**CHAPTER - 1**

**INTRODUCTION**

* 1. **ABOUT MACHINE LEARNING**

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than have human programmers specify every needed step.

The discipline of machine learning employs various approaches to help computers learn to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset has often been used.

**1.1.1 Machine learning approaches**

Early classifications for machine learning approaches sometimes divided them into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system. These were:

Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent) As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

Other approaches or processes have since developed that don't fit neatly into this three-fold categorization, and sometimes more than one is used by the same machine learning system. For example topic modeling, dimensionality reduction or meta learning. [8] As of 2020, deep learning has become the dominant approach for much ongoing work in the field of machine learning.

**1.2. ABOUT PROJECT**

To identify weather conditions that are deterrent for the production of apples. It then efficiently predicts the yield of apples on the basis of monthly weather patterns. The use of several algorithms like Artificial Neural Network, K Nearest Neighbors, and Regularized Greedy Forest is demonstrated in [5] to select a crop based on the pre- diction yield rate, which, in turn, is influenced by multiple parameters. Additional features included in the system are pesticide prediction and online trading based on agricultural commodities. 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 crop 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.

A farmer’s decision about which crop to grow is generally clouded by his intuition and other irrelevant factors like making instant profits, lack of awareness about market demand, overestimating a soil’s potential to support a particular crop, and so on. A very misguided decision on the part of the farmer could place a significant strain on his family’s financial condition. Perhaps this could be one of the many reasons contributing to the countless suicide cases of farmers that we hear from media on a daily basis. In a country like India, where agriculture and related sector contributes to approximately 20.4 per cent of its Gross Value Added (GVA) [2], such an erroneous judgment would have negative implications on not just the farmer’s family, but the entire economy of a region. For this reason, we have identified a farmer’s dilemma about which crop to grow during a particular season, as a very grave one. The need of the hour is to design a system that could provide predictive insights to the Indian farmers, thereby helping them make an informed decision about which crop to grow. With this in mind, we propose a system, an intelligent system that would consider environmental parameters (temperature, rainfall, geographical location in terms of state) and soil characteristics (pH value, soil type and nutrients concentration) before recommending the most suitable crop to the user .

The processing part also takes into consideration two more datasets i.e. one obtained from weather department, forecasting the weather expected in current year and the other data being static data. This static data is the crop production and data related to demands of various crops obtained from various government websites. The proposed system applies machine learning and prediction algorithm like Random Forest algorithm to identify the pattern among data and then process it as per input conditions. This in turn will propose the best feasible crops according to given environmental conditions. Thus, this system will only require the location of the user and it will suggest number of profitable crops providing a choice directly to the farmer about which crop to cultivate. As past year production is also taken into account, the prediction will be more accurate.

**1.3. OBJECTIVES**

Machine learning can bring a boom in the agriculture field by changing the income scenario through growing the optimum crop .Nowadays many experts are Experimenting on automated farming ideology's. Use of Random Forest Algorithm or decision tree or makes the model very efficient in terms of computation . we can use various machine learning algorithms like Random Forest Algorithm or Linear Regression etc. This would be helpful for us to learn which crop suitable to grow and it will gives us the statistics about the crops based on the factors.

Factors that are to be considered:

* Soil Parameters: Soil Type, Soil Nutrients like N, P, K values.
* Climatic Parameters: Humidity Temperature, PH and Rainfall.
* To build a robust model to give correct and accurate prediction of crop sustain- ability in a given state for the particular soil type and climatic conditions.
* Provide recommendation of the best suitable crops in the area so that the farmer does not incur any losses
* Provide profit analysis of various crops based on previous years data.

**1.4. EXISTING SYSTEM**

The inconsistent trends developed from the side effects of global warming make it cumbersome for the farmers to clearly predict the temperature and rainfall patterns thus affecting their crop productivity. In order to perform accurate prediction and handle inconsistent trends in temperature and rainfall various machine learning algorithms like decision tree, KNN etc can be applied to get a pattern. To identify weather conditions that are deterrent for the production of apples. It then efficiently predicts the yield of apples on the basis of monthly weather patterns. The use of several algorithms like Artificial Neural Network, K Nearest Neighbors, and Regularized Greedy Forest is demonstrated in [5] to select a crop based on the pre- diction yield rate, which, in turn, is influenced by multiple parameters. Additional features included in the system are pesticide prediction and online trading based on agricultural commodities.

**1.5. DISADVANTAGES OF EXISTING SYSTEM**

One shortcoming that we identified in all these notable published works was that the authors of each paper concentrated on a single parameter (either weather or soil) for predicting the suitability of crop growth. However, in our opinion, both these factors should be taken together into consideration concomitantly for the best and most accurate prediction. This is because, a particular soil type may be fit for supporting one type of crop, but if the weather conditions of the region are not suitable for that crop type, then the yield will suffer.

**1.6. PROPOSED SYSTEM**

We to eliminate the aforementioned drawbacks, we propose an Intelligent Crop Recommendation system- which takes into consideration all the appropriate parameters, including temperature, rainfall, location and soil condition, to predict crop suitability. This system is fundamentally concerned with performing the primary function of AgroConsultant, which is, providing crop recommendations to farmers algorithms. We also provide the profit analysis on crops grown in different states which gives the user an easy and reliable insight to decide and plan the crops.

**1.7. ADVANTAGES OF PROPOSED SYSTEM**

* Better prediction
* Better accuracy

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 INTRODUCTION:**

The agriculture plays a dominant role in the growth of the country’s economy. Climate and other environmental changes has become a major threat in the agriculture field. Machine learning (ML) is an essential approach for achieving practical and effective solutions for this problem. Crop Yield Prediction involves predicting yield of the crop from available historical available data like weather parameter, soil parameter and historic crop yield. This paper focus on predicting the yield of the crop based on the existing data by using Random Forest algorithm. Real data of Tami nadu were used for building the models and the models were tested with samples. The prediction will helps to the farmer to predict the yield of the crop before cultivating onto the agriculture field. To predict the crop yield in future accurately Random Forest, a most powerful and popular supervised machine learning algorithm is used.

Literature [survey](http://www.blurtit.com/q876299.html) is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company Traffic Redundancy Elimination, once these things are satisfied, then next steps are to determine which operating system and language can be used for developing the tool. Once the [programmers](http://www.blurtit.com/q876299.html) start building the tool the programmers need lot of external support.

This paper has been prepared as an effort to reassess the research studies on the relevance of machine learning techniques in the domain of agricultural crop production. Methods/Statistical Analysis: This method is a new approach for production of agricultural crop management. Accurate and timely forecasts of crop production are necessary for important policy decisions like import-export, pricing marketing distribution etc. which are issued by the directorate of economics and statistics. However one has understand that these prior estimates are not the objective estimates as these estimate requires lots of descriptive assessment based on many different qualitative factors. Hence there is a requirement to develop statistically sound objective prediction of crop production. That development in computing and information storage has provided large amount of data. Findings: The problem has been to intricate knowledge from this raw data, this has lead to the development of new approach and techniques such as machine learning that can be used to unite the knowledge of the data with crop yield evaluation. This research has been intended to evaluate these innovative techniques such that significant relationship can be found by their applications to the various variables present in the data base. Application/Improvement: The few techniques like artificial neural networks, Information Fuzzy Network, Decision Tree, Regression Analysis, Bayesian belief network. Time series analysis, Markov chain model, k-means clustering, k nearest neighbor, and support vector machine are applied in the domain of agriculture were presented.

**2.2 DIGITAL SIGNATURES**

##### Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique Authors: Rakesh Kumar, M.P. Singh, Prabhat Kumar.

##### J.P. Singh

This paper proposed a method named Crop Selection Method (CSM) to solve crop selection problem, and maximize net yield rate of crop over season and subsequently achieves maximum economic growth of the country. The proposed method may im- prove net yield rate of crops.

##### AgroConsultant: Intelligent Crop Recommendation System Using Ma- chine Learning Algorithms Authors: Zeel Doshi, Subhash Nadkarni, Rashi Agrawal, Prof. Neepa Shah

This paper, proposed and implemented an intelligent crop recommendation sys- tem, which can be easily used by farmers all over India. This system would assist the farmers in making an informed decision about which crop to grow depending on a variety of environmental and geographical factors. We have also implemented a secondary system, called Rainfall Predictor, which predicts the rainfall of the next 12 months.

##### Development of Yield Prediction System Based on Real-time Agricultural meteorological Information Haedong Lee \*, Aekyung Moon\* \* ETRI, 218 Gajeong-ro, Yuseong-gu, 305-700, Korea

This paper contains about the research and the building of an effective agricultural yield forecasting system based on real-time monthly weather. It is difficult to predict the agricultural crop production because of the abnormal weather that happens every year and rapid regional climate change due to global warming. The development of agricultural yield forecasting system that leverages real-time weather information is urgently required. In this research, we cover how to process the number of weatherand how to configure the prediction

##### Analysis of Soil Behaviour and Prediction of Crop Yield using Data Mining Approach Monali Paul, Santosh K. Vishwakarma, Ashok Verma Computer science and Engineering GGITS, Jabalpur

This work presents a system, which uses data mining techniques in order to predict the category of the analyzed soil datasets. The category, thus predicted will indicate the yielding of crops. The problem of predicting the crop yield is formalized as a classification rule, where Naive Bayes and K-Nearest Neighbor methods are used.

##### Crop Recommendation System for Precision Agriculture S.Pudumalar\*, E.Ramanujam\*, R.Harine Rajashree, C.Kavya, T.Kiruthika, J.Nisha.

This paper, proposes a recommendation system through an ensemble model with majority voting technique using Random tree, CHAID, K-Nearest Neighbor and Naive Bayes as learners to recommend a crop for the site specific parameters with high accuracy and efficie

**CHAPTER - 3**

**SYSTEM ANANYSIS**

**3.1 INTRODUCTION**

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI’S at the top level have been categorized as

1. Administrative user interface.
2. The operational or generic user interface.

The ‘Administrative user interface’ concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updating along with the extensive data search capabilities.

The ‘operational or generic user interface’ helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included flexibilities

**3.2 SOFTWARE REQUIREMENTS**

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

* **Python ide 3.7 version (or)**
* **Anaconda 3.7 ( or)**
* **Jupiter (or)**

**3.3 HARDWARE REQUIREMENTS**

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

* **Operating system : windows,**
* **Processor : minimum intel i3**
* **Ram : minimum 4 gb**
* **Hard disk : minimum 250gb**

**3.4 USER REQUIERMENTS**

**3.4.1 FUNCTIONAL REQUIREMENTS**

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. Following are the functional requirements on the system:

1. All the data must be in the same format as a structured data.
2. The data collected will be vectorized and sent across to the classifier

**3.4.2 NON FUNCTIONAL REQUIREMENTS**

Non functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system properties such as reliability, response time and store occupancy. Non functional requirements arise through the user needs, because of budget constraints, organizational policies and the need for interoperability with other software and hardware systems.

**3.5. FEASIBILITY STUDY**

The possibility of the project is analyzed during this part and business proposal is place forth with a awfully general arrange for the project and a few value estimates. throughout system analysis the FEASIBILITY study of the projected system is to be distributed. this can be to make sure that the projected system isn't a burden to the corporate. For risk analysis, some understanding of the key needs for the system is important.

Three key issues concerned within the FEASIBILITY analysis area unit

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**3.5.1 ECONOMICAL FEASIBILITY:**  This study is distributed to visualize the economic impact that the system can wear the society. the quantity of fund that the corporate will pour into the analysis and development of the system is restricted. The expenditures should be even. so the developed system additionally inside the budget and this was achieved as a result of most of the technologies used area unit freely out there. solely the bespoken product had to be purchased.

**3.5.2 TECHNICAL FEASIBILITY:** This study is distributed to visualize the technical risk, that is, the technical needs of the system. Any system developed should not have a high demand on the out there technical resources. this can result in high demands on the out there technical resources. this can result in high demands being placed on the shopper. The developed system should have a modest demand, as solely borderline or null changes area unit needed for implementing this technique.

**3.5.3 SOCIAL FEASIBILITY:** The facet of study is to visualize the amount of acceptance of the system by the user. This includes the method of coaching the user to use the system expeditiously. The user should not feel vulnerable by the system, instead should settle for it as a necessity. the amount of acceptance by the users entirely depends on the ways that area unit used to teach the user regarding the system and to form him well-known with it. His level of confidence should be raised so he's additionally ready to build some helpful criticism, that is welcome, as he's the ultimate user of the system.

**CHAPTER - 4**

**SYSTEM DESIGN**

**4.1 INTRODUCTION**

To identify weather conditions that are deterrent for the production of apples. It then efficiently predicts the yield of apples on the basis of monthly weather patterns. The use of several algorithms like Artificial Neural Network, K Nearest Neighbors, and Regularized Greedy Forest is demonstrated in [5] to select a crop based on the pre- diction yield rate, which, in turn, is influenced by multiple parameters. Additional features included in the system are pesticide prediction and online trading based on agricultural commodities. 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 crop 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.

**4.2 MODULES DESCRIPTION**

**Random Forest** Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

**Decision Tree**

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.The decisions or the test are performed on the basis of features of the given dataset.

**Naïve Bayes**

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.It is mainly used in text classification that includes a high-dimensional training dataset.Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.It is a probabilistic classifier, which means it predicts on the basis of the probability of an object

This is system is of divided as four modules they are

**1.Import Dataset**

**2. Preprocess Dataset**

**3.Train With Machine Learning Algorithms**

**4.Upload Test Data & Verify Statistics.**

**IMPORT DATASET :** This phase it will import the dataset into the project. The dataset has the values of Rainfall, Humidity, Temperature , Soil Parameters and PH values .The crop production dataset that is used to predict the name of the crop is fed into classification and regression algorithms.

