**A PROJECT REPORT ON**

**CROP YIELD PREDICTION USING NEURAL NETWORKS & MACHINE LEARNING**

**A Dissertation Submitted in Partial Fulfillment of the Academic Requirements for the Award of the Degree of**

**MASTER OF COMPUTER APPLICATION**

**In**

**COMPUTER SCIENCE**

***By***

**Yarasani Lokesh Reddy (Regd. No : 122162010)**

**Dara Jashua (Regd. No : 122162060)**

**Sanivarapu Koti Reddy (Regd.No : 122162080)**

**Komire Siva Sankar (Regd.No : 122162083)**

**Under The Esteemed Guidance of**

**Dr.A.RAMESH BABU**

**Head of the Department of MCA**

**Professor**  ****

**DEPARTMENT OF COMPUTER SCIENCE**

**CHAITANYA INSTITUTE OF TECHNOLOGY AND SCIENCE**

**(Affiliated to Kakatiya University, Warangal)**

**Kishanpura, Hanamkonda, Warangal Urban, Telangana-506001  
Academic Year 2022-24**

**CHAITANYA INSTITUTE OF TECHNOLOGY AND SCIENCE**

**(Affiliated to Kakatiya University, Warangal)**

**Kishanpura, Hanamkonda, Warangal Urban, Telangana-506001  
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

****

**CERTIFICATE** This is to certify that the Dissertation entitled **"CROP YIELD PREDICTION USING NEURAL NETWORKS AND MACHINE LEARNING”** is a Bonafide work done by YARASANI LOKESH REDDY **(**Regd**.**No**:**122162010**),** DARA JASHUA (Regd No:122162060),SANIVARAPU KOITREDDY(RegdNo:122162080),KOMIRESIVASANKAR(RegdNo:122162083) In partial fulfillment of the academic requirements for the award of the degree of Master of Technology in Software engineering submitted to the Department of Computer Science and Engineering, Chaitanya Institute of Technology and Science, Hanamkonda during the period 2022-24.

**Internal Guide Head of the Department Principal**

**Dr.A.RAMESH BABU Dr.A.RAMESH BABU S. Kavitha   
 Professor Professor Professor,Dean,Admin**

**Internal Examiner External Examiner**

**ACKNOWLEDGMENT**

With great pleasure We want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Dr.A.Ramesh Babu, Professor, of MCA Department** for his/her valuable and timely suggestions during the execution of this project work. We express Our deep sense of gratitude for his/her constant guidance throughout the project work.

We would like to thank **Dr.A.Ramesh Babu, Professor & Head of the Department of MCA**, for being moral support during the course of my project.

We are highly indebted to **S. Kavitha, Principal, Dean Administration, Chaitanya**

**Institute of Technology & Science**, Warangal for her kind Patronage and for giving us the

Permission to carry out this project.

We extend our sincere thanks to the staff of MCA Department and all my friends for their

good wishes, their helping hands and constructive criticism, which led the successful

Completion of our project report.

**YARASANI LOKESH REDDY Regd No:122162010**

**DARA JASHUA Regd No:122162060**

**SANIVARAPU** **KOTI REDDY Regd No:122162080**

**KOMIRE SIVA SANKAR Regd No:122162083**

**TABLE OF CONTENTS**

1.ABSTRACT

2.INTRODUCTION

2.1Motivation

2.2Problem Definition

3.SYSTEM REQUIREMENTS

3.1Hardware Requirements

3.2 Software Requirements

3.3 Existing System

3.4 Proposed System

4.SYSTEM DESIGN

4.1 UML Diagrams

4.1.1 Use Case Diagram

4.1.2 Class Diagram

4.1.3 Sequence Diagram

4.1.4 Activity Diagram

5.IMPLEMENTATION

5.1 Technologies to be used

5.2 Screenshots

5.3 Code

6.FUTURE ENHANCEMENT

**7.**CONCLUSION

8.BIPILIOGRAPHY

**1.ABSTRACT**

The impact of climate change in India, most of the agricultural crops are being badly affected in terms of their performance over a period of the last two decades. Predicting the crop yield in advance of its harvest would help the policy makers and farmers for taking appropriate measures for marketing and storage. This project will help the farmers to know the yield of their crop before cultivating onto the agricultural field and thus help them to make the appropriate decisions. It attempts to solve the issue by building a prototype of an interactive prediction system.

Implementation of such a system with an easy-to-use web based graphic user interface and the machine learning algorithm will be carried out. The results of the prediction will be made available to the farmer. Thus, for such kind of data analytics in crop prediction, there are different techniques or algorithms, and with the help of those algorithms we can predict crop yield. Random forest algorithm is used. By analyzing all these issues and problems like weather, temperature, humidity, rainfall, moisture, there is no proper solution and technologies to overcome the situation faced by us. In India, there are many ways to increase the economic growth in the field of agriculture. Data mining is also useful for predicting crop yield production.

Generally, data mining is the process of analyzing data from various viewpoint and summarizing it into important information. Random forest is the most popular and powerful supervised machine learning algorithm capable of performing both classification and regression tasks, that operate by constructing a multitude of decision trees during training time and generating output of the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

**2.INTRODUCTION**

**2.1 MOTIVATION**

The history of agriculture in India[1] dates back to the Indus Valley Civilization Era. India ranks second in this sector. Agriculture and allied sectors like forestry and fisheries account for 15.4 percent of the GDP (gross domestic product) with about 31 percent of the workforce. India ranks first globally with the highest net cropped area followed by US and China. Agriculture is demographically the broadest economic sector and plays a significant role in the overall socio-economic fabric of India. Due to the revolution in industrialization, the economic contribution of agriculture to India’s GDP is steadily declining with the country’s broad-based economic growth.

India is ranked 2nd worldwide in farm output . Agriculture and allied sectors like

forestry and fisheries accounted for 16.6 percent of the GDP 2009, about 50 percent of the

overall workforce . The monetary contribution of agriculture to India‘s GDP is regularly

declining. The crop yield of plants relies on different factors like on climatic, geographical,

organic, political and financial elements. For farmers, it is difficult when there is more than one crop to grow especially when the market prices are unknown to them. Citing the Wikipedia

statistics, the farmer suicide rate in India has ranged between 1.4 and 1.8 per 100000 total

population, over a 10-year period through 2005. While 2014 saw 5650 farmer suicides, the figure crossed 8000 in 2015 .

In recent times, it has become inevitable to use technology to create awareness about

cultivation. The seasonal climatic conditions are also being changed against the fundamental

assets like soil, water and air which lead to insecurity of food. In a scenario, crop yield rate is

falling short of meeting the demand consistently and there is a need for a smart system which can solve the problem of decreasing crop yield.

Therefore, to eliminate this problem, weproposea system which will provide crop selection based on economic and environmental factors to reap the maximum yield out of it for the farmers which will sequentially help meet the elevating demands for the food supplies in the country. The proposed system uses machine learning to make the predictions. The system will provide crop yield and crop selection based on weather attributes suitable for the crop to get the maximum yield out of it for the farmers. The system makes predictions of the productions of crops by studying the factors such as rainfall, temperature, area (in hectares), season, etc. The system also helps in suggesting whether a particular time is the right one to use fertilizers.

