

CHAPTER 1

INTRODUCTION

India being a rural nation, its economy transcendentally 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.

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.

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 Neighbor to identify the most accurate model and then process it. This in turn will help predict the price of the crop.

1.1 PROBLEM STATEMENT:

Agricultural crop prices are highly volatile and influenced by various factors such as supply and demand, weather conditions, government policies, and market trends. Farmers, traders, and policymakers often face difficulties in making informed decisions regarding crop production, storage, and marketing due to the unpredictability of crop prices. This unpredictability can lead to financial losses, food wastage, and inefficient resource allocation.

1.2 SCOPE AND OBJECTIVE:

The objective of this project is to develop a machine learning model that can predict the future prices of crops based on historical data and relevant influencing factors. By providing accurate and timely price forecasts, the model aims to:

1. Empower farmers to decide when to sell their crops to maximize profits.
2. Help traders and wholesalers optimize inventory and logistics.
3. Assist policymakers in planning subsidies, imports, and exports.
4. Contribute to reducing food waste and enhancing agricultural sustainability.

The model will leverage historical price data, weather patterns, crop yield, market trends, and other socio-economic factors to predict prices. The output should be easy to interpret and accessible to stakeholders with varying levels of technical expertise.

CHAPTER 2

LITERATURE SURVEY

The following project 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. 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 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 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 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 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, Support Vector Regression Algorithm, and technologies having a general easy to use interface to the clients. The training datasets acquired give sufficient bits of knowledge to foreseeing the appropriate price 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.

2.1 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 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. Google API is used for providing a multilingual website for ease to read.

2.2 PROPOSED SYSTEM:

In this project we are using various machine learning algorithms such as Random Forest, Decision Tree and KNN to predict crop prices. All these algorithms get train on Crop Prices dataset which contains crop details weather details such as Rainfall with crop name, market name with prices and Rainfall. We have used Python for basic programming in all modules. Django is used for hosting.

We use dataset details with crop name, market name with prices and Rainfall. 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.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS:

- **System** : i3, ryzen3 or above.
- **Hard Disk** : 256 GB or above.
- **RAM** : 8 GB or above.

3.2 SOFTWARE REQUIREMENTS:

- **Operating System** : Windows 11
- **Coding Language** : Python 3.7
- **Server** : XAMPP server
- **IDE** : Visual Studio Code

3.3 UTILISED SYSTEM FOR THE PROJECT:

- **System** : Ryzen 5
- **SSD** : 1TB
- **RAM** : 16 GB
- **Operating System** : Windows 11
- **Coding Language** : Python 3.7
- **Server** : XAMPP server
- **IDE** : Visual Studio Code

CHAPTER 4

SYSTEM STUDY

4.1 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**

4.1.1 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.

4.1.2 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.

4.1.3 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.



Figure 4.1: Types of Feasibility studies

CHAPTER 5

SYSTEM DESIGN

5.1 UML DIAGRAMS:

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

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

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

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

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

5.1.1 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.

