

A MAJOR PROJECT REPORT
ON
FARMING MADE EASY USING MACHINE LEARNING

Submitted to
Sri Indu College Of Engineering & Technology, Hyderabad
In partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Submitted by

Y. POOJA SRI	(19D41A05N5)
S.LAVANYA	(19D41A05J6)
A.RAKESH REDDY	(19D41A05P3)
V.SIDDHARTHA REDDY	(19D41A05M6)

Under the esteemed guidance of

R.MAHENDAR

(Assistant.Proffesor)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution under UGC, Accredited by NBA, Affiliated to JNTUH)
Sheriguda, Ibrahimpatnam (2022-2023)

SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution under UGC, Accredited by NBA, Affiliated to JNTUH)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

Certified that the Major Project entitled **“FARMING MADE EASY USING MACHINE LEARNING”** is a bonafide work carried out by **Y.POOJASRI(19D41A05N5)** , **S.LAVANYA(19D41A05J6)** , **A.RAKESH REDDY(19D41A05P3)** , **V.SIDDHARTHA REDDY(19D41A05M6)** in partial fulfilment for the award of **Bachelor of Technology in Computer Science and Engineering** of SICET, Hyderabad for the academic year **2019-2023**. The Project has been approved as it satisfies academic requirements in respect of the work prescribed for **IV YEAR, II-SEMESTER** of **B.TECH** course.

INTERNAL GUIDE

(R.Mahendar)

HEAD OF THE DEPARTMENT

(Dr.T.Charan singh)

DEAN
(Dr.K.S.Sadasiva Rao)

EXTERNAL EXAMINER



CONSCIENCE TECHNOLOGIES

A Right Platform For All Engineers...

Certificate

This is to certify that

Y.POOJA SRI

19D41A05N5

S.LAVANYA

19D41A05J6

A.RAKESH REDDY

19D41A05P3

V.SIDDHARTHA REDDY

19D41A05M6

Students of final year B.Tech, **Computer Science and Engineering, SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY, HYDERABAD** (*affiliated to J.N.T.University, Hyderabad*), have successfully completed their project titled **“FARMING MADE EASY USING MACHINE LEARNING ”** in PYTHON DIVISION, CONSCIENCE TECHNOLOGIES (CT), Hyderabad.

For CONSCIENCE TECHNOLOGIES

Project Manager

K.V. REDDY

ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We are thankful to Principal **Dr. G. SURESH** garu and Dean CSE & Allied Branches **Dr. K. S. Sadasiva Rao** garu for giving us the permission to carry out this project and for providing necessary infrastructure and labs.

We are highly indebted to **Dr. T. CHARAN SINGH, Professor& Head** , Department of Computer Science Engineering, for providing valuable guidance at every stage of this project.

We are grateful to our internal project guide **R.MAHENDAR, Assistant Professor** for her constant motivation and guidance given by him during the execution of this project work.

We would like to thank the Teaching & Non-Teaching staff of Department of Computer Science and engineering for sharing their knowledge with us, last but not least we express our sincere thanks to everyone who helped directly or indirectly for the completion of this project.

Y.POOJA SRI [19D41A05N5]

S.LAVANYA [19D41A05J6]

A.RAKESH REDDY [19D41A05P3]

V.SIDDHARTHA REDDY [19D41A05M6]

ABSTRACT

Agriculture is the primary mainstay of the economy in our country. In recent years because of uncertain trends in climate and other fluctuations in the price trends, the price of the crop has varied to a larger level. Farmers remain oblivious of these uncertainties, which spoils the crops and causes massive loss. They are unaware of the crop type which would benefit them most. Due to their limited knowledge of different crop diseases and their specific remedies, crops get damaged. This system is handy, easy-to-use. It provides accurate results in predicting the price of the crop. This framework utilizes Machine Learning's Decision Tree Regression Algorithm to predict crop price. The attributes considered for prediction are rainfall, wholesale price index, month, and year. Consequently, the system gives an advance forecast to the farmers' which grows the speed of profit to them and consequently the country's economy. This system also incorporates other modules like weather forecast, crop recommendation, fertilizer recommendation, and shop, chat portal, and guide are also implemented.

CONTENTS

S.NO	CHAPTER NAME	PAGENO.
	LIST OF FIGURES	
	LIST OF SCREENSHOTS	
1.	INTRODUCTION	1
1.1	LITERATURE SURVEY	3
2.	SYSTEM ANALYSIS	5
2.1	EXISTING SYSTEM	5
2.2	PROPOSED SYSTEM	5
2.3	SYSTEM SPECIFICATION	6
3.	SYSTEM STUDY	7
3.1	FEASIBILITY STUDY	7
4.	TECHNOLOGIES USED	9
4.1	MACHINE LEARNING	9
4.1.1	INTRODUCTION TO MACHINE LEARNING	9
4.1.2	GOALS OF MACHINE LEARNING	9
4.1.3	APPLICATIONS OF MACHINE LEARNING	10
5.	MODULES	11
5.1	NEW FARMER SIGN UP	11
5.2	FARMER LOGIN	11
5.3	ADMIN LOGIN	11

6.SYSTEM DESIGN	13
6.1 ARCHITECTURE	13
6.2 UML DIAGRAMS	14
6.2.1 USE CASE DIAGRAM	15
6.2.2 CLASS DIAGRAM	16
6.2.3 ACTIVITY DIAGRAM	17
6.2.4 SEQUENCE DIAGRAM	18
7.IMPLEMENTATION	19
SOFTWARE ENVIRONMENT	19
PYTHON	19
MACHINE LEARNING	24
SAMPLE CODE	43
8.OUTPUT SCREENS	56
9.TESTING	64
10.CONCLUSION	67
11.REFERENCES	68

LIST OF FIGURES

Fig No	Name	Page No
Fig.1	Architecture diagram	13
Fig.2	Use case diagram-user	15
Fig.3	Class diagram-user	16
Fig.4	Activity diagram-user	17
Fig.5	Sequence diagram-user	18

LIST OF SCREEN SHOTS

Fig No	Name	Page No
Fig.1	open browser and enter url	56
Fig.2	Admin Login	56
Fig.3	After login Admin login screen	57
Fig.4	Add government schemes	57
Fig.5	Login schemes with dates	58
Fig.6	View schemes details	58
Fig.7	Farmer sign up	59
Fig.8	Farmer login	59
Fig.9	View login	60
Fig.10	View schemes	60
Fig.11	Predict crop prices	61
Fig.12	Select crop	61
Fig.13	Graph of predicted and original price	62
Fig.14	Prediction screen	62

1. INTRODUCTION

India being a rural nation, its economy transcendently relies upon agricultural yield development and unified agroindustry items. It is currently quickly advancing towards a specialized turn of events. India now is rapidly progressing towards technical development. Smart farming is changing the face of agriculture in India. Technology can provide a solution to most challenges farmers face. It can help them predict weather more accurately, decrease waste, boost output and increase their profit margins. In the status quo, the farmers and the consumers find it difficult in the real world to determine the accurate prices of crops without having prior knowledge of the fluctuating trend prices or weather conditions. Accordingly, innovation will end up being helpful to agriculture. The paper aims to predict crop prices in advance. This work is based on finding proper regional datasets that help us in achieving high accuracy and better performance. Our system, Agro-Genius, is using Machine Learning to build the Price Predicting Model.

In the past few years, a lot of fluctuation in the prices of the crop has been seen. This has increased the rate of crop damage produced each year. The main aim of this prediction system is to ensure that the farmers get a better idea about their yield and deal with the value risk.

Weather is also highly unpredictable these days. It also affects the crop production. The proposed system will also forecast the weather helping the farmer make correct decisions regarding field ploughing, field harvesting etc. Similarly, fertilizers play an important role. Fertilizers load the soil with the required nutrients that the crops eliminate from the soil. Crop yields and production will be fundamentally decreased if fertilizers are not used. That is the reason fertilizers are utilized to enhance the soil's supplement stocks with minerals that can be immediately assimilated and utilized by crops. Our system will provide fertilizer consumption based on different crops and provide a portal to buy the fertilizers and seeds from the user's location. They can even get the exact location along with the address of the fertilizer and seed shop. The provided fertilizers will get more profit to the farmers on the growing system suggested crop. It will also show the best suited crop based on cultivation date and month and location details, thereby maximizing the yield.

It will provide multilingual and region specific guide books for the farmers. Any farmer who is new to this field and who wishes to gain information from his ancestors but having the same methods documented

will be highly beneficial. We have also provided maps for the farmers to gain knowledge. Our system will provide two different types of maps for the farmer to gain the knowledge about how the land and where they should start their farming. Irrigation maps show the irrigated-non irrigated area over the country. Agriculture land view map will provide an overview of agricultural land present in various states of India and help farmers to analyze the non Agricultural land which can further be improved. Maps make the farmers easy to understand they have to just hover on the state they are thinking of starting their farming and they will get the information about that state and they can decide whether they should change the place or should start farming. If the farmers are new in this field it is the best thing for them as the most important thing in farming is to firstly choose the land and place of farming.

Moving in the same direction, our system will incorporate a chat application which helps in information sharing. Often farmers have certain queries which cannot be solved due to their limited knowledge, hence we are building a platform where information can be exchanged. Language can pose as a barrier to the users. Since the majority of non- English speaking farm workers in India are native Hindispeakers, we anticipate that once these resources are developed they might be translated to other languages as well. Hence, to make the website user friendly, we have provided language translation. Farmers should know about their location, date of cultivation of their crop. Our system is a web application, which is developed based on machine learning concepts. The proposed system applies machine learning and prediction algorithms like Naive Bayes, Decision Trees and K-Nearest Neighbour to identify the most accurate model and then process it. This in turn will help predict the price of the crop.

1.1 LITERATURE SURVEY

The following papers focused on predicting crop price using Machine Learning and providing results. In April 2019, the exploration targets foreseeing both the cost and benefit of the given harvest before planting. The preparing datasets so acquired give enough bits of knowledge to foresee the suitable cost and request in the business sectors[1]. The authors have predicted the most profitable crops and its expected price during harvesting time according to the location, by predicting different historical raw datasets using different machine learning algorithms. The work shown by Nishiba [2] is the expected utilization of data mining procedures in foreseeing the harvest yield dependent on the input parameters average rainfall and area of the field. The easy-to-use website page created for anticipating crop yield can be utilized by any client by giving the normal precipitation and region of that place. Different Data Mining techniques are applied to different datasets. This paper can also include certain modules [11] which can help farmers to make certain decisions based on the harvested area or current trends in the market. The system can be extended by visualizing the crop details in a map with details, which will help farmers to view the nearby district cultivation details. Proposed system can be enhanced by providing a graphical visualization of predicted prices for better understanding.

This system is proposed to provide help to the farmers for expecting the best amount for their crops and for predicting the best price for the crops. This also helps the farmers to check previous prices of different commodities. The system can predict crops using [9] Random forest, Polynomial Regression and Decision Tree algorithms. The best crop and its required fertilizers make the farmer more confident about the crop and its yield and also our system will do marketing work [4] by estimating total value of the crop based on current market price. The idea of the system can be extended by adding some extra features to the system like providing a nearby shop location portal for purchasing seeds and fertilizers.