Crop yield prediction is an important agricultural problem. Every farmer always tries to

know how much yield will be produced and whether it meets their expectations. In the past, yield prediction was calculated by analyzing a farmer's previous experience on a particular crop.

The Agricultural yield is primarily dependent on weather conditions pests and planning of harvest operation. Accurate information about the history of crop yield is an important thing for making decisions related to agricultural risk management.Relevance of the Project In recent years, India has been shaken by economic and social forces related to higher suicide rates amongst small and marginal farmers . Our aim is to offer assistance and tools to help such farmers and communities and address these issues.

Generally, they face challenges accessing and trusting educational outreach and training to better understand how to increase crop yields and improve financial standing. Because of the serious nature of the issues at stake and general hesitance to trust help from outside the community, any service or product meant to help must be carefully designed and tested in order to ensure positive outcomes and successful adoption.

There is no existing software solution which recommends crops based on multiple factors

such as type of the soil and weather components which include temperature and rainfall. And the systems that already exist are hardware based which makes them expensive and difficult to

maintain. The proposed system suggests a Mobile based application, which can precisely predict the most profitable crop to the farmer by predicting the yield. The user location is identified with the help of GPS and the Area & soil type are taken as user input. According to user location, the crop yield in the respective location is identified from the soil and weather database. After the processing is done at the server side, the result is sent to the user‘s application.

The previous production of the crops is also taken into account which in turn leads to precise crop yield results. Depending on the numerous scenarios and additional filters according to the user requirement, the most producible crop is suggested based on the yield.

While there are many ways to contribute to improvements in the lives of our target

audience, our task was to leverage data to predict a valuable result so that farmers and aid

workers would be able to make informed planning decisions. Ultimately, the focus of the work

during this project was to both conduct audience research that would direct the design of the

product and design a data model that would produce the desired results.

**2.2 PROBLEM DEFINITION**

The problem that the Indian Agriculture sector is facing is the integration of technology to bring the desired outputs. With the advent of new technologies and overuse of non-renewable energy resources patterns of rainfall and temperature are disturbed. 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 yield productivity. In order to perform accurate prediction and handle inconsistent trends in temperature and rainfall various machine learning algorithms like RNN, LSTM, etc can be applied to get a pattern. It will complement the agricultural growth in India and all together augment the ease of living for farmers. In past, many researchers have applied machine learning techniques to enhance agricultural growth of the country. Prediction of crop yield using Machine Learning Techniques. The goal of the project is to help the users choose a suitable crop to grow in order to maximize the yield and hence the profit.

The system proposed tries to overcome the drawbacks of existing systems and make

predictions by analyzing structured data. The solution we are proposing is to design a system

taking into consideration the most influencing parameters to grow a crop and to get a better

selection of crops which can be grown over the season. This would help reduce the difficulties

faced by the farmers in selecting the crop to get high yield and thus maximize profits which in

turn will reduce the suicide rates.

There are multiple ways to increase and improve the crop yield and the quality of the crops. Data mining is also useful for predicting crop yield production. The main objectives are

* To use machine learning techniques to predict crop yield.
* To provide easy to use User Interface.
* To increase the accuracy of crop yield prediction.
* To analyze different climatic parameters (cloud cover, rainfall, temperature)

**3.SYSTEM REQUIREMENTS**

**3.1 HARDWARE REQUIREMENTS:**

* System : Pentium IV 2.4 GHz.
* Ram : 8GB

**3.2 SOFTWARE REQUIREMENTS**

* **Operating System:** Windows 11
* **Coding Language**: Python 3.7

**3.3 EXISTING SYSTEM:**

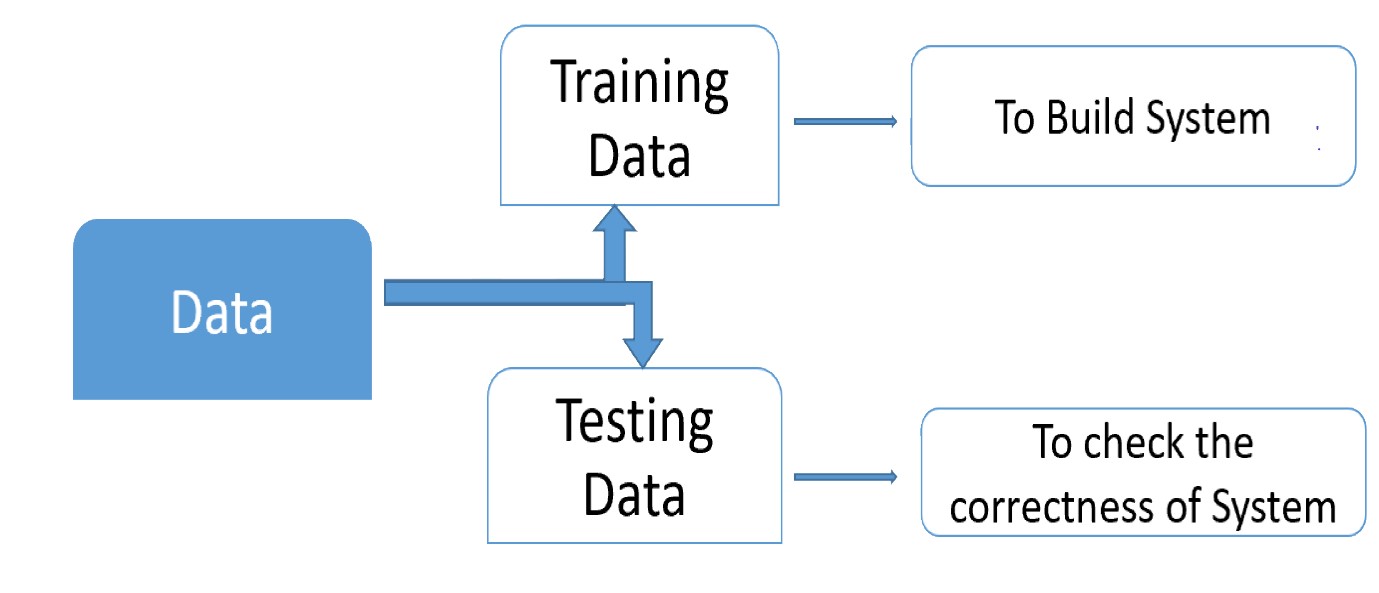
Due to the revolution in industrialization, the economic contribution of agriculture to India’s GDP is steadily declining with the country’s broad-based economic growth. The problem that the Indian Agriculture sector is facing is the integration of technology to bring the desired outputs. The inconsistent 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 yield productivity. In order to perform accurate prediction and handle inconsistent trends in temperature and rainfall various machine learning algorithms like RNN, LSTM, etc can be applied to get a pattern. It will complement the agricultural growth in India and all together augment the ease of living for farmers. In past, many researchers have applied machine learning techniques to enhance agricultural growth of the country.