5.2 USE CASE DIAGRAM:

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

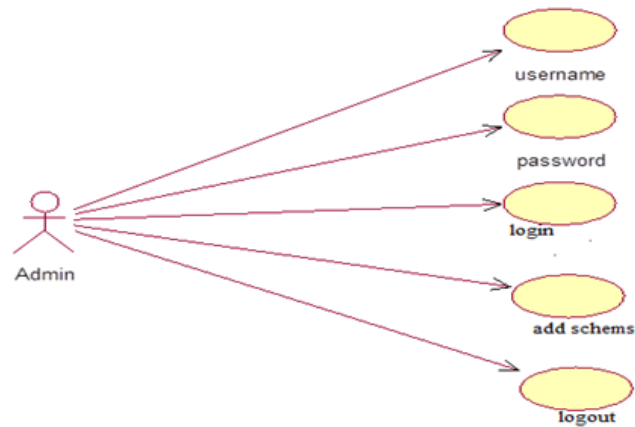


Figure 5.2.1: Use case diagram for Admin

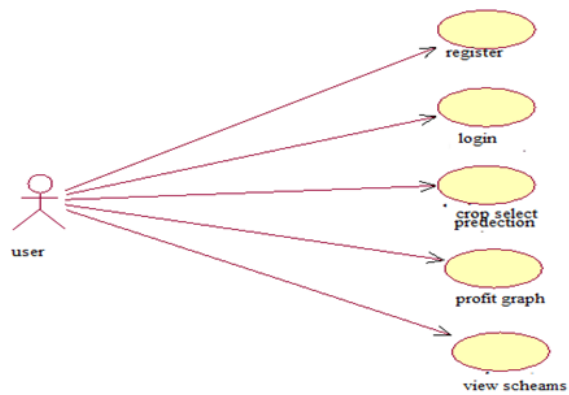


Figure 5.2.2: Use case diagram for User

5.3 CLASS DIAGRAM:

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

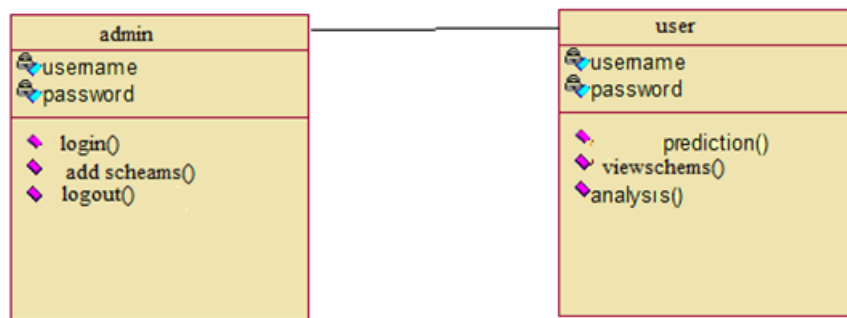


Figure 5.3: Class diagram

5.4 SEQUENCE DIAGRAM:

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

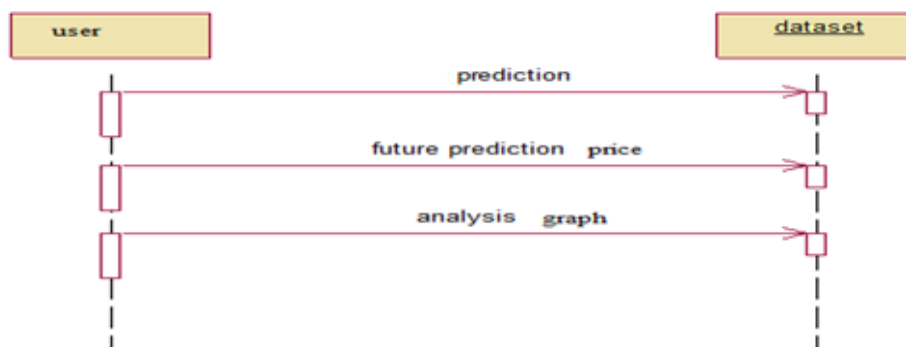


Figure 5.4: Sequence diagram

5.5 ACTIVITY DIAGRAM:

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

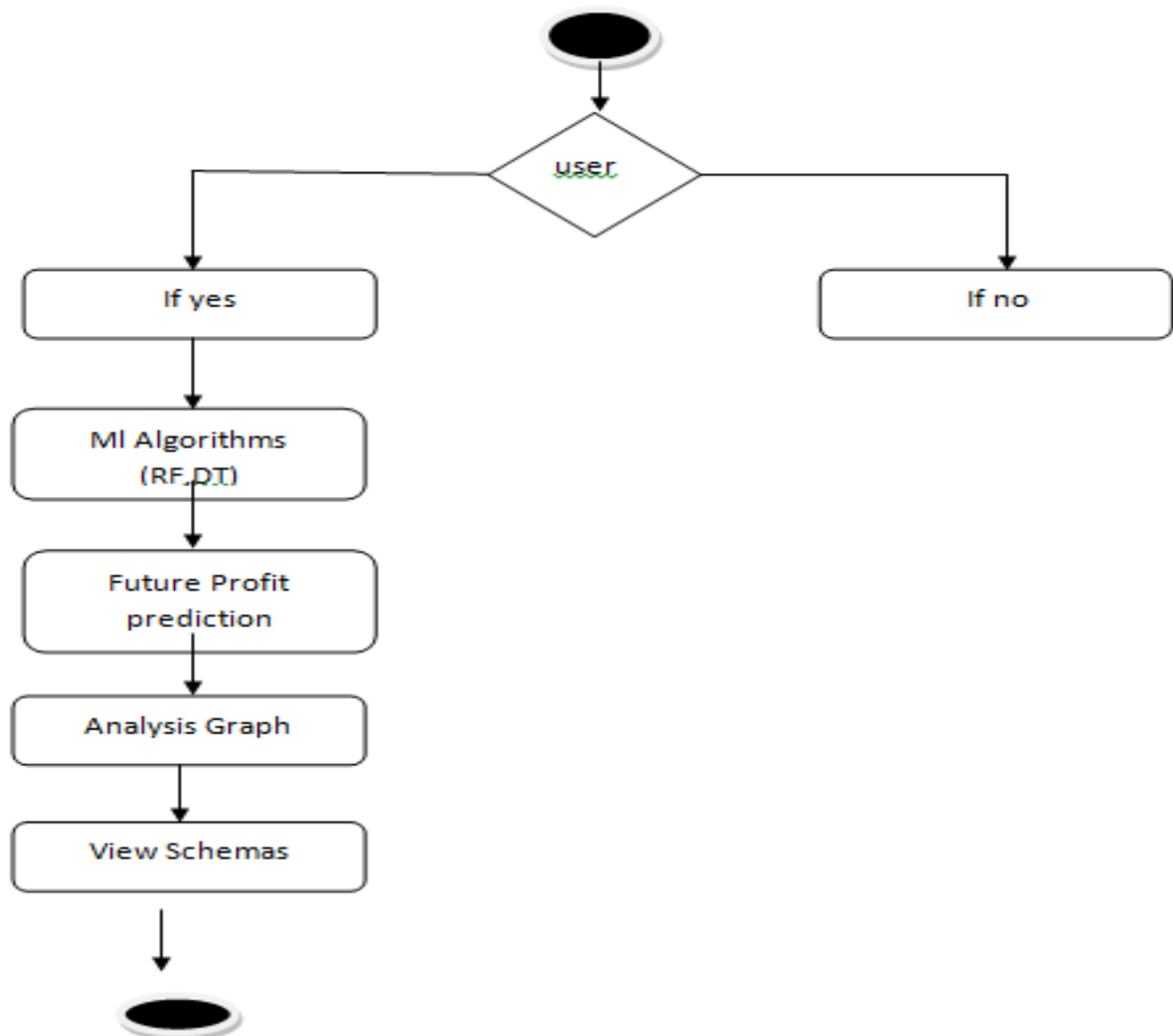


Figure 5.5: Activity diagram

CHAPTER 6

ALGORITHMS

6.1 DECISION TREE ALGORITHM:

A decision tree is a flowchart-like structure used to make decisions or predictions. It consists of nodes representing decisions or tests on attributes, branches representing the outcome of these decisions, and leaf nodes representing final outcomes or predictions. Each internal node corresponds to a test on an attribute, each branch corresponds to the result of the test, and each leaf node corresponds to a class label or a continuous value.

6.1.1 STRUCTURE OF DECISION TREE:

1. **Root Node:** Represents the entire dataset and the initial decision to be made.
2. **Internal Nodes:** Represent decisions or tests on attributes. Each internal node has one or more branches.
3. **Branches:** Represent the outcome of a decision or test, leading to another node.
4. **Leaf Nodes:** Represent the final decision or prediction. No further splits occur at these nodes.

6.1.2 HOW DECISION TREE WORKS?

The process of creating a decision tree involves:

1. **Selecting the Best Attribute:** Using a metric like Gini impurity, entropy, or information gain, the best attribute to split the data is selected.
2. **Splitting the Dataset:** The dataset is split into subsets based on the selected attribute.
3. **Repeating the Process:** The process is repeated recursively for each subset, creating a new internal node or leaf node until a stopping criterion is met (e.g., all instances in a node belong to the same class or a predefined depth is reached).

6.1.3 ADVANTAGES OF DECISION TREES:

- **Simplicity and Interpretability:** Decision trees are easy to understand and interpret. The visual representation closely mirrors human decision-making processes.
- **Versatility:** Can be used for both classification and regression tasks.
- **No Need for Feature Scaling:** Decision trees do not require normalization or scaling of the data.
- **Handles Non-linear Relationships:** Capable of capturing non-linear relationships between features and target variables.

6.1.4 DISADVANTAGES OF DECISION TREES:

- **Overfitting:** Decision trees can easily overfit the training data, especially if they are deep with many nodes.
- **Instability:** Small variations in the data can result in a completely different tree being generated.
- **Bias towards Features with More Levels:** Features with more levels can dominate the tree structure.

6.1.5 APPLICATIONS OF DECISION TREES:

- **Business Decision Making:** Used in strategic planning and resource allocation.
- **Healthcare:** Assists in diagnosing diseases and suggesting treatment plans.
- **Finance:** Helps in credit scoring and risk assessment.
- **Marketing:** Used to segment customers and predict customer behavior.