These papers aim at predicting the price and forecast through web application and it runs on efficient machine learning algorithms like using an Autoregressive Integrated Moving Average (ARIMA) model, Traditional ARIMA [6], Support Vector Regression Algorithm[8], and technologies having a general easy to use interface to the clients. The training datasets [7] acquired give sufficient bits of knowledge to foreseeing the appropriate price [10] and request in the markets. The results are displayed as web applications in order that poor farmers can access easily. Models can be improved by integrating this with other departments like horticulture, sericulture, and others towards the agricultural development

of our country. Different agriculture departments have various problems in the current time. Incorporating them will not only increase the scope but also help the farmers new to this part of the spectrum. Their work may be expanded by building a framework for suggesting agriculture produce and dispersion for farmers. Utilizing this framework, We ought to get the same accuracy indeed when an information autonomous framework is utilized. Further, can be enhanced by making an android application for the same.

2.SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

We have used Python for basic programming in all modules. Flask is used for hosting. Socket Programming is used for a chat application. Chart.js is used for visualizing the maps. JavaScript is used for validation purposes.

For Weather Forecast [12] and fertilizer shop location, we have used APIs. Using the self-made dataset and concept of linear regression in machine learning we have implemented a Crop recommendation model so that a farmer can learn about the best suited crop for a particular region. In Fertilizer Recommendation we have used a dataset for predicting which fertilizer should be used for the disease present on crops. Socket programming is used for farmers interaction using provided chat application [3]. Google API is used for providing a multilingual website for ease to read.

2.2 PROPOSED SYSTEM:

In this paper author is using various machine learning algorithms such as Random Forest, Decision Tree and KNN to predict crop prices. All this algorithms get train on Crop Prices dataset which contains crop details weather details such as Rainfall and below screen showing dataset details with crop name, market name with prices and Rainfall.

2.3 SYSTEM SPECIFICATION:

HARDWARE REQUIREMENTS:

- System : MINIMUM i3.
- Hard Disk : 40 GB.
- RAM : 4 GB.

SOFTWARE REQUIREMENTS:

- **Operating System:** Windows 8
- **Coding Language:** Python 3.7

3.SYSTEM STUDY

3.1 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.

4. TECHNOLOGIES USED

4.1 MACHINE LEARNING:

4.1.1 INTRODUCTION TO 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.

4.1.2 CHALLENGES IN MACHINE 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 overfitting & underfitting – If the model is overfitting or underfitting, 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.

4.1.3 APPLICATIONS OF MACHINE 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

5.MODULES

5.1 NEW FARMER SIGN UP

Using this module farmers can signup with application

5.2 FARMER LOGIN

Farmer can login to application by using username and password given at signup time and then farmer select crop name to get its predicted prices in different market. Farmer can view all schemes details launched from the government

5.3 ADMIN LOGIN

Admin can login to application by using 'admin' as username and password and then can add new schemes details.

6.SYSTEM DESIGN

6.1 ARCHITECTURE

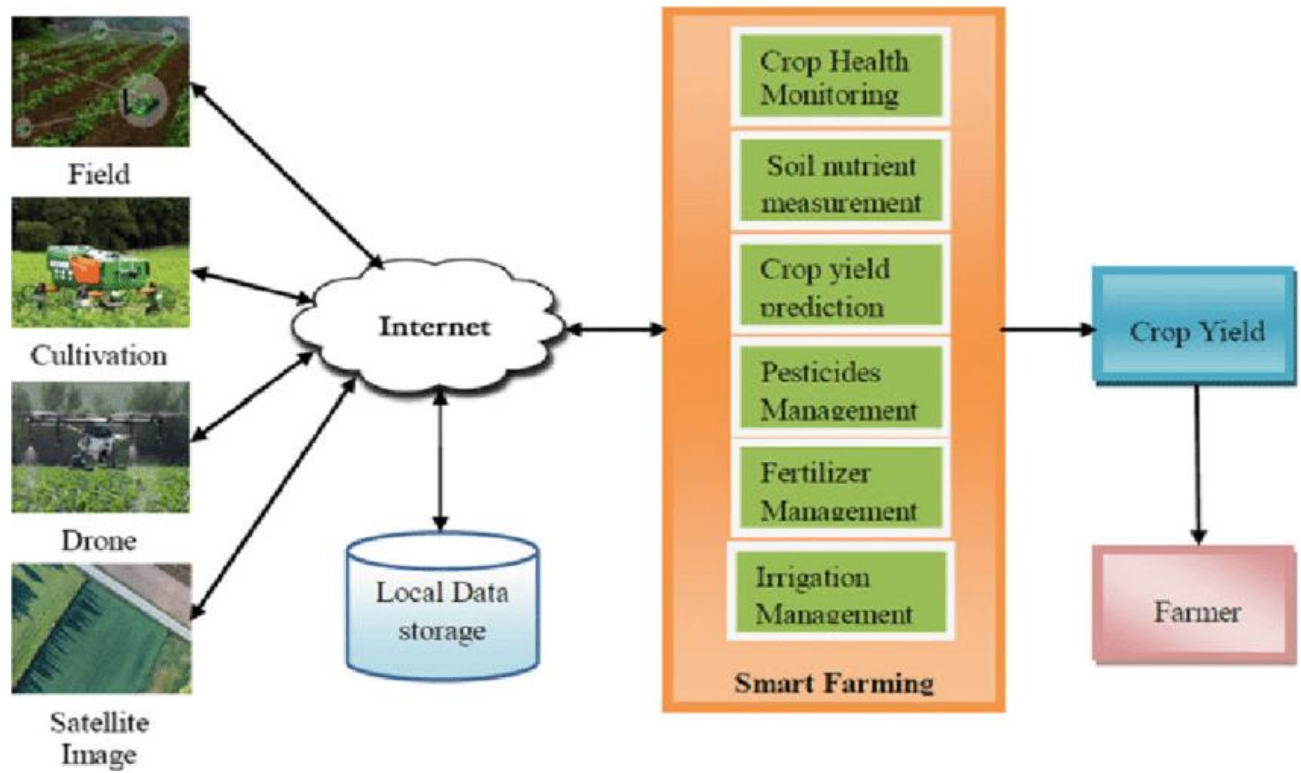


Fig1:Architecture diagram

6.2 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.

6.2.1 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 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.

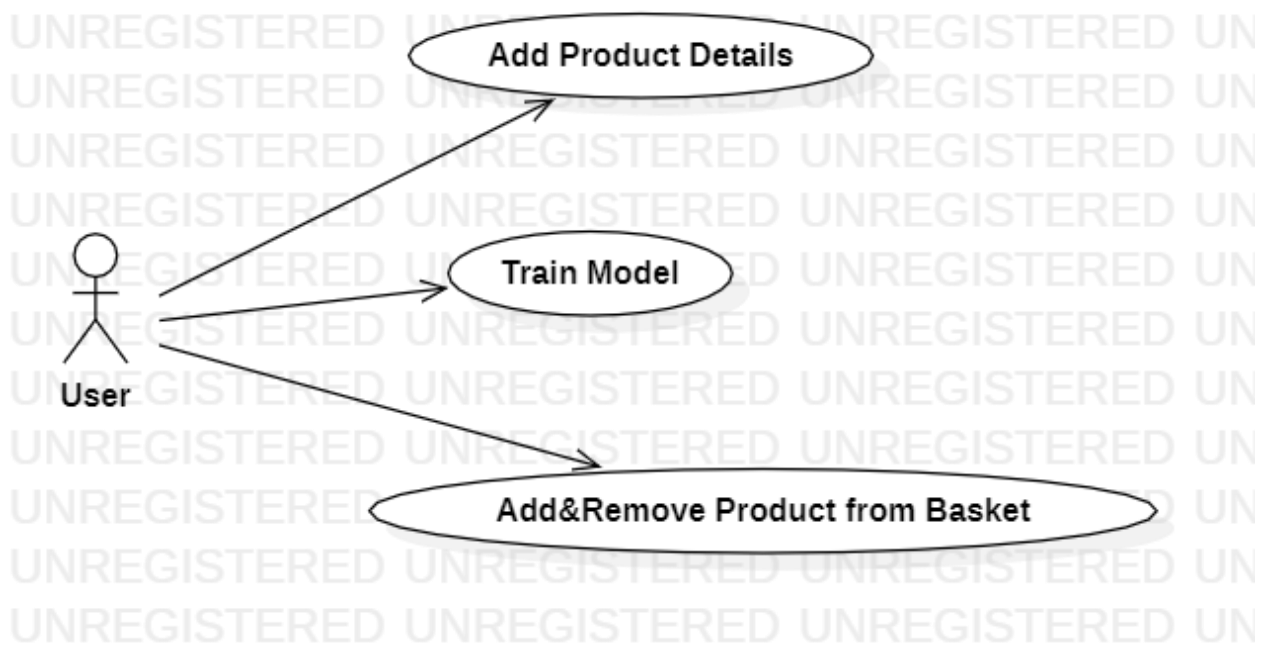


Fig2:Use Case Diagram

6.2.2 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.

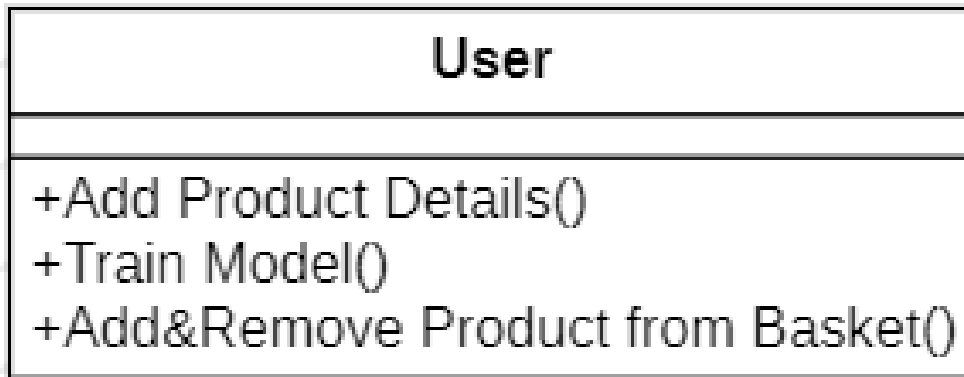


Fig3:Class Diagram

6.2.3 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.

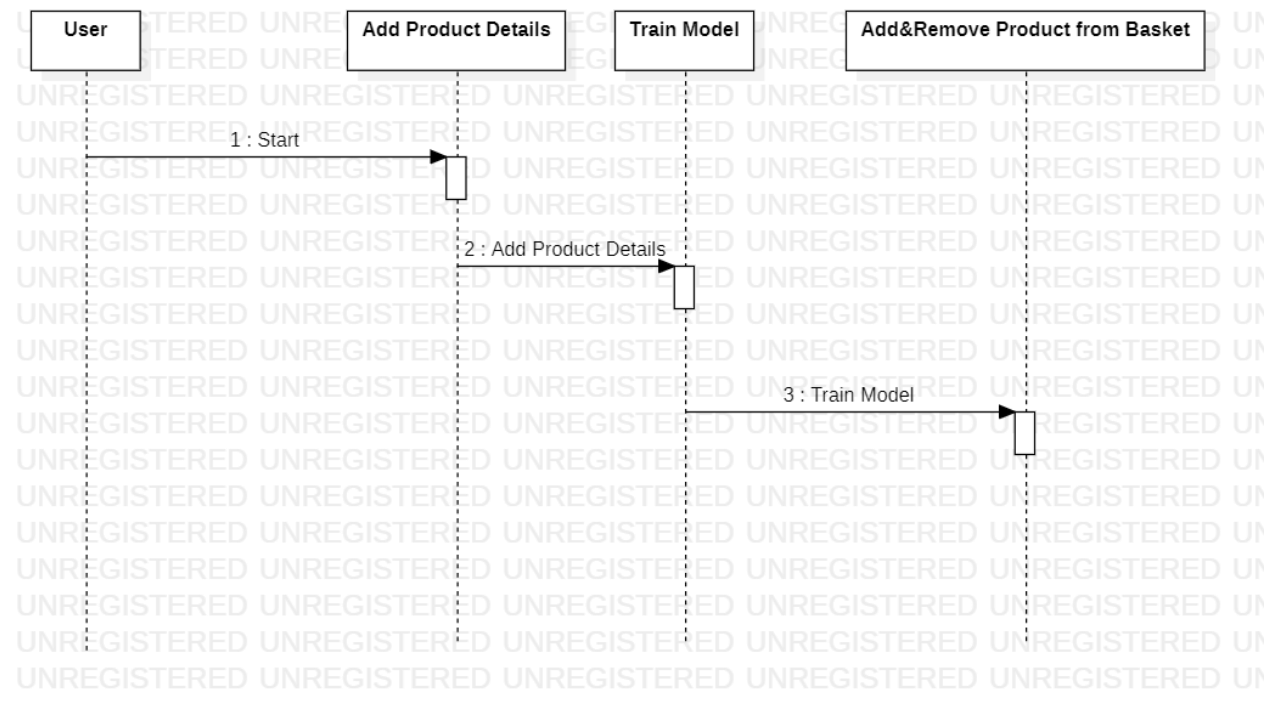


Fig4:Sequence Diagram

6.2.4 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.

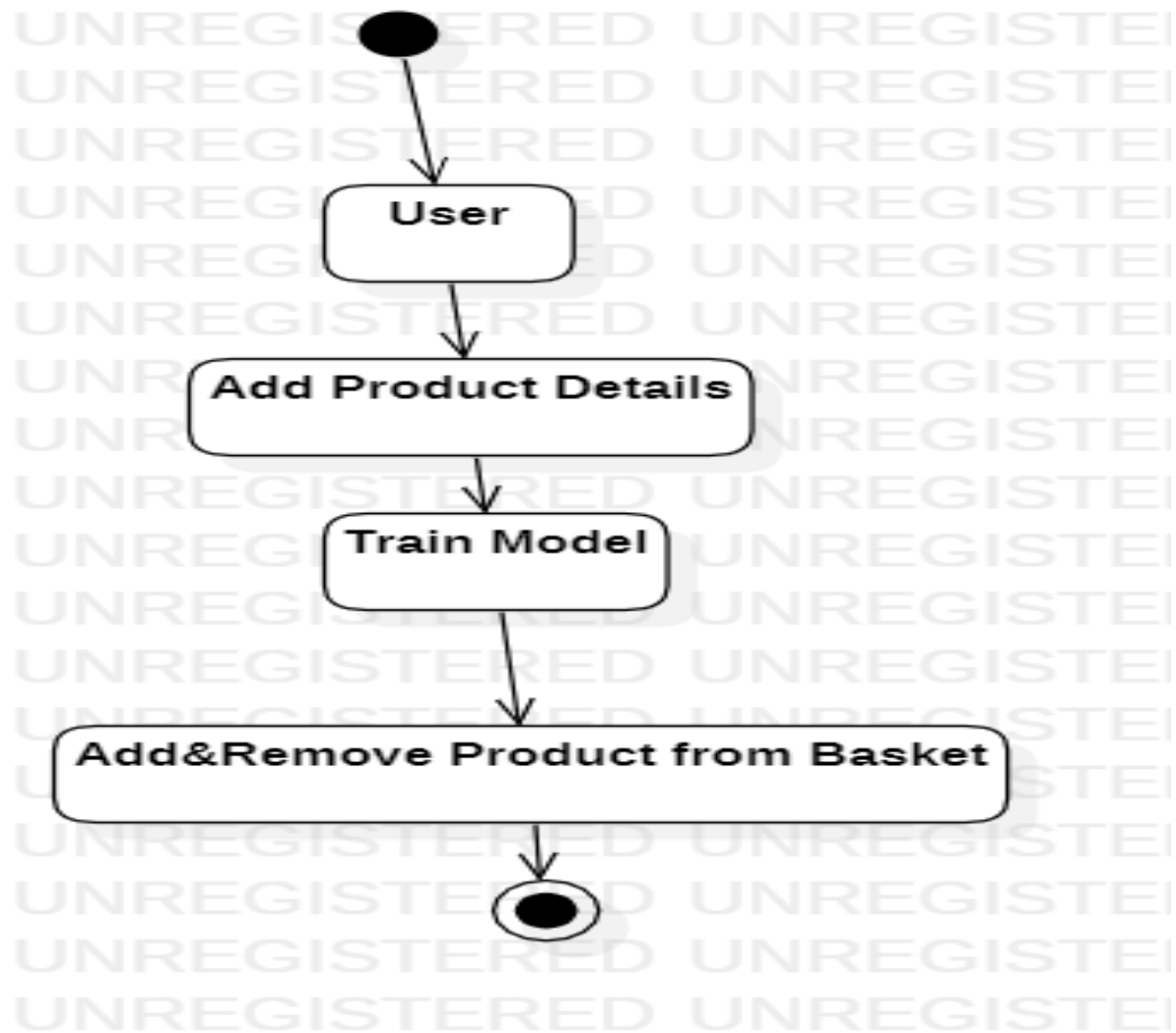


Fig5:Activity Diagram

7.IMPLEMENTATION

SOFTWARE ENVIRONMENT

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

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, BeautifulSoup, 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** 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.

Any doubts till now in the advantages of Python? Mention in the comment section.

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.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

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**, 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 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 **Carbannelle**.

The reason it is not so famous despite the existence of Brython 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."

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 programming 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 overfitting & underfitting – If the model is overfitting or underfitting, 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

Recommendation of products to customer in online shopping

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, 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 maths 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! 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 [Keras](#), [TensorFlow](#), [Scikit-learn](#), etc.

So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on GeeksforGeeks.

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** 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 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 and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 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 IPython shells, the Jupyter 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 and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. 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**

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.

Install Python Step-by-Step in Windows and Mac :

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace. The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](#) The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: <https://www.python.org>



Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.



Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 3.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b3db4ae77b9ab01b0f09be	23017663	SiG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SiG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a4c2c8a8cee08e6	34898416	SiG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SiG
Windows help file	Windows		063999573a2c56b2ac56cade68477cd2	8131761	SiG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9800c3cf6d9ec0b0abe83184a0728a2	7504391	SiG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76d4bdc3043a383e563e00	26880348	SiG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28c91c6088bd73ae0e53a3bd351b4bd2	1362904	SiG
Windows x86 embeddable zip file	Windows		9fab3bd19841879fda9413574139d8	6741626	SiG
Windows x86 executable installer	Windows		33cc602942a54446a3d8451476394789	25663848	SiG
Windows x86 web-based installer	Windows		1b670cfa5d3117d882c30983ea371d87c	1324608	SiG

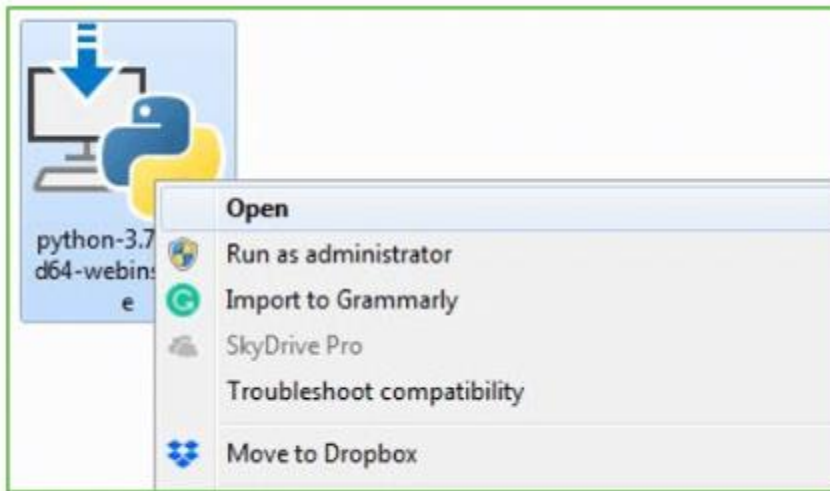
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



Step 3: Click on Install NOW After the installation is successful. Click on Close



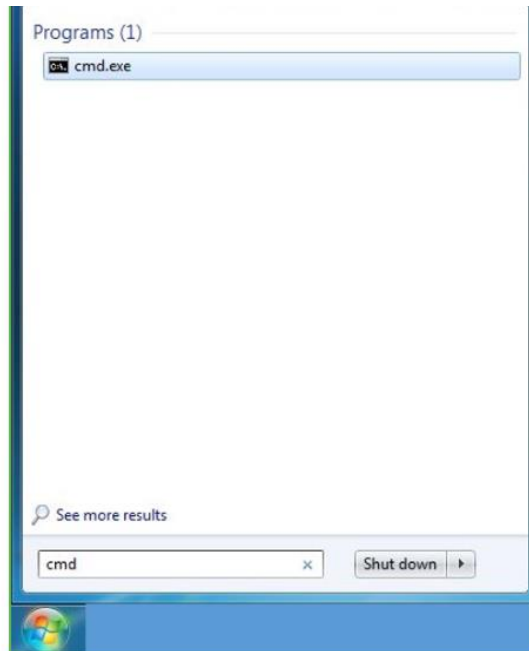
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python -V** and press Enter.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\DELL>python -V
Python 3.7.4
C:\Users\DELL>_
```

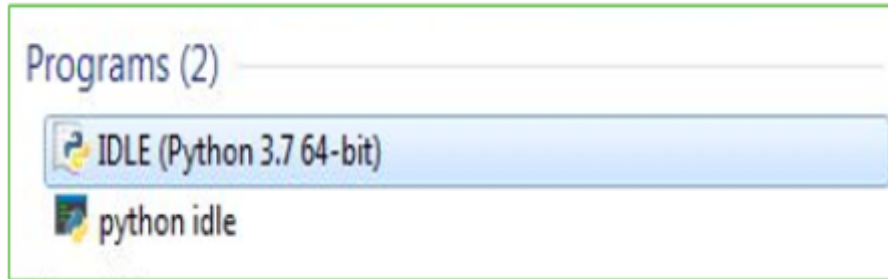
Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

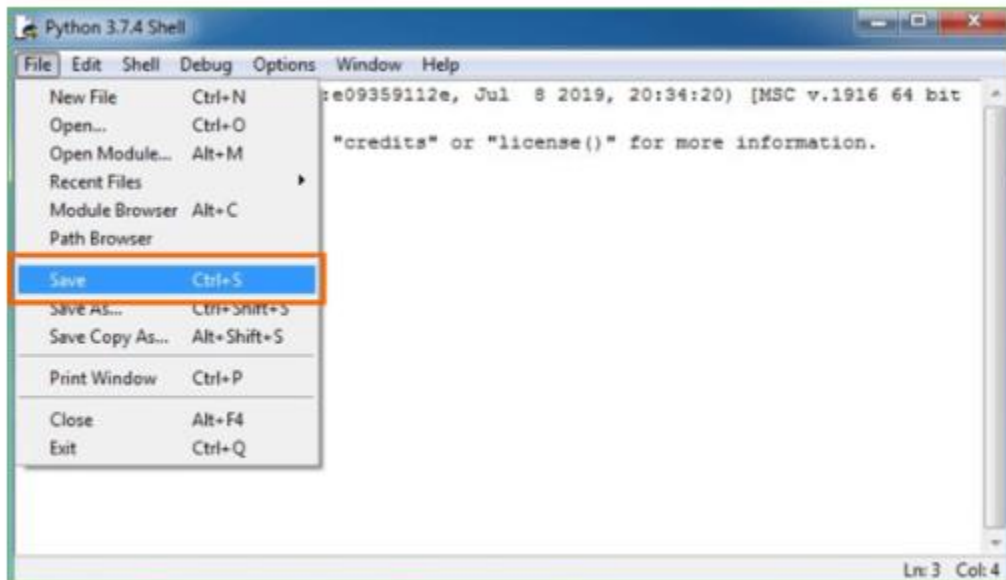
Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.



Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. **enter print**

SAMPLE CODE

urls.py