**3.4 PROPOSED SYSTEM:**

This paper focuses on the practical application of machine learning algorithms and its quantification. The work presented here also takes into account the inconsistent data from rainfall and temperature datasets to get a consistent trend. Crop yield prediction is determined by considering all the features in contrast with the usual trend of determining the prediction considering one feature at a time.

**4.SYSTEM DESIGN**

**4.1 SYSTEM ARCHITECTURE:**

****

**4.1.1 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

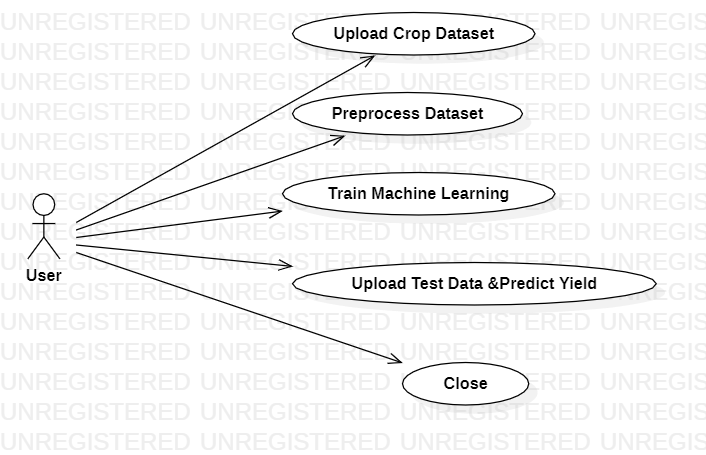
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

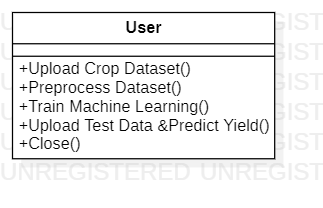
**4.1.2 USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



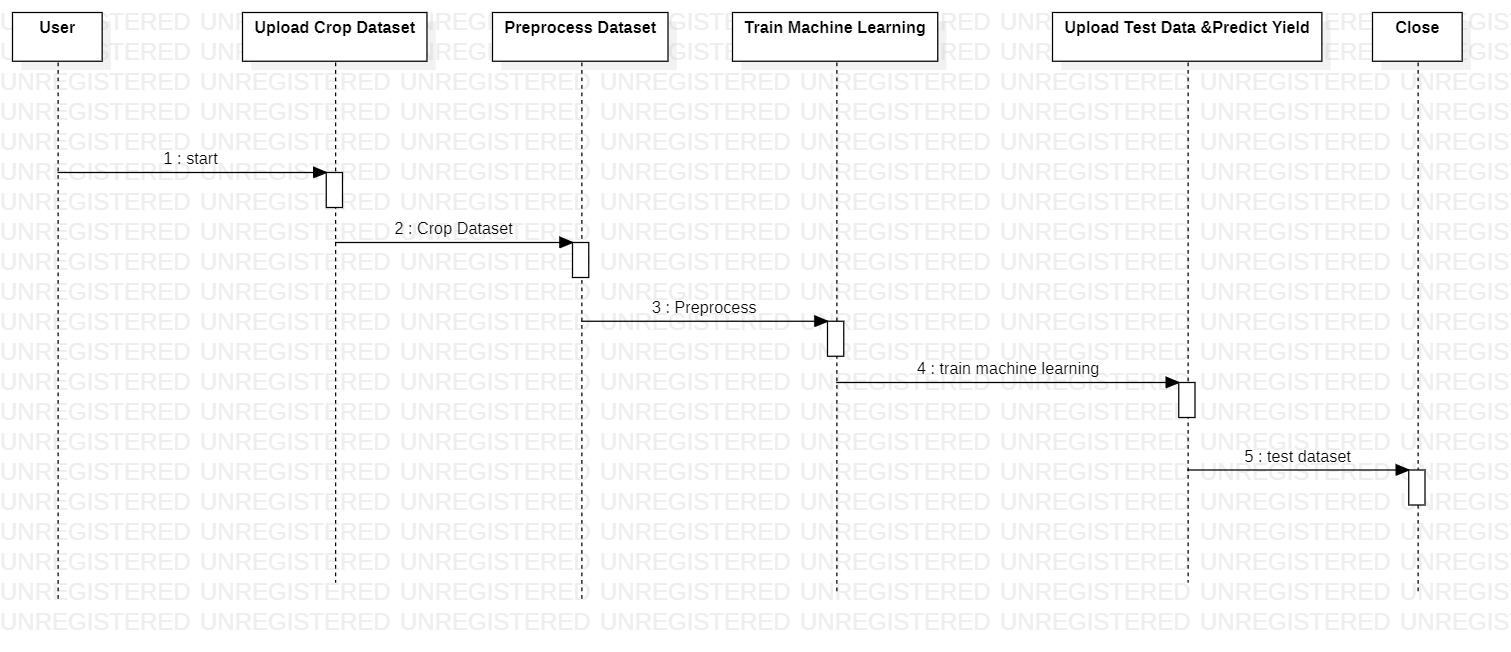
**4.1.3 CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

****

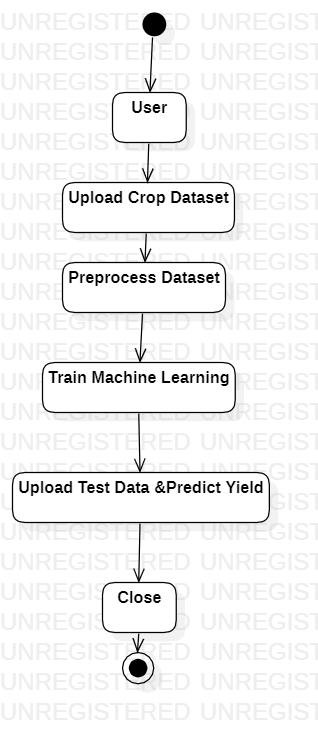
**4.1.4 SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

****

**4.1.5 ACTIVITY DIAGRAM:**

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.

****

**5 .IMPLEMENTATION**

**MODULES:**

**Upload Crop Dataset**

The crop production dataset that is used to predict the name and yield of the crop is fed into classification and regression algorithms.

**Preprocess Dataset**

Experiments were conducted on Indian government dataset and it has been established that Random Forest Regressor gives the highest yield prediction accuracy. Sequential model that is Simple Recurrent Neural Network performs better on rainfall prediction while LSTM is good for temperature prediction. By combining rainfall, temperature along with other parameters like season and area, yield prediction for a certain district can be made.

**Train Machine Learning**

This 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 Data &Predict Yield**

Results reveal 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.

**5.1 TECHNOLOGIES TO BE USED**

# What is Python :-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
  + GUI Applications (like Kivy, Tkinter, PyQt etc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, Beautiful Soup, Selenium)
  + Test frameworks
  + Multimedia
  + **Advantages of Python** :-

Let’s see how Python dominates over other languages.