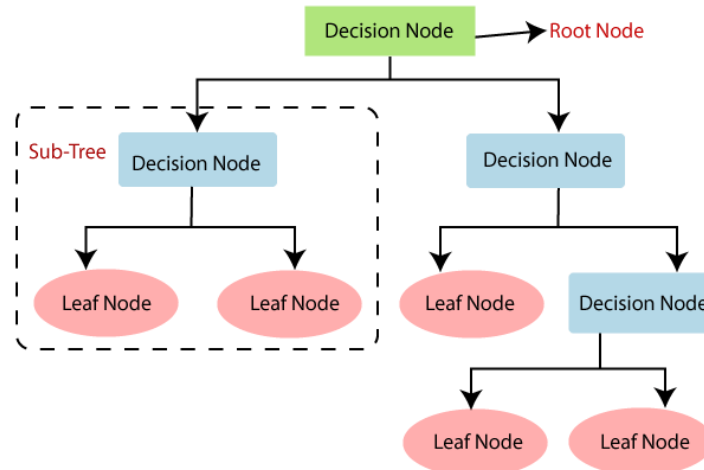


Figure 6.1: Structure of Decision Tree

6.2 RANDOM FOREST ALGORITHM:

Random Forest algorithm is a powerful tree learning technique in Machine Learning. It works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance.

In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks). This collaborative decision-making process, supported by multiple trees with their insights, provides an example stable and precise results. Random forests are widely used for classification and regression functions, which are known for their ability to handle complex data, reduce overfitting, and provide reliable forecasts in different environments.

6.2.1 ALGORITHM FOR RANDOM FOREST WORK:

1. **Step 1:** Select random K data points from the training set.
2. **Step 2:** Build the decision trees associated with the selected data points(Subsets).
3. **Step 3:** Choose the number N for decision trees that you want to build.
4. **Step 4:** Repeat Step 1 and 2.
5. **Step 5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

6.2.2 APPLICATIONS OF RANDOM FOREST IN REAL-WORLD:

Some of the widely used real-world application of Random Forest is discussed below:

1. **Finance Wizard:** Imagine Random Forest as our financial superhero, diving into the world of credit scoring. Its mission? To determine if you're a credit superhero or, well, not so much. With a knack for handling financial data and sidestepping overfitting issues, it's like having a guardian angel for robust risk assessments.
2. **Health Detective:** In healthcare, Random Forest turns into a medical Sherlock Holmes. Armed with the ability to decode medical jargon, patient records, and test results, it's not just predicting outcomes; it's practically assisting doctors in solving the mysteries of patient health.
3. **Environmental Guardian:** Out in nature, Random Forest transforms into an environmental superhero. With the power to decipher satellite images and brave noisy data, it becomes the go-to hero for tasks like tracking land cover changes and safeguarding against potential deforestation, standing as the protector of our green spaces.
4. **Digital Bodyguard:** In the digital realm, Random Forest becomes our vigilant guardian against online trickery. It's like a cyber-sleuth, analyzing our digital footsteps for any hint of suspicious activity. Its ensemble approach is akin to having a team of cyber-detectives, spotting subtle deviations that scream "fraud alert!" It's not just protecting our online transactions; it's our digital bodyguard.

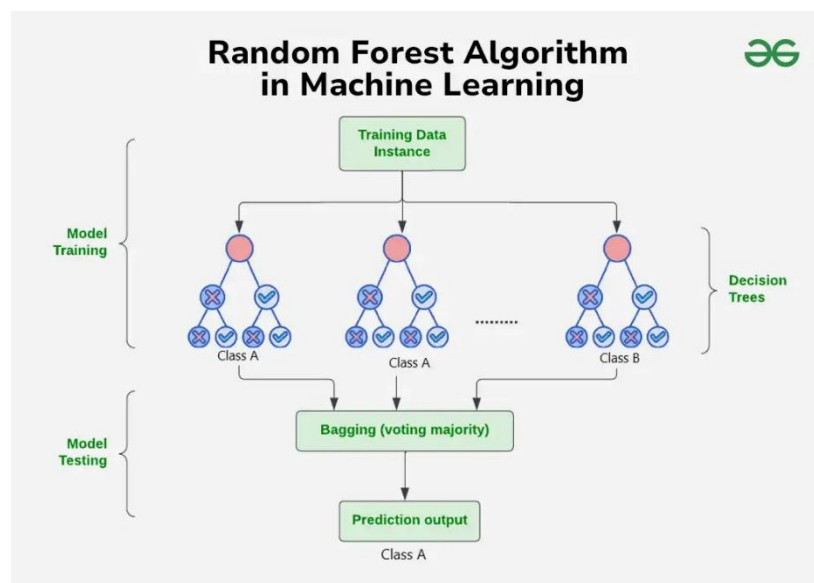


Figure 6.2: Structure of Random Forest

6.3 K-NEAREST NEIGHBOR ALGORITHM:

KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data). We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

6.3.1 HOW DOES KNN WORK?

The K-NN working can be explained on the basis of the below algorithm:

