mean higher value of true negative and lower false positive rate. The lower value of specificity would mean lower value of true negative and higher value of false positive.

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. We need the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

$$\text{True Positive Rate(TPR)} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{False Positive rate(FPR)} = \text{FP} / (\text{FP} + \text{TN})$$

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct. (When the model predicts default: how often is correct?)

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

General Formula:

$$F\text{- Measure} = 2TP / (2TP + FP + FN)$$

F1-Score Formula:

$$F1 \text{ Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

### Used Python Packages:

#### **sklearn:**

- In python, sklearn is a machine learning package which include a lot of ML algorithms.
- Here, we are using some of its modules like `train_test_split`, `DecisionTreeClassifier` or `Logistic Regression` and `accuracy_score`.

#### **NumPy:**

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

#### **Pandas:**

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

#### **Matplotlib:**

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

#### **tkinter:**

- Standard python interface to the GUI toolkit.
- Accessible to everybody and reusable in various contexts.

## 10. ADVANTAGES & DISADVANTAGES

### Advantages:

- Our goal is push for assisting farmers, 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. If there are some genuine problems in rolling out that work to next stage, then identify those problems and try solving them.
- It is targeted to those farmers who wish to professionally manage their farm by planning, monitoring and analyzing all farming activities.
- Achieving the maximum crop at minimum yield is the ultimate Aim of the project.
- Early detection of problems and management of that problems can help the farmers for better crop yield.
- For the better understanding of the crop yield, we need to study of the huge data with the help of machine learning algorithm so it will give the accurate yield for that crop and suggest the farmer for a better crop.

#### Disadvantages:

- The obtained result for the crop yield prediction using SMO classifier gives less accuracy when compared to naïve Bayes, multilayer perceptron and Bayesian network.
- Previously yield is predicted on the bases of the farmers prior experience but now weather conditions may change drastically so they cannot guess the yield.

## 11. CONCLUSION:

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. Finally we predict the crop using machine learning algorithm with different results. This brings some of the following insights about crop prediction. As maximum types of crops will be covered under this system, farmer may get to know about the crop which may never have been cultivated and lists out all possible crops, it helps the farmer in decision making of which crop to cultivate. Also, this system takes into consideration the past production of data which will help the farmer get insight into the demand and the cost of various crops in market.

## 12. FUTURE SCOPE:

- Remaining SMLT algorithms will be involve to finding the best accuracy with applying to predict the crop yield and cost.
- Agricultural department wants to automate the detecting the yield crops from eligibility process (real time).
- To automate this process by show the prediction result in web application or desktop application.
- To optimize the work to implement in Artificial Intelligence environment.

### 13. APPENDIX

#### Source Code:

```
#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")
listcrops = p.Categorical(df['Crop'])
listcropss
df['Crop'].value_counts()
df['Crop'].nunique()
#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()
#find sum of duplicate data
sum(df.duplicated())
#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()
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)
df.rename(columns={'cost of production per yield':'CPPY'}, inplace=True)
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)
#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')
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['CPY'].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.boxplot(column="CP", 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')
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))
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')
#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='YPr', axis=1)
#Response variable
y = df.loc[:, 'YPr']
#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)
from sklearn.linear_model import LogisticRegression
logR= LogisticRegression()
logR.fit(X_train,y_train)
predictR = logR.predict(X_test)
print("")
print('Classification report of Logistic Regression Results:')
print("")
print(classification_report(y_test,predictR))
x = (accuracy_score(y_test,predictR)*100)
print('Accuracy result of Logistic Regression is:', x)
print("")
cm1=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Logistic Regression is:\n',cm1)

```

```

print("")
sensitivity1 = cm1[0,0]/(cm1[0,0]+cm1[0,1])
print('Sensitivity : ', sensitivity1 )
print("")
specificity1 = cm1[1,1]/(cm1[1,0]+cm1[1,1])
print('Specificity : ', specificity1)
print("")
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()
dtree.fit(X_train, y_train)

predictDT = dtree.predict(X_test)
print("")
print('Classification report of Decision Tree Classifier Results:')
print("")
print(classification_report(y_test,predictDT))
x = (accuracy_score(y_test,predictDT)*100)
print('Accuracy result of Decision Tree Classifier is', x)
print("")
cm2=confusion_matrix(y_test,predictDT)
print('Confusion Matrix result of Decision Tree Classifier is:\n',
confusion_matrix(y_test,predictDT))
print("")
sensitivity1 = cm2[0,0]/(cm2[0,0]+cm2[0,1])
print('Sensitivity : ', sensitivity1 )
print("")
specificity1 = cm2[1,1]/(cm2[1,0]+cm2[1,1])
print('Specificity : ', specificity1)

```

### **GitHub Repo Link:**

<https://github.com/IBM-EPBL/IBM-Project-29899-1660133258>