```
from django.urls import path

from . import views

urlpatterns = [path("index.html", views.index, name="index"),
                path("AdminLogin.html", views.AdminLogin, name="AdminLogin"),
                path("AdminLoginAction", views.AdminLoginAction,
name="AdminLoginAction"),
                path("FarmerLogin.html", views.FarmerLogin, name="FarmerLogin"),
                path("FarmerLoginAction", views.FarmerLoginAction,
name="FarmerLoginAction"),
                path("Signup.html", views.Signup, name="Signup"),
                path("SignupAction", views.SignupAction, name="SignupAction"),
                path("AddScheme.html", views.AddScheme, name="AddScheme"),
                path("AddSchemeAction", views.AddSchemeAction,
name="AddSchemeAction"),
                path("PredictCropPrices.html", views.PredictCropPrices,
name="PredictCropPrices"),
                path("PredictCropPricesAction", views.PredictCropPricesAction,
name="PredictCropPricesAction"),
                path("ViewSchemes", views.ViewSchemes, name="ViewSchemes"),
]
```

Views.py

```
from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
from django.http import HttpResponse
import os
from django.core.files.storage import FileSystemStorage
```

```

import pymysql
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor

global uname

le1 = LabelEncoder()

dataset = pd.read_csv("Dataset/CropDataset.csv",encoding='iso-8859-1',usecols=['variety','max_price','Rainfall'])
dataset.fillna(0, inplace = True)

def PredictCropPricesAction(request):
    if request.method == 'POST':
        item = request.POST.get('item', False)
        dataset1 = pd.read_csv("Dataset/CropDataset.csv",encoding='iso-8859-1',usecols=['variety','max_price','Rainfall','district'])
        dataset1.fillna(0, inplace = True)
        df = dataset1.loc[(dataset1['variety'] == item)]
        print(df)
        Y = df.values[:,2:3]
        district = df.values[:,0:1].ravel()
        df.drop(['max_price'], axis = 1,inplace=True)
        df.drop(['district'], axis = 1,inplace=True)

```

```

df['variety'] = pd.Series(le1.fit_transform(df['variety'].astype(str)))
df.fillna(0, inplace = True)
print(Y)
X = df.values
sc = MinMaxScaler(feature_range = (0, 1))
X = sc.fit_transform(X)
Y = sc.fit_transform(Y)
X_train = X
Y_train = Y
X_test = X
Y_test = Y

dt_regression = DecisionTreeRegressor()
dt_regression.fit(X_train, Y_train.ravel())
predict = dt_regression.predict(X_test)
dt_mse = mean_squared_error(Y_test.ravel(),predict.ravel())
dt_accuracy = 1.0 - dt_mse

knn_regression = KNeighborsRegressor(n_neighbors=2)
knn_regression.fit(X_train, Y_train.ravel())
predict = knn_regression.predict(X_test)
knn_mse = mean_squared_error(Y_test.ravel(),predict.ravel())
knn_accuracy = 1.0 - knn_mse

rf_regression = RandomForestRegressor()
rf_regression.fit(X_train, Y_train.ravel())
predict = rf_regression.predict(X_test)
rf_mse = mean_squared_error(Y_test.ravel(),predict.ravel())
rf_accuracy = 1.0 - rf_mse
predict1 = predict.reshape(predict.shape[0],1)
predict1 = sc.inverse_transform(predict1)

```

```

predict1 = predict1.ravel()
labels = sc.inverse_transform(Y_test)
labels = labels.ravel()

output = '<table border=1><tr><th><font size="" color="black">District Market</th><th><font
size="" color="black">Crop Name</th><th><font size="" color="black">Original Price</th>'
output += '<th><font size="" color="black">Predicted Price</th></tr>'
for i in range(len(predict1)):
    output += '<tr><td><font size="" color="black">'+district[i]+'</td>'
    output += '<td><font size="" color="black">'+item+'</td>'
    output += '<td><font size="" color="black">'+str(labels[i])+</td>'
    output += '<td><font size="" color="black">'+str(predict1[i])+</td></tr>'
output += '<tr><td><font size="" color="black">Random Forest Accuracy</td>'
output += '<td><font size="" color="black">'+str(rf_accuracy)+'</td></tr>'
output += '<tr><td><font size="" color="black">Decision Tree Accuracy</td>'
output += '<td><font size="" color="black">'+str(dt_accuracy)+'</td></tr>'
output += '<tr><td><font size="" color="black">KNN Accuracy</td>'
output += '<td><font size="" color="black">'+str(knn_accuracy)+'</td></tr>'
context= {'data':output}
print(output)
plt.plot(Y_test.ravel(), color = 'red', label = 'Original Price')
plt.plot(predict.ravel(), color = 'green', label = 'Predicted Price')
plt.title('Crop Price Forecasting')
plt.xlabel('Current Price for Crop '+item)
plt.ylabel('Predicted Price for Crop '+item)
plt.legend()
plt.show()
return render(request, 'ViewPrices.html', context)

```

```

def PredictCropPrices(request):
    if request.method == 'GET':
        variety = np.unique(dataset['variety'])

```

```

        output = '<tr><td><font size="'
color="black">Choose Crop Name</font></td><td><select name=item>
        for i in range(len(variety)):
            output += '<option value="'+str(variety[i])+'">'+str(variety[i])+'</option>'
        output += "</select></td></tr>"
        context= {'data1':output}
        return render(request, 'PredictCropPrices.html', context)
def index(request):
    if request.method == 'GET':
        return render(request, 'index.html', { })
def AdminLogin(request):
    if request.method == 'GET':
        return render(request, 'AdminLogin.html', { })
def FarmerLogin(request):
    if request.method == 'GET':
        return render(request, 'FarmerLogin.html', { })
def Signup(request):
    if request.method == 'GET':
        return render(request, 'Signup.html', { })
def getOutput(table,length):
    font = '<font size=" color=black>'
    output=""
    con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database =
'cropinfo',charset='utf8')
    with con:
        cur = con.cursor()
        cur.execute("select * from "+table)
        rows = cur.fetchall()
        for row in rows:
            output+="

```

```

        for i in range(0,length):
            output+="<td><font size=" color=black>"+font+row[i]+"</td>"
        return output

def ViewSchemes(request):
    if request.method == 'GET':
        output = '<table border=1 align=center width=100%>'
        font = '<font size=" color=black>'
        arr = ['Scheme ID','Scheme Name','Scheme Description','Required Documents','Scheme Launch
Date','Scheme End Date']
        output += "<tr>"
        for i in range(len(arr)):
            output += "<th>"+font+arr[i]+"</th>"
        output += getOutput("addscheme",len(arr))
        context= {'data':output}
        return render(request, 'ViewSchemes.html', context)

def AdminLoginAction(request):
    global uname
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        if username == 'admin' and password == 'admin':
            uname = username
            context= {'data':'welcome '+username}
            return render(request, 'AdminScreen.html', context)
        else:
            context= {'data':'login failed'}
            return render(request, 'ExpertLogin.html', context)

    def FarmerLoginAction(request):
        global uname
        if request.method == 'POST':

```

```

username = request.POST.get('t1', False)
password = request.POST.get('t2', False)
index = 0

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database =
'cropinfo',charset='utf8')

with con:
    cur = con.cursor()
    cur.execute("select username,password FROM signup")
    rows = cur.fetchall()
    for row in rows:
        if row[0] == username and password == row[1]:
            uname = username
            index = 1
            break
    if index == 1:
        context= {'data':'welcome '+uname}
        return render(request, 'FarmerScreen.html', context)
    else:
        context= {'data':'login failed'}
        return render(request, 'FarmerLogin.html', context)

def SignupAction(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        contact = request.POST.get('t3', False)
        gender = request.POST.get('t4', False)
        email = request.POST.get('t5', False)
        address = request.POST.get('t6', False)
        output = "none"

        con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database =
'cropinfo',charset='utf8')

```

```

with con:
    cur = con.cursor()
    cur.execute("select username FROM signup")
    rows = cur.fetchall()
    for row in rows:
        if row[0] == username:
            output = username+" Username already exists"
            break
    if output == 'none':
        db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root',
database = 'cropinfo',charset='utf8')
        db_cursor = db_connection.cursor()
        student_sql_query = "INSERT INTO
signup(username,password,contact_no,gender,email,address)
VALUES('"+username+"','"+password+"','"+contact+"','"+gender+"','"+email+"','"+address+"')"
        db_cursor.execute(student_sql_query)
        db_connection.commit()
        print(db_cursor.rowcount, "Record Inserted")
        if db_cursor.rowcount == 1:
            output = 'Signup Process Completed'
            context= {'data':output}
            return render(request, 'Signup.html', context)
def AddScheme(request):
    if request.method == 'GET':
        return render(request, 'AddScheme.html', {})
def AddSchemeAction(request):
    if request.method == 'POST':
        sid = request.POST.get('t1', False)
        name = request.POST.get('t2', False)
        desc = request.POST.get('t3', False)
        document = request.POST.get('t4', False)

```

```

start = request.POST.get('t5', False)
end = request.POST.get('t6', False)
    output = "none"
    db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root',
database = 'cropinfo',charset='utf8')
    db_cursor = db_connection.cursor()
    student_sql_query = "INSERT INTO
addscheme(scheme_id,scheme_name,description,document,start_date,end_date)
VALUES('"+sid+"','"+name+"','"+desc+"','"+document+"','"+start+"','"+end+"')"
    db_cursor.execute(student_sql_query)
    db_connection.commit()
    print(db_cursor.rowcount, "Record Inserted")
    if db_cursor.rowcount == 1:
        output = 'New Scheme details added'
    context= {'data':output}
    return render(request, 'AddScheme.html', context)

```

manage.py

```

#!/usr/bin/env python
import os
import sys
if __name__ == '__main__':
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'CropPrice.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc

```