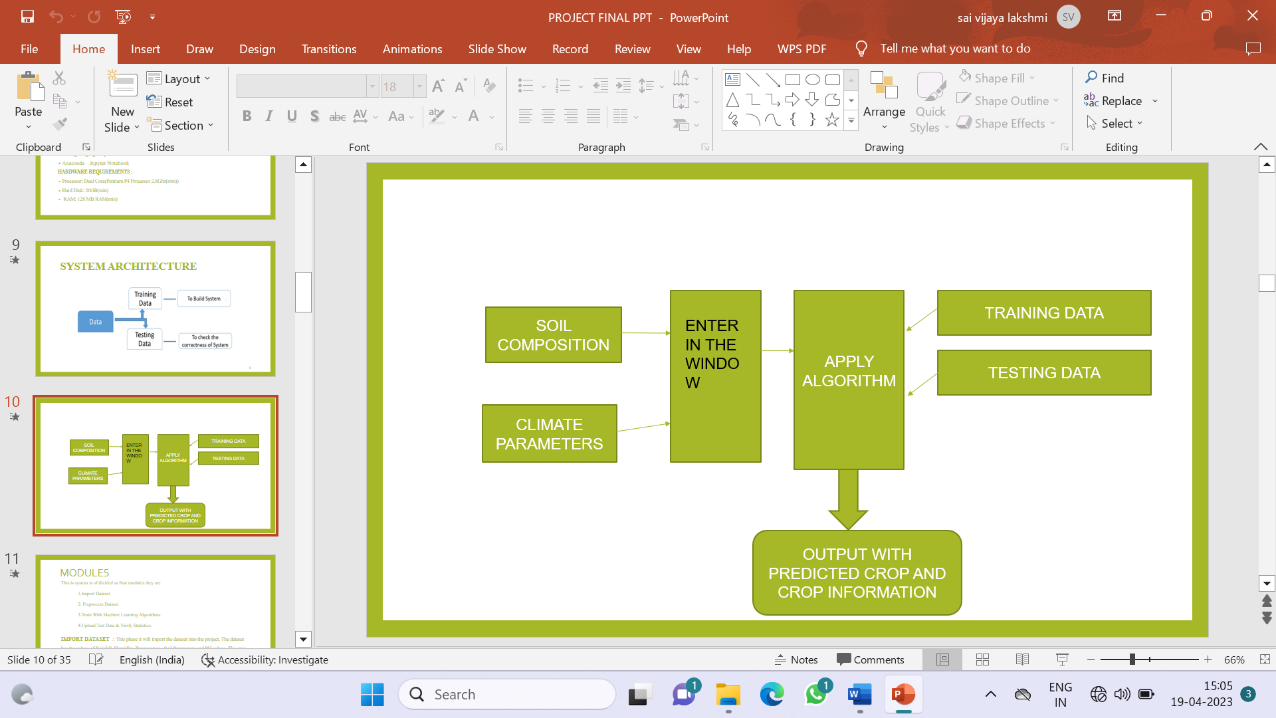
**PREPROCESS DATASET :**Experiments were conducted on dataset and it has been established that Random Forest Regressor gives the highest yield prediction accuracy. By combining rainfall, temperature along with other parameters like season and area, Crop and yield prediction for a certain district can be made.

**TRAIN WITH MACHINE LEARNING ALGORITHM:**This focus on the crop prediction according to the soil parameters like Potassium, Phosphorus and Nitrogen and Climate parameters like Rainfall, Humidity, Temperature and PH. This also focuses on district wise yield prediction according to the crop sown in the district. Yield is being predicted for given crops district wise and crops with best yield.

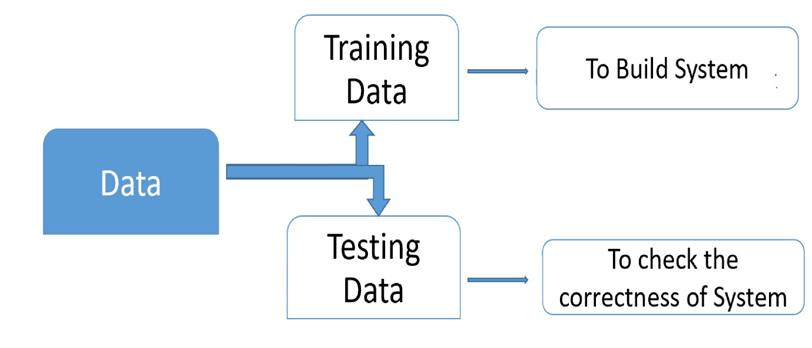
**UPLOAD TEST AND VERIFY THE STATISTICS:**

Results reveals that Random Forest is the best classifier when all parameters are combined. This will not only help farmers in choosing the right crop to grow in the next season but also bridge the gap between technology and the agriculture sector.

**4.3. SYSTEM ARCHITECTURE**

****

**Fig: 4.1. System Model**

****

**Fig 4.2 architecture diagram**

**4.4 INPUT & OUTPOUT REPRESENTETION**

## Input design :

Input design includes data mediums used for inputting data and validations that are to be done during data entry. Different messages regarding data are given to guide users during data entry. Validation checks are done for each input. Data entry screens are designed so that the system interacts with the user in providing an effective dialogue. Fields in the screen are logically arranged to help the user.

The design is the process of converting the user-originated inputs into a compute-based format. The goal of the input design is to make the data entry easier, logical and free from error. Errors in the input data are controlled by input design.The application has been developed in a user-friendly manner. The windows have been designed in such a way that during the processing the cursor is placed in the position where the data must be entered. If any of the data going into the system is wrong then the process and output will magnify these errors.

The decisions made during design of input are:

1. To achieve the highest possible level of accuracy.
2. To provide a list of possible choices and help while accepting the input for an important field wherever possible outputs from computer system are required primarily to communicate the results of processing to the users. They are also used to provide a permanent copy of these results for later consultation/verification.

**Output Design:**

Output refers to the results and information that are generated by the system. Output is the main reason for developing the system and based on this, the usefulness and applicability of system are evaluated.Outputs from computer systems are required primarily to communicate the results of processing to users. Efficiently designed outputs enhance the understandability of the information.

According to the requirements of the system, various types of outputs are considered and designed as follows.Internal outputs, whose destination is within the organization and which require careful design because they are the user’s main interface with the computer.Interactive outputs, in which the user communication with the Computer is essential.

**4.4. UML DIAGRAMS**

UML remains for Unified Modeling Language. UML is an institutionalized broadly useful displaying dialect in the field of protest situated programming designing. The standard is overseen, and was made by the Object Management Group. The objective is for UML to end up a typical dialect for making models of protest arranged PC programming. In its present shape UML is contained two noteworthy segments: a Meta-display and a documentation. Later on, some type of technique or process may likewise be added to; or connected with UML.

* The Unified Modeling Language is a standard dialect for indicating, Visualization, Constructing and recording the antiques of programming framework, and additionally for business displaying and other non-programming frameworks.
* The UML speaks to an accumulation of best building rehearses that have demonstrated effective in the displaying of vast and complex frameworks.The UML is an imperative piece of creating articles arranged programming and the product improvement prepare. The UML utilizes for the most part graphical documentations to express the plan of programming tasks.

**GOALS**

The Primary objectives in the plan of the UML are as per the following:

1. Provide clients a prepared to-utilize, expressive visual displaying Language with the goal that they can create and trade important models.

2. Provide extendibility and specialization instruments to develop the center ideas.

3. Be free of specific programming dialects and improvement handle.

4. Provide a formal reason for comprehension the displaying dialect.

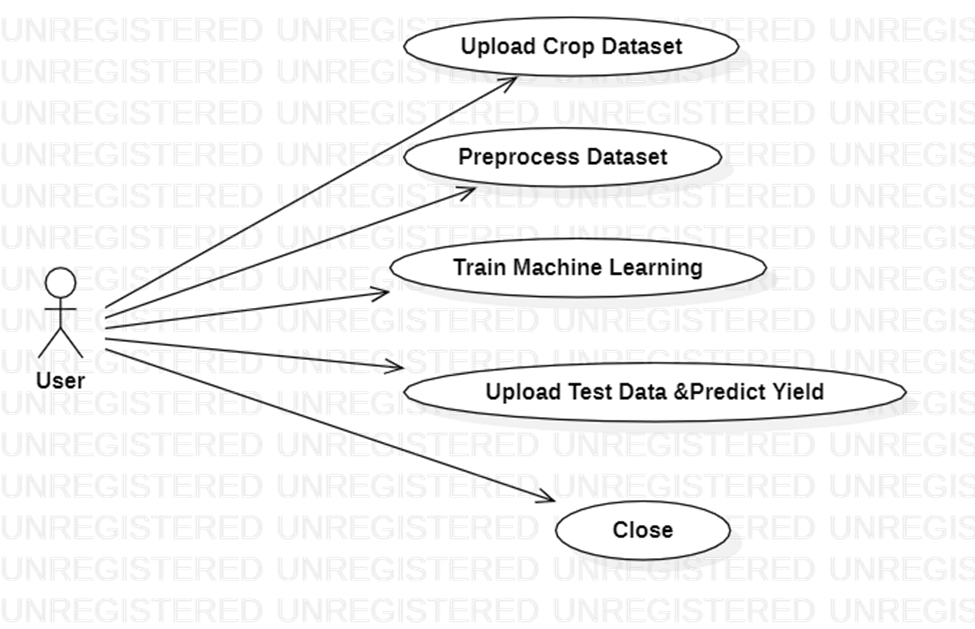
5. Encourage the development of OO devices showcase.

6. Support more elevated amount improvement ideas, for example, coordinated efforts, systems, examples and parts.

7. Integrate best practices.

**4.4.1. USE CASE DIAGRAM**

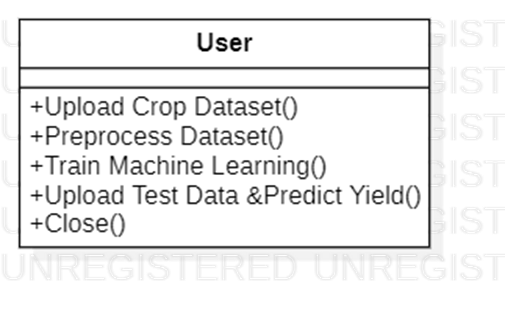
A use case diagram within the unified modeling language (UML) may be a kind of activity diagram outlined by and created from a use-case analysis. Its purpose is to gift a graphical summary of the practicality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. the most purpose of a use case diagram is to indicate what system functions area unit performed that actor. Roles of the actors within the system is represented.

****

**Fig: 4.3. Use Case Diagram**

**4.4.2. CLASS DIAGRAM**

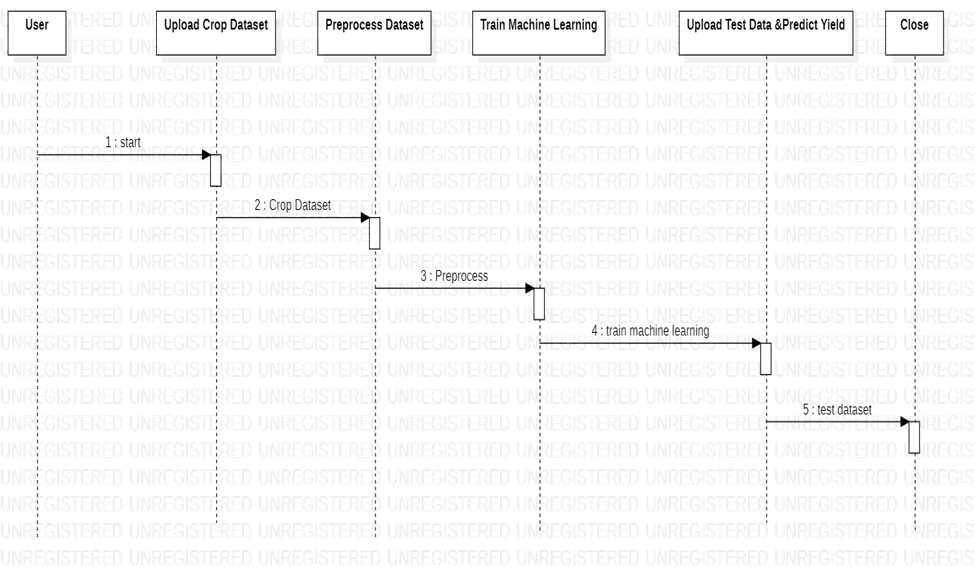
In computer code engineering, a category diagram within the Unified Modeling Language (UML) may be a kind of static structure diagram that describes the structure of a system by showing the system's categories, their attributes, operations (or methods), and also the relationships among the categories. It explains that category contains data.

****

**Fig: 4.4. Class Diagram**

**4.4.3. SEQUENCE DIAGRAM**

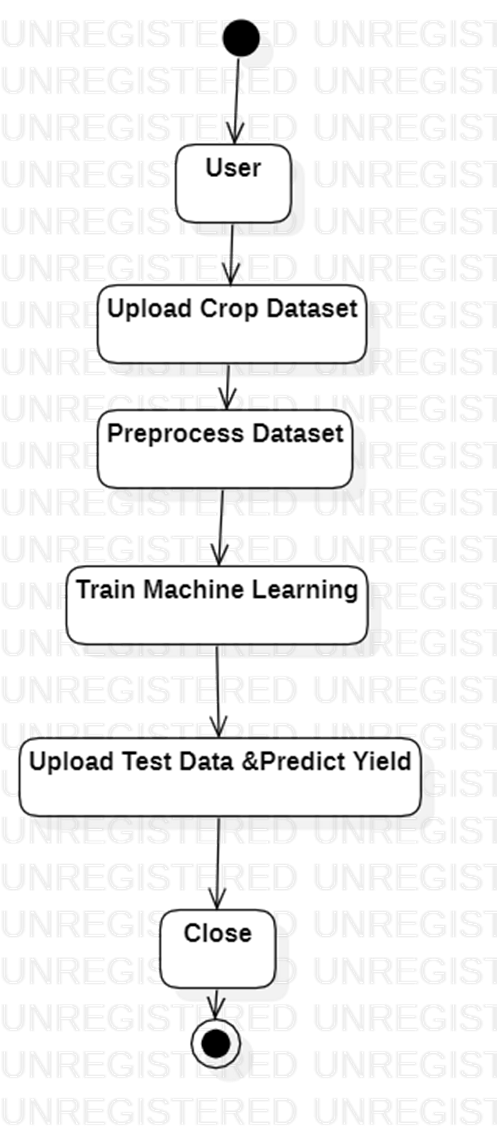
A sequence diagram in Unified Modeling Language (UML) may be a quite interaction diagram that shows however processes operate with each other and in what order. it's a construct of a Message Sequence Chart. Sequence diagrams ar generally known as event diagrams, event situations, and temporal order diagrams.