#### 1. Extensive Libraries:

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

#### 2. Extensible:

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

#### 3. Embeddable:

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

#### 4. Improved Productivity:

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### 5. IOT Opportunities:

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

#### 6. Simple and Easy:

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### 7. Readable:

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

#### 8. Object-Oriented:

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

#### 9. Free and Open-Source:

Like we said earlier, Python is **freely available.** But not only can you[**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### 10. Portable:

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

#### 11. Interpreted:

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

### **Advantages of Python Over Other Languages**

#### 1. Less Coding:

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### 2. Affordable:

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

#### 3. Python is for Everyone:

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### **Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations:

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers:

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Bryton is that it isn’t that secure.

#### 3. Design Restrictions:

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers:

Compared to more widely used technologies like **JDBC (Java Database Connectivity)** and **ODBC (Open Database Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple:

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**History of Python : -**

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system.

In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

**What is Machine Learning : -**

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

**Categories Of Machine Learning :-**

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

## **Need for Machine Learning**

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

## **Challenges in Machines Learning** :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of over-fitting & under-fitting** − If the model is over fitting or under fitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

## **Applications of Machines Learning :-**

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention

# How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of  **$146,085** per year. But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### **How to start learning ML?**

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites:

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### (a) Learn Linear Algebra and Multivariate Calculus:

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on math’s as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### (b) Learn Statistics:

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!  
 Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

#### (c) Learn Python:

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Kera’s](https://keras.io/), [TensorFlow](https://www.tensorflow.org/), [Scikit-learn](https://scikit-learn.org/stable/), etc.

### Step 2 – Learn Various ML Concepts:

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### (a) Terminologies of Machine Learning:

* **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

#### (b) Types of Machine Learning:

* **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### **Advantages of Machine learning :-**

#### 1. Easily identifies trends and patterns :

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### 2. No human intervention needed (automation):

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### 3. Continuous Improvement :

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### 4. Handling multi-dimensional and multi-variety data:

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### 5. Wide Applications:

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### **Disadvantages of Machine Learning** :

#### 1. Data Acquisition:

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

#### 2. Time and Resources:

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

#### 3. Interpretation of Results:

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

#### 4. High error-susceptibility:

[**Machine Learning**](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
 Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Modules Used in Project :-**

**TensorFlow:**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

**NumPy:**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas:**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib:**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [I Python](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with I Python. For the power user, you have full control of line styles, font properties, axes properties, etc., via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn:**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

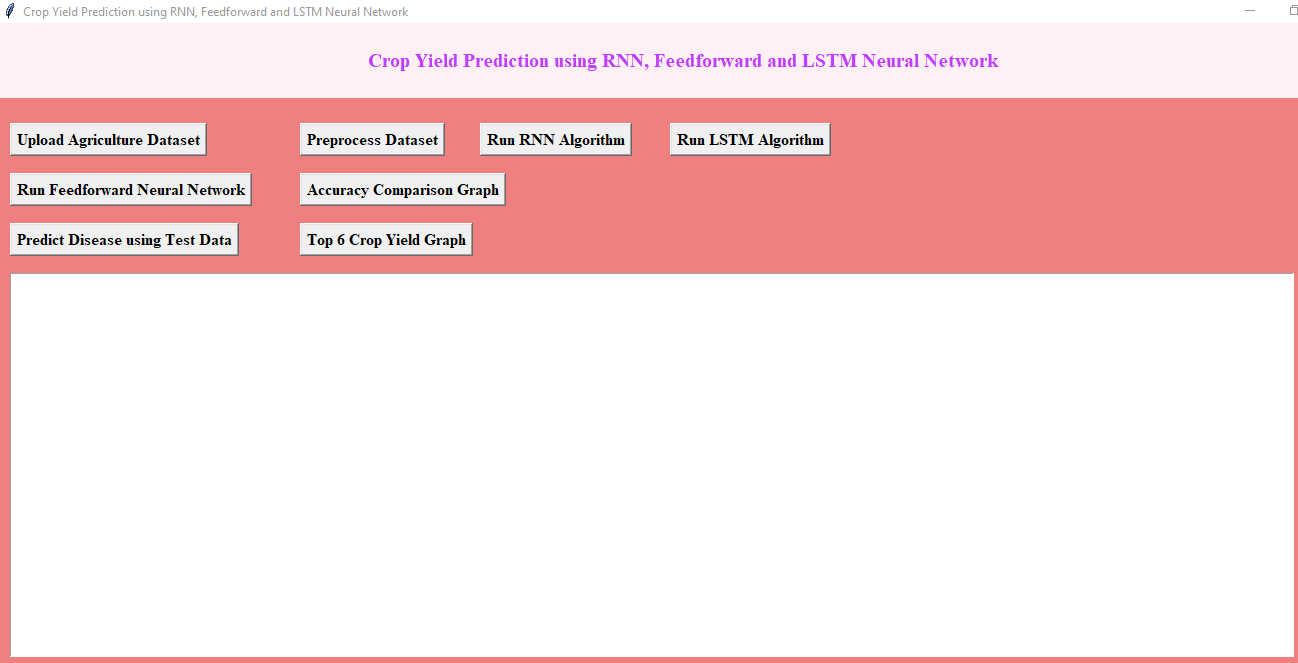
Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

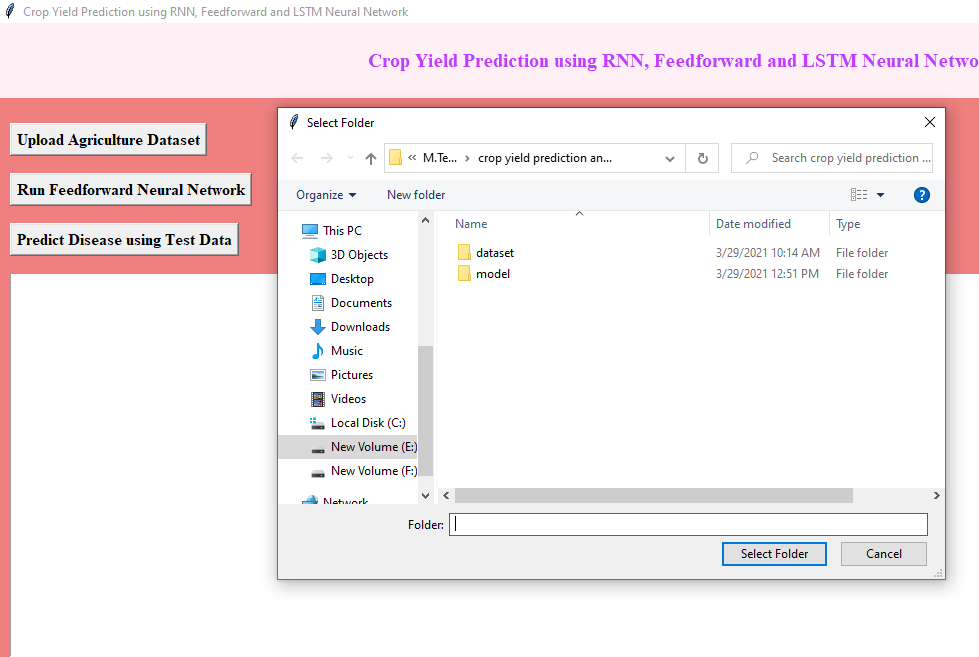
Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills.

**5.2 SCREENSHOTS**

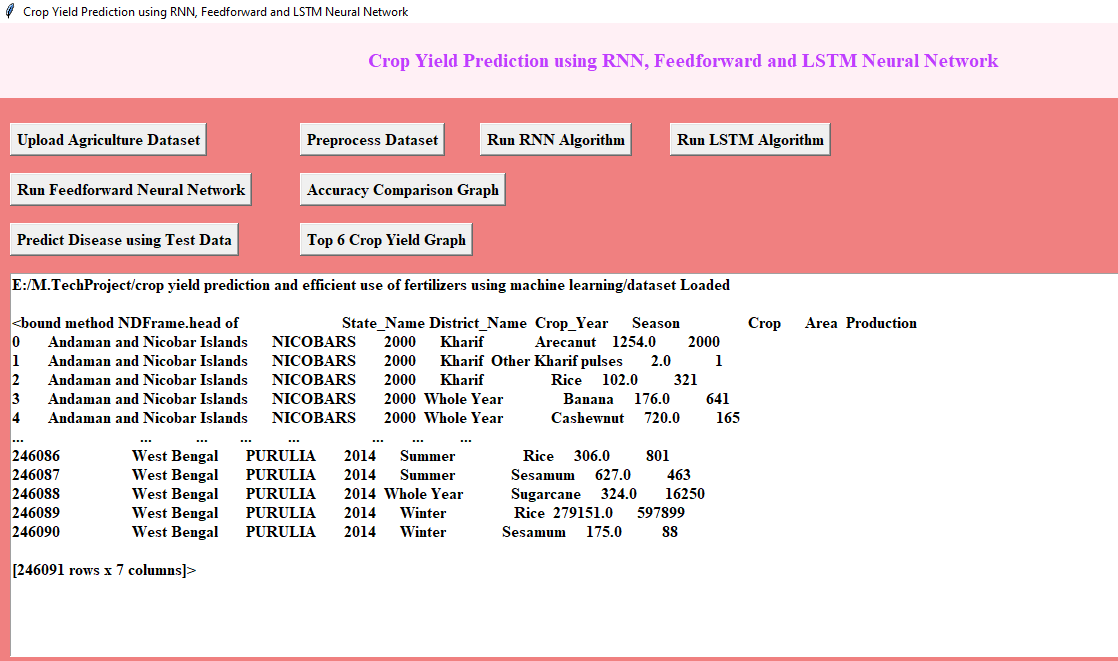
To run project double click on ‘run.bat’ file to get below screen



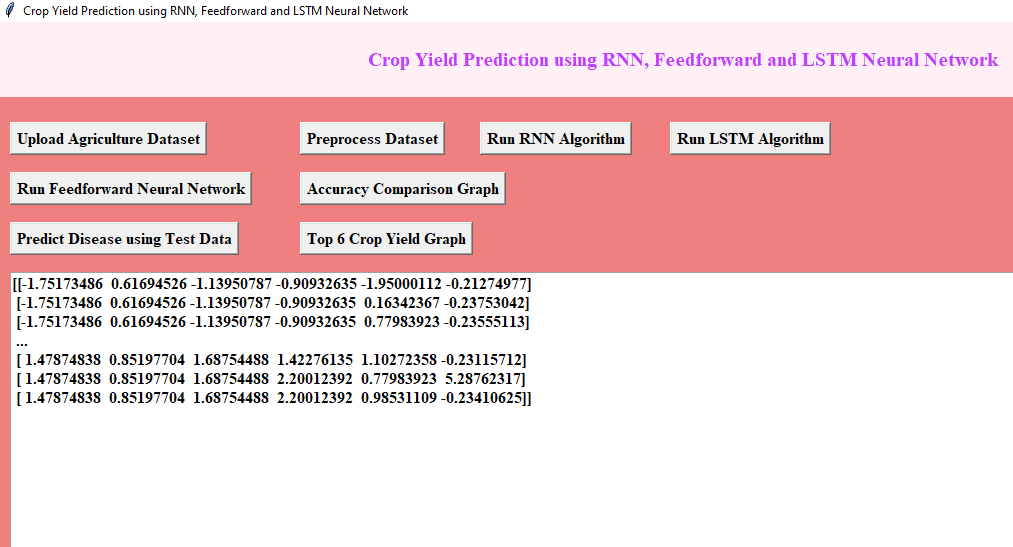
In above screen click on ‘Upload Crop Dataset’ button to upload dataset



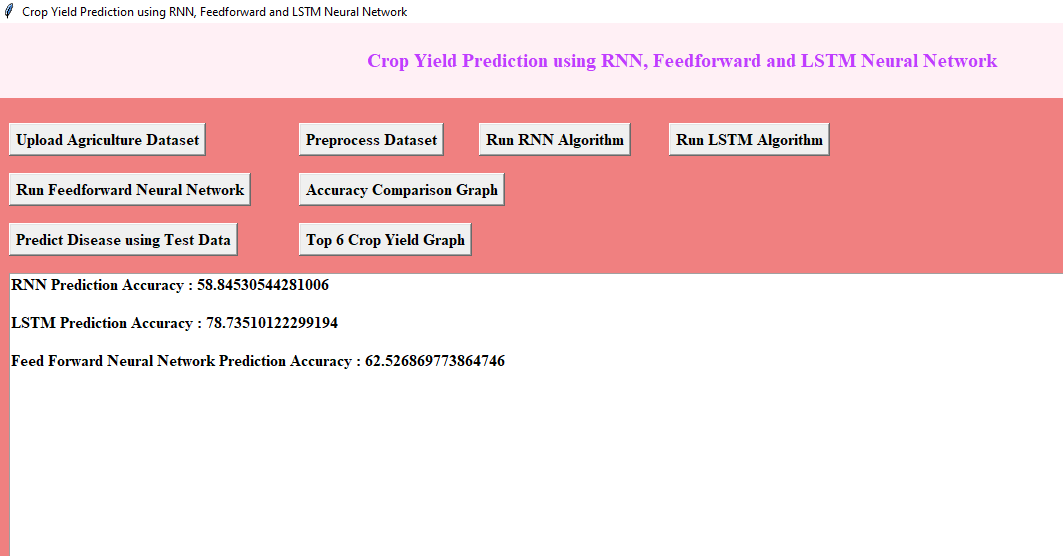
In above screen selecting and uploading ‘Dataset.csv’ file and then click on ‘Open’ button to load dataset and to get below screen



Dataset’ button to process dataset

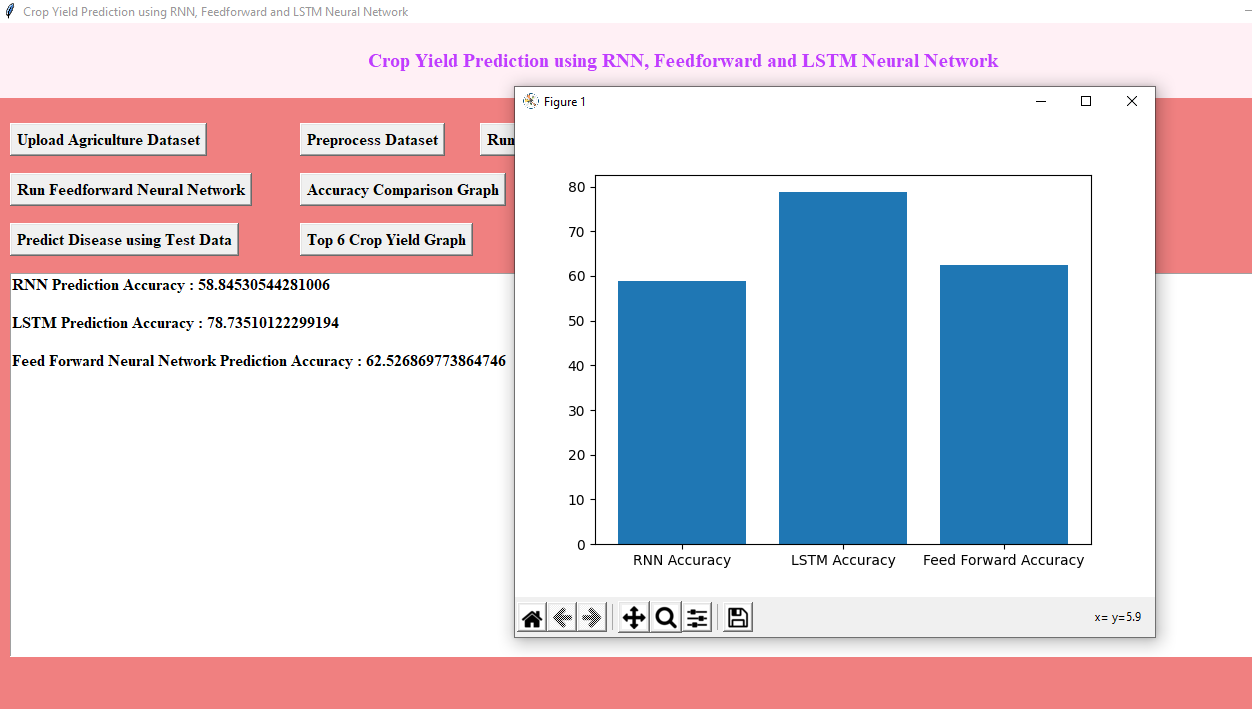


In above screen all non-numeric values converted to numeric format and in below lines we can see dataset contains total 246091 records and application using (80%) 196872 records to train ML and using (20%) 49219 records to test ML prediction error rate (RMSE (root mean square error)). Now click on ‘Train Machine Learning Algorithm’ button to train Decision Tree Machine learning algorithm on above dataset and then calculate prediction error rate

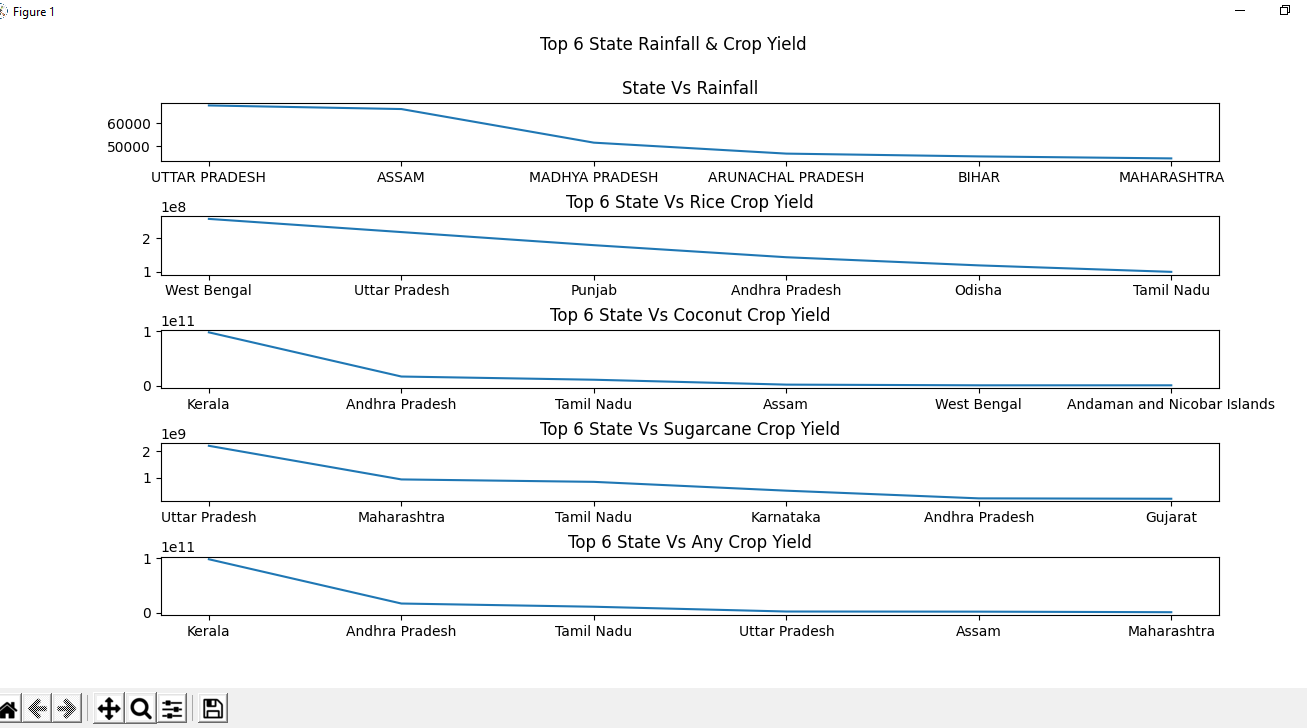


In above screen ML is trained and we got prediction error rate as 0.067% and now Decision Tree model is ready and now click on ‘Upload Test Data & Predict Yield’ button to upload test data and then application will predict production





In above screen selecting and uploading ‘test.csv’ file and then click on ‘Open’ button to load test data and then application will give below prediction result



In above screen each test record is separated with newline and in above screen in square bracket we can see test data values and after square bracket we can see predicted production and after that we can see predicted YIELD per acre. So each test record and its prediction is separated with newline.

**5.3 CODE:**

from tkinter import messagebox

from tkinter import \*

from tkinter.filedialog import askopenfilename

from tkinter import simpledialog

import tkinter

import numpy as np

from tkinter import filedialog

import pandas as pd

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

import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import normalize

from sklearn.metrics import mean\_squared\_error

from sklearn.tree import DecisionTreeRegressor

import numpy as np

import pandas as pd

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

from sklearn.preprocessing import MinMaxScaler, StandardScaler

from sklearn.ensemble import RandomForestClassifier

main = tkinter.Tk()

main.title("Crop Yield Prediction for Fertilizer, Crop and Price")

main.geometry("1300x1200")

global filename

global X\_train, X\_test, y\_train, y\_test

global X,Y

global dataset

global le

global model

global ms, sc, rfc

def upload():

global filename

global dataset

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir = "Dataset")

pathlabel.config(text=filename)

text.delete('1.0', END)

text.insert(END,'CropYield dataset loaded\n')

dataset = pd.read\_csv(filename,encoding='unicode\_escape')

#crop = pd.read\_csv("C:/Users/User/Desktop/python/Crop\_recommendation.csv",encoding='unicode\_escape')

v=dataset.head()

text.insert(END,str(dataset.head(10))+"\n")

def processDataset():

global le

global dataset

global X\_train, X\_test, y\_train, y\_test

global X,Y

global ms, sc, rfc

text.delete('1.0', END)

le = LabelEncoder()

X = dataset.drop(['label', 'Soil'], axis=1)

y = dataset['label']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

ms = MinMaxScaler()

sc = StandardScaler()

X\_train = ms.fit\_transform(X\_train)

X\_train = sc.fit\_transform(X\_train)

X\_test=ms.transform(X\_test)

X\_test=sc.transform(X\_test)

rfc = RandomForestClassifier()

rfc.fit(X\_train, y\_train)

text.insert(END,"\n\nTotal records found in dataset is : "+str(len(X))+"\n")

text.insert(END,"all the data are trained with machine learning algorithm : "+str(X\_train.shape[0])+"\n")

def trainModel():

global model

text.delete('1.0', END)

global X\_train, X\_test, y\_train, y\_test

global X,Y

model = DecisionTreeRegressor(max\_depth=100,random\_state=0,max\_leaf\_nodes=20,max\_features=5,splitter="random")

model.fit(X,Y)

predict = model.predict(X\_test)

mse = mean\_squared\_error(predict,y\_test)

rmse = np.sqrt(mse)/ 1000

text.insert(END,"Training process completed\n")

text.insert(END,"Decision Tree Machine Learning Algorithm Training RMSE Error Rate : "+str(rmse)+"\n\n")

def cropYieldPredict():

global model

global le

global ms, sc, rfc

crop\_dict = {

'rice': 1,'maize': 2,'jute': 3,'cotton': 4,'coconut': 5,'papaya': 6,'orange': 7,'apple': 8,'muskmelon': 9,'watermelon': 10,

'grapes': 11,'mango': 12,'banana': 13,'pomegranate': 14,'lentil': 15,'blackgram': 16,'mungbean': 17,'mothbeans': 18,

'pigeonpeas': 19,'kidneybeans': 20,'chickpea': 21,'coffee': 22

}

soil\_dict = {

'clay loams': 1,'maize': 2,'jute': 3,'cotton': 4,'coconut': 5,'papaya': 6,'orange': 7,'apple': 8,'muskmelon': 9,'watermelon': 10,

'grapes': 11,'mango': 12,'banana': 13,'pomegranate': 14,'lentil': 15,'blackgram': 16,'mungbean': 17,'mothbeans': 18,

'pigeonpeas': 19,'kidneybeans': 20,'chickpea': 21,'coffee': 22

}

dataset['crop\_num']=dataset['label'].map(crop\_dict)

natural\_fertilizers = {

"rice": ["Compost", "Manure", "NPK Fertilizer"],"maize": ["Compost", "Manure", "NPK Fertilizer"],

"jute": ["Organic Fertilizer", "NPK Fertilizer"],"cotton": ["Compost", "Manure", "NPK Fertilizer"],

"coconut": ["Organic Fertilizer", "Palm Ash", "Fish Emulsion"],"papaya": ["Compost", "Manure", "Organic Fertilizer"],

"orange": ["Compost", "Manure", "Citrus Fertilizer"],"apple": ["Compost", "Manure", "Fruit Tree Fertilizer"],

"muskmelon": ["Compost", "Manure", "Melon Fertilizer"],"watermelon": ["Compost", "Manure", "Melon Fertilizer"],

"grapes": ["Compost", "Manure", "Grape Vine Fertilizer"],"mango": ["Compost", "Manure", "Fruit Tree Fertilizer"],

"banana": ["Compost", "Manure", "Banana Fertilizer"],"pomegranate": ["Compost", "Manure", "Fruit Tree Fertilizer"],

"lentil": ["Compost", "Manure", "Legume Fertilizer"],"blackgram": ["Compost", "Manure", "Legume Fertilizer"],

"mungbean": ["Compost", "Manure", "Legume Fertilizer"],"mothbeans": ["Compost", "Manure", "Legume Fertilizer"],

"pigeonpeas": ["Compost", "Manure", "Legume Fertilizer"],"kidneybeans": ["Compost", "Manure", "Legume Fertilizer"],

"chickpea": ["Compost", "Manure", "Legume Fertilizer"],"coffee": ["Compost", "Manure", "Coffee Plant Fertilizer"],

# Add more crops and their suitable natural fertilizers here

}

price = {

"rice": ["23,450 per Acer"],"maize": ["19,450 per Acer"],

"jute": ["33,450 per Acer"],"cotton": ["19,250 per Acer"],

"coconut": ["43,450 per Acer"],"papaya": ["12,47 per Acer"],

"orange": ["53,450 per Acer"],"apple": ["29,450 per Acer"],

"muskmelon": ["63,450 per Acer"],"watermelon": ["83,450 per Acer"],

"grapes": ["73,450 per Acer"],"mango": ["73,450 per Acer"],

"banana": ["83,450 per Acer"],"pomegranate": ["63,450 per Acer"],

"lentil": ["93,450 per Acer"],"blackgram": ["53,450 per Acer"],

"mungbean": ["13,450 per Acer"],"mothbeans": ["43,450 per Acer"],

"pigeonpeas": ["23,450 per Acer"],"kidneybeans": ["33,450 per Acer"],

"chickpea": ["33,450 per Acer"],"coffee": ["Compost", "23,450 per Acer"],

# Add more crops and their suitable natural fertilizers here

}

text.delete('1.0', END)

def recommendation(N, P, K, temperature, humidity, pH, rainfall):

# Predict the crop as before

features = np.array([[N, P, K, temperature, humidity, pH, rainfall]])

transformed\_features = ms.transform(features)

transformed\_features = sc.transform(transformed\_features)

predicted\_crop = rfc.predict(transformed\_features)[0]

# Get the recommended natural fertilizers for the predicted crop from the dictionary

recommended\_fertilizers = natural\_fertilizers.get(predicted\_crop,[])

cost = price.get(predicted\_crop,[])

return predicted\_crop, recommended\_fertilizers, cost

if not recommended\_fertilizers:

recommended\_fertilizers=["Unknown Crop"]

cost =["0"]

return predicted\_crop,recommended\_fertilizers, cost

text.insert(END,str("prediction is started")+"\n")

n=data.get()

n1=data1.get()

n2=data2.get()

n3=data3.get()

n4=data4.get()

n5=data5.get()

n6=data6.get()

if(n1=="" or n2=="" or n== "" or n3=="" or n4=="" or n5== "" or n6== "" or not(n.isdigit()) or not(n1.isdigit()) or not(n2.isdigit()) or not(n3.isdigit()) or not(n4.isdigit()) or not(n5.isdigit()) or not(n6.isdigit())):

text.insert(END,str("Please enter valid data only")+"\n")

else:

N = int(n)

P = int(n1)

K = int(n2)

temperature = float(n3)

humidity = float(n4)

pH = float(n5)

rainfall = float(n6)

predicted\_crop, recommended\_fertilizers, price = recommendation(N, P, K, temperature, humidity, pH, rainfall)

if predicted\_crop != "Unknown Crop":

print(f"{predicted\_crop} is a best crop to be cultivated.")

print(f"Suggested natural fertilizers for {predicted\_crop}: {', '.join(recommended\_fertilizers)}")

print(f"Estimated Price for {predicted\_crop}: {', '.join(price)}")

else:

print("Sorry, we are not able to recommend a proper crop for this environment.")

text.insert(END,str("Sorry, we are not able to recommend a proper crop for this environment.")+"\n")

text.insert(END,str(f"{predicted\_crop} is a best crop to be cultivated.")+"\n")

text.insert(END,str(f"Suggested natural fertilizers for {predicted\_crop}: {', '.join(recommended\_fertilizers)}")+"\n")

text.insert(END,str(f"Estimated Price for {predicted\_crop}: {', '.join(price)}")+"\n")

text.insert(END,str("prediction is done")+"\n")

def close():

main.destroy()

font = ('times', 16, 'bold')

title = Label(main, text='Crop Yield Prediction for Fertilizer, Crop and Price')

title.config(bg='#345ae7', fg='#F4D02a')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 14, 'bold')

upload = Button(main, text="Upload Crop Dataset", command=upload)

upload.place(x=700,y=100)

upload.config(font=font1)

pathlabel = Label(main)

pathlabel.config(bg='DarkOrange1', fg='white')

pathlabel.config(font=font1)

#pathlabel.place(x=700,y=100)

processButton = Button(main, text="Preprocess Dataset", command=processDataset)

processButton.place(x=700,y=150)

processButton.config(font=font1)

'''mlButton = Button(main, text="Train Machine Learning Algorithm", command=trainModel)

mlButton.place(x=700,y=200)

mlButton.config(font=font1) temperature, humidity, pH, rainfall'''

data=StringVar()

data1=StringVar()

data2=StringVar()

data3=StringVar()

data4=StringVar()

data5=StringVar()

data6=StringVar()

user\_name = Label(main, text = "N").place(x=700,y=200)

N\_val = Entry(main, textvariable=data, width = 30).place(x = 780,y = 200)

p = Label(main, text = "P").place(x=700,y=230)

P\_val = Entry(main, textvariable=data1, width = 30).place(x = 780,y = 230)

k = Label(main, text = "K").place(x=700,y=260)

k\_val = Entry(main, textvariable=data2, width = 30).place(x = 780,y = 260)

t = Label(main, text = "temperature").place(x=700,y=290)

t\_val = Entry(main, textvariable=data3, width = 30).place(x = 780,y = 290)

h = Label(main, text = "humidity").place(x=700,y=320)

h\_val = Entry(main, textvariable=data4, width = 30).place(x = 780,y = 320)

ph = Label(main, text = "pH").place(x=700,y=350)

ph\_val = Entry(main, textvariable=data5, width = 30).place(x = 780,y = 350)

ra = Label(main, text = "rainfall").place(x=700,y=380)

ra\_val = Entry(main, textvariable=data6, width = 30).place(x = 780,y = 380)

predictButton = Button(main, text="Predict Yield", command=cropYieldPredict)

predictButton.place(x=700,y=410)

predictButton.config(font=font1)

closeButton = Button(main, text="Close", command=close)

closeButton.place(x=700,y=460)

closeButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=30,width=80)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=100)

text.config(font=font1)

main.config(bg='#98ae45')

main.mainloop()

**6. FEATURE ENHANCEMENT**

This project will help the farmers to know the yield of their crop before cultivating onto the agricultural field and thus help them to make the appropriate decisions. It attempts to solve the issue by building a prototype of an interactive prediction system.

Implementation of such a system with an easy-to-use web based graphic user interface and the machine learning algorithm will be carried out. The results of the prediction will be made available to the farmer. Thus, for such kind of data analytics in crop prediction, there are different techniques or algorithms, and with the help of those algorithms we can predict crop yield. Random forest algorithm is used. By analyzing all these issues and problems like weather, temperature, humidity, rainfall, moisture, there is no proper solution and technologies to overcome the situation faced by us. In India, there are many ways to increase the economic growth in the field of agriculture. Data mining is also useful for predicting crop yield production.

**7.CONCLUSION**

The paper presented the various machine learning algorithms for predicting the yield of the crop on the basis of temperature, rainfall, season and area. Experiments were conducted on Indian government dataset and it has been established that Random Forest Regressor gives the highest yield prediction accuracy. Sequential model that is Simple Recurrent Neural Network performs better on rainfall prediction while LSTM is good for temperature prediction. By combining rainfall, temperature along with other parameters like season and area, yield prediction for a certain district can be made. Results reveal 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.

The future work is focused on providing the sequence of crops to be grown depending on the soil and weather conditions and to update the datasets time to time to produce accurate predictions. The Future Work targets a fully automated system that will do the same. Another functionality that we are trying to implement is to provide the correct fertilizer for the given crop and location. To implement this through study of fertilizers and their relationship with soil and climate is required.

**8 .BIPILIOGRAPHY**

1. Agriculture Role on Indian Economy Madhusudhan L - <https://www.omicsonline.org/open-access/agriculture-role-on-indianeconomy-2151-6219-1000176.php?aid=62176>

2. Priya, P., Muthaiah, U., Balamurugan, M. International Journal of Engineering Sciences Research Technology Predicting Yield of the Crop Using Machine Learning Algorithm.

4.Mishra, S., Mishra, D., Santra, G. H. (2016). Applications of machine learning techniques in agricultural crop production: a review paper. Indian J. Sci. Technol,9(38), 1-14.

4. Manjula, E., Djodiltachoumy, S. (2017). A Model for Prediction of Crop Yield. International Journal of Computational Intelligence and Informatics,6(4), 2349-6364.

5. Dahikar, S. S., Rode, S. V. (2014). Agricultural crop yield prediction using artificial neural network approach. International journal of innovative research in electrical, electronics, instrumentation and control engineering,2(1), 683-686.

6. Gonzlez Snchez, A., Frausto Sols, J., Ojeda Bustamante, W. (2014). Predictive ability of machine learning methods for massive crop yield prediction.

7.Mandic, D. P., Chambers, J. (2001). Recurrent neural networks for prediction: learning algorithms, architectures and stability. John Wiley Sons, Inc..

8. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

9.Sak, H., Senior, A., Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association.

10. Liaw, A., Wiener, M. (2002). Classification and regression by random Forest. R news, 2(3), 18-22.