CROP YIELD PREDICTION BASED INDIAN AGRUCULTURE USING PYTHON

A project work Submitted in partial fulfilment of

the requirements for the degree of

Bachelor of Computer Application & Computer Science

to the

Bharathidasan University, Tiruchirapalli

By

R.VINOTHINI

REG.NO.CB20S262979

M.MIRANYA

REG.NO.CB20S262234

R.KAVIYA

REG.NO.CB20S262233

S.SENEGA

REG.NO.CB20S262240

MIT COLLEGE OF ARTS & SCIENCE FOR WOMAN

MUSIRI -621 211

CONTENT

SYNOPSIS

In society the population is increasing at a high rate, people are not aware of the advancement of technologies. Machine learning is used to predict best crop to grow in a particular region based on the parameters like state, district, rainfall and temperature. By analyzing these parameters optimal crop to grow in particular region will be predicted accurately. Initially dataset preprocessing is done which removes irrelevant data and required field has been extracted through feature extraction. Next classification step is implemented which plays important role in prediction. In our proposed method Regression Random Forest is implemented which achieves maximum accuracy. Therefore by selecting particular region as input crop for its respective region will be recommended will be used for improving production. Hence our project will be useful for farmers to predict accurate crop for production. Compared to existing methods our proposed methods achieve better results.

INTROUDUCTION

From ancient days, agriculture is considered as the main source of supply to satisfy the daily needs of human lives. It is also considered a primary occupation, and also one of the India's major industrial sectors. The farmers are ought to follow a traditional naked eye observation and yielded healthy crops without the involvement of chemicals for animals and also to their cultivation land in order to keep healthy diversity. But nowadays, weather conditions are being rapidly changing against the elemental assets to deplete the food and increase the security. In meantime, the GDP in agricultural sector is keep on decreasing, where in 2005 it was about 17.2%, in 2012 it was 11.1, in 2018 it was 5% and in first quarterly year of 2109- 2020 it came down to 2%. Approximately 80 percent of farmers come from rural areas, and if the revenue from crop production goes down, their lifestyle would be influenced by the farms at industry level. This makes sense to farmers in India to show some special concern towards effective and precision farming. In India there are multiple ways to rise the crop learn profit and improve the standard of the crops so as to keep up the economic growth within the field of agriculture. So, the deployment of one of the recent advancement in technology such as, Machine learning is one among the answer for predicting the crop with relation to atmospheric & soil parameter of the agricultural land.

Agriculture is the backbone of every economy. In a country like India, which has ever increasing demand of food due to rising population, advances in agriculture sector are required to meet the needs. From ancient period, agriculture is considered as the main and the foremost culture practiced in India. Ancient people cultivate the crops in their own land and so they have been accommodated to their needs. Therefore, the natural crops are cultivated and have been used by many creatures such as human beings, animals and birds. The greenish goods produced in the land which have been taken by the creature leads to a healthy and welfare life. Since the invention of new innovative technologies and techniques the agriculture field is slowly degrading. Due to these, abundant invention people are been concentrated on cultivating artificial products that is hybrid products where there leads to an unhealthy life. Nowadays,

modern people don't have awareness about the cultivation of the crops in a right time and at a right place. Because of these cultivating techniques the seasonal climatic conditions are also being changed against the fundamental assets like soil, water and air which lead to insecurity of food. By analysing all these issues and problems like weather, temperature and several factors, there is no proper solution and technologies to overcome the situation faced by us. In India there are several ways to increase the economic growth in the field of agriculture. There are multiple ways to increase and improve the crop yield and the quality of the crops. Data mining also useful for predicting the crop yield production. Generally, data mining is the process of analysing data from different perspectives and summarizing it into useful information.

Since, now-a-day's climatic conditions aren't predictable like decades ago. It is changing day by day due to globalization. Hence, the farmers are facing difficulties in forecasting the weather and crops based on climate data. In recent years the advancement of Machine Learning plays a crucial role in every field including agriculture, here the crop prediction process done with consolidating the preceding data and the present data of a particular month to prove the accuracy of climatic data. Machine learning may be a methodology of analysing information to automatize the given model and may be a branch of AI depend on the concept that systems will study from data to form selections with minimal human intervention. There may be a logical classifier, where a naive mathematician who predicts membership opportunities for each group, such as the possibility that knowledge belongs to a specific class. The proposed system analyses the application of supervised machine learning approaches the class with the very best chance is taken into account as the possibly class. Here the category is nothing however the crop that get foretold for the given input parameters.

Data mining software is an analytical tool that allows users to analyse data from many different dimensions or angles, categorize, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. The patterns, associations, or relationships among all this data can provide information. Information can be converted into knowledge about historical patterns and

future trends. For example, summary information about crop production can help the farmers identify the crop losses and prevent it in future. Crop yield prediction is an important agricultural problem. Each and Every farmer is always tries to know, how much yield will get from his expectation. In the past, yield prediction was calculated by analysing farmer's previous experience on a particular crop. The Agricultural yield is primarily depends on weather conditions, pests and planning of harvest operation. Accurate information about history of crop yield is an important thing for making decisions related to agricultural risk management. Therefore, this paper proposes an idea to predict the yield of the crop .The farmer will check the yield of the crop as per the acre, before cultivating onto the field.

Once the crop is foretold, it will facilitate the farmers to predict the affordable crop for their individual land. Then, the farmers is guided with an application in mobile tend to make them to understand that what quite seeds we will tend to sow in land to induce higher yielding. Within the past preceding data, crop prediction was calculated by analyzing farmer's previous expertise on climatic condition. So, the correct data regarding history of climatic condition is a vital factor for creating selections in choosing crops. Therefore, this paper proposes a thought to predict the affordable crop for the given input parameter for the poor farmers using machine learning. Thereby this proposed work will suggest the farmers with effective solutions for more profitable cultivation.

Machine Learning

Machine Learning may be a sub-area of AI, whereby the term refers to the power of IT systems to independently find solutions to problems by recognizing patterns in databases. In other words: Machine Learning enables IT systems to acknowledge patterns in the idea of existing algorithms and datasets and to develop adequate solution concepts. Therefore, in Machine Learning, artificial knowledge is generated on the idea of experience. In orderto enable the software to independently generate solutions, the prior action .

For example, the required algorithms and data must be fed into the systems in advance and the respective analysis rules for the recognition of patterns in the data stock must be defined. Once these two steps have been completed, the system can perform the following tasks by Machine Learning:

- Finding, extracting and summarizing relevant data
- Making predictions based on the analysis data
- Calculating probabilities for specific results

Basically, algorithms play a crucial role in Machine Learning: On the one hand, they're liable for recognizing patterns and on the opposite hand, they will generate solutions. Algorithms can be divided into different categories:

Supervised learning:

In the course of monitored learning, example models are defined beforehand. So as to make sure an adequate allocation of the knowledge to the respective model groups of the algorithms, these then need to be specified. In other words, the system learns on the idea of given input and output pairs. within the course of monitored learning, a programmer, who acts as a sort of teacher, provides the acceptable values for specific input. The aim is to coach the system within the context of successive calculations with different inputs and outputs to determine connections.

Supervised learning is where you've got input variables (X) and an output variable (Y) and you employ an algorithm to find out the mapping function from the input to the output. Y = f(X) The goal is to approximate the mapping function so well that once you have a new input file (X) that you simply can predict the output variables (Y) for that data. It's called supervised learning because the method of an algorithm

learning from the training dataset is oftenthought of as an educator supervising the training process. We all know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected. Learning stops when the algorithm achieves a suitable level of performance. Techniques of Supervised Machine Learning algorithms include linear and logistic regression, multi-class classification, Decision Tree, and SupportVector Machine.

Supervised Learning problems are a kind of machine learning technique often further grouped into Regression and Classification problems. The difference between these two is that the dependent attribute is numerical for regression and categorical for classification.

Regression:

Linear regression could also be a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and thus the only output variable (y). More specifically, that y is usually calculated from a linear combination of the input variables (x).

When there's one input variable (x), the tactic is mentioned as simple linear regression. When there are multiple input variables, literature from statistics often refers to the tactic as multiple linear regression.

Classification:

Classification could also be a process of categorizing a given set of data into classes, It is often performed on both structured or unstructured data. the tactic starts with predicting the category of givendata points. The classes are often mentioned as target, label, or categories.

In short, classification either predicts categorical classalabels orclassification data supported the training set and thus the values(class labels) in classifying attributes and uses it in classifying new data.

There is a variety of classification models. Classification models include Logistic Regression, Decision Tree, Random Forest, Gradient Boosted Tree, One-vs.-One, and Naïve Bayes.

Unsupervised learning:

In unsupervised learning, AI learns without predefined target values and without rewards. It's mainly used for learning segmentation (clustering). The machine tries to structure and type the info entered consistent with certain characteristics. For instance, a machine could (very simply) learn that coins of various colors are often sorted consistent with the characteristic "color" so asto structure them. Unsupervised Machine Learning algorithms are used when the knowledge used to train is neither classified nor labeled. The system doesn't determine the right output but it explores the data and should draw inferences from datasets to elucidate hidden structures from unlabeled data. Unsupervised Learning is that the training of Machines using information that's neither classified nor labeled and allowing the algorithm to act thereon information without guidance.

Unsupervised Learning is accessed into two categories of algorithms:

- Clustering: A clustering problem is where you would like to get the inherent grouping in the data such as grouping customers by purchasing behavior.
- Association: An Association rule learning problem is where you would wish to get rules that describe large portions of your data such as folks that buy X also tend to shop for Y.

Applications of Machine Learning:

Virtual Personal Assistants:

Siri, Alexa, Google Now are a number of the favored samples of virtual personal assistants. As the name suggests, they assist find information, when asked over voice. Machine learning is a crucial apart of these personal assistants as they collect and refine the knowledge on the idea of your previous involvement with them.

Later, this set of knowledge is employed to render results that aretailored to your preferences.

Virtual Assistants are integrated to a spread of platforms. For example:

- Smart Speakers : Amazon Echo and Google Home
- Smartphone : Samsung Bixby on Samsung S8
- Mobile Apps : Google Allo

Videos Surveillance:

- Imagine one person monitoring multiple video cameras! Certainly, adifficult job to try to do and boring also. This is why the thought of coaching computers to try to do this job is sensible.
- The video closed-circuit television nowadays is powered by AI that createsit possible to detect crimes before they happen. They track unusual behavior of individuals like standing motionless for an extended time, stumbling, or napping on benches, etc. The system can thus give an awareness of human attendants, which may ultimately help to avoid mishaps. And when such activities are reported and counted to be true, they assist to enhance the surveillance services. This happens with machine learning doing its job at the backend.

Social Media Services:

From personalizing your news feed to raised ads targeting, socialmedia platforms are utilizing machine learning for his or her ownand user benefits.

- People You May Know

Search Engine Result Refining:

Google and other search engines use machine learning to enhance the search results for you. Every time you execute an inquiry, the algorithms at the backend keep a watch on how you answer the results. If you open the highest results and stay on the online page for long, the program assumes that the results it displayed were in accordance with the query. Similarly, if you reach the second or thirdpage of the search results but don't open any of the results, the program

estimates that the results served did not match the requirement. This way, the algorithms performing at the backend improve the search results.