\

**Fig: 4.5 Sequence Diagram**

**4.4.4 ACTIVITY DIAGARAM :**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**Fig : 4.6 ACTIVITY DIAGRAM**

**CHAPTER - 5**

**SYSTEM IMPLEMENTATION**

**5.1. INTRODUCTION**

The production of crops may depend on geographical conditions of the region like river ground, hill areas or the depth areas. Weather conditions like humidity, rainfall, temperature, cloud. Soil type may be clay, sandy, saline, or peaty. Soil composition can be copper, potassium, phosphate, nitrogen, manganese, iron, calcium. 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 do not have enough knowledge about the new crops and are not completely aware of the benefits they get while farming them. The scope of this project is the proposed system applies machine learning and prediction algorithm to identify the pattern among data and then process it as per input conditions .Thus, this system will only require the location of the user and it will suggest number of profitable crops providing a choice directly to the farmer about which crop to cultivate

**5.2 TECHNOLOGY DESCRIPTION**

**Introduction to Python**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3. The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it.[32][33] With Python 2's end-of-life, only Python 3.5.x and later are supported. Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source[35] reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

**SYNTAX AND SEMANTICS**

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation.

Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

**INDENTATION**

Main article: Python syntax and semantics § Indentation

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure. This feature is sometimes termed the off-side rule, which some other languages share, but in most languages indentation doesn't have any semantic meaning.

**STATEMENTS AND FLOW DATA**

Python's statements include (among others):

The assignment statement (token '=', the equals sign). This operates differently than in traditional imperative programming languages, and this fundamental mechanism (including the nature of Python's version of variables) illuminates many other features of the language. Assignment in C, e.g., x = 2, translates to "typed variable name x receives a copy of numeric value 2". The (right-hand) value is copied into an allocated storage location for which the (left-hand) variable name is the symbolic address. The memory allocated to the variable is large enough (potentially quite large) for the declared type. In the simplest case of Python assignment, using the same example, x = 2, translates to "(generic) name x receives a reference to a separate, dynamically allocated object of numeric (int) type of value 2." This is termed binding the name to the object. Since the name's storage location doesn't contain the indicated value, it is improper to call it a variable.

Names may be subsequently rebound at any time to objects of greatly varying types, including strings, procedures, complex objects with data and methods, etc. Successive assignments of a common value to multiple names, e.g., x = 2; y = 2; z = 2 result in allocating storage to (at most) three names and one numeric object, to which all three names are bound.

Since a name is a generic reference holder it is unreasonable to associate a fixed data type with it. However at a given time a name will be bound to some object, which will have a type; thus there is dynamic typing.

* The if statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).
* The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
* The while statement, which executes a block of code as long as its condition is true.
* The try statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.
* The raise statement, used to raise a specified exception or re-raise a caught exception.
* The class statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.
* The def statement, which defines a function or method.
* The with statement, from Python 2.5 released in September 2006, which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing Resource Acquisition Is Initialization (RAII)-like behavior and replaces a common try/finally idiom.
* The break statement, exits from the loop.
* The continue statement, skips this iteration and continues with the next item.
* The pass statement, which serves as a NOP. It is syntactically needed to create an empty code block.
* The assert statement, used during debugging to check for conditions that ought to apply.
* The yield statement, which returns a value from a generator function. From Python 2.5, yield is also an operator. This form is used to implement coroutines.

The import statement, which is used to import modules whose functions or variables can be used in the current program. There are three ways of using import: import <module name> [as <alias>] or from <module name> import \* or from <module name> import <definition 1> [as <alias 1>], <definition 2> [as <alias 2>],

The print statement was changed to the print() function in Python 3.

Python does not support tail call optimization or first-class continuations, and, according to Guido van Rossum, it never will. However, better support for coroutine-like functionality is provided in 2.5, by extending Python's generators. Before 2.5, generators were lazy iterators; information was passed unidirectionally out of the generator. From Python 2.5, it is possible to pass information back into a generator function, and from Python 3.3, the information can be passed through multiple stack levels.

**EXPRESSIONS**

Some Python expressions are similar to languages such as C and Java, while some are not: Addition, subtraction, and multiplication are the same, but the behavior of division differs. There are two types of divisions in Python. They are floor division (or integer division) // and floating point/division. Python also added the \*\* operator for exponentiation.

From Python 3.5, the new @ infix operator was introduced. It is intended to be used by libraries such as NumPy for matrix multiplication.

From Python 3.8, the syntax :=, called the 'walrus operator' was introduced. It assigns values to variables as part of a larger expression.

In Python, == compares by value, versus Java, which compares numerics by valueand objects by reference. (Value comparisons in Java on objects can be performed with the equals() method.) Python's is operator may be used to compare object identities (comparison by reference). In Python, comparisons may be chained, for example a <= b <= c.

Python uses the words and, or, not for its boolean operators rather than the symbolic &&, ||, !used in Java and C.

Python has a type of expression termed a list comprehension. Python 2.4 extended list comprehensions into a more general expression termed a generator expression.

Anonymous functions are implemented using lambda expressions; however, these are limited in that the body can only be one expression.

Conditional expressions in Python are written as x if c else y (different in order of operands from the c ? x : y operator common to many other languages).

Python makes a distinction between lists and tuples. Lists are written as [1, 2, 3], are mutable, and cannot be used as the keys of dictionaries (dictionary keys must be immutable in Python). Tuples are written as (1, 2, 3), are immutable and thus can be used as the keys of dictionaries, provided all elements of the tuple are immutable. The + operator can be used to concatenate two tuples, which does not directly modify their contents, but rather produces a new tuple containing the elements of both provided tuples. Thus, given the variable t initially equal to (1, 2, 3), executing t = t + (4, 5) first evaluates t + (4, 5), which yields (1, 2, 3, 4, 5), which is then assigned back to t, thereby effectively "modifying the contents" of t, while conforming to the immutable nature of tuple objects. Parentheses are optional for tuples in unambiguous contexts.

Python features sequence unpacking wherein multiple expressions, each evaluating to anything that can be assigned to (a variable, a writable property, etc.), are associated in the identical manner to that forming tuple literals and, as a whole, are put on the left hand side of the equal sign in an assignment statement.

The statement expects an iterable object on the right hand side of the equal sign that produces the same number of values as the provided writable expressions when iterated through, and will iterate through it, assigning each of the produced values to the corresponding expression on the left.

Python has a "string format" operator %. This functions analogous to printf format strings in C, e.g. "spam=%s eggs=%d" % ("blah", 2) evaluates to "spam=blah eggs=2".

In Python 3 and 2.6+, this was supplemented by the format() method of the str class, e.g. "spam={0} eggs={1}".format("blah", 2). Python 3.6 added "f-strings": blah = "blah"; eggs = 2; f'spam={blah} eggs={eggs}'.

Python has various kinds of string literals:

Strings delimited by single or double quote marks. Unlike in Unix shells, Perl and Perl-influenced languages, single quote marks and double quote marks function identically. Both kinds of string use the backslash (\) as an escape character. String interpolation became available in Python 3.6 as "formatted string literals".

Triple-quoted strings, which begin and end with a series of three single or double quote marks. They may span multiple lines and function like here documents in shells, Perl and Ruby.

Raw string varieties, denoted by prefixing the string literal with an r. Escape sequences are not interpreted; hence raw strings are useful where literal backslashes are common, such as regular expressions and Windows-style paths. Compare "@-quoting" in C#.

Python has array index and array slicing expressions on lists, denoted as a[key], a[start:stop] or a[start:stop:step]. Indexes are zero-based, and negative indexes are relative to the end. Slices take elements from the start index up to, but not including, the stop index. The third slice parameter, called step or stride, allows elements to be skipped and reversed. Slice indexes may be omitted, for example a[:] returns a copy of the entire list. Each element of a slice is a shallow copy.

In Python, a distinction between expressions and statements is rigidly enforced, in contrast to languages such as Common Lisp, Scheme, or Ruby. This leads to duplicating some functionality. For example:

List comprehensions vs. for-loops

Conditional expressions vs. if blocks

The eval() vs. exec() built-in functions (in Python 2, exec is a statement); the former is for expressions, the latter is for statements.

Statements cannot be a part of an expression, so list and other comprehensions or lambda expressions, all being expressions, cannot contain statements. A particular case of this is that an assignment statement such as a = 1 cannot form part of the conditional expression of a conditional statement. This has the advantage of avoiding a classic C error of mistaking an assignment operator = for an equality operator == in conditions: if (c = 1) { ... } is syntactically valid (but probably unintended) C code but if c = 1: ... causes a syntax error in Python.

**METHODS**

Methods on objects are functions attached to the object's class; the syntax instance.method(argument) is, for normal methods and functions, syntactic sugar for Class.method(instance, argument). Python methods have an explicit self parameter to access instance data, in contrast to the implicit self (or this) in some other object-oriented programming languages (e.g., C++, Java, Objective-C, or Ruby).

**APPLICATIONS OF PYTHON**

As mentioned before, Python is one of the most widely used language over the web. I'm going to list few of them here:

**Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

**Easy-to-read** − Python code is more clearly defined and visible to the eyes.

**Easy-to-maintain** − Python's source code is fairly easy-to-maintain.

**A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

**Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

**Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

**Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

**Databases** − Python provides interfaces to all major commercial databases.

**GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

**Scalable** − Python provides a better structure and support for large programs than shell scripting.

INSTALLATION STEPS OF PYTHON

Installing and using Python on Windows 10 is very simple. The installation procedure involves just three steps:

* Download the binaries
* Run the Executable installer
* Add Python to PATH environmental variables

To install Python, you need to download the official Python executable installer. Next, you need to run this installer and complete the installation steps. Finally, you can configure the PATH variable to use python from the command line.

**Step 1**: Download the Python Installer binaries

* Open the official Python website in your web browser. Navigate to the Downloads tab for Windows.
* Choose the latest Python 3 release. In our example, we choose the latest Python 3.7.3 version. Click on the link to download Windows x86 executable installer if you are using a 32-bit installer.
* In case your Windows installation is a 64-bit system, then download Windows x86-64 executable installer.

**Step 2:** Run the Executable Installer

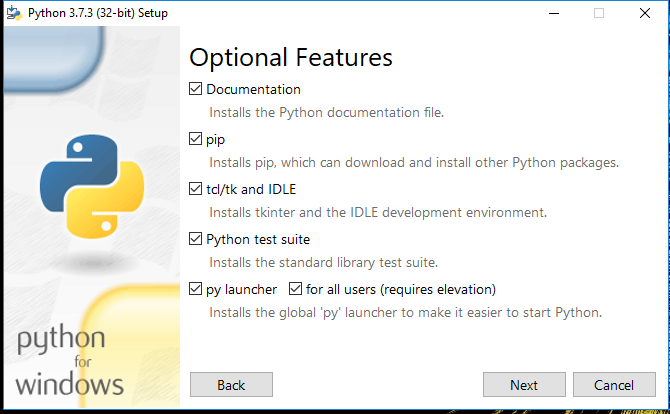
1. Once the installer is downloaded, run the Python installer.
2. Check the Install launcher for all users check box. Further, you may check the Add Python 3.7 to path check box to include the interpreter in the execution path.



1. Select **Customize installation**.

Choose the optional features by checking the following check boxes:

1. Documentation
2. pip
3. tcl/tk and IDLE (to install tkinter and IDLE)
4. Python test suite (to install the standard library test suite of Python)
5. Install the global launcher for `.py` files. This makes it easier to start Python
6. Install for all users.

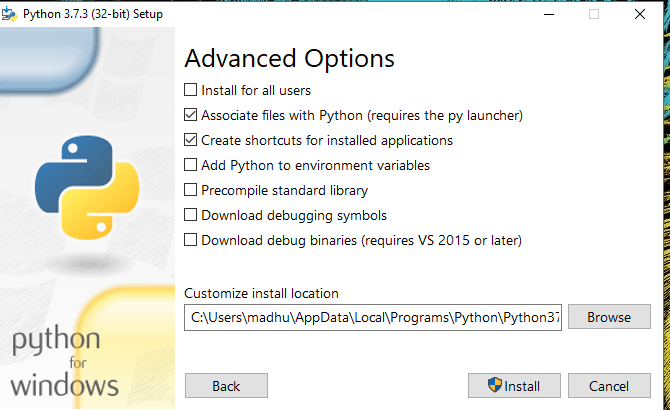


Click Next.

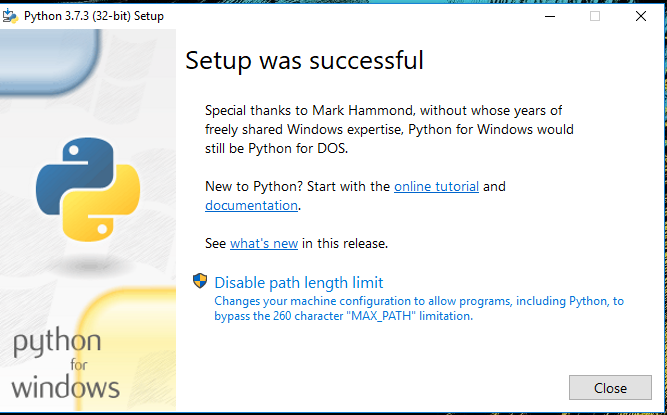
1. This takes you to Advanced Options available while installing Python. Here, select the Install for all users and Add Python to environment variables check boxes.

Optionally, you can select the Associate files with Python, Create shortcuts for installed applications and other advanced options. Make note of the python installation directory displayed in this step. You would need it for the next step.

After selecting the Advanced options, click Install to start installation.



1. Once the installation is over, you will see a Python Setup Successful window.



**Step 3:** Add Python to environmental variables

The last (optional) step in the installation process is to add Python Path to the System Environment variables. This step is done to access Python through the command line. In case you have added Python to environment variables while setting the Advanced options during the installation procedure, you can avoid this step. Else, this step is done manually as follows.

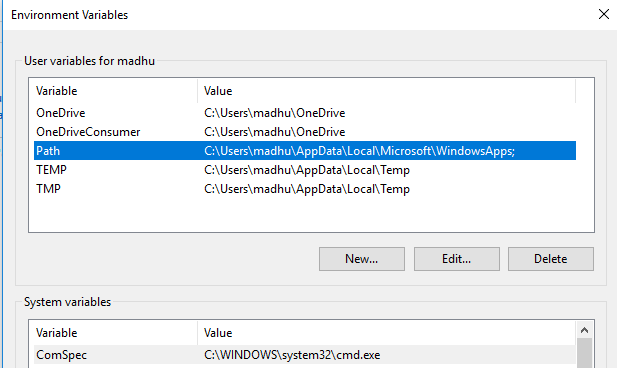
In the Start menu, search for “advanced system settings”. Select “View advanced system settings”. In the “System Properties” window, click on the “Advanced” tab and then click on the “Environment Variables” button.