```
execute_from_command_line(sys.argv)
```

test.py

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt

le1 = LabelEncoder()
le2 = LabelEncoder()

dataset = pd.read_csv("Dataset/CropDataset.csv",usecols=['variety','max_price','Rainfall'])
dataset.fillna(0, inplace = True)
df = dataset.loc[(dataset['variety'] == 'Coriander Seed')]
Y = df.values[:,1:2]
df.drop(['max_price'], axis = 1,inplace=True)
df['variety'] = pd.Series(le1.fit_transform(df['variety'].astype(str)))
df.fillna(0, inplace = True)
X = df.values
sc = MinMaxScaler(feature_range = (0, 1))
X = sc.fit_transform(X)
Y = sc.fit_transform(Y)
print(X.shape)
X_train = X
Y_train = Y
X_test = X
Y_test = Y
svr_regression = RandomForestRegressor()
#training SVR with X and Y data
```

```

svr_regression.fit(X_train, Y_train.ravel())
#performing prediction on test data
predict = svr_regression.predict(X_test)
svm_mse = mean_squared_error(Y_test.ravel(),predict.ravel())
print(svm_mse)
#plotting comparison graph between original values and predicted values
plt.plot(Y_test.ravel(), color = 'red', label = 'Observed Crime Location')
plt.plot(predict.ravel(), color = 'green', label = 'Predicted Crime Location')
plt.title('SVM Crime Location Forecasting')
plt.xlabel('Original Observed Crimes Locations')
plt.ylabel('Forecasting Crimes Locations')
plt.legend()
plt.show()
print(predict)
predict = predict.reshape(predict.shape[0],1)
predict = sc.inverse_transform(predict)
predict = predict.ravel()
labels = sc.inverse_transform(Y_test)
labels = labels.ravel()
for i in range(len(labels)):
    print(str(predict[i])+" "+str(labels[i]))

```

Adminscreen.html

```

    { % load static % }
<html>
<head>
<title>Farming</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="{ % static 'style.css' % }" rel="stylesheet" type="text/css" />
</head>
<body>

```

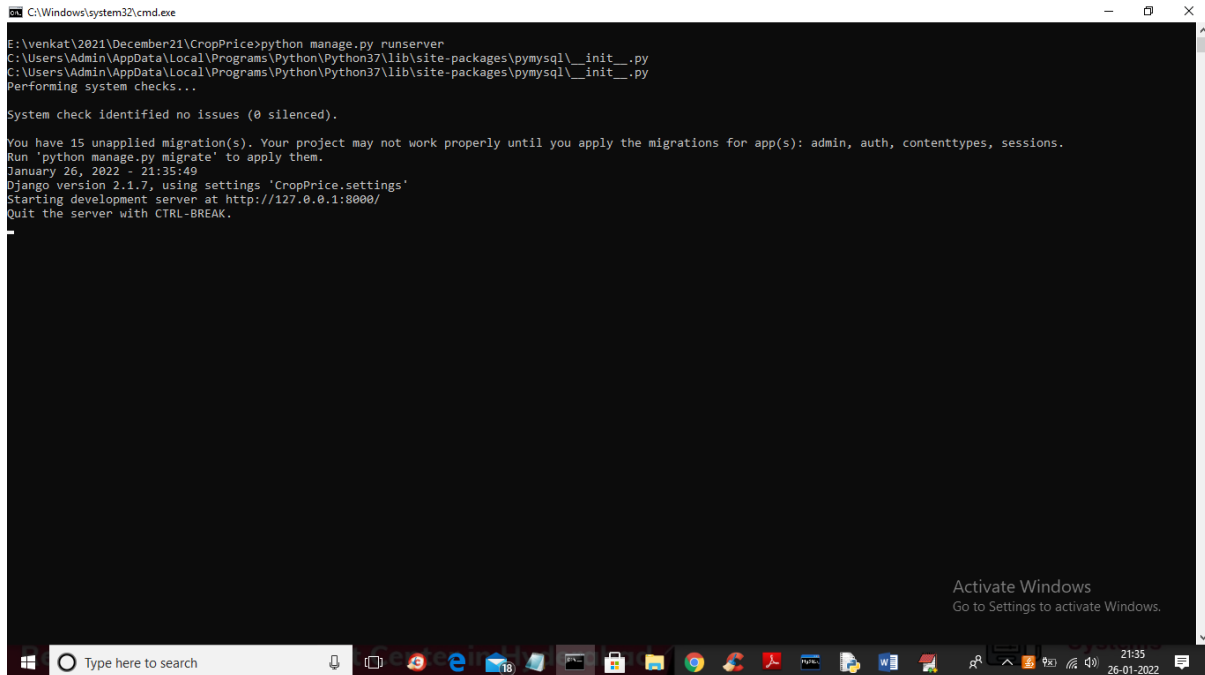
```

<div class="main">
  <div class="main_resize">
    <div class="header">
      <div class="logo">
        <h1><span>Farming Made Easy using Machine Learning</span><small></small></h1>
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
          <ul>
            <ul>
              <li><a href="{ % url 'AddScheme' % }">Add Government Schemes</a></li>
              <li><a href="{ % url 'index' % }">Logout</a></li>
            </ul>
          </ul>
        </div>
        <div class="hbg"></div>

        <font size="3" color="red"><center>{ { data|safe } }</center></font>
      </body>
    </html>

```

8.OUT PUT SCREENS



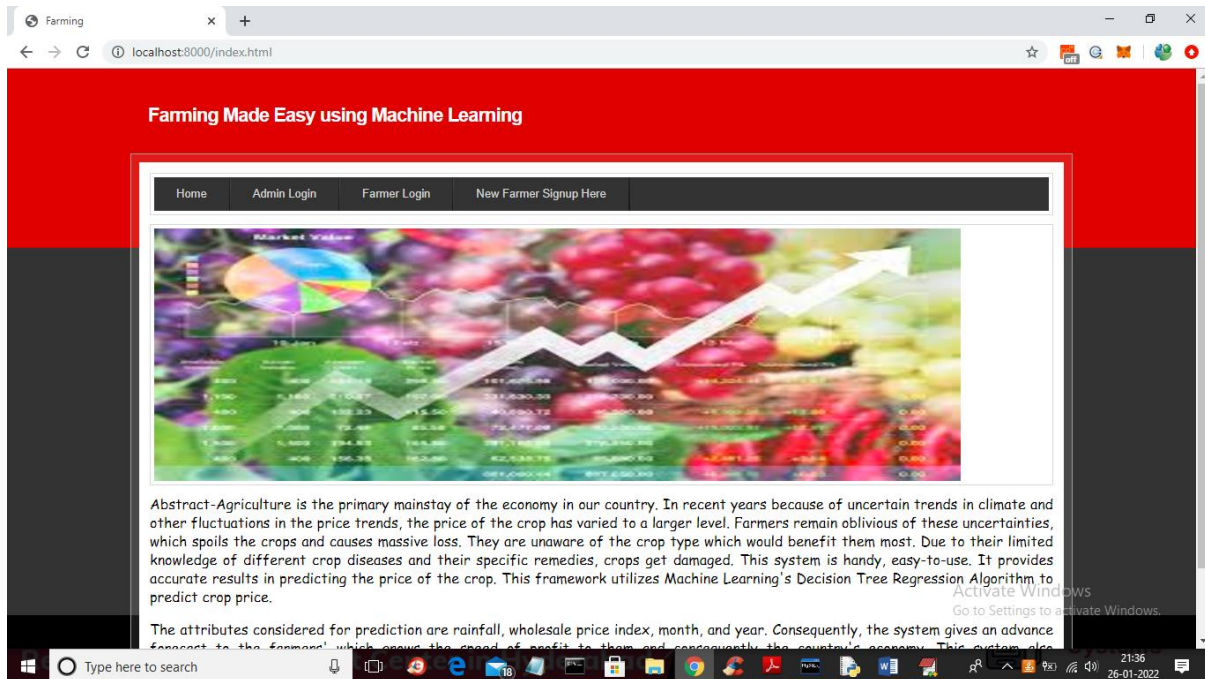
```
C:\Windows\system32\cmd.exe

E:\venkat\2021\December21\CropPrice>python manage.py runserver
C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
Performing system checks...

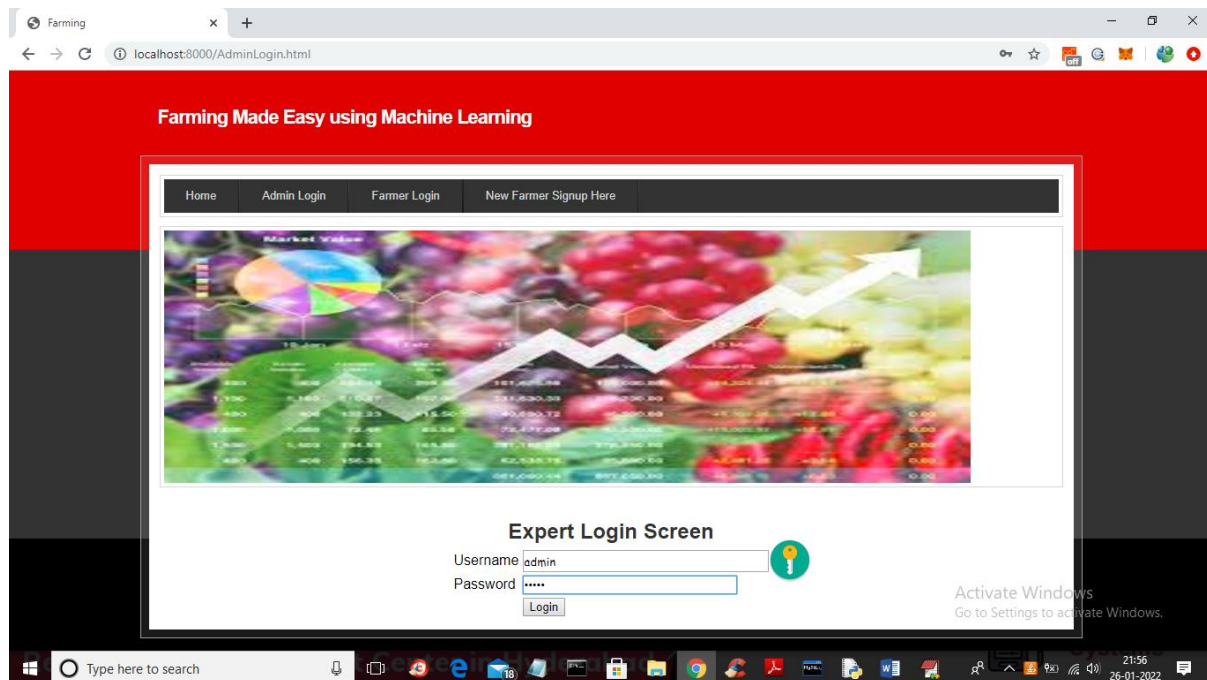
System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
January 26, 2022 - 21:35:49
Django version 2.1.7, using settings 'CropPrice.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

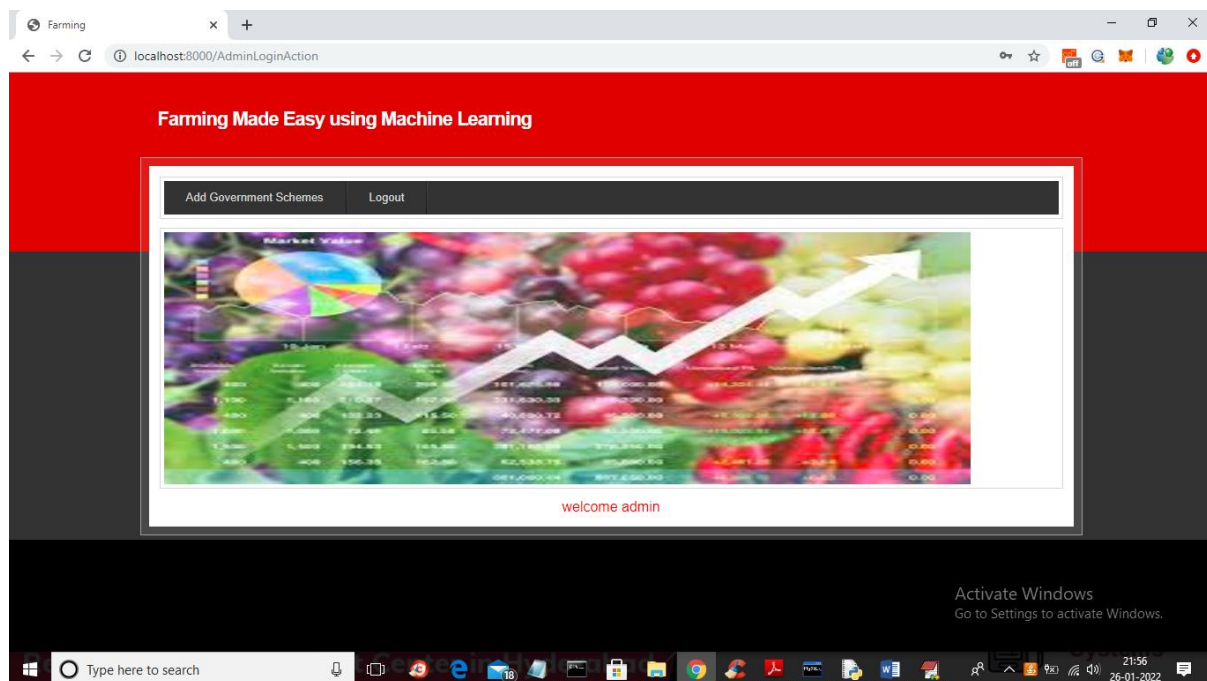
1. In above screen server started and now open browser and enter URL



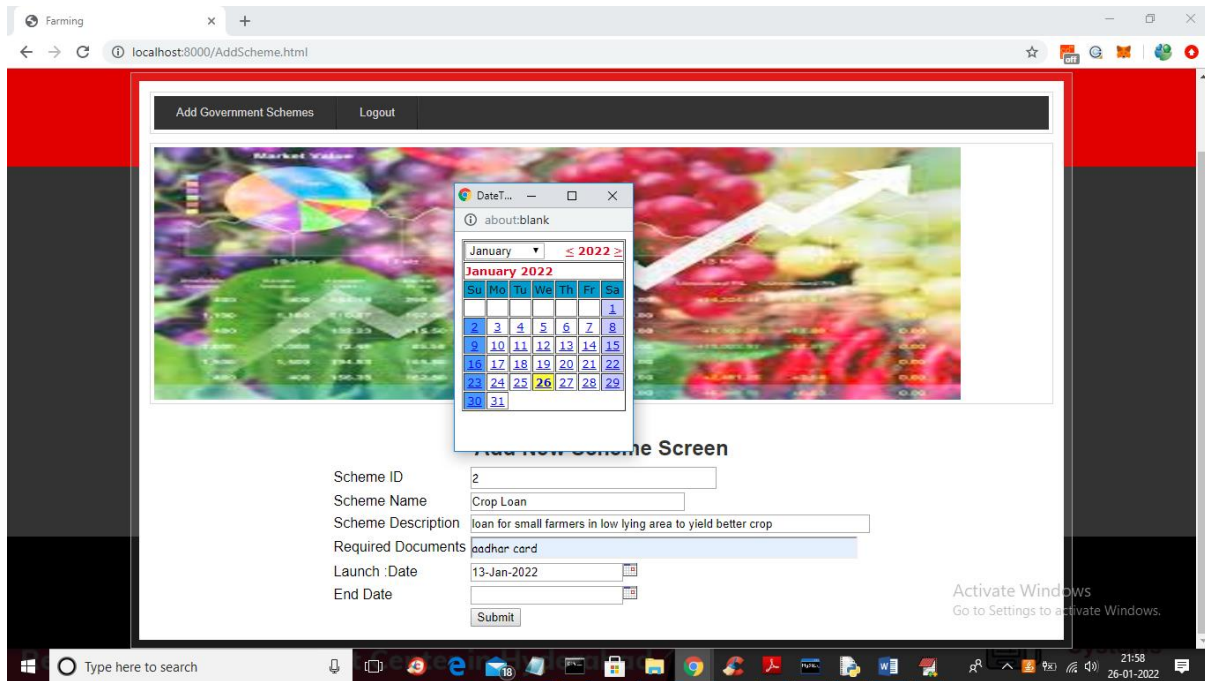
2. In above screen click on 'Admin Login' link to get below login screen



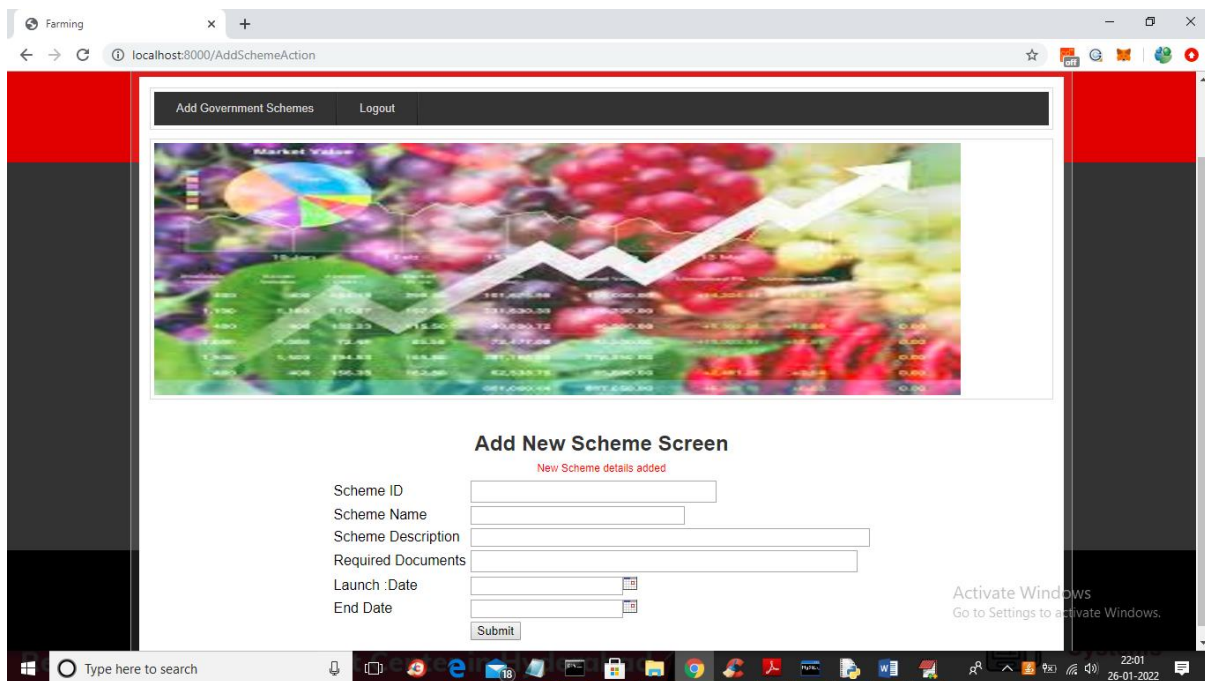
3. In above screen admin is login and after login will get below screen



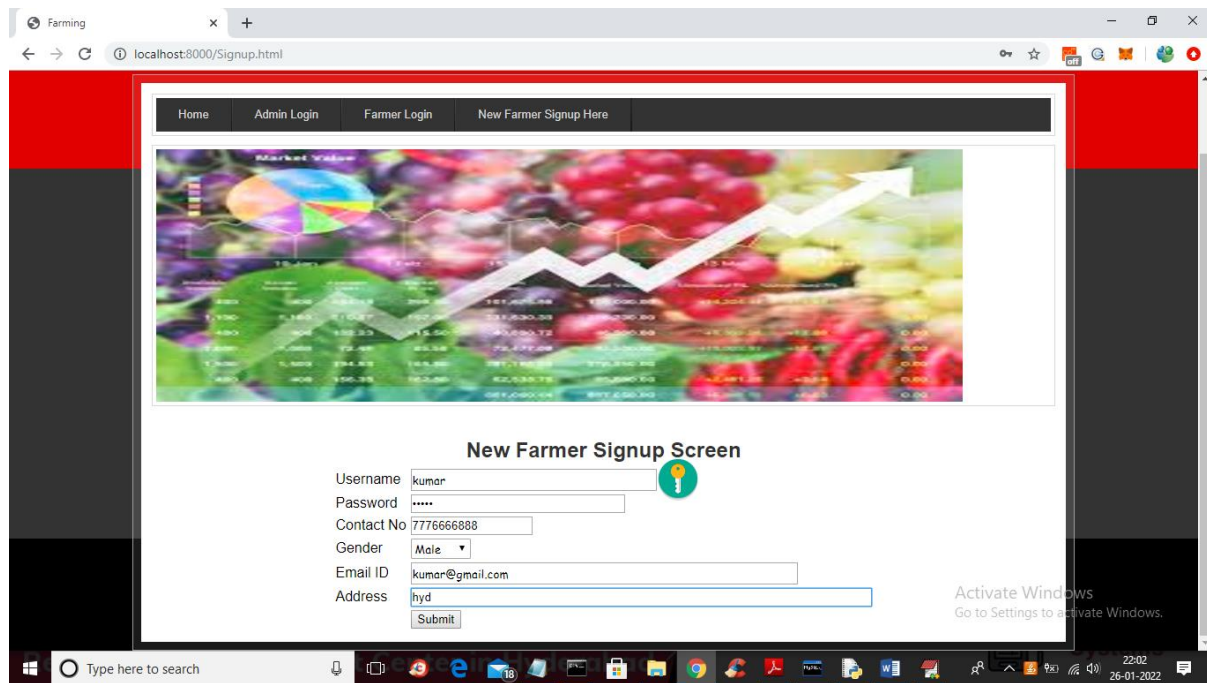
4. In above screen click on 'Add Government Schemes' link to add new schemes



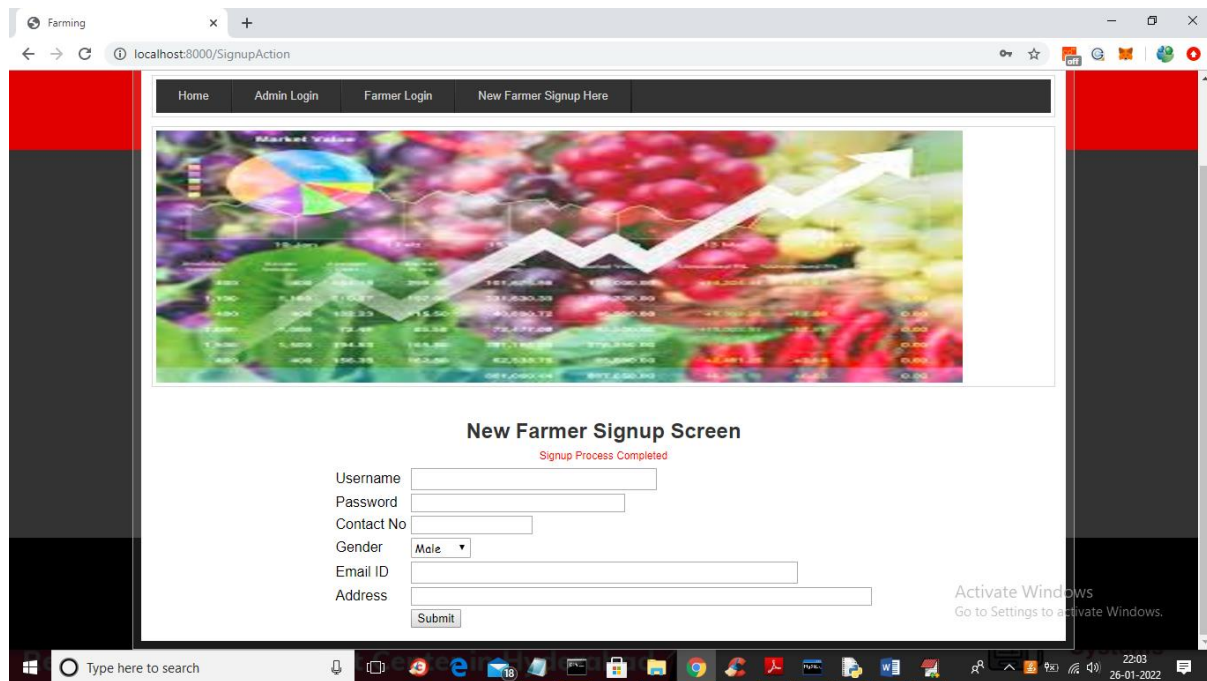
5. In above screen admin will add schemes details with start and end date and then click on 'Submit' button to save schemes details



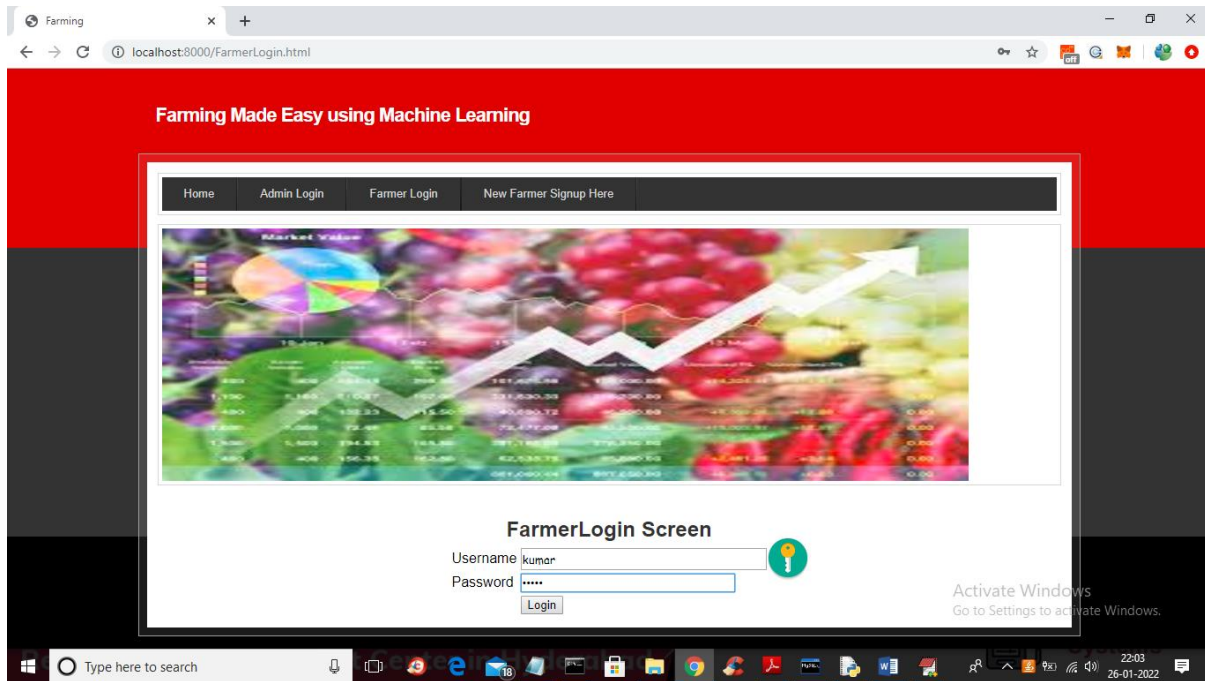
6. In above screen in red colour text we can see scheme details added and now logout and sign up new farmer



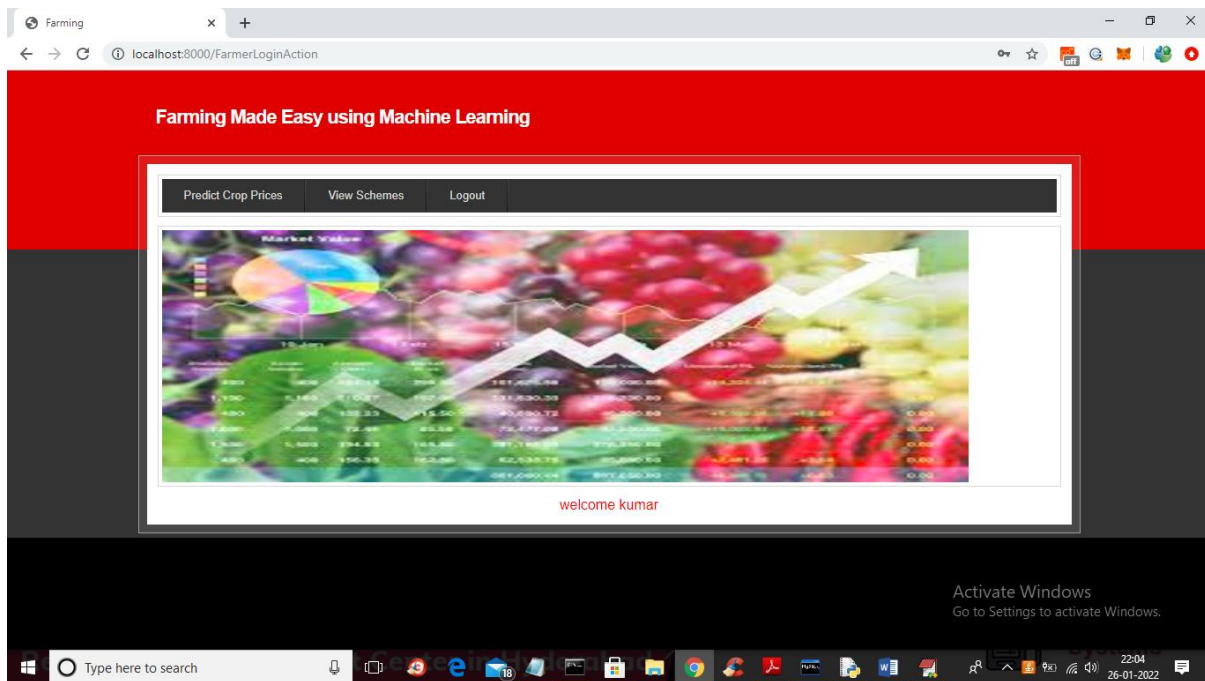
7. In above screen farmer is signup and click on 'Submit' button to complete signup process



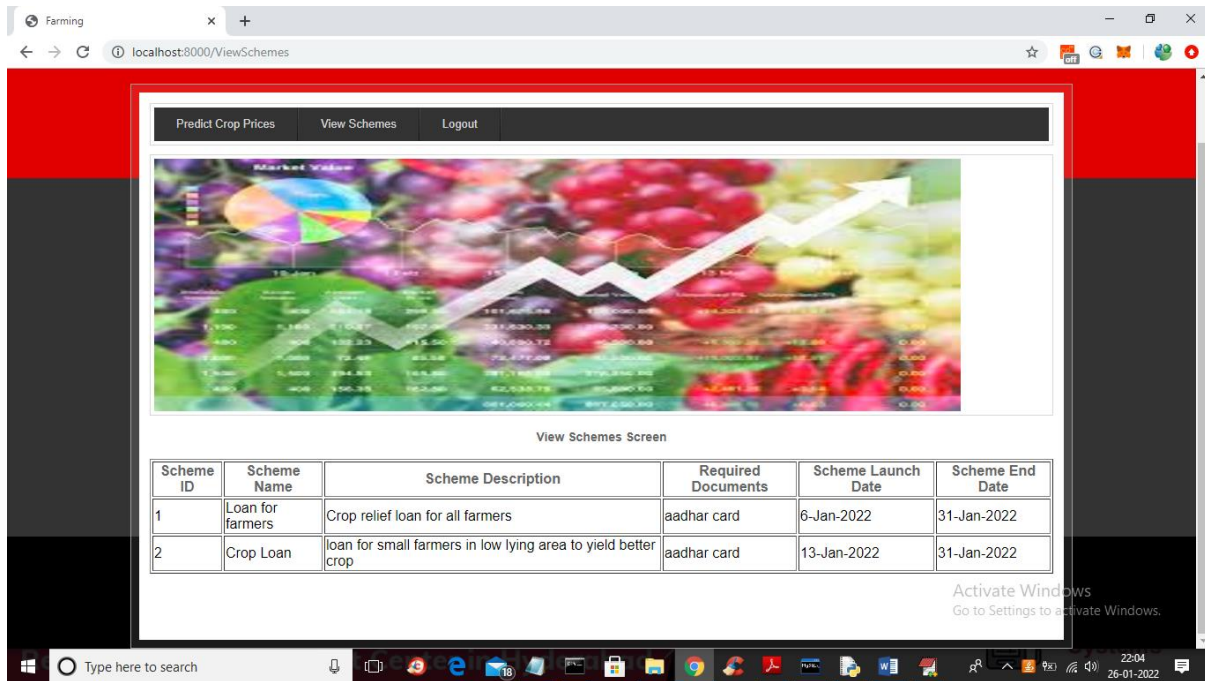
8. In above screen signup is completed and now click on 'Farmer Login' link to get below screen



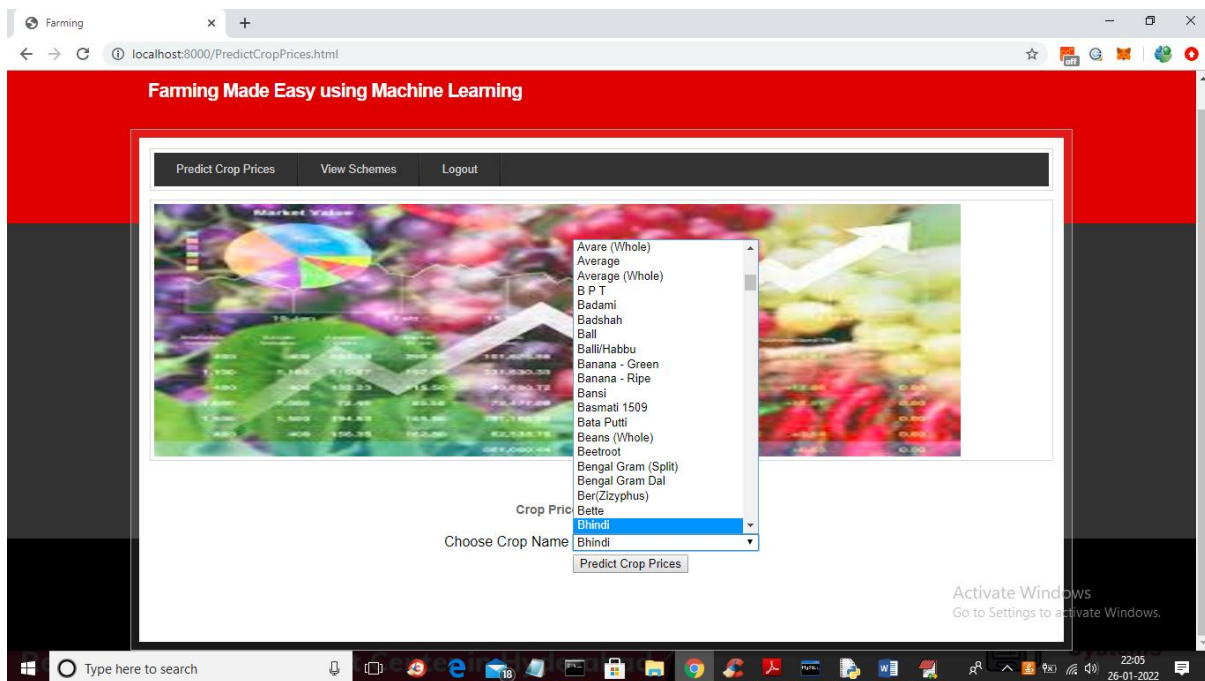
9. In above screen farmer is login and click on 'Login' button to get below screen



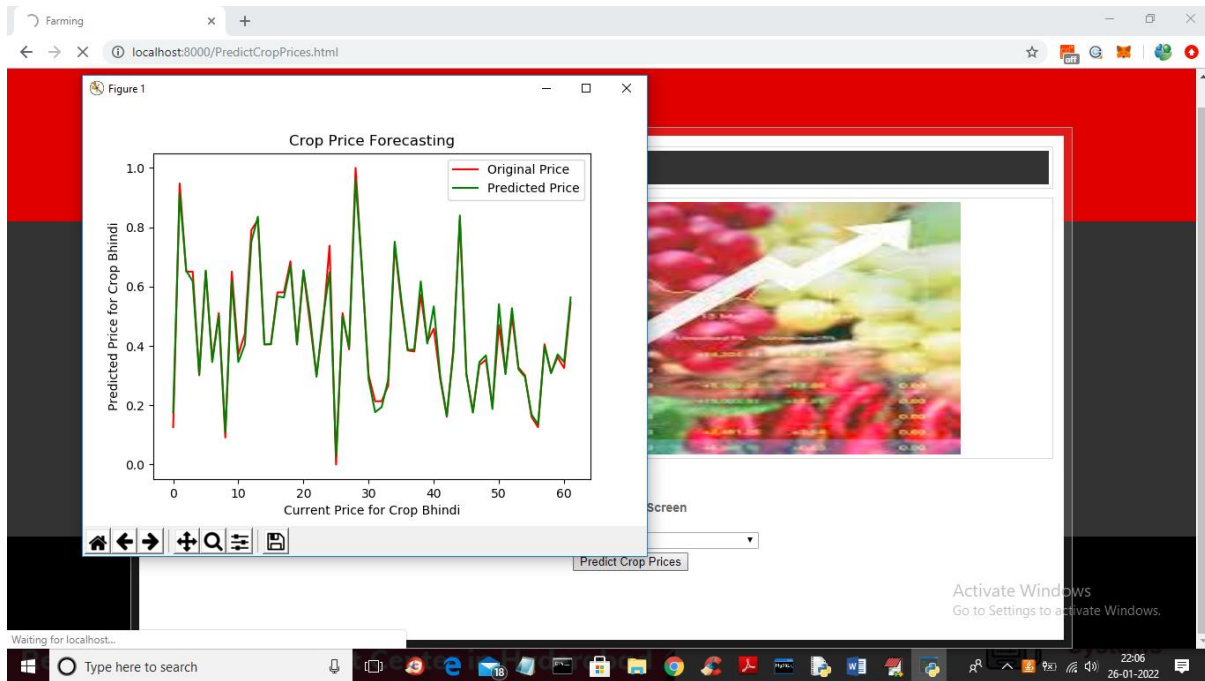
10. In above screen farmer can click on 'View Schemes' link to get all schemes details



11. In above screen all schemes details can be viewed by farmer and now click on 'Predict Crop Prices' link to get below screen



12. In above screen farmer can select desired crop and then click on 'Predict Crop Prices' link to get below prediction



13. In above graph red line represents Original prices and green line represents predicted prices and by seeing above graph farmer can understand what is current price and what will be future price and now close above graph to view predicted values

District Market	Crop Name	Original Price	Predicted Price
Rajnandgaon	Bhindi	1000.0	1289.58333333333348
Ambala	Bhindi	5700.0	5501.00000000000007
Ambala	Bhindi	4000.0	4022.00000000000008
Kurukshetra	Bhindi	4000.0	3810.7690476190496
Mahendragarh-Narnaul	Bhindi	2000.0	2021.4687770562768
Rohtak	Bhindi	4000.0	4022.00000000000008
Bangalore	Bhindi	2300.0	2256.0720634920635
Bangalore	Bhindi	3200.00000000000005	3143.797619047621
Davangere	Bhindi	800.00000000000001	900.83333333333342
Ernakulam	Bhindi	4000.0	3810.7690476190496
Ernakulam	Bhindi	2400.0	2256.0720634920635
Ernakulam	Bhindi	2800.0	2592.619047619047
Ernakulam	Bhindi	4800.0	4576.45000000000001
Kasargod	Bhindi	5000.0	5057.00000000000006
Kollam	Bhindi	2600.0	2593.60000000000003
Kottayam	Bhindi	2600.0	2600.300000000000025
Kottayam	Bhindi	3600.0	3522.6
Kottayam	Bhindi	3600.0	3501.59999999999995
Kottayam	Bhindi	4200.0	4116.00000000000006
Kozhikode(Calicut)	Bhindi	2600.0	2593.60000000000003
Palakad	Bhindi	4000.0	4028.00000000000008

14. In above screen first column represents 'district market name' and second column represents 'Crop Name' and third column represents 'Original Crop Price' and fourth column represents 'predicted prices'

using Machine learning algorithms and now scroll down above screen to view machine learning algorithms predicted accuracy

The screenshot shows a web browser window with the address bar displaying 'localhost:8000/PredictCropPricesAction'. The main content area contains a table with crop names, their districts, and predicted prices. At the bottom of the table, there are three rows showing the accuracy of different machine learning models: Random Forest, Decision Tree, and KNN.

Barabanki	Bhindi	1920.0	1969.1316666666666
Bijnor	Bhindi	1200.0	1225.9999999999982
Bulandshahar	Bhindi	2500.0	2434.6190476190463
Bulandshahar	Bhindi	5000.0	5083.0000000000006
Deoria	Bhindi	2025.0	2021.4687770562768
Etawah	Bhindi	1300.0	1280.9999999999999
Faizabad	Bhindi	2200.0	2256.0720634920635
Fatehpur	Bhindi	2300.0	2385.6190476190473
Firozabad	Bhindi	1400.0	1349.9999999999989
Firozabad	Bhindi	2970.0000000000005	3372.6666666666667
Mainpuri	Bhindi	2050.0	2021.4687770562768
Mathura	Bhindi	3140.0	3295.609523809524
Meerut	Bhindi	2150.0	2115.5049999999997
Mirzapur	Bhindi	2000.0	1969.1316666666666
Muzaffarnagar	Bhindi	1200.0	1243.9999999999977
Muzaffarnagar	Bhindi	1000.0	1057.5000000000016
Sant Kabir Nagar	Bhindi	2600.0	2557.0000000000014
Siddharth Nagar	Bhindi	2050.0	2040.7098484848502
Sultanpur	Bhindi	2350.0	2404.0000000000002
Varanasi	Bhindi	2140.0	2256.0720634920635
Malda	Bhindi	3400.0	3501.5999999999995
Random Forest Accuracy	0.9993409853957524		
Decision Tree Accuracy	0.9997952633406897		
KNN Accuracy	0.9986281918756577		

15. In above screen in last 3 lines we can see Random forest, KNN and decision tree prediction accuracy.

Similarly you can select any crop and get prediction prices

Note: some crop contains only 3 or 4 records so prediction may not be correct as to train we need minimum 50 to 100 records. Here bhindi and coriander crop contains more records

9.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.

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.

10.CONCLUSION

This project is undertaken using machine learning and evaluates the performance by using KNN, Naive Bayes, and Decision Tree algorithms. In our proposed model among all the three algorithm Decision Tree gives the better yield prediction as compared to other algorithms

As most extreme sorts of harvests will be secured under this system, farmers may become more acquainted with the yield which may never have been developed. The work exhibited the expected utilization of machine learning methods in foreseeing the harvest cost dependent on the given attributes. The created web application is easy to understand and the testing accuracy is over 90%.

11.REFERENCES

- [1] Rachana, Rashmi, Shravani, Shruthi, Seema Kousar, Crop Price Forecasting System Using Supervised Machine Learning Algorithms, International Research Journal of Engineering and Technology (IRJET), Apr 2019
- [2] Nishiba Kabeer, Dr.Loganathan.D, Cowsalya.T, Prediction of Crop Yield Using Data Mining, International Journal of Computer Science and Network, June 2019
- [3] J. Vijayalakshmi, K. PandiMeena, Agriculture TalkBot Using AI, International Journal of Recent Technology and Engineering (IJRTE), July 2019
- [4] Gamage, A., & Kasthurirathna, D. Agro-Genius: Crop Prediction Using Machine Learning, International Journal of Innovative Science and Research Technology, Volume 4, Issue 10, October – 2019
- [5] Vohra Aman, Nitin Pandey, and S. K. Khatri. "Decision Making Support System for Prediction of Prices in Agricultural Commodity." 2019 Amity International Conference on Artificial Intelligence (AICAI). IEEE, 2019.
- [6] Nguyen, Huy Vuong, et al. "A smart system for short-term price prediction using time series models." Computers & Electrical Engineering 76 (2019)
- [7] Sangeeta, Shruthi G, Design And Implementation Of Crop Yield Prediction Model In Agriculture, International Journal Of Scientific & Technology Research Volume 8, Issue 01, January 2020
- [8] Rohith R, Vishnu R, Kishore A, Deeban Chakkarawartha, Crop Price Prediction and Forecasting System using Supervised Machine Learning Algorithms, International Journal of Advanced Research in Computer and Communication Engineering, March 2020
- [9] Naveen Kumar P R, Manikanta K B, Venkatesh B Y, Naveen Kumar R, Amith Mali Patil, Journal of Xi'an University of Architecture & Technology, 2020.
- [10] Kumar, Y. Jeevan Nagendra, et al. "Supervised Machine learning Approach for Crop Yield Prediction in the Agriculture Sector." 2020 5th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2020.
- [11] Pandit Samuel, B.Sahithi , T.Saheli , D.Ramanika , N.Anil Kumar, Crop Price Prediction System using Machine learning Algorithms, Quest Journals Journal of Software Engineering and Simulation, 2020

[12] Rubhi gupta, Review on weather prediction using machine learning, International Journal of Engineering Development and Research, 2020