1.1ORGANIZING PROFILE

NEXT TECHNOLIGIES is proud to introduce ourselves as an end to training solution providing company for skill development of various sectors.

The training programs of NEXT TECHNOLOGIES encompassa wide range of skills that are integral and necessary parts of today's competitive global business atmosphere.

OUR MOTTO

Discover your potential!! Realize your dreams!!

OUR VISION

Our vision is to emphasize high priority in providing training solutions to meet the demand of the ever growing skilled manpower of global standards along with high moral, ethical, social values of our country.

OUR MISSION

Our mission is to accelerate economic development in India, by creating employment opportunities, and making our youth employable, we believe we can impact large scale economic growth.

OUR TEAM

Our trainers are a team of experienced, passionate, and realistic with strong academic background. We formally build a team who can live it to the best of it and for us commercial part is always a secondary note.

OUR EXPERTISE

NEXT TECHNOLOGIES Trainers team served across industries and levels. Our familiarity and confidence in this space has given us the ability to maintain the highest standards of quality to me

1.2SYSTEM REQUIREMENTS

1.2.1HARDWARE REQUIREMENTS (Minimum requirement):

The section of hardware configuration is an important task related to the software development insufficient random access memory may affect adversely on the speed and efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application.

- System : Pentium Dual Core.
- Hard Disk : 120 GB.
- Ram : 1GB.

1.2.2SOFTWARE REQUIREMENTS:

A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system. This document gives a detailed description of the software requirement specification. The study of requirement specification is focused specially on the functioning of the system. It allow the developer or analyst to understand the system, function to be carried out the performance level to be obtained and corresponding interfaces to be established.

- Operating system : Windows 7.
- Coding Language : Python
- Database : MYSQL

1.3 SOFTWARE DESCRIPTION

PYTHON

In technical terms, Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development. It is extremely attractive in the field of Rapid Application Development because it offers dynamic typing and dynamic binding options.

Python is relatively simple, so it's easy to learn since it requires a unique syntax that focuses on readability. Developers can read and translate Python code much easier than other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers.

Additionally, Python supports the use of modules and packages, which means that programs can be designed in a modular style and code can be reused across a variety of projects. Once you've developed a module or package you need, it can be scaled for use in other projects, and it's easy to import or export these modules.

One of the most promising benefits of Python is that both the standard library and the interpreter are available free of charge, in both binary and source form. There is no exclusivity either, as Python and all the necessary tools are available on all major platforms. Therefore, it is an enticing option for developers who don't want to worry about paying high development costs.

If this description of Python over your head, don't worry. You'll understand it soon enough. What you need to take away from this section is that Python is a programming language used to develop software on the web and in app form,

including mobile. It's relatively easy to learn, and the necessary tools are available to all free of charge.

That makes Python accessible to almost anyone. If you have the time to learn, you can create some amazing things with the language.

Readable and Maintainable Code

While writing a software application, you must focus on the quality of its source code to simplify maintenance and updates. The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional code. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.

Multiple Programming Paradigms:

Like other modern programming languages, Python also supports several programming paradigm. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help you to use Python for developing large and complex software applications.

Compatible with Major Platforms and Systems:

At present, Python is supports many operating systems. You can even use Python interpreters to run the code on specific platforms and tools. Also, Python is an

interpreted programming language. It allows you to you to run the same code on multiple platforms without recompilation. Hence, you are not required to recompile the code after making any alteration. You can run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for you to make changes to the code without increasing development time.

Robust Standard Library:

Its large and robust standard library makes Python score over other programming languages. The standard library allows you to choose from a wide range of modules according to your precise needs. Each module further enables you to add functionality to the Python application without writing additional code. For instance, while writing a web application in Python, you can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols. You can even gather information about various modules by browsing through the Python Standard Library documentation.

Many Open Source Frameworks and Tools:

As an open source programming language, Python helps you to curtail software development cost significantly. You can even use several open source Python frameworks, libraries and development tools to curtail development time without increasing development cost. You even have option to choose from a wide range of open source Python frameworks and development tools according to your precise needs. For instance, you can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and Cherrypy. Likewise, you can accelerate desktop GUI application development using

Python GUI frameworks and toolkits like PyQT, PyJs, PyGUI, Kivy, PyGTK and WxPython.

Simplify Complex Software Development:

Python is a general purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. You can take advantage of the data analysis features of Python to create custom big data solutions without putting extra time and effort. At the same time, the data visualization libraries and APIs provided by Python help you to visualize and present data in a more appealing and effective way. Many Python developers even use Python to accomplish artificial intelligence (AI) and natural language processing tasks.

Adopt Test Driven Development:

You can use Python to create prototype of the software application rapidly. Also, you can build the software application directly from the prototype simply by refactoring the Python code. Python even makes it easier for you to perform coding and testing simultaneously by adopting test driven development (TDD) approach. You can easily write the required tests before writing code and use the tests to assess the application code continuously. The tests can also be used for checking if the application meets predefined requirements based on its source code.

However, Python, like other programming languages, has its own shortcomings. It lacks some of the built-in features provided by other modern programming language. Hence, you have to use Python libraries, modules, and frameworks to accelerate custom software development. Also, several studies have shown that Python is

slower than several widely used programming languages including Java and C++. You have to speed up the Python application by making changes to the application code or using custom runtime. But you can always use Python to speed up software development and simplify software maintenance.

Benefits of Learning Python

There are many benefits of learning Python, especially as your first language, which we will discuss.

It is a language that is remarkably easy to learn, and it can be used as a stepping stone into other programming languages and frameworks. If you're an absolute beginner and this is your first time working with any type of coding language, that's something you definitely want.

Python is widely used, including by a number of big companies like Google, Pinterest, Instagram, Disney, Yahoo!, Nokia, IBM, and many others. The Raspberry Pi - which is a mini computer and DIY lover's dream - relies on Python as it's main programming language too. You're probably wondering why either of these things matter, and that's because once you learn Python, you'll never have a shortage of ways to utilize the skill. Not to mention, since a lot of big companies rely on the language, you can make good money as a Python developer.

Other benefits include:

1) Python can be used to develop prototypes, and quickly because it is so easy to work with and read.

2) Most automation, data mining, and big data platforms rely on Python. This is because it is the ideal language to work with for general purpose tasks.

3) Python allows for a more productive coding environment than massive languages like C# and Java. Experienced coders tend to stay more organized and productive when working with Python, as well.

4) Python is easy to read, even if you're not a skilled programmer. Anyone can begin working with the language, all it takes is a bit of patience and a lot of practice. Plus, this makes it an ideal candidate for use among multi-programmer and large development teams.

5) Python powers Django, a complete and open source web application framework. Frameworks - like Ruby on Rails - can be used to simplify the development process.

6) It has a massive support base thanks to the fact that it is open source and community developed. Millions of like-minded developers work with the language on a daily basis and continue to improve core functionality. The latest version of Python continues to receive enhancements and updates as time progresses. This is a great way to network with other developers.

Python Environment Setup

One of the most important things you'll do when working with any programming language is setup a development environment which allows you to execute the code you write. Without this, you will never be able to check your work and see if your website or application is free of syntax errors.

With Python, you also need something called an interpreter that converts your code - which makes up the entirety of your application - to something the computer can read and execute. Without this interpreter, you'll have no way to run your code.

To convert your code, you must first use a Python shell, which calls upon the interpreter through something called a "bang" line.

As for creating an application or file, there are two ways to do this. You can create a program using a simple text editor like WordPad, or Notepad++. You can also create a program using a Python shell. There are advantages and disadvantages to each method, which we'll discuss next.

Python Shell versus Text File

A shell is a program or tool that can be used to interact with a system. For instance, the Windows operating system shell can be tapped into by using a "terminal" or command line to submit commands and arguments.

With Python, things work a bit differently than an operating system shell. The Python shell is used to interact with an interpreter, which feeds code to a computer in a form that it can understand.

When you execute a Python program that you've written, the interpreter reads the code and converts it into usable commands. The important thing to note is that all of this is done after the program has been executed.

With a shell, the interpreting - or conversion - happens in real-time as you type the code into the computer or system. This means that the actual program is executing as you type. This gives you some idea of how your final code will look, and what your program is actually going to do.

When you write code in a text file, none of that happens until you feed the document into an interpreter. If you have Python installed on your computer you can call upon the interpreter using a command line, but this step is done after you've already written the code.

This makes it more difficult to spot errors in your code, and it can also be frustrating if the interpreter runs into issues, because they may not be as apparent as they would if you had used a shell. Still, a lot of developers prefer to use a text editing tool because it is simple and easy to do.

Python Features:

Python is often comparable to Perl, Ruby, PHP, Scheme, and Java. This is because it is is an incredibly powerful object-oriented language.

Python also has several notable features which make it an enticing language to work with for developers.

1) Python makes use of an elegant syntax, meaning the programs you write are much easier to read. This is because they are closer to the human language, or how we write our words, instead of a language that computers use to read and interpret code. For example, the "print" command will display anything proceeding it - and in quotes - at runtime.

2) Python is simple and easy-to-use, which means that it's much easier to get your programs up and running. That is why Python is considered ideal for prototype development and similar ad-hoc programming tasks. It does not compromise maintainability either.

3) It comes with the Standard Python Library, offering integrated support for a variety of common programming tasks like syncing with web servers, searching through text, and modifying files. For a majority of other languages, you have to create this content from scratch.

4) It includes an interactive mode that simplifies testing for short snippets of code. There's even a development environment bundled with it called IDLE. The dev environment makes setup so much easier and faster

5) The language can be extended by adding new modules, even if they've been compiled in C or C++. Even better, the modules can be used as shortcuts in future projects once they've been created.

6) Python can be embedded into an application, which will provide a programmable interface for users of that app. This is a great feature if you're putting together an app that will teach coding, or requires working with Python in a terminal.

7) It is compatible with a long list of computers and operating systems like Windows, Linux, MacOS, many brands of Unix, OS/2, and more. Furthermore, it uses a similar interface on each one of those platforms, which means you can jump between them easily if necessary.

8) It is truly free because it doesn't cost anything to download or use, and there are no licensing fees. Plus, it can be freely modified and redistributed, since the language is available under an open source license - despite the fact that it is copyrighted.

Differ from Other Languages Like PHP or Ruby

Python stands out because it is easy to learn and easy to understand. Many consider Ruby a great place to start, like Python, yet the latter has a four-year head start. This means that it has a big foothold in the enterprise world, and it's much more popular with C developers. This is because it's easy to crossover between the two languages.

Both Ruby and Python share a significant amount of growth in the job market, so choosing either language would be beneficial in terms of a career. PHP is also used often though the application is different.

Ultimately, it comes down to what you will be developing, as each language has its niche.

Languages Used:

PHP is a server scripting language that is primarily used to create dynamic and interactive websites. It is the best language for creating HTML content, and can be used to build anything from a simple blog to a huge, corporate style website.

Python is a high-level, object-oriented general-purpose language, that is versatile and can be used for nearly anything. It is commonly used to develop web and mobile applications, website crawlers, indexers, daemons, and desktop GUI apps.

Ruby is a high-level, object-oriented language that is used to work with web application and data entities; it exists to take the focus away from query tasks. Ruby is most famous for its dynamic type system which performs type checking during runtime. It also features automatic memory management.

TIP: When a language is referred to as "high-level" it is because the syntax and commands it recognizes are closer to human language instead of that of a computer. The term high-level was initially used to describe languages that are not locked down to a particular type of computer.

Out of the three languages, Python is the best for absolute beginners and is often recommended by programmers because it uses a syntax that emphasizes simplicity and ease of use. Whereas, Ruby is better used by programmers that have experience with other languages. PHP, on the other hand, is best suited for developers who are used to working with C languages.

2.SYSTEM STUDY

FEASIBILITY STUDY:

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

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

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

TECHNICAL FEASIBILITY

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

SOCIAL FEASIBILITY

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

2.1 EXISTING SYSTEM

Agriculture is a major contributor to the Indian economy. The mainstream Indian population depends either explicitly or implicitly on agriculture for their livelihood. It is, thus, irrefutable that agriculture plays a vital role in the country. A vast majority of the Indian farmers believe in depending on their intuition to decide which crop to sow in a particular season. They find comfort in simply following the ancestral farming patterns and norms without realizing the fact that crop output is circumstantial, depending heavily on the present-day weather and soil conditions. However, a single farmer cannot be expected to take into account all the innumerable factors that contribute to crop growth before reaching a consensus about which one to grow. A single misguided or imprudent decision by the farmer can have undesirable ramifications on both himself as well as the agricultural economy of the region. A combination of Big Data Analytics and Machine Learning can effectively help alleviate this issue. In existing method, an intelligent system, called Agro Consultant, which intends to assist the Indian farmers in making an informed decision about which crop to grow depending on the sowing season, his farm's geographical location, soil characteristics as well as environmental factors.

2.1.1DRAW BACK:

- Prediction of crop with respect to soil has been developed in addition geographical location also included. Here based on soil crop will be recommended incase environment condition is not proper then yield will be affected.
- Prediction is done based on soil and location in case proper fertilizer is not used then it will affect crop yield prediction.
- Accurate prediction of crop with respect to soil with fertilizer is not possible.

2.2 PROPOSED WORK:

As recommendation of crop and soil is important for farmers in farming decision making, the paper proposes the use of data mining techniques to provide recommendations. Recommendations of suitable crop in the field and soil to farmers are provided with the help of data stored as datasets. Proposed system provides crop recommendation based on state, district rainfall and temperature available. Thus aim of this system is to increase the production of crops by recommending perfect crop and soil according to the region. The performance evaluation shown, that the accuracy of developed system is reasonably high. In our proposed system dataset has been loaded and pre-processed to avoid irrelevant and noisy data and undergone for further process. Feature extraction is used for extracting the mandatory parameter from the dataset which increases the accuracy in prediction. Through Regression Random Forest algorithm accurate classification based on region has been implemented. The result extracts recommended crop based on region with respect to specific temperature and rainfall.

2.2.1FEATURE:

- Increases accuracy in prediction of crop based on region with temperature and rainfall.
- Time consumption is low and fast recovery of data.
- Increase yield for farmers.

3. SYSTEM DEVELOPMENT

3.1 INPUT DESIGN:

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

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be

free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

3.2OUTPUT DESIGN:

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

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

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system. The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the

- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

3.3 Database Design:

The Database Management System (DBMS) consists of a collection of interrelated data and a set of programs to access that data. The Collection of data usually referred to as database. The primary key goal of DBMS is to provide an environment that is both convenient and efficient to use in retrieving and storing data information.

3.4 Code design:

A code design is a document that sets rules for the design of a new development. It is a tool that can be used in the design and planning process, but goes further and is more regulatory than other forms of guidance commonly used in the English planning system over recent decades. It can be thought of as a process and document – and therefore a mechanism – which operationalises design guidelines or standards which have been established through a master plan process. The master plan or framework is the vision. It should be accompanied by a design rationale that explains why, followed by a code that gives instructions to the appropriate degree or precision and that is operational.

In this way a design code may be a tool which helps ensure that the aspirations for quality and quantity for housing developments, particularly for large-scale projects, sought by the Government and other agencies are actually realised in the final schemes.

3.5 SYSTEM DEVELOPMENT

3.5.1 INTRODUCTION

Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications

Predicting crop yield based on the environmental, soil, water and crop parameters has been a potential research topic. Deep-learning-based models are broadly used to extract significant crop features for prediction. Though these methods could resolve the yield prediction problem there exist the following inadequacies: Unable to create a direct non-linear or linear mapping between the raw data and crop yield values; and the performance of those models highly relies on the quality of the extracted features. Deep reinforcement learning provides direction and motivation for the aforementioned shortcomings. Combining the intelligence of reinforcement learning and deep learning, deep reinforcement learning builds a complete crop yield prediction framework that can map the raw data to the crop prediction values. The proposed work constructs a Deep Recurrent Q-Network model which is a Recurrent Neural Network deep learning algorithm over the Q-Learning reinforcement learning algorithm to forecast the crop yield. The sequentially stacked layers of Recurrent Neural network is fed by the data parameters. The Q- learning network constructs a crop yield prediction environment based on the input parameters. A linear layer maps the Recurrent Neural Network output values to the Q-values. The reinforcement learning agent incorporates a combination of parametric features with the threshold that assist in predicting crop yield. Finally, the agent receives an aggregate score for the actions performed by minimizing the error and maximizing the forecast accuracy.

Reinforced random forest

Reinforcement learning improves classification accuracy. But use of reinforcement learning is relatively unexplored in case of random forest classifier. We propose a reinforced random forest (RRF) classifier that exploits reinforcement learning to improve classification accuracy. Our algorithm is initialized with a forest. Then the entire training data is tested using the initial forest. In order to reinforce learning, we use mis-classified data points to grow certain number of new trees. A subset of the new trees is added to the existing forest using a novel graph-based approach. We show that addition of these trees ensures improvement in classification accuracy. This process is continued iteratively until classification accuracy saturates. The proposed RRF has low computational burden. We achieve at least 3% improvement in F-measure compared to random forest in three breast cancer datasets. Results on benchmark datasets show significant reduction in average classification error.

Random Forests for Global and Regional Crop Yield Predictions

Accurate predictions of crop yield are critical for developing effective agricultural and food policies at the regional and global scales. We evaluated a machine-learning method, Random Forests (RF), for its ability to predict crop yield responses to climate and biophysical variables at global and regional scales in wheat, maize, and potato in comparison with multiple linear regressions (MLR) serving as a benchmark. We used crop yield data from various sources and regions for model training and testing: 1) gridded global wheat grain yield, 2) maize grain yield from US counties over thirty years, and 3) potato tuber and maize silage yield from the northeastern seaboard region. RF was found highly capable of predicting crop yields and outperformed MLR benchmarks in all performance statistics that were compared. For example, the root mean square errors (RMSE) ranged between 6 and

14% of the average observed yield with RF models in all test cases whereas these values ranged from 14% to 49% for MLR models. Our results show that RF is an effective and versatile machine-learning method for crop yield predictions at regional and global scales for its high accuracy and precision, ease of use, and utility in data analysis. RF may result in a loss of accuracy when predicting the extreme ends or responses beyond the boundaries of the training data.

Crop Price prediction using Random Forest and Decision Tree Regression

Machine learning with the Prediction model has gained its popularity through its promising results. Its application has been incorporated in this paper too where various regression model has been studied to predict the Crop prices. The crop price prediction assist the farmers to plan their next crop to be grown and avoid hyperinflation. The dataset has 330 different crops altogether. Different models have been investigated for their performance and compared. The Results shows that the Random Forest Regression and Decision Tree Regressor hasthe best prediction model among all with an accuracy of around 99%.

Evaluation of random forest method for agricultural crop classification

This study aims to examine the performance of Random Forest (RF) and Maximum Likelihood Classification (MLC) method to crop classification through pixel-based and parcel-based approaches. Analyses are performed on multispectral SPOT 5 image. First, the SPOT 5 image is classified using the classification methods in pixel-based manner. Next, the produced thematic maps are overlaid with the original agricultural parcels and the frequencies of the pixels within the parcels are computed. Then, the majority of the pixels are assigned as class label to the parcels. Results indicate that the overall accuracies of the parcel-based approach computed

for the Random Forest method is 85.89%, which is about 8% better than the corresponding result of MLC.

Random Forests for Global and Regional Crop Yield Predictions

Accurate predictions of crop yield are critical for developing effective agricultural and food policies at the regional and global scales. We evaluated a machine-learning method, Random Forests (RF), for its ability to predict crop yield responses to climate and biophysical variables at global and regional scales in wheat, maize, and potato in comparison with multiple linear regressions (MLR) serving as a benchmark. We used crop yield data from various sources and regions for model training and testing: 1) gridded global wheat grain yield, 2) maize grain yield from US counties over thirty years, and 3) potato tuber and maize silage yield from the northeastern seaboard region. RF was found highly capable of predicting crop yields and outperformed MLR benchmarks in all performance statistics that were compared. For example, the root mean square errors (RMSE) ranged between 6 and 14% of the average observed yield with RF models in all test cases whereas these values ranged from 14% to 49% for MLR models. Our results show that RF is an effective and versatile machine-learning method for crop yield predictions at regional and global scales for its high accuracy and precision, ease of use, and utility in data analysis. RF may result in a loss of accuracy when predicting the extreme ends or responses beyond the boundaries of the training data.

Evaluation of random forest method for agricultural crop classification

This study aims to examine the performance of Random Forest (RF) and Maximum Likelihood Classification (MLC) method to crop classification through pixel-based and parcel-based approaches. Analyses are performed on multispectral SPOT 5 image. First, the SPOT 5 image is classified using the classification methods

in pixel-based manner. Next, the produced thematic maps are overlaid with the original agricultural parcels and the frequencies of the pixels within the parcels are computed. Then, the majority of the pixels are assigned as class label to the parcels. Results indicate that the overall accuracies of the parcel-based approach computed for the Random Forest method is 85.89%, which is about 8% better than the corresponding result of MLC.

Reinforcement Learning Trees

The innovations are three-fold. First, the new method implements reinforcement learning at each selection of a splitting variable during the tree construction processes. By splitting on the variable that brings the greatest future improvement in later splits, rather than choosing the one with largest marginal effect from the immediate split, the constructed tree utilizes the available samples in a more efficient way. Moreover, such an approach enables linear combination cuts at little extra computational cost. Second, we propose a variable muting procedure that progressively eliminates noise variables during the construction of each individual tree. The muting procedure also takes advantage of reinforcement learning and prevents noise variables from being considered in the search for splitting rules, so that towards terminal nodes, where the sample size is small, the splitting rules are still constructed from only strong variables. Last, we investigate asymptotic properties of the proposed method under basic assumptions and discuss rationale in general settings.