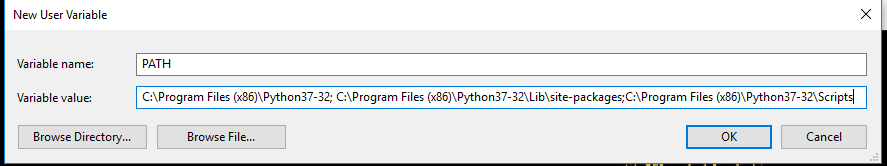
Locate the Python installation directory on your system. If you followed the steps exactly as above, python will be installed in below locations:

* C:\Program Files (x86)\Python37-32: for 32-bit installation
* C:\Program Files\Python37-32: for 64-bit installation

The folder name may be different from “Python37-32” if you installed a different version. Look for a folder whose name starts with Python.

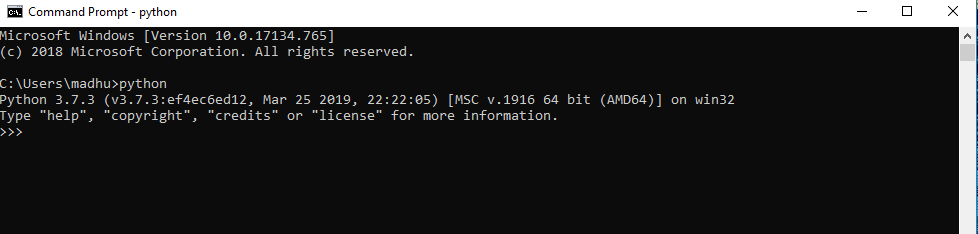
Append the following entries to PATH variable as shown below:



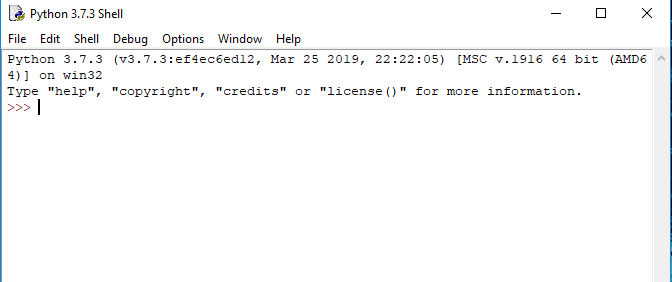


**Step 4:** Verify the Python Installation

You have now successfully installed Python 3.7.3 on Windows 10. You can verify if the Python installation is successful either through the command line or through the IDLE app that gets installed along with the installation. Search for the command prompt and type “python”. You can see that Python 3.7.3 is successfully installed.



An alternate way to reach python is to search for “Python” in the start menu and clicking on IDLE (Python 3.7 64-bit). You can start coding in Python using the Integrated Development Environment(IDLE).



**Uses**

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index where, as of February 2020, it is the third most popular language (behind Java, and C). It was selected Programming Language of the Year in 2007, 2010, and 2018.

* An empirical study found that scripting languages, such as Python, are more productive than conventional languages, such as C and Java, for programming problems involving string manipulation and search in a dictionary, and determined that memory consumption was often "better than Java and not much worse than C or C++".
* Large organizations that use Python include Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify and some smaller entities like ILM and ITA. The social news networking site Reddit is written entirely in Python.
* Python can serve as a scripting language for web applications, e.g., via mod\_wsgi for the Apache web server. With Web Server Gateway Interface, a standard API has evolved to facilitate these applications. Web frameworks like Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications.
* SQLAlchemy can be used as data mapper to a relational database. Twisted is a framework to program communications between computers, and is used (for example) by Dropbox.
* Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python in scientific computing, with specialized libraries such as Biopython and Astropy providing domain-specific functionality. SageMath is a mathematical software with a notebook interface programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus.
* Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go.
* Python is commonly used in artificial intelligence projects with the help of libraries like TensorFlow, Keras, Pytorch and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.
* Many operating systems include Python as a standard component. It ships with most Linux distributions, AmigaOS 4, FreeBSD (as a package), NetBSD, OpenBSD (as a package) and macOS and can be used from the command line (terminal). Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.
* Python is used extensively in the information security industry, including in exploit development.
* Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python. The Raspberry Pi single-board computer project has adopted Python as its main user-programming language.
* Due to Python's user-friendly conventions and easy-to-understand language, it is commonly used as an intro language into computing sciences with students. This allows students to easily learn computing theories and concepts and then apply them to other programming languages.
* LibreOffice includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature[169] since Version 4.0 from 7 February 2013.

.

**5.3 MODULE IMPLEMENTATION**

**5.3.1 SOURCE CODE**

# importing necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# loading the dataset

crop\_data=pd.read\_csv("Crop\_recommendation.csv")

crop\_data

#rows and columns

crop\_data.shape

#checking basic information against columns

crop\_data.info()

# dataset columns

crop\_data.columns

#Changing the name of label to Crop for readability

crop\_data.rename(columns = {'label':'Crop'}, inplace = True)

crop\_data

# statistical inference of the dataset

crop\_data.describe()

#Dropping missing values

crop\_data = crop\_data.dropna()

crop\_data

# Visualizing the features

ax = sns.pairplot(crop\_data)

ax

crop\_data.Crop.unique()

# get top 5 most frequent growing crops

n = 5

crop\_data['Crop'].value\_counts()[:5].index.tolist()

sns.barplot(crop\_data["Crop"], crop\_data["temperature"])

plt.xticks(rotation = 90)

sns.barplot(crop\_data["Crop"], crop\_data["ph"])

plt.xticks(rotation = 90)

sns.barplot(crop\_data["Crop"], crop\_data["humidity"])

plt.xticks(rotation = 90)

sns.barplot(crop\_data["Crop"], crop\_data["rainfall"])

plt.xticks(rotation = 90)

crop\_data.corr()

sns.heatmap(crop\_data.corr(), annot =True)

plt.title('Correlation Matrix')

# shuffling the dataset to remove order

from sklearn.utils import shuffle

df = shuffle(crop\_data,random\_state=5)

df.head()

# Selection of Feature and Target variables.

x = df[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]

target = df['Crop']

# Encoding target variable

y = pd.get\_dummies(target)

y

# Splitting data set - 25% test dataset and 75%

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.25, random\_state= 0)

print("x\_train :",x\_train.shape)

print("x\_test :",x\_test.shape)

print("y\_train :",y\_train.shape)

print("y\_test :",y\_test.shape)

from sklearn.datasets import make\_classification

from sklearn.multioutput import MultiOutputClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive\_bayes import GaussianNB

# creating a confusion matrix

from sklearn.metrics import confusion\_matrix

cm=confusion\_matrix(y\_test.values.argmax(axis=1), gnb\_pred.argmax(axis=1))

#cm = confusion\_matrix(y\_test, gnb\_pred)

ax= plt.subplot()

sns.heatmap(cm, annot=True, fmt='g', ax=ax);

# labels, title and ticks

ax.set\_xlabel('Predicted labels');ax.set\_ylabel('True labels');

ax.set\_title('Confusion Matrix');

from sklearn import metrics

# Print the confusion matrix

print(metrics.confusion\_matrix(y\_test.values.argmax(axis=1), gnb\_pred.argmax(axis=1)))

# Print the precision and recall, among other metrics

print(metrics.classification\_report(y\_test.values.argmax(axis=1), gnb\_pred.argmax(axis=1), digits=3))

#Random Foresrt Algorithm

# Training

forest = RandomForestClassifier(random\_state=1)

multi\_target\_forest = MultiOutputClassifier(forest, n\_jobs=-1)

multi\_target\_forest.fit(x\_train, y\_train)

# Predicting test results

forest\_pred = multi\_target\_forest.predict(x\_test)

forest\_pred

# Calculating Accuracy

from sklearn.metrics import accuracy\_score

a3 = accuracy\_score(y\_test.values.argmax(axis=1), forest\_pred.argmax(axis=1))

a3

# creating a confusion matrix

from sklearn.metrics import confusion\_matrix

cm=confusion\_matrix(y\_test.values.argmax(axis=1), forest\_pred.argmax(axis=1))

#cm = confusion\_matrix(y\_test, gnb\_pred)

ax= plt.subplot()

sns.heatmap(cm, annot=True, fmt='g', ax=ax);

# labels, title and ticks

ax.set\_xlabel('Predicted labels');ax.set\_ylabel('True labels');

ax.set\_title('Confusion Matrix');

from sklearn import metrics

# Print the confusion matrix

print(metrics.confusion\_matrix(y\_test.values.argmax(axis=1), forest\_pred.argmax(axis=1)))

# Print the precision and recall, among other metrics

print(metrics.classification\_report(y\_test.values.argmax(axis=1), forest\_pred.argmax(axis=1), digits=3))

# loading the dataset2

crop\_data=pd.read\_csv("crop\_production.csv")

crop\_data

crop\_data.shape

crop\_data.columns

crop\_data.describe()

crop\_data.isnull().sum()

# Dropping missing values

crop\_data = crop\_data.dropna()

crop\_data

#checking

crop\_data.isnull().values.any()

# Displaying State Names present in the dataset

crop\_data.State\_Name.unique()

# Adding a new column Yield which indicates Production per unit Area.

crop\_data['Yield'] = (crop\_data['Production'] / crop\_data['Area'])

crop\_data.head(10)

# Visualizing the features

ax = sns.pairplot(crop\_data)

ax

from sklearn.model\_selection import train\_test\_split

x = dummy.drop(["Production","Yield"], axis=1)

y = dummy["Production"]

# Splitting data set - 25% test dataset and 75%

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.25, random\_state=5)

print("x\_train :",x\_train.shape)

print("x\_test :",x\_test.shape)

print("y\_train :",y\_train.shape)

print("y\_test :",y\_test.shape

#Random Forest Algorithm

from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor(n\_estimators = 11)

model.fit(x\_train,y\_train)

rf\_predict = model.predict(x\_test)

rf\_predict

# Calculating R2 score

from sklearn.metrics import r2\_score

r1 = r2\_score(y\_test,rf\_predict)

print("R2 score : ",r1)

# Calculating Adj. R2 score:

Adjr2\_1 = 1 - (1-r)\*(len(y\_test)-1)/(len(y\_test)-x\_test.shape[1]-1)

print("Adj. R-Squared : {}".format(Adjr2\_1))

ax = sns.distplot(y\_test, hist = False, color = "r", label = "Actual value ")

sns.distplot(rf\_predict, hist = False, color = "b", label = "Predicted Values", ax = ax)

plt.title('Random Forest Regression')

import numpy as np

import matplotlib.pyplot as plt

# create a dataset

Algorithms = ['Random Forest', 'Decision-tree']

Accuracy = [r1, r2]

x\_pos = np.arange(len(Accuracy))

# Create bars with different colors

plt.bar(x\_pos, Accuracy, color=['#488AC7','#ff8c00'])

# Create names on the x-axis

plt.xticks(x\_pos, Algorithms)

plt.ylabel('R-Squared Score')

plt.xlabel('Machine Learning Regression Techniques')

# Show graph

plt.show()

plt.savefig('SD.png')

import numpy as np

import matplotlib.pyplot as plt

# create a dataset

Algorithms = ['Random Forest', 'Decision-tree']

Accuracy = [Adjr2\_1, Adjr2\_2]

x\_pos = np.arange(len(Accuracy))

# Create bars with different colors

plt.bar(x\_pos, Accuracy, color=['#488AC7','#ff8c00'])

# Create names on the x-axis

plt.xticks(x\_pos, Algorithms)

plt.ylabel('Adjusted R-Squared Score')

plt.xlabel('Machine Learning Regression Techniques')

# Show graph

plt.show()

plt.savefig('SD.png'

**5.4 SCREENSHOTS**

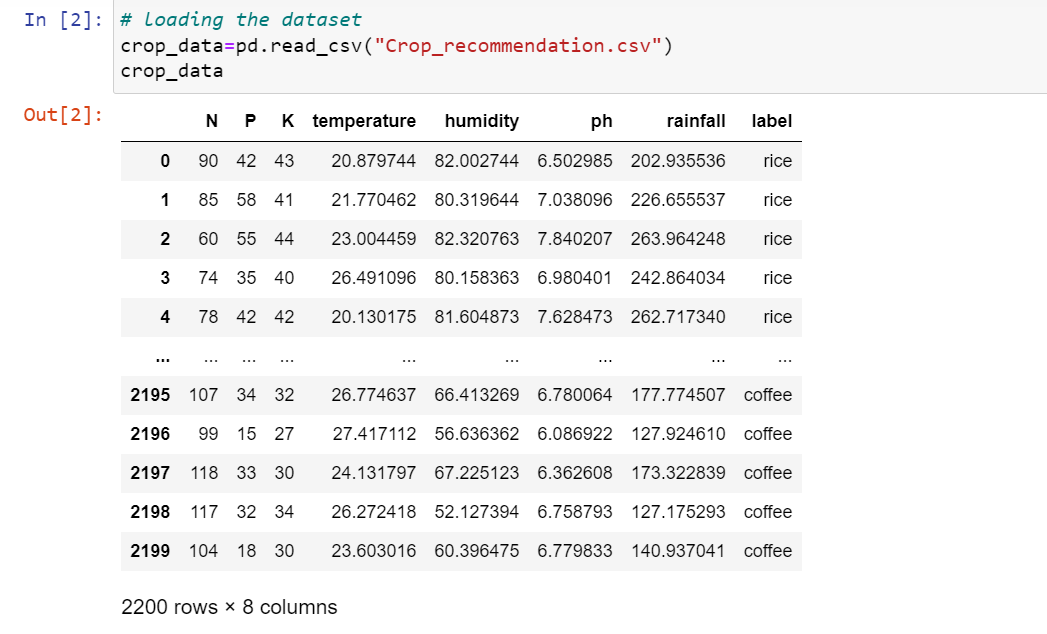


FIG :**5.1** Data set

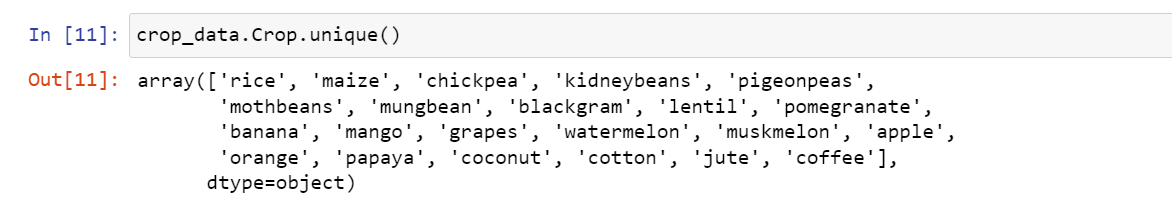


FIG **5.2**: Names of the crops

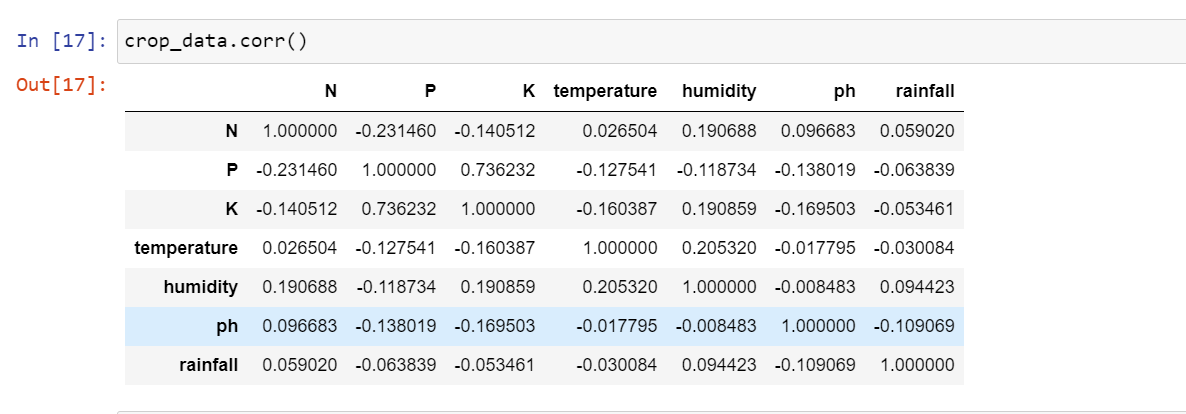
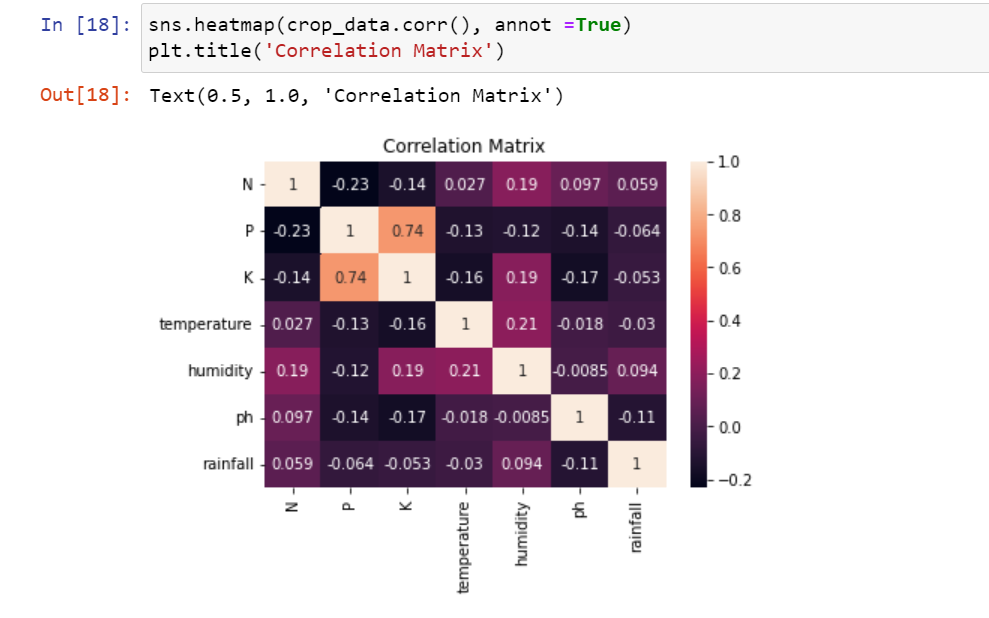


FIG **5.3**: Correlation Of Data



FIG **5.4**: Dropping The Null Data



**FIG** 5.5**: Correlation Matrix**

# 

FIG **5.6** : Statistics Of Data Set

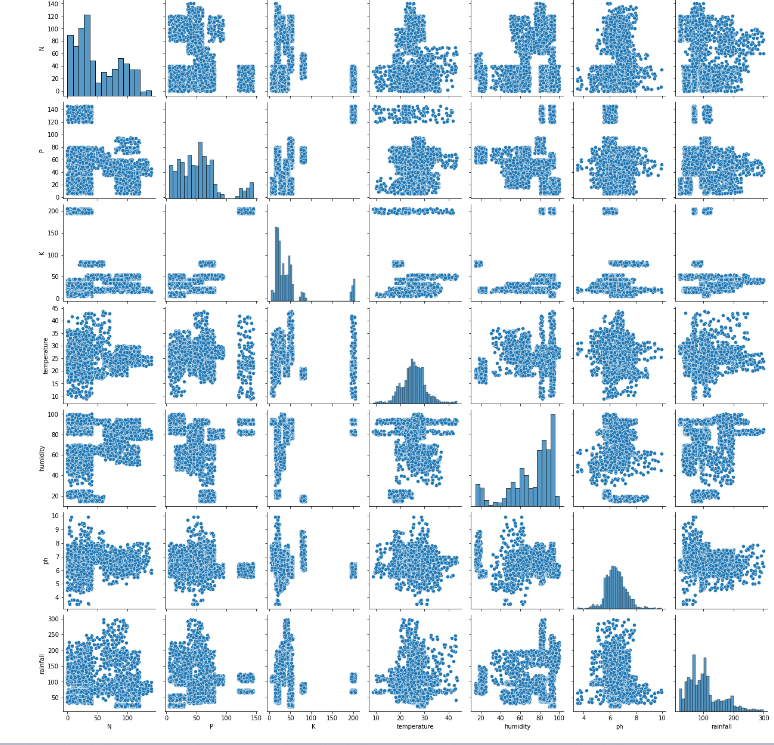


FIG **5.7**: Statistics of data set

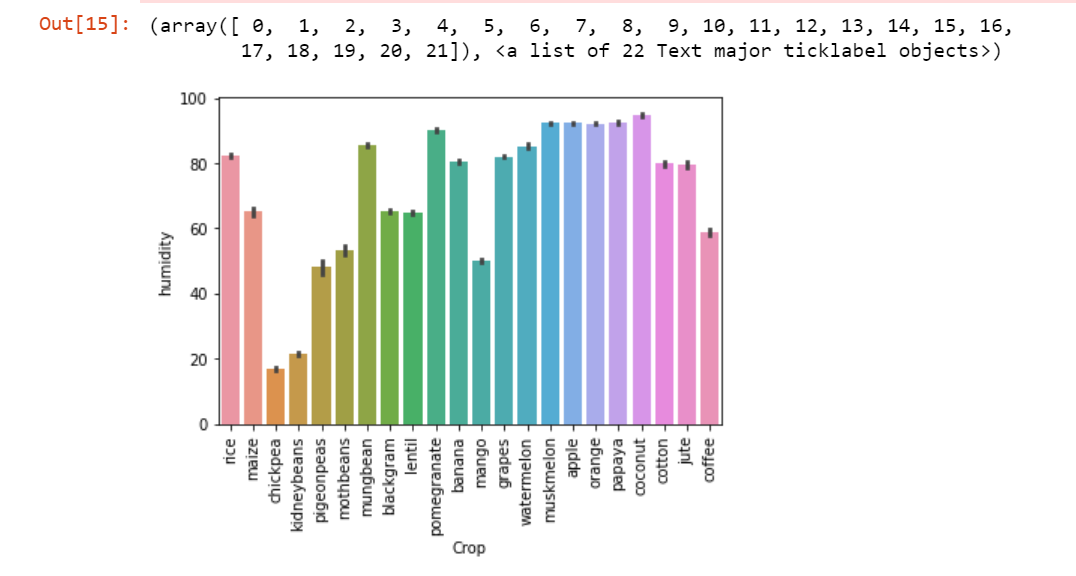


FIG **5.8** : HUMIDITY

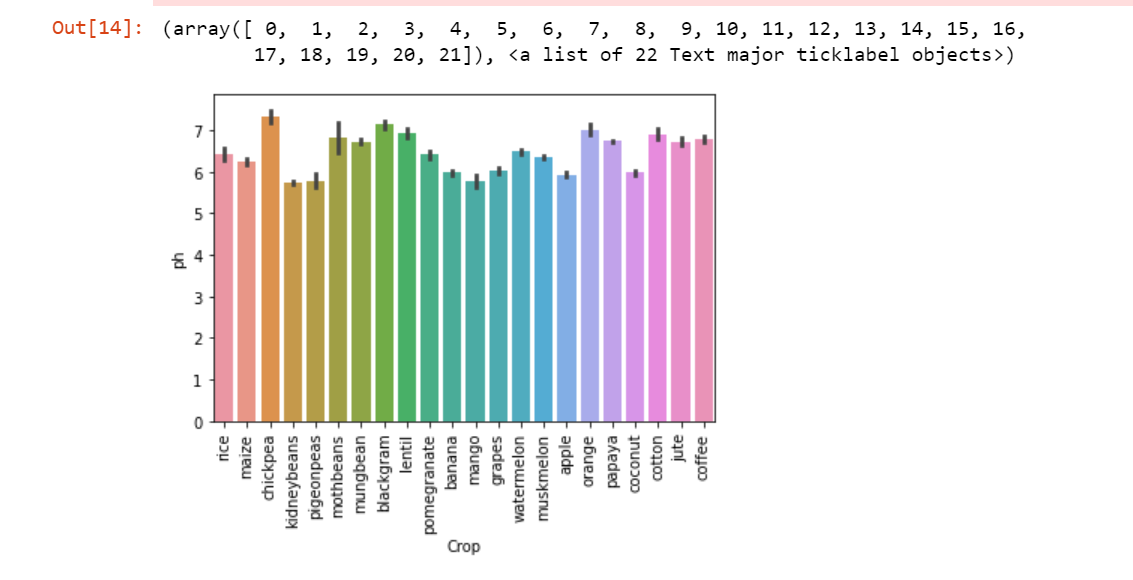


FIG **5.9**: PH

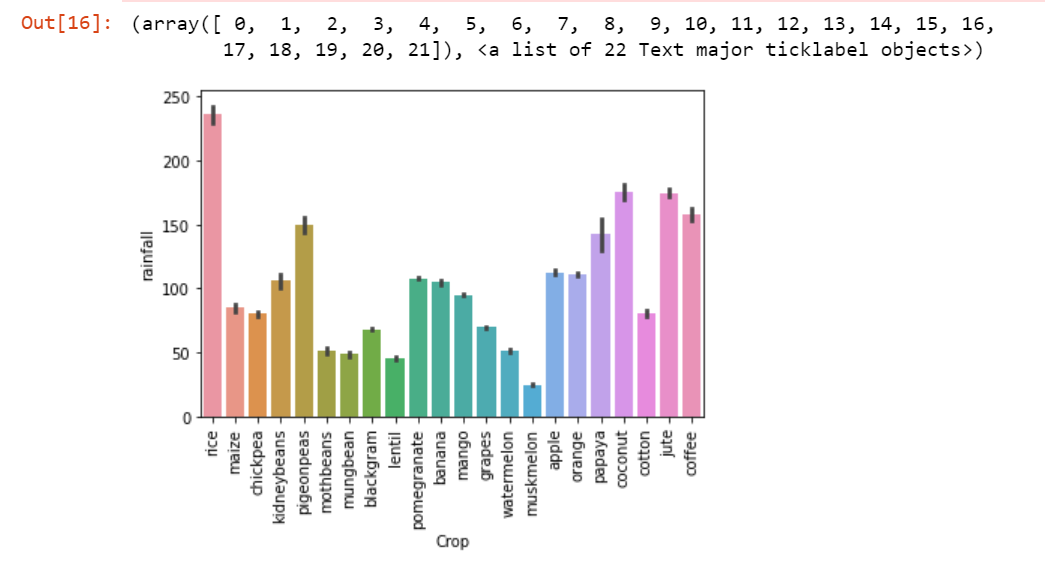


FIG **5.10** : RAINFALL

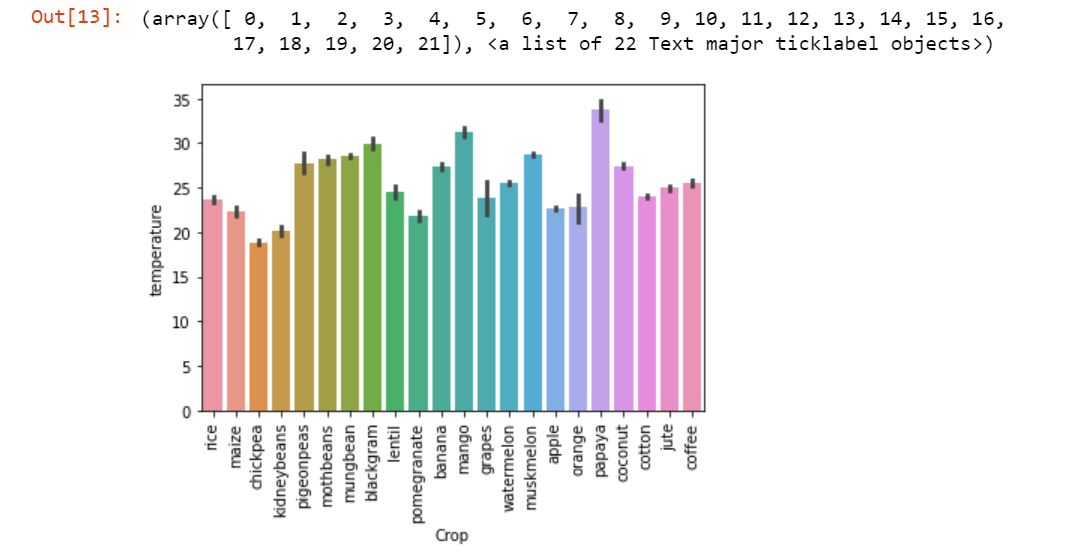


FIG **5.11** : TEMPERATURE

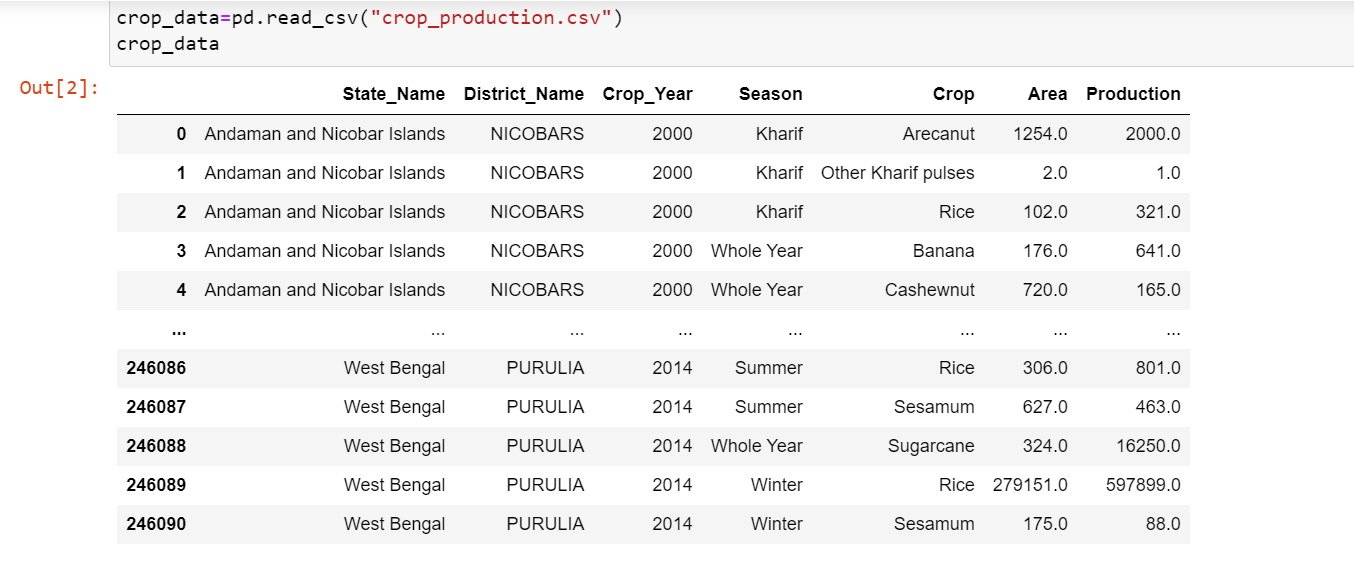


FIG **5.12** : UPLOADING DATASET2

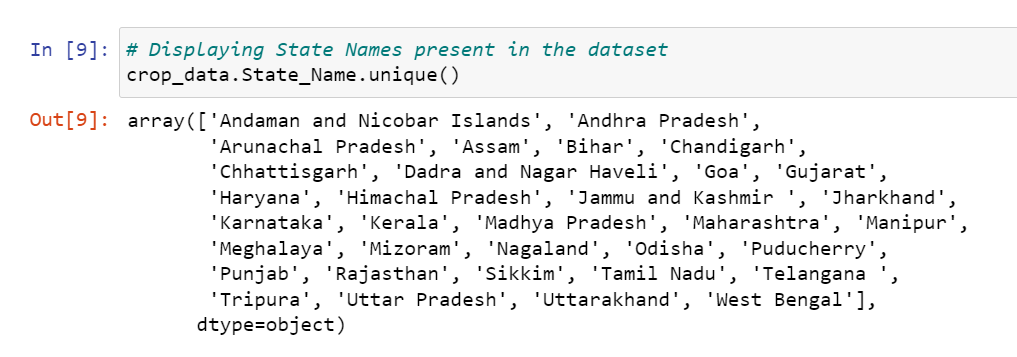


FIG **5.13** : STATES IN DATA SET 2

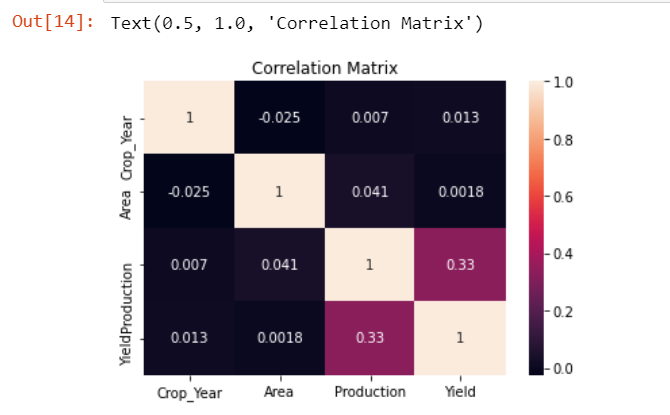


FIG **5.14** : CORRELATION MATRIX OF DATASET 2

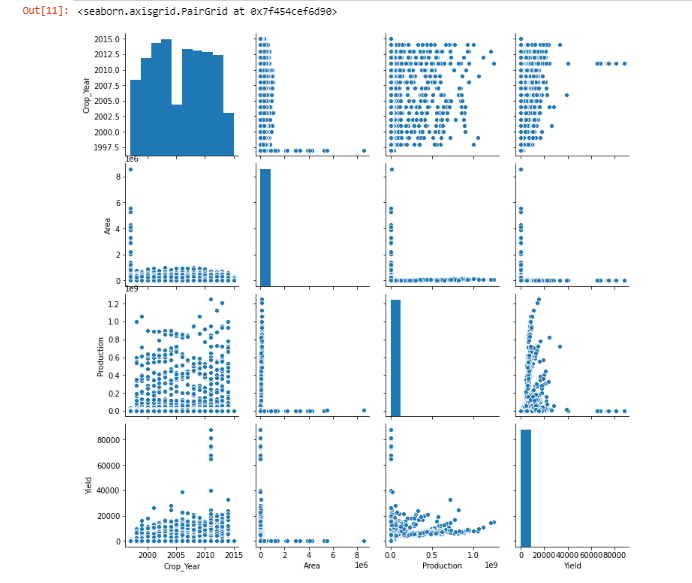


FIG **5.15** : STATISTICS OF DATASET 2

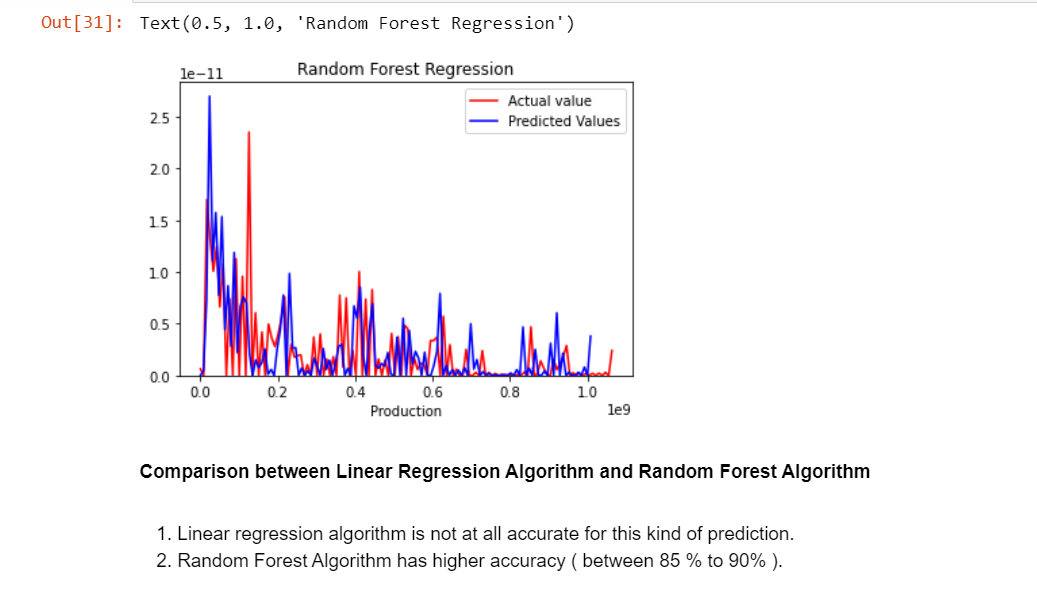


FIG **5.16** : RANDOM FOREST

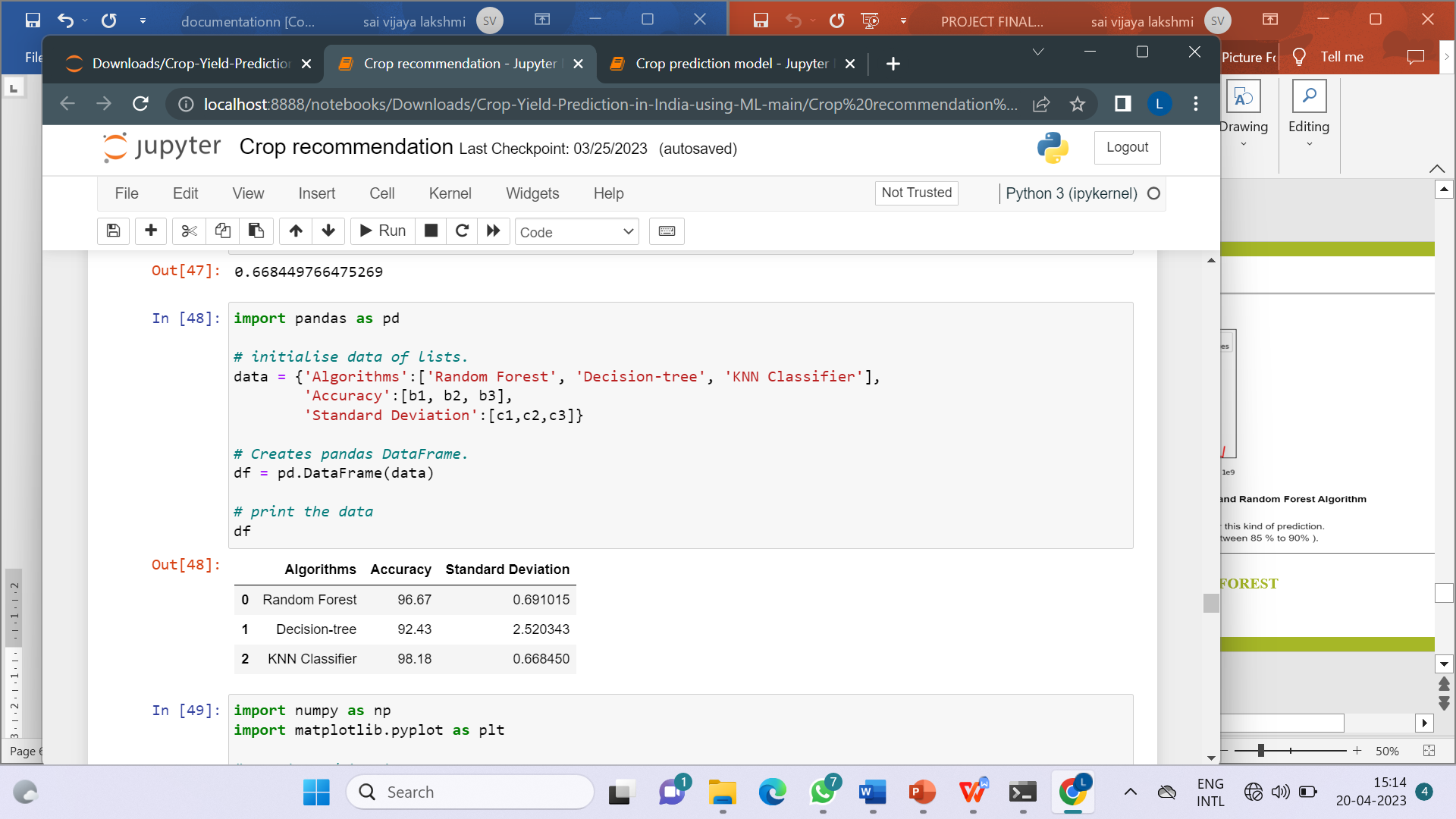


FIG **5.17** : ACCURACY OF RANDOM FOR EST AND DECISION TREE

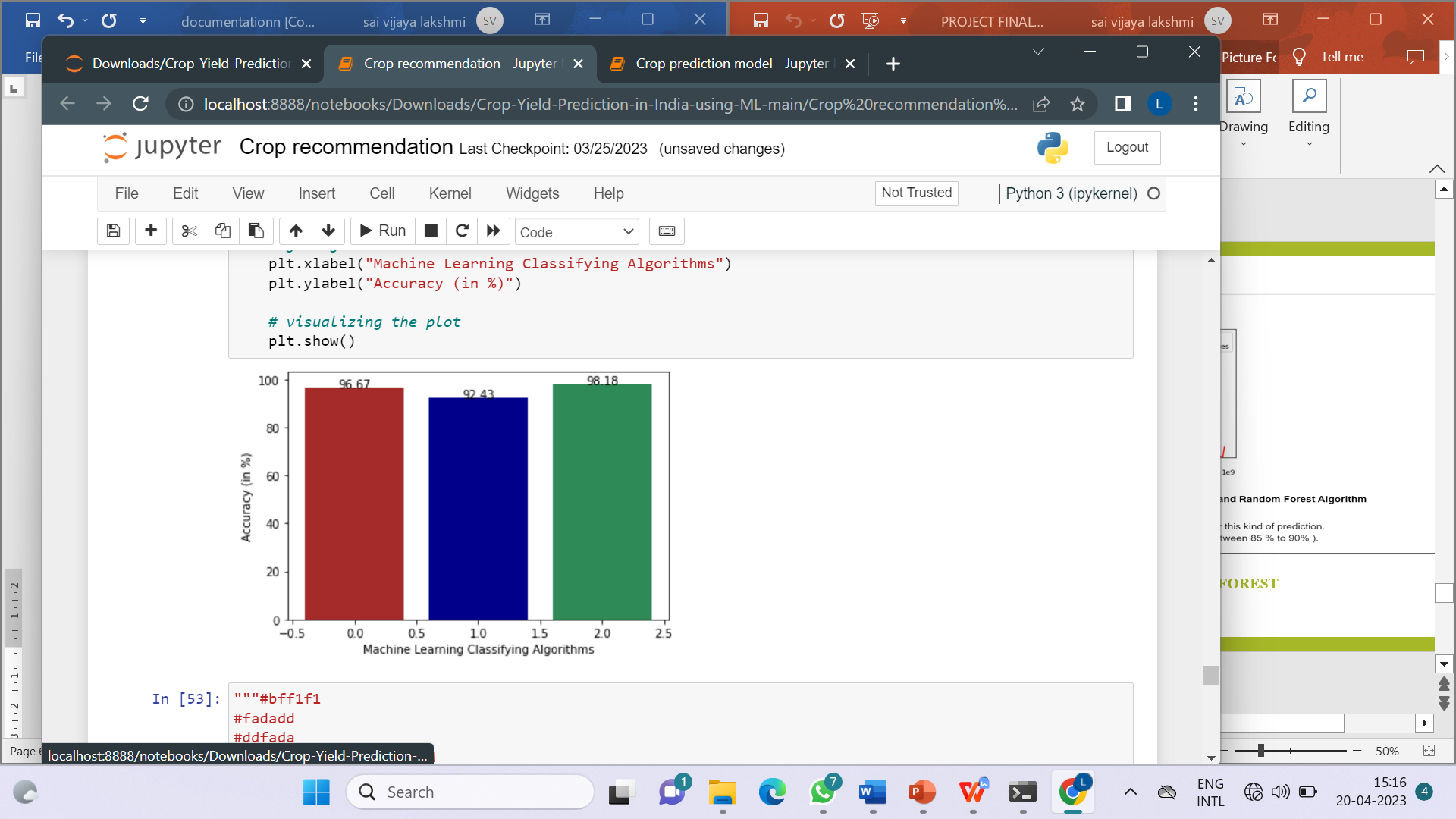


FIG **5.18** : RED -RANDOM FOREST AND BLUE IS DECISSION TREE.

**CHAPTER - 6**

**TESTING AND VALIDATIONS**

Testing is that the debugging program is one amongst the leading crucial aspects of the pc programming triggers, while not programming that works, the system would ne'er turn out relate in Nursing output of that it had been designed .Testing is best performed once user development is asked to help in characteristic all errors and bugs. The sample knowledge are used for testing. It is not amount however quality of the information used the matters of testing .Testing is aimed toward guaranteeing that the system was accurately relate in Nursing with efficiency before live operation commands.

**Testing objectives:** The most objective of testing is to uncover a bunch of errors, consistently and with minimum effort and time. Stating formally ,we can say, testing may be a method of corporal punishment a program with intent of finding miscalculation.

1. A productive check is one that uncovers Associate in Nursing hitherto undiscovered error.

2. A decent legal action is one that has likelihood of finding miscalculation, if it exists.

3. The check is insufficient to find probably gift errors.

4. The code additional or less confirms to the standard and reliable standards.

**6.1. TYPES OF TESTING**

**6.1.1. UNIT TESTING**

Unit testing we have a tendency to test every module separately and integrate with the general system. Unit testing focuses verification efforts on the littlest unit of code style within the module. this is often conjointly called module testing.

The module of the system is tested individually. as an example the validation check is completed for variable the user input given by the user that validity of the information entered. it's terribly straightforward to search out error rectify the system. Every Module will be tested victimization the subsequent 2 Strategies: recording machine Testing and White Box Testing.

**6.1.2. BLACK BOX TESTING**

Recording machine checking may be a code testing techniques during which practicality of the code below test (SUT) is tested while not staring at the interior code structure, implementation details and data of internal ways of the code .This type of testing is predicated entirely on the code needs and specifications .In recording machine Testing we have a tendency to simply concentrate on inputs and output of the package while not bothering concerning internal data of the code program. The on top of recording machine will be any package you wish to check. For example, Associate in Nursing software like Windows, a web site like Google ,a information like Oracle or maybe your own custom application. Under recording machine testing, you can check these applications by simply that specialize in the inputs and outputs while not knowing their internal code implementation.

**Types of Black Box Testing**

There are many varieties of recording machine Testing however following ar the outstanding ones.

**• Functional testing:** This recording machine testing kind is said to purposeful needs of a system; it's done by code testers.

**• Non-Functional testing:** This sort of recording machine testing isn't associated with testing of a selected practicality, however non-functional needs like performance, measurability, usability.

**• Regression testing:** Regression testing is completed once code fixes, upgrades or the other system maintenance to visualize the new code has not affected the prevailing code.

**6.1.3. WHITE BOX TESTING**

White Box Testing is that the testing of a code solution's internal committal to writing and infrastructure. It focuses totally on Traffic Redundancy Elimination ngthening security, the flow of inputs and outputs through the applying, and rising style and value. White box testing is additionally called clear, open, structural, and glass box testing. It is one amongst 2 elements of the "box testing" approach of code testing.

**System Testing:**

Once the individual module testing is completed, modules are assembled and integrated to perform as a system. The top down testing, that began from higher level to lower level module, was allotted to visualize whether or not the whole system is playacting satisfactorily. There are 3 main types of System testing: Alpha Testing, Beta Testing, Acceptance Testing.

**Alpha Testing:** This refers to the system checking that's allotted by the test team with the Organization.

**Beta Testing:** This refers to the system testing that's performed by a particular cluster of friendly customers.

**Acceptance Testing:** This refers to the system testing that's performed by the client to see whether or not or to not settle for the delivery of the system.

**6.2. TEST STRATEGY AND APPROACH**

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

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

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

**Test Scenarios**

General Scenarios

• All mandatory fields should be validated and indicated by asterisk (\*) symbol

• Validation error messages should be displayed properly at correct position

• All error messages should be displayed in same CSS style (e.g. using red color)

• General confirmation messages should be displayed using CSS style other than

error messages style (e.g. using green color)

• Dropdown fields should have first entry as blank or text like ‗Select‘

• Delete functionality for any record on page should ask for confirmation

**GUI and Usability Test Scenarios**

• All fields on page (e.g. text box, radio options, dropdown lists) should be aligned properly.

• Scroll bar should be enabled only when necessary

• Description text box should be multi-line

• User should be able to submit the form again by correcting the errors

• Default radio options should be pre-selected on page load

• Check all pages for broken images

**Test Scenarios for a Window**

• Check if default window size is correct

• Check if child window size is correct

• Check if child windows are getting closed on closing parent/opener window

• Check window minimize, maximize and close functionality

• Check if window is re-sizable

**Database Testing Test Scenarios**

* Check if correct data is getting saved in database upon successful page submit
* Check values for columns which are not accepting null values
* Check for data integrity. Data should be stored in single or multiple tables based on design
* For every database add/update operation log should be added
* Required table indexes should be created

**Security Testing Test Scenarios**

• Secure pages should use HTTPS protocol

• Check application logout functionality

• Check for Brute Force Attacks

• Password should not be stored in cookies

• Test for Denial of Service attacks

CHAPTER - 7

**RESULTS AND DISCUSSION**

**7.1 DATASET 1 :**

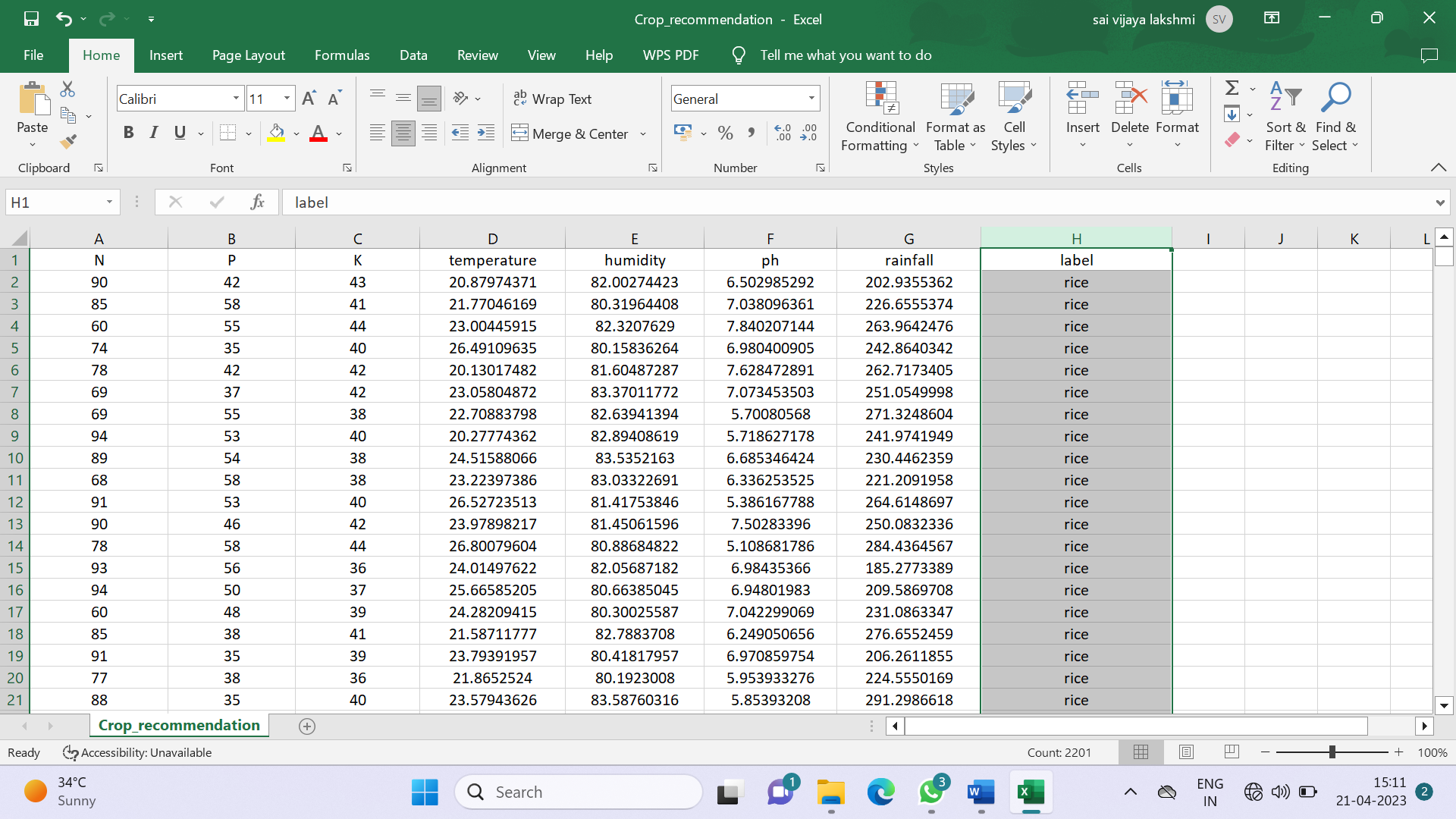
****

FIG : 7.1 Data set

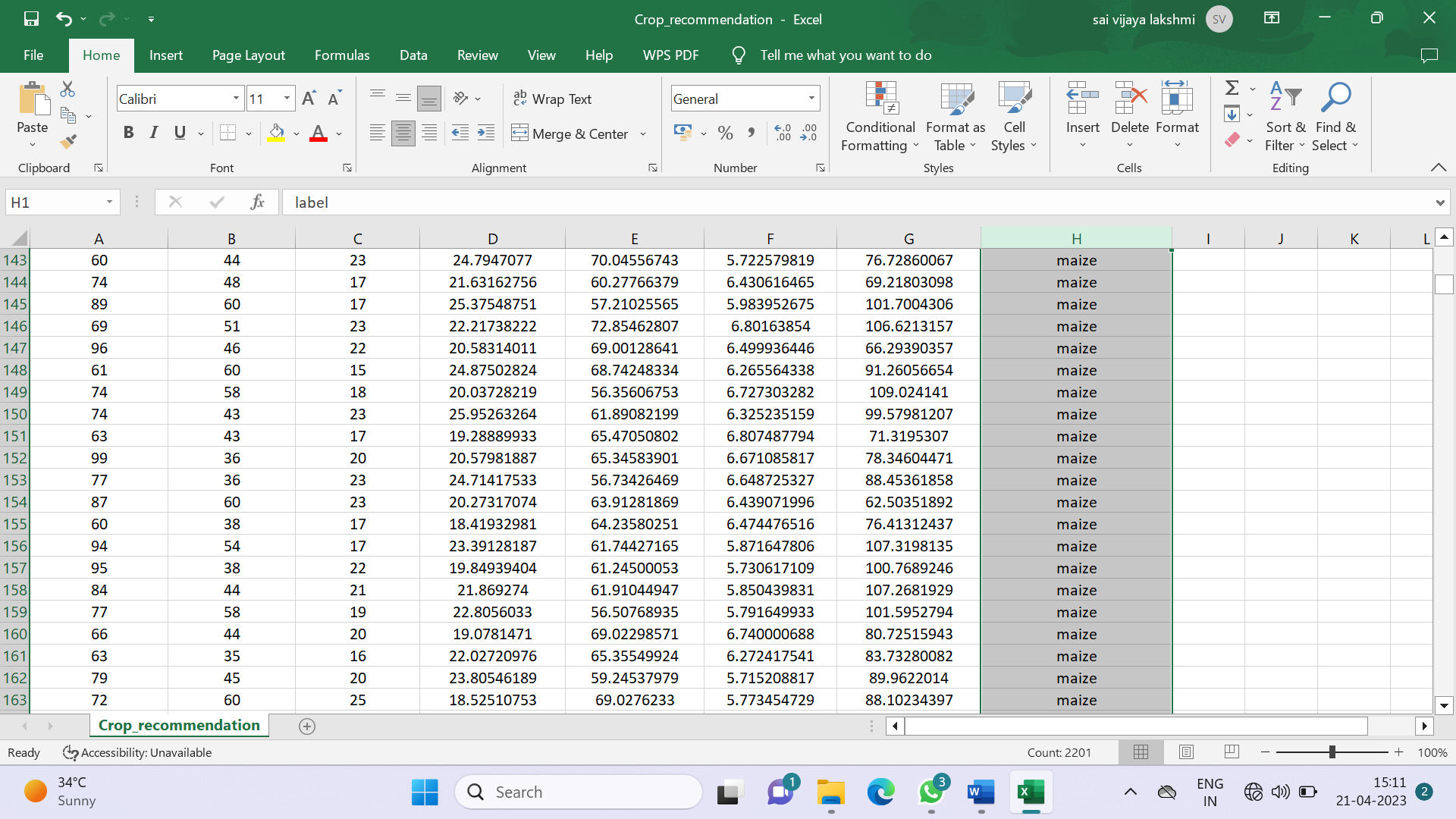
****

Fig : 7.2 Data Set screenshot

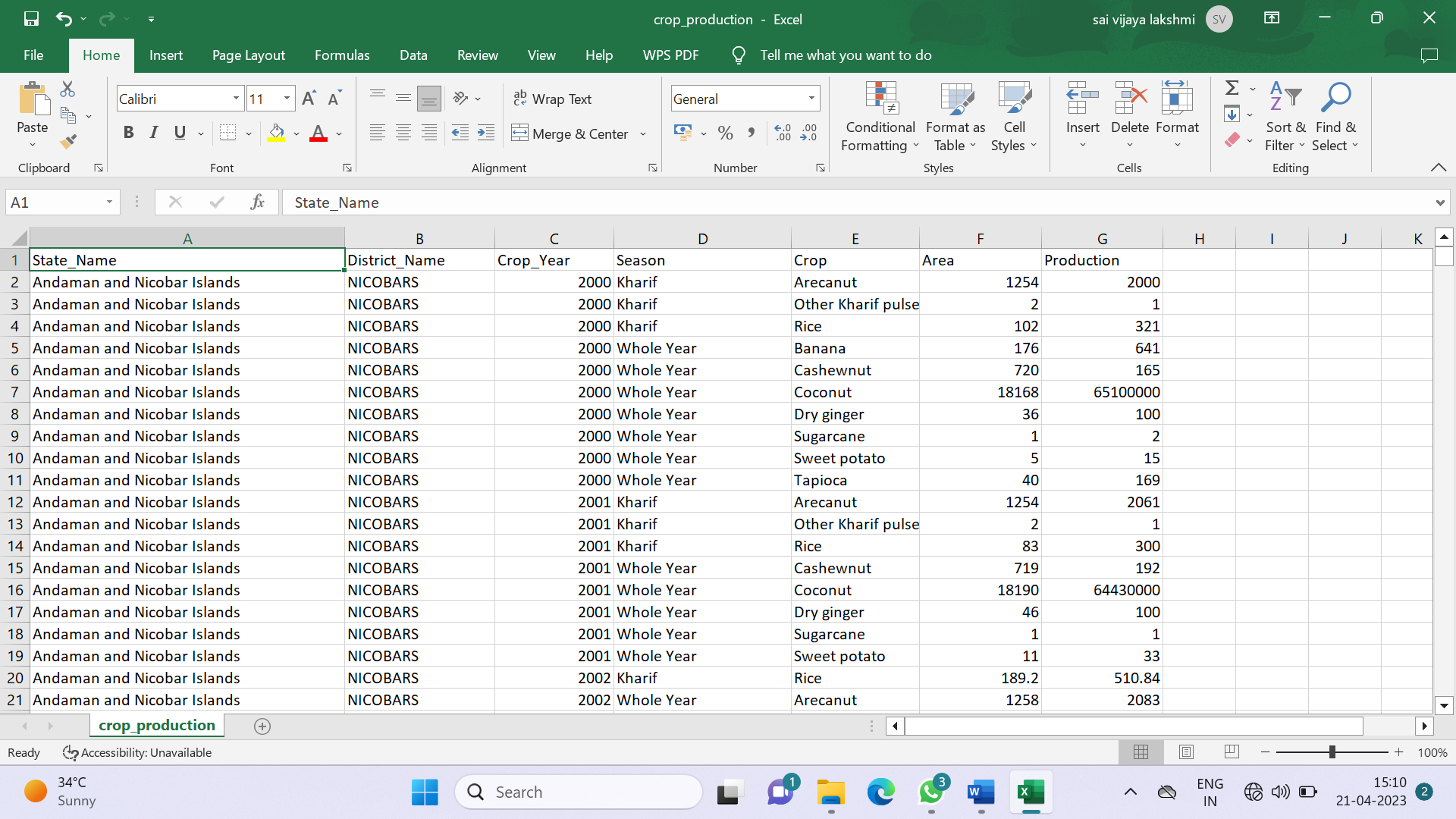


Fig : 7.3 Data Set 2

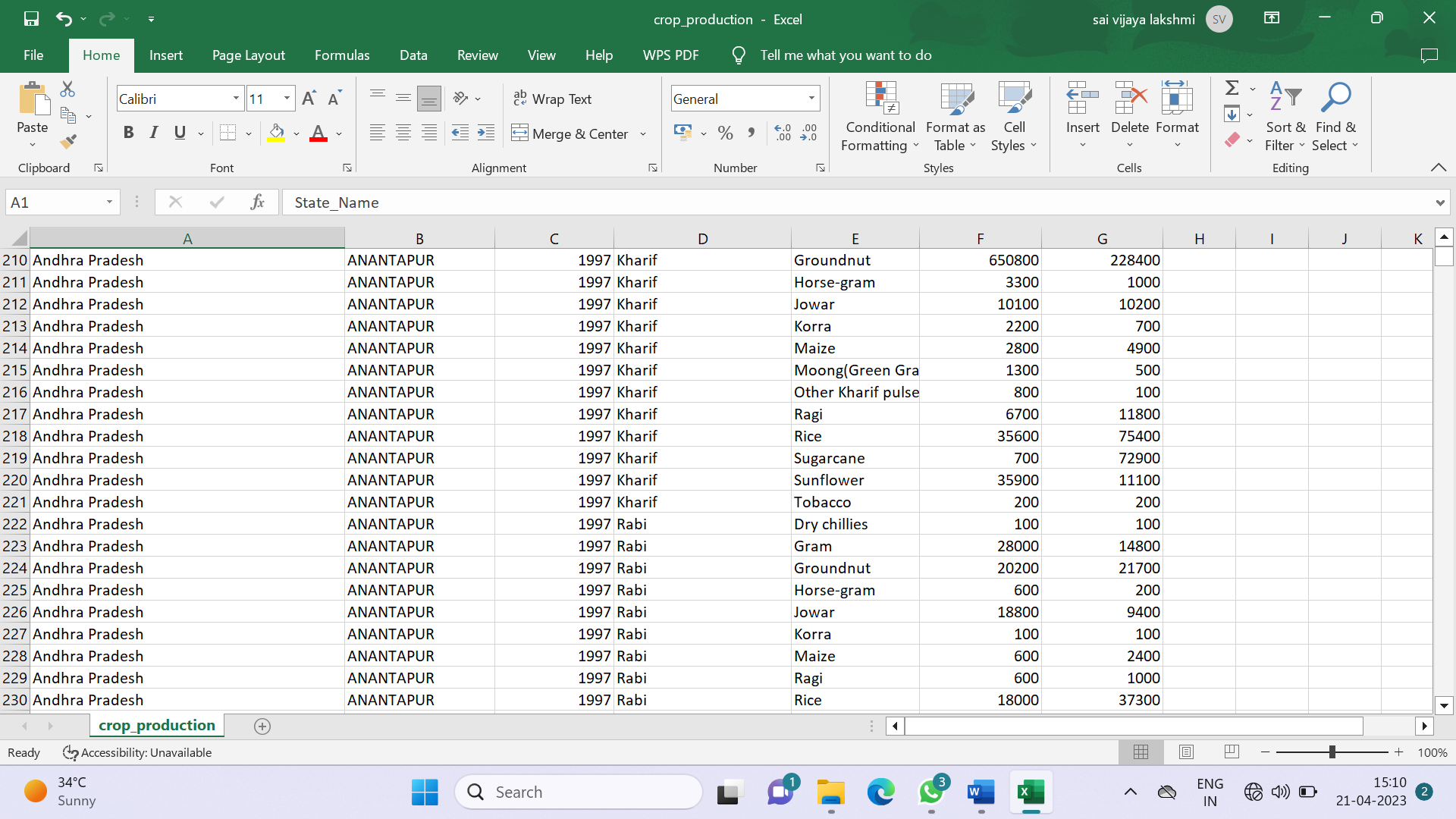


Fig 7.4 : Data Set Screenshot 2

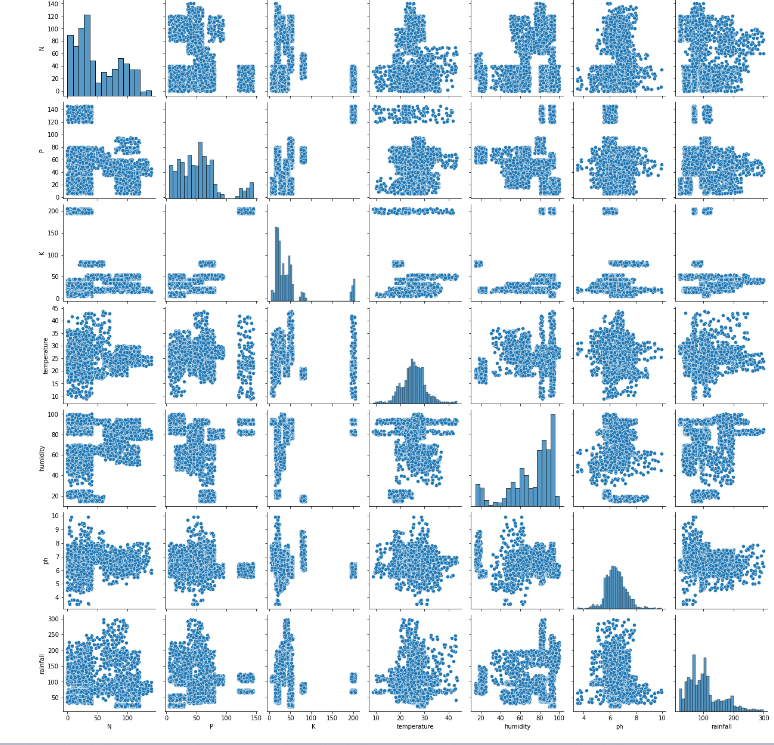


Fig : 7.5 Statistical output

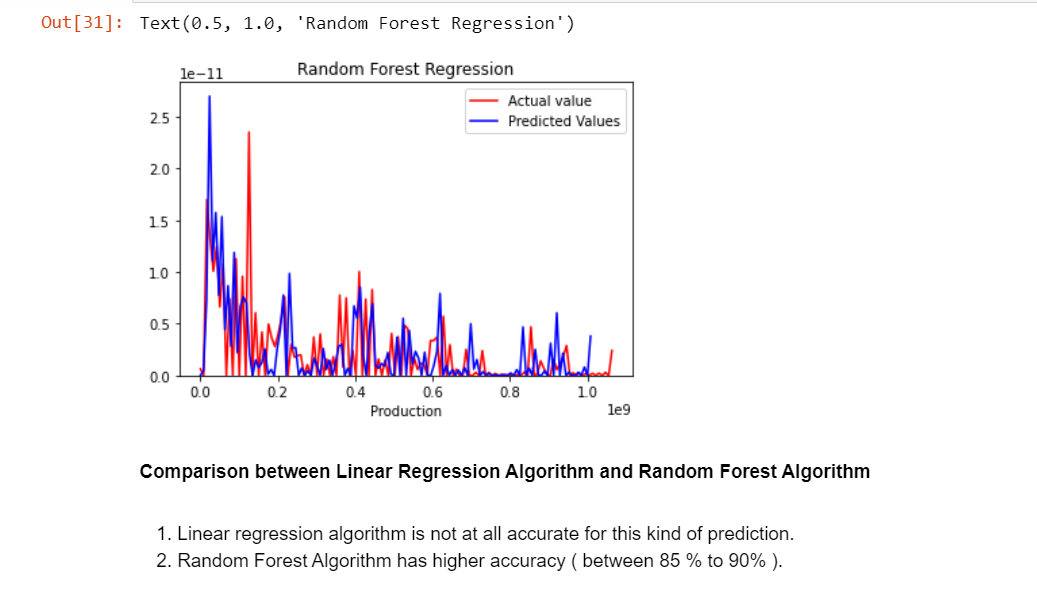


Fig ; 7.6 Random Forest Output

# CHAPTER 8

# CONCLUSION & FRAMEWORK

**CONCLUSION**

This system helps the farmer to choose the right crop by providing insights that ordinary farmers don’t keep track of thereby decreasing the chances of crop failure and increasing productivity. It also prevents them form incurring losses. The system can be extended to the web and can be accessed by millions of farmers across the country. Further development is to integrate the crop prediction system with another subsystem, yield predictor that would also provide the farmer an estimate of production if he plants the recommended crop.

**ADVANTAGES:**

* Select good crops for the region
* Good yield
* Model is of low cost

**CHAPTER 9**

**BIBILOGRAPHY**

### Rakesh Kumar , M.P. Singh, Prabhat Kumar and J.P. Singh, “Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique”, International Conference on Smart Technolo- gies and Management for Computing, Communication, Controls,

### Energy and Materials, 2015

### Haedong Lee and Aekyung Moon, “Development of Yield Predic- tion System Based on Real-time Agricultural Meteorological Infor- mation”, 16th International Conference on Advanced Communica- tion Technology, 2014

### T.R. Lekhaa, “Efficient Crop Yield and Pesticide Prediction for Improving Agricultural Economy using Data Mining Techniques”, International Journal of Modern Trends in Engineering and Science (IJMTES), 2016, Volume 03, Issue 10

### Jay Gholap, Anurag Ingole, Jayesh Gohil, Shailesh Gargade and Vahida Attar, “Soil Data Analysis Using Classification Techniques and Soil Attribute Prediction”, International Journal of Computer Science Issues, Volume 9, Issue 3

### Anshal Savla, Parul Dhawan, Himtanaya Bhadada, Nivedita Is- rani, Alisha Mandholia, Sanya Bhardwaj (2015), ‘Survey of clas- sification algorithms for formulating yield prediction accuracy in precision agriculture’, Innovations in Information, Embedded and Communication systems (ICIIECS).

### AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms. Zeel Doshi , Subhash Nadkarni , Rashi Agrawal, Prof. Neepa Shah.

### Crop Recommendation System for Precision Agriculture S.Pudumalar\*, E.Ramanujam\*, R.Harine Rajashree, C.Kavya, T.Kiruthika, J.Nisha.

### Tom M. Mitchell, Machine Learning, India Edition 2013, McGraw Hill Educa- tion.

### <https://data.gov.in/g>

### Kaggle<https://www.kaggle.com/notebook>