Using reinforcement learning to find an optimal set of features

Identifying the most characterizing features of observed data is critical for minimizing the classification error. Feature selection is the process of identifying a small subset of highly predictive features out of a large set of candidate features. In

the literature, many feature selection methods approach the task as a search problem, where each state in the search space is a possible feature subset. In this study, we consider feature selection problem as a reinforcement learning problem in general and use a well-known method, temporal difference, to traverse the state space and select the best subset of features. Specifically, first, we consider the state space as a Markov decision process, and then we introduce an optimal graph search to overcome the complexity of the problem of concern. Since this approach needs a state evaluation paradigm as an aid to traverse the promising regions in the state space, the presence of a low-cost evaluation function is necessary. This method initially explores the lattice of feature sets, and then exploits the obtained experiments. Finally, two methods, based on filters and wrappers, are proposed for the ultimate selection of features.

Fuzzy Forests: Extending Random Forest Feature Selection for Correlated, High-Dimensional Data

Fuzzy forests is specifically designed to provide relatively unbiased rankings of variable importance in the presence of highly correlated features, especially when the number of features, p, is much larger than the sample size, n (p n). We introduce our implementation of fuzzy forests in the R package, fuzzyforest. Fuzzy forests works by taking advantage of the network structure between features. First, the features are partitioned into separate modules such that the correlation within modules is high and the correlation between modules is low. The package fuzzyforest allows for easy use of the package WGCNA (weighted gene coexpression network analysis, alternatively known as weighted correlation network analysis) to form modules of features such that the modules are roughly uncorrelated. Then recursive feature elimination random forests (RFE-RFs) are used on each module, separately. From the surviving features, a final group is selected

and ranked using one last round of RFE-RFs. This procedure results in a ranked variable importance list whose size is pre-specified by the user. The selected features can then be used to construct a predictive model.

3.6 DESCRIPTION OF MODULES

MODULES:

1. Data collection
2. Data preprocessing
3. Feature extraction
4. Classifier

MODULE DESCRIPTION:

Data collection:

As a part of this machine learning problem, the first step is to gather the data and perform the feature engineering. For attaining a better accuracy in the crop prediction, it is required to tune the parameter of the model. Data is composed from a different source and optimized for data sets. And the data is used to evaluate descriptively. Several abstract online outlets, like Kaggle, Google weather forestation and data government, provide the data for up to 10years in series. The data sets such as soil nature, climatic conditions and region data are used for the crop prediction and better crop yields. In our proposed work it will focus on the optimizing the Reinforcement Random Forest. In any Machine learning model, the model parameters such as state, district, rainfall and temperature which has to be set before training the model.

Pre-processing:

Pre-processing the data is considered as a significant step machine learning phase. Pre-processing involves adding the missing values, the correct set of data, and extracting the functionality. Data set form is important to the process of analysis.

The data collected in this step will induced in platform in the form of python programming in order to get the desired output.

The statistical data along with the agricultural production data and weather data are the two sets of data collected for this work. It provides predictive value for crop yield in the particular area. The area under analysis has an average climate. The agricultural production data contains planting area, irrigation area, and fertilizer usage and irrigation details. The weather dataset comprises climatologically features including rainfall, and temperature. Each instance of the dataset contains the details about the crop with the cultivated area, annual production and weather features monitored during the year. The data set is pre-processed for identification and understanding of features, missing values treatment and outlier treatment. The quality of data decided the quality of the output. The data was cleaned using missing value treatment.

Feature selection:

Extraction of the features would reduce the data size involved to characterize a wide collection of data. The characteristics of soil, crop and weather collected from the pre-treatment process establish the final training data collection. This approach selects the features based on the correlation matrix i.e. the features that has more correlation value is selected as an important predictive function for crop.

Selections of high level features that contribute for prediction accuracy play a major role in obtaining the accurate prediction. By applying different feature selection algorithms such as sequential forward feature selection, correlation based feature selection, variance inflation factor and random forest variable importance, different feature subsets were selected. From a large datasets more number of parameters is taken into consideration in order to attain detail in perfect manner.

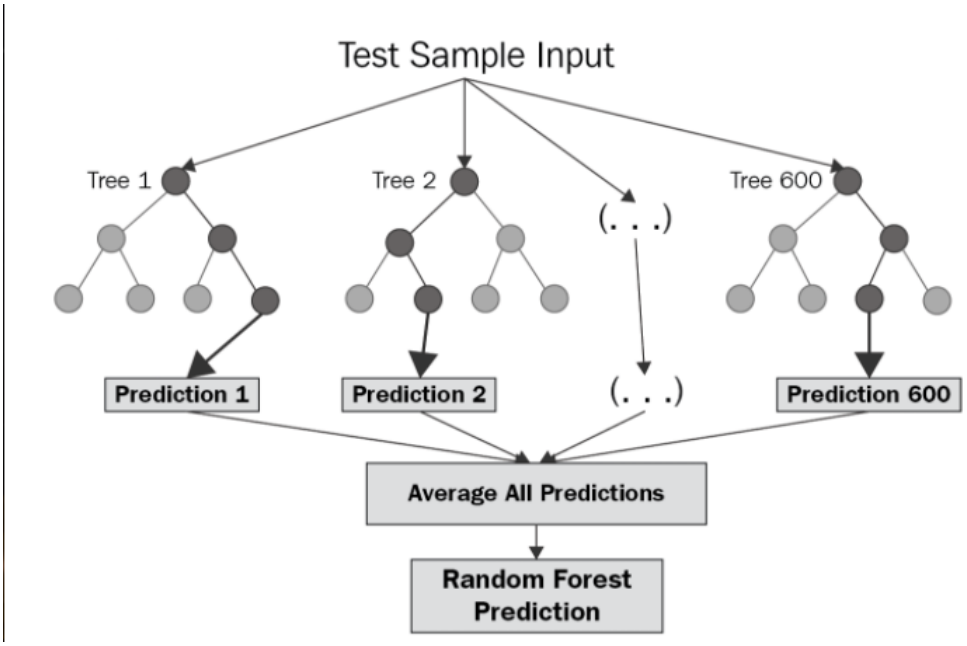
However needed parameter alone extracted from database and loaded for processing hence this process could be done through feature selection module. The parameter here selected must be in mandatory one to analyse then only efficient suggestions will be provided to user.

Classifier:

In advance to this step there need to split the data into train dataset and test dataset. By applying the Regression random forest classifier the data is trained with available input and output data. In the test phase, the data are tested if the accuracy of e model is satisfied. Then the new data is predicted by machine learning module.

Regression Random Forest :

Regression Random Forest is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.Random forest is an ensemble of decision trees. This is to say that many trees, constructed in a certain “random” way form a Random Forest. Each tree is created from a different sample of rows and at each node, a different sample of features is selected for splitting. Each of the trees makes its own individual prediction. These predictions are then averaged to produce a single result. The averaging makes a Random Forest better than a single Decision Tree hence improves its accuracy and reduces overfitting. A prediction from the Random Forest Regressor is an average of the predictions produced by the trees in the forest.



The diagram above shows the structure of a Random Forest. You can notice that the trees run in parallel with no interaction amongst them. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees. To get a better understanding of the Random Forest algorithm, let's walk through the steps:

- Pick at random k data points from the training set.
- Build a decision tree associated to these k data points.
- Choose the number N of trees you want to build and repeat steps 1 and
- For a new data point, make each one of your N-tree trees predict the value of y for the data point in question and assign the new data point to the average across all of the predicted y values.

A Random Forest Regression model is powerful and accurate. It usually performs great on many problems, including features with non-linear relationships. Disadvantages, however, include the following: there is no interpretability,

overfitting may easily occur, we must choose the number of trees to include in the model.

4. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate

that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

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

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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

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

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An

example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which 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.

Black Box Testing

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

Unit Testing:

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

Test strategy and approach

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

Test objectives

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

Features to be tested

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

Integration Testing

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

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

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

Acceptance Testing

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

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

5.CONCLUSION

The Results shows that we can attain an accurate crop prediction using the Reinforcement Random Forest algorithm. Reinforcement Random Forest algorithm achieves a largest number of crop models with a lowest models. It is suitable for massive crop yield prediction in agricultural planning. This makes the farmers to take the right decision for right crop such that the agricultural sector will be developed by innovative ideas. By inputting the parameters like state, district, rainfall and temperature one can accurately predict optimal crop to grow in that particular region to current situation. Compared to SVM classifier our proposed method achieved accurate prediction and attains maximum accuracy and it improves performance.

6.REFERENCES

1. Abolpour B, Javan M, Karamouz M (2007) Water allocation improvement in river basin using Adaptive Neural Fuzzy Reinforcement Learning approach. Appl Soft Comput 7:265–285.
2. Al-Dayaa HS, Megherbi DB (2012) Reinforcement learning technique using agent state occurrence frequency with analysis of knowledge sharing on the agent's learning process in multiagent environments. J Supercomput 59:526–547.
3. Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Machine Intell 39:2481–2495.
4. Bergez JE, Eigenraam M, Garcia F (2001) Comparison between dynamic programming and reinforcement learning: a case study on maize irrigation management. In Proceedings of the 3rd European Conference on Information Technology in Agriculture (EFITA01) 343–348.
5. Bone C, Dragićević S (2010) Simulation and validation of a reinforcement learning agent-based model for multi-stakeholder forest management. Comput Environ Urban Syst 34:162–174.
6. Bonneau M, Gaba S, Peyrard N, Sabbadin R (2014) Reinforcement learning-based design of sampling policies under cost constraints in Markov random felds: application to weed map reconstruction. Comput Stat Data Anal 72:30.
7. Chen W, Xie X, Wang J, Pradhan B, Hong H, Bui DT, Duan Z, Ma J (2017) A comparative study of logistic model tree, random forest, and classifcation and regression tree models for spatial prediction of landslide susceptibility. CATENA 151:147–160.

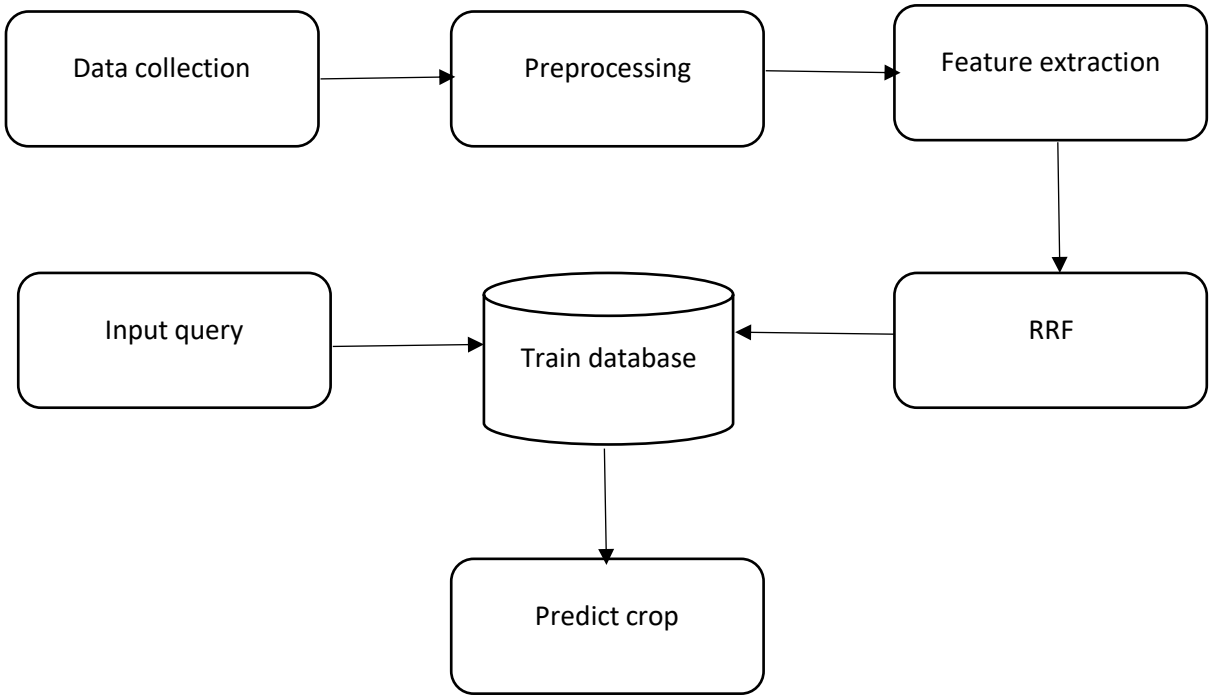
8. Chen H, Huang JJ, McBean E (2020) Partitioning of daily evapotran- spiration using a modifed shuttleworth-wallace model, random Forest and support vector regression, for a cabbage farmland. Agric Water Manag 228:105923.
9. Dantas TM, Oliveira FLC (2018) Improving time series forecasting: an approach combining bootstrap aggregation, clusters and exponential smoothing. Int J Forecast 34:748–761.
10. Elavarasan D, Vincent PMD (2020a) Crop yield prediction using deep reinforcement learning model for sustainable agrarian applications. IEEE Access 8:86886–86901.
11. Elavarasan D, Vincent DR (2020b) Reinforced XGBoost machine learning model for sustainable intelligent Agrarian Applications. J Intell Fuzzy Syst.
12. Elavarasan D, Vincent DR, Sharma V, Zomaya AY, Srinivasan K (2018) Forecasting yield by integrating agrarian factors and machine learning models: a survey. Comput Electron Agric 155:257–282.
13. Fanyu Bu, Wang X (2019) A smart agriculture IoT system based on deep reinforcement learning. Future Generation Comput Syst 99:500–507.
14. Fukuda S, Spreer W, Yasunaga E, Yuge K, Sardsud V, Müller J (2013) Random Forests modelling for the estimation of mango (Mangifera indica L. cv. Chok Anan) fruit yields under diferent irrigation regimes. Agric Water Manag 116:142–150.
15. Govindan R, Al-Ansari T (2019) Simulation-based reinforcement learning for delivery feet optimisation in CO2 fertilisation networks to enhance food production systems. Comput Aided Chem Eng 46:1507–1512.
16. Hapfelmeier A, Ulm K (2014) Variable selection by Random For- ests using data with missing values. Comput Stat Data Anal 80:129–139.
17. Hong Y, Shen R, Cheng H, Chen Y, Zhang Y, Liu Y, Zhou M, Lei Yu, Liu Yi, Liu Y (2019) Estimating lead and zinc concentrations in peri-urban

agricultural soils through refectance spectroscopy: efects of fractional-order derivative and random forest. Sci Total Environ 651:1969–1982.

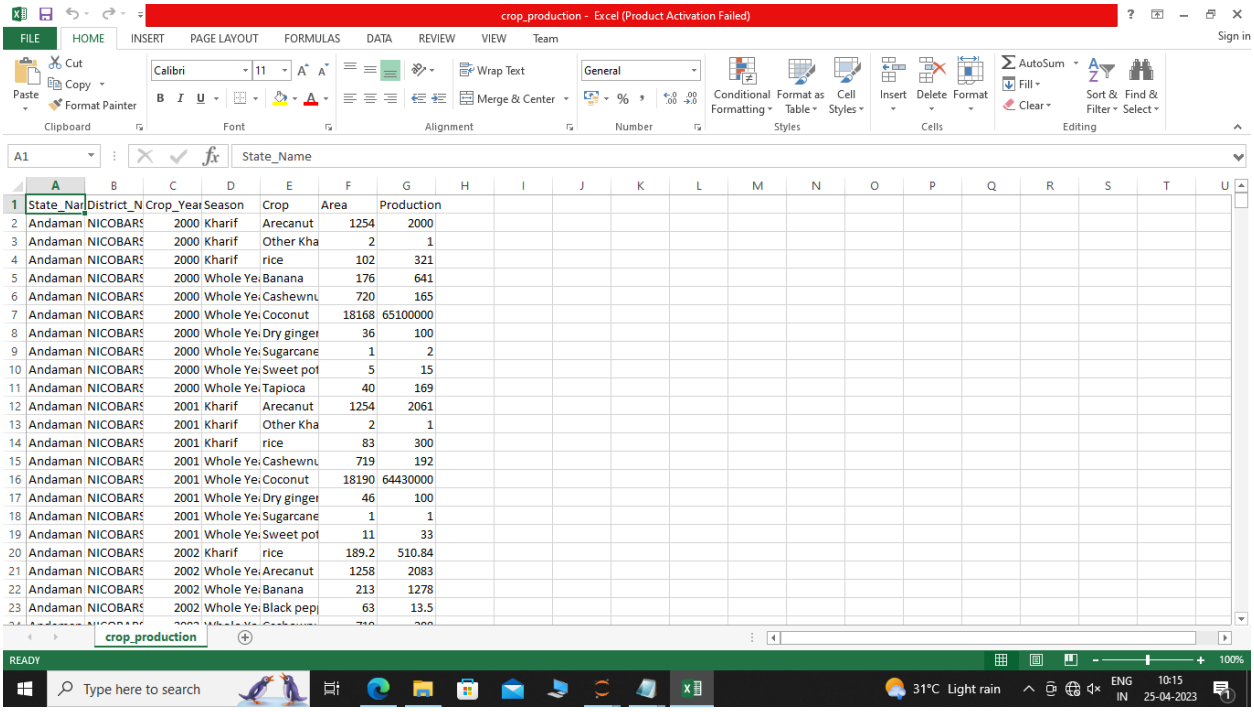
18. Huang T, Yang R, Huang W, Huang Y, Qiao Xi (2018) Detecting sugarcane borer diseases using support vector machine. InformProc Agric 5:74–82.
19. Iqbal F, Lucieer A, Barry K (2018) Poppy crop capsule volume esti- mation using UAS remote sensing and random forest regression. Int J Appl Earth Obs Geoinf.
20. Jiang Y, Hao K, Cai X, Ding Y (2018) An improved reinforcement immune algorithm for agricultural resource allocation optimization. J Comput Sci 27:320–328

7.APPENDICES

A.DATA FLOW DIAGRAM

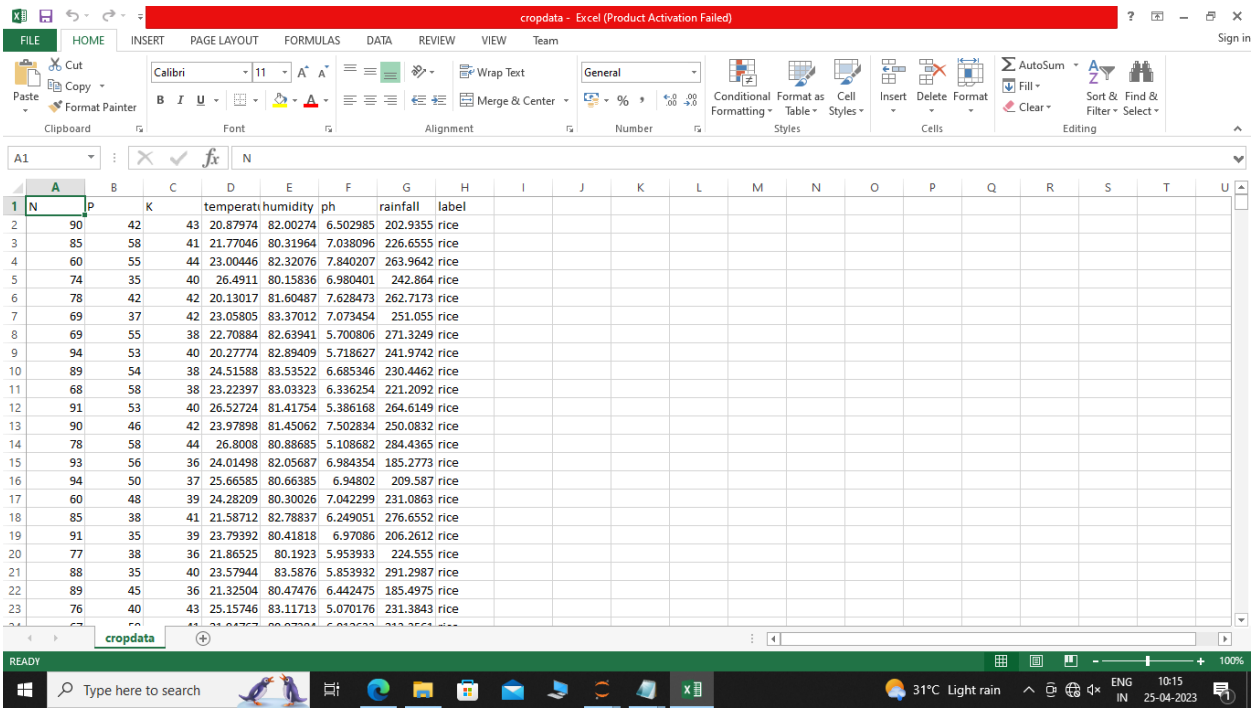


B. DATASET



crop_production - Excel (Product Activation Failed)

State_Name	District_N	Crop_Year	Season	Crop	Area	Production
Andaman	NICOBARS	2000	Kharif	Arecanut	1254	2000
Andaman	NICOBARS	2000	Kharif	Other Kha	2	1
Andaman	NICOBARS	2000	Kharif	rice	102	321
Andaman	NICOBARS	2000	Whole Ye	Banana	176	641
Andaman	NICOBARS	2000	Whole Ye	Cashewnu	720	165
Andaman	NICOBARS	2000	Whole Ye	Coconut	18168	65100000
Andaman	NICOBARS	2000	Whole Ye	Dry ginger	36	100
Andaman	NICOBARS	2000	Whole Ye	Sugarcane	1	2
Andaman	NICOBARS	2000	Whole Ye	Sweet pot	5	15
Andaman	NICOBARS	2000	Whole Ye	Tapioca	40	169
Andaman	NICOBARS	2001	Kharif	Arecanut	1254	2061
Andaman	NICOBARS	2001	Kharif	Other Kha	2	1
Andaman	NICOBARS	2001	Kharif	rice	83	300
Andaman	NICOBARS	2001	Whole Ye	Cashewnu	719	192
Andaman	NICOBARS	2001	Whole Ye	Coconut	18190	64430000
Andaman	NICOBARS	2001	Whole Ye	Dry ginger	46	100
Andaman	NICOBARS	2001	Whole Ye	Sugarcane	1	1
Andaman	NICOBARS	2001	Whole Ye	Sweet pot	11	33
Andaman	NICOBARS	2002	Kharif	rice	189.2	510.84
Andaman	NICOBARS	2002	Whole Ye	Arecanut	1258	2083
Andaman	NICOBARS	2002	Whole Ye	Banana	213	1278
Andaman	NICOBARS	2002	Whole Ye	Black pep	63	13.5



cropdata - Excel (Product Activation Failed)

N	P	K	temperat	humidity	ph	rainfall	label
90	42	43	20.87974	82.00274	6.502985	202.9355	rice
85	58	41	21.77046	80.31964	7.038096	226.6555	rice
60	55	44	23.00446	82.32076	7.840207	263.9642	rice
74	35	40	26.4911	80.15836	6.980401	242.864	rice
78	42	42	20.13017	81.60487	7.628473	262.7173	rice
69	37	42	23.05805	83.37012	7.073454	251.055	rice
69	55	38	22.70884	82.63941	5.700806	271.3249	rice
94	53	40	20.27774	82.89409	5.718627	241.9742	rice
89	54	38	24.51588	83.53522	6.685346	230.4462	rice
68	58	38	23.22397	83.03323	6.336254	221.2092	rice
91	53	40	26.52724	81.41754	5.386168	264.6149	rice
90	46	42	23.97898	81.45062	7.502834	250.0832	rice
78	58	44	26.8008	80.88685	5.108682	284.4365	rice
93	56	36	24.01498	82.05687	6.984354	185.2773	rice
94	50	37	25.66585	80.66385	6.94802	209.587	rice
60	48	39	24.28209	80.30026	7.042299	231.0863	rice
85	38	41	21.58712	82.78837	6.249051	276.6552	rice
91	35	39	23.79392	80.41818	6.97086	206.2612	rice
77	38	36	21.86525	80.1923	5.953933	224.555	rice
88	35	40	23.57944	83.5876	5.853932	291.2987	rice
89	45	36	21.32504	80.47476	6.442475	185.4975	rice
76	40	43	25.15746	83.11713	5.070176	231.3843	rice

C.SAMPLE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report
from sklearn import metrics
import warnings
warnings.simplefilter(action = "ignore")

dataset = pd.read_csv("cropdata.csv")
print(dataset.head())

print(dataset.shape)

print(dataset.size)

df = dataset.copy()

print(df.head())

print(df.describe(include = "all"))

print(df.columns)
print(df['label'].unique)
print(df['label'].value_counts())
```

```
print(df.dtypes)

sns.distplot(df['N'])
plt.show()

sns.distplot(df['P'])
plt.show()

sns.distplot(df['K'])
plt.show()

sns.distplot(df['temperature'])
plt.show()

sns.distplot(df['humidity'])
plt.show()

sns.distplot(df['ph'])
plt.show()

sns.distplot(df['rainfall'])
##plt.show()
sns.heatmap(df.corr(),annot=True)
##plt.show()

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

```
print(x)

print(y)

from sklearn.model_selection import train_test_split

Xtrain, Xtest, Ytrain, Ytest = train_test_split(x,y,test_size = 0.2,random_state =2)

print(Xtrain)

print(Ytest)
print(Xtest)
print(Ytrain)

from statsmodels.stats.outliers_influence import variance_inflation_factor

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

vif = pd.DataFrame()

vif["VIF"] = [variance_inflation_factor(variables.values,i) for i in range(variables.shape[1])]
vif["Feature"]=variables.columns

print(vif)

# Initialzing empty lists to append all model's name and corresponding name
acc = []
model = []
```

```
from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

DecisionTree.fit(Xtrain,Ytrain)

Predicted_values = DecisionTree.predict(Xtest)

x1 = metrics.accuracy_score(Ytest,Predicted_values)
acc.append(x1)
model.append('Decision Tree')

print("Decision Tree accuracy is : " , x1*100)
print(classification_report(Ytest,Predicted_values))

from sklearn.model_selection import cross_val_score

score = cross_val_score(DecisionTree,x,y,cv = 5)
print(score)

import pickle
DT_Model_pkl = open('DT_pkl_file','wb')
pickle.dump(DecisionTree , DT_Model_pkl)
DT_Model_pkl.close()
```

```
from sklearn.naive_bayes import GaussianNB

NaiveBayes = GaussianNB()

NaiveBayes.fit(Xtrain,Ytrain)

predicted_values = NaiveBayes.predict(Xtest)
x1 = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x1)
model.append('Naive Bayes')

print("Naive Bayes's Accuracy is: ", x1)

print(classification_report(Ytest,predicted_values))

score = cross_val_score(NaiveBayes,x,y,cv=5)
score

NB_Model_pkl = open('NB_pkl_file', 'wb')
pickle.dump(NaiveBayes, NB_Model_pkl)
NB_Model_pkl.close()

from sklearn.svm import SVC
from sklearn.preprocessing import MinMaxScaler
norm = MinMaxScaler().fit(Xtrain)
X_train_norm = norm.transform(Xtrain)
X_test_norm = norm.transform(Xtest)
```

```
SVM = SVC(kernel='poly', degree=3, C=1)
SVM.fit(X_train_norm,Ytrain)

predicted_values = SVM.predict(X_test_norm)
x1 = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x1)
model.append('SVM')

print("SVM's Accuracy is: ", x1)

print(classification_report(Ytest,predicted_values))

SVM_Model_pkl = open('SVM_pkl_file', 'wb')
pickle.dump(SVM, SVM_Model_pkl)
SVM_Model_pkl.close()

#from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression

#RF = RandomForestClassifier(n_estimators = 20 , random_state = 0)
LR = LogisticRegression()
LR.fit(Xtrain,Ytrain)

Predicted_values = LR.predict(Xtest)
x1 = metrics.accuracy_score(Ytest,Predicted_values)
acc.append(x1)
model.append('LR')
```

```
print("Logistic Regression accuracy is : " , x1)
print(classification_report(Ytest,Predicted_values))

score = cross_val_score(LR,x,y,cv=5)
score

LR_Model_pkl = open('LR_pkl_file', 'wb')
pickle.dump(LR, LR_Model_pkl)
LR_Model_pkl.close()

plt.figure(figsize=[10,5],dpi = 100)
plt.title('Accuracy Comparison')
plt.xlabel('Accuracy')
plt.ylabel('Algorithm')
sns.barplot(x = acc,y = model,palette='dark')
plt.show()

data = np.array([[104,18, 30, 23.603016, 60.3, 6.7, 140.91]])
prediction = SVM.predict(data)
print(prediction)

df = pd.read_csv("D:/m.divya/newcrop/crop_production.csv")
data = df.dropna()
test = df[~df["Production"].notna()].drop("Production",axis=1)
sum_maxp = data["Production"].sum()
data["percent_of_production"] = data["Production"].map(lambda x:(x/sum_maxp)*100)
```

```
rice_df = data[data["Crop"]==prediction[0]]
print(rice_df.shape)
print(rice_df[:3])

sns.barplot("Season","Production",data=rice_df)
plt.show()
```

D.SAMPLE INPUT

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

(2200, 8)

17600

D.SAMPLE OUTPUT

```
    N   P   K  temperature   humidity        ph    rainfall label
0  90  42  43    20.879744  82.002744  6.502985  202.935536  rice
1  85  58  41    21.770462  80.319644  7.038096  226.655537  rice
2  60  55  44    23.004459  82.320763  7.840207  263.964248  rice
3  74  35  40    26.491096  80.158363  6.980401  242.864034  rice
4  78  42  42    20.130175  81.604873  7.628473  262.717340  rice
                N           P  ...     rainfall  label
count   2200.000000  2200.000000  ...  2200.000000   2200
unique          NaN          NaN  ...          NaN     22
top             NaN          NaN  ...          NaN   rice
freq            NaN          NaN  ...          NaN    100
mean      50.551818    53.362727  ...   103.463655    NaN
std       36.917334    32.985883  ...    54.958389    NaN
min        0.000000     5.000000  ...    20.211267    NaN
25%       21.000000    28.000000  ...    64.551686    NaN
50%       37.000000    51.000000  ...    94.867624    NaN
75%       84.250000    68.000000  ...   124.267508    NaN
max      140.000000   145.000000  ...   298.560117    NaN

[11 rows x 8 columns]
Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label'], dtype='object')
<bound method Series.unique of 0         rice
1         rice
2         rice
3         rice
4         rice
        ...
```

2195 coffee

2196 coffee

2197 coffee

2198 coffee

2199 coffee

Name: label, Length: 2200, dtype: object>

rice 100

maize 100

jute 100

cotton 100

coconut 100

papaya 100

orange 100

apple 100

muskmelon 100

watermelon 100

grapes 100

mango 100

banana 100

pomegranate 100

lentil 100

blackgram 100

mungbean 100

mothbeans 100

pigeonpeas 100

kidneybeans 100

chickpea 100

coffee 100

Name: label, dtype: int64

N int64

P int64

K int64

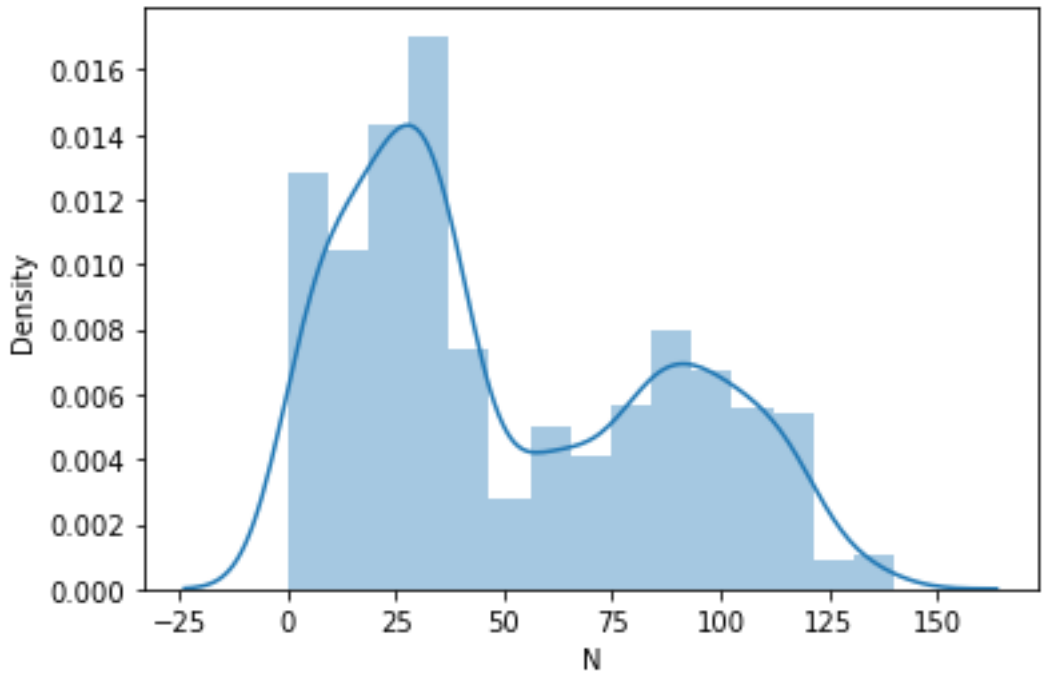
temperature float64

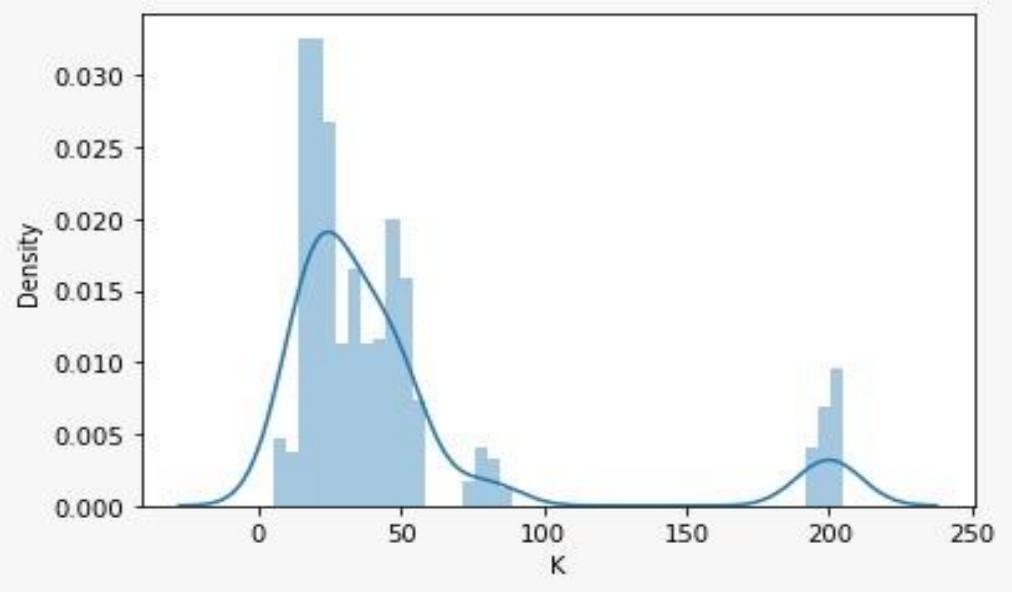
humidity float64

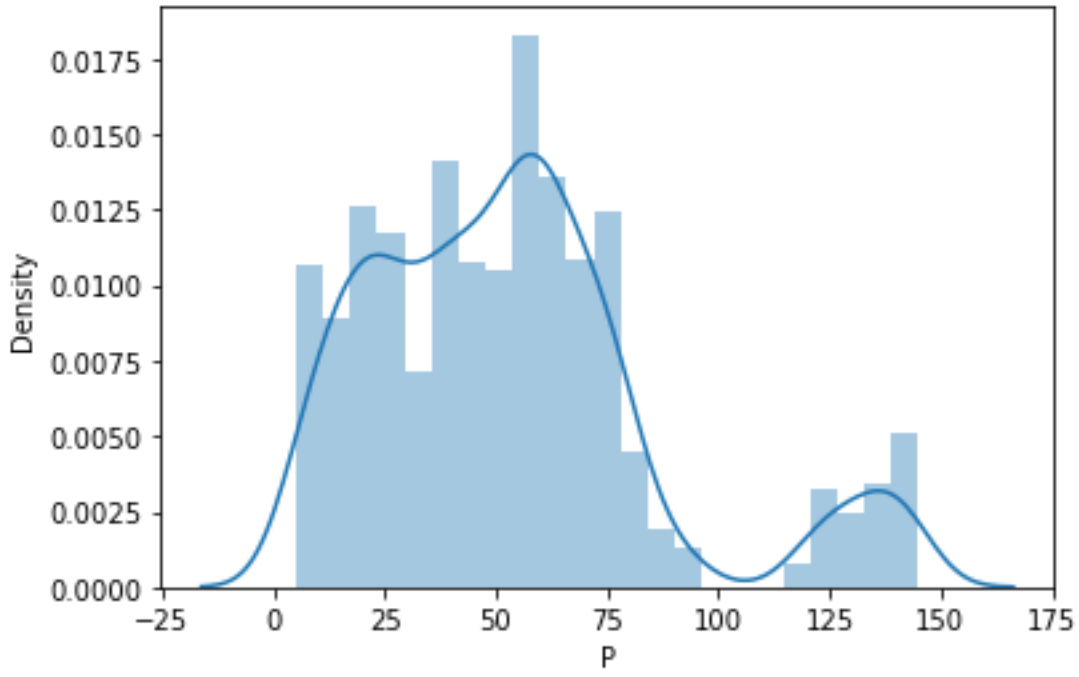
ph float64

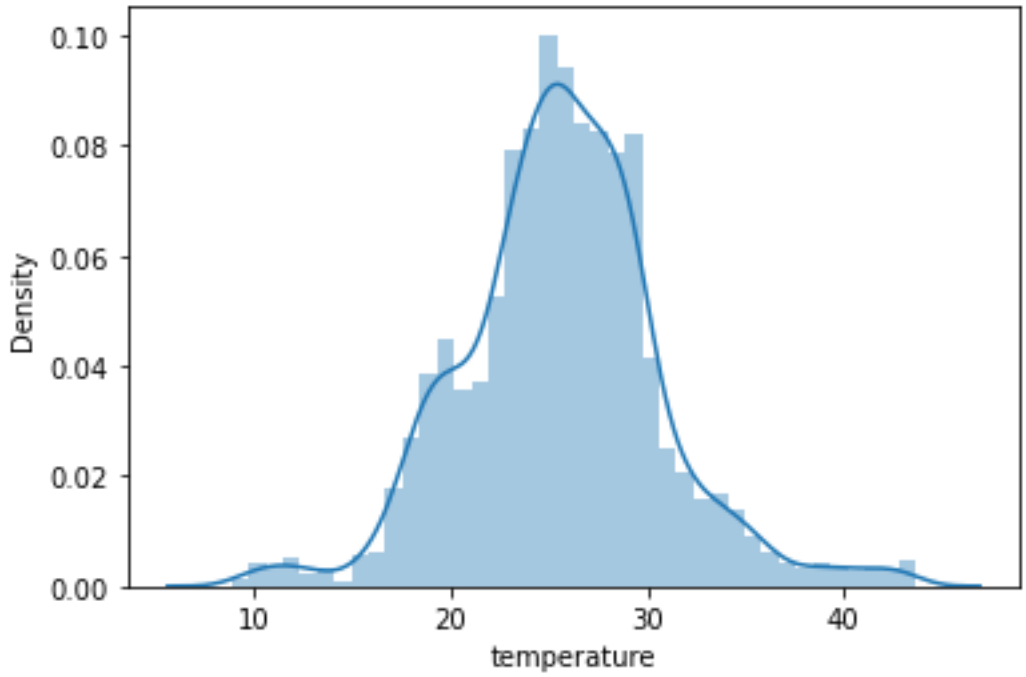
rainfall float64

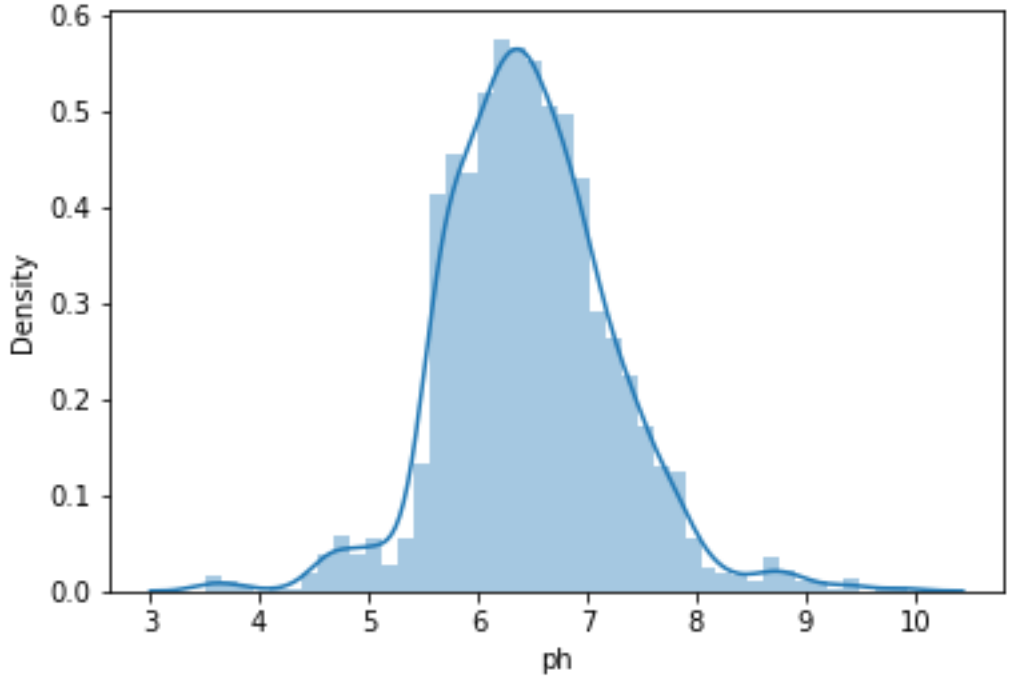
label objectd type: object’

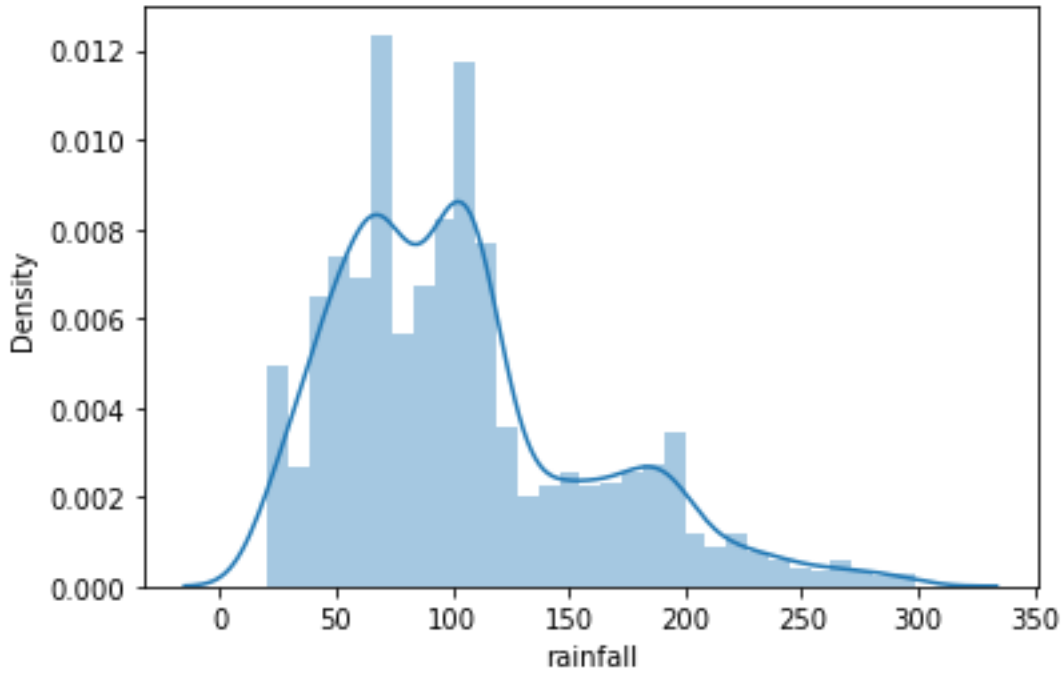


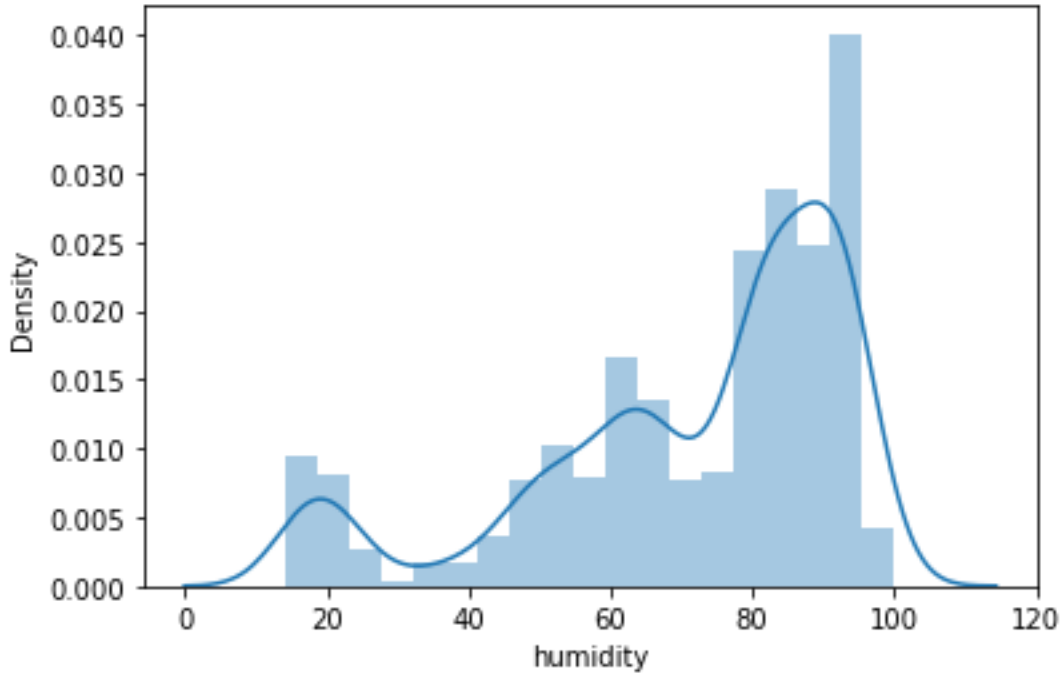

0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
0
50
100
150
200
250
K

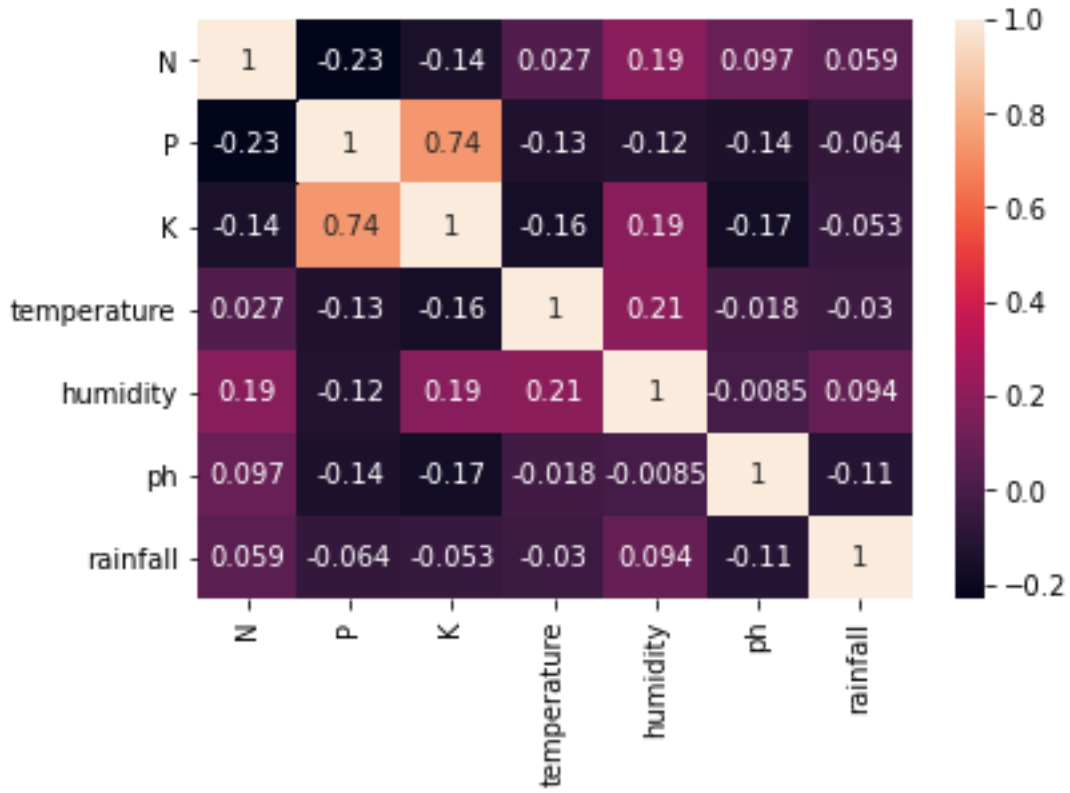

0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
Density
−25
0
25
50
75
100
125
150
175
P

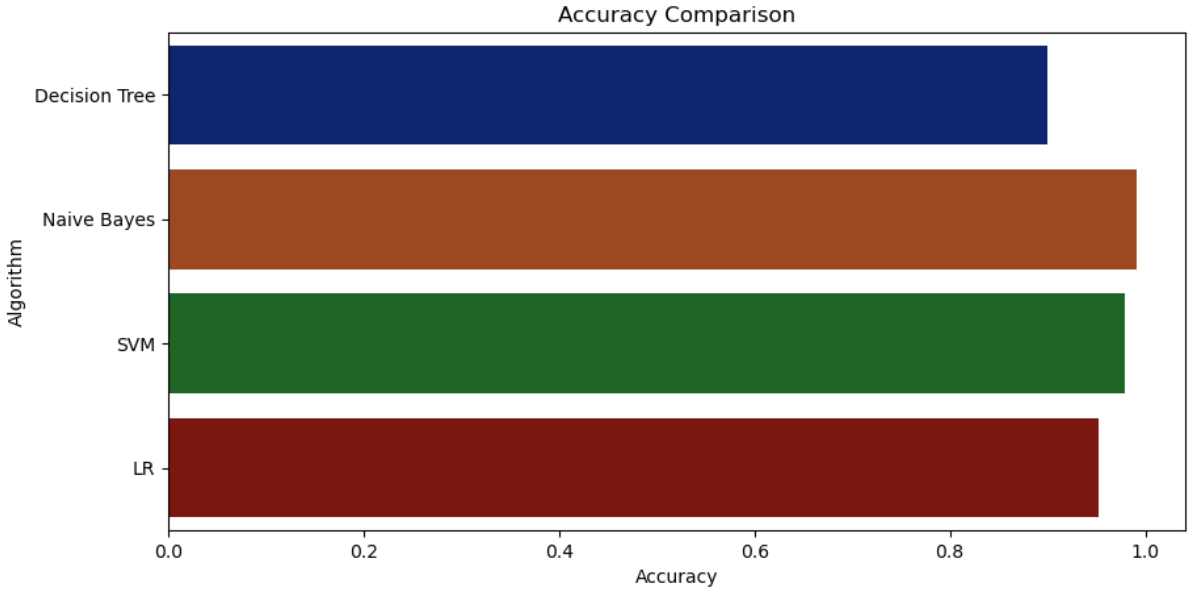

0.10
0.08
0.06
0.04
0.02
0.00
Density
10
20
30
40
temperature

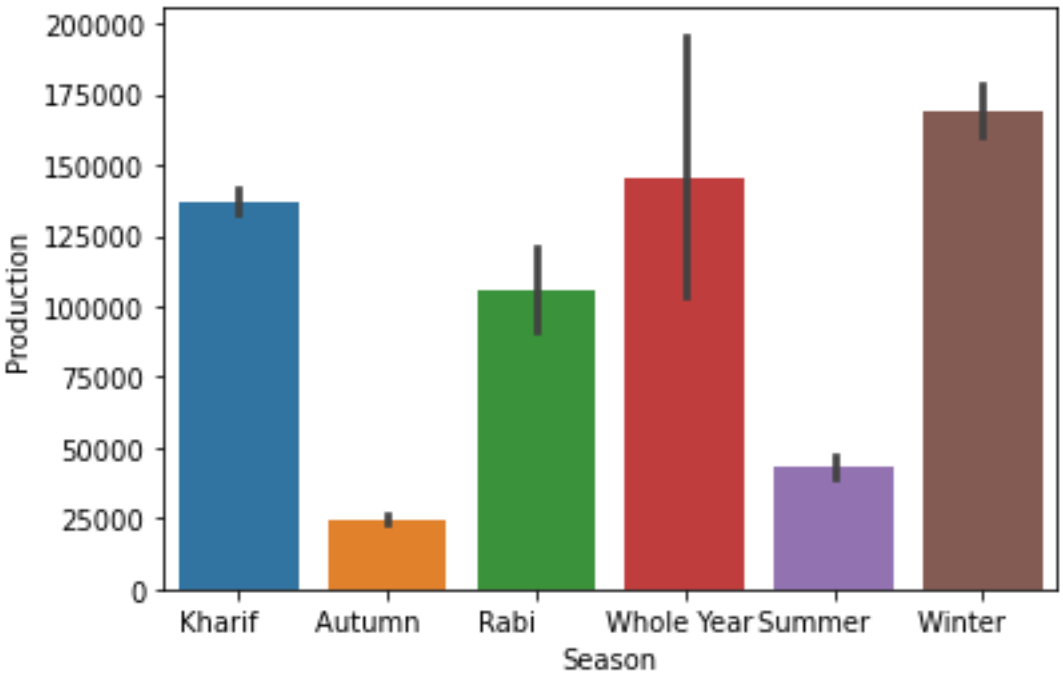

0.6
0.5
0.4
0.3
0.2
0.1
0.0
Density
3
4
5
6
7
8
9
10
ph


0.012
0.010
0.008
0.006
0.004
0.002
0.000
Density
0
50
100
150
200
250
300
350
rainfall


0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
0
20
40
60
80
100
120
humidity


N
P
K
temperature
humidity
ph
rainfall
1 -0.23 -0.14 0.027 0.19 0.097 0.059
-0.23 1 0.74 -0.13 -0.12 -0.14 -0.064
-0.14 0.74 1 -0.16 0.19 -0.17 -0.053
0.027 -0.13 -0.16 1 0.21 -0.018 -0.03
0.19 -0.12 0.19 0.21 1 -0.0085 0.094
0.097 -0.14 -0.17 -0.018 -0.0085 1 -0.11
0.059 -0.064 -0.053 -0.03 0.094 -0.11 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


Accuracy Comparison
Decision Tree
Naive Bayes
SVM
LR
Algorithm
0.0
0.2
0.4
0.6
0.8
1.0
Accuracy


200000
175000
150000
125000
100000
75000
50000
25000
0
Production
Kharif
Autumn
Rabi
Whole Year
Summer
Winter
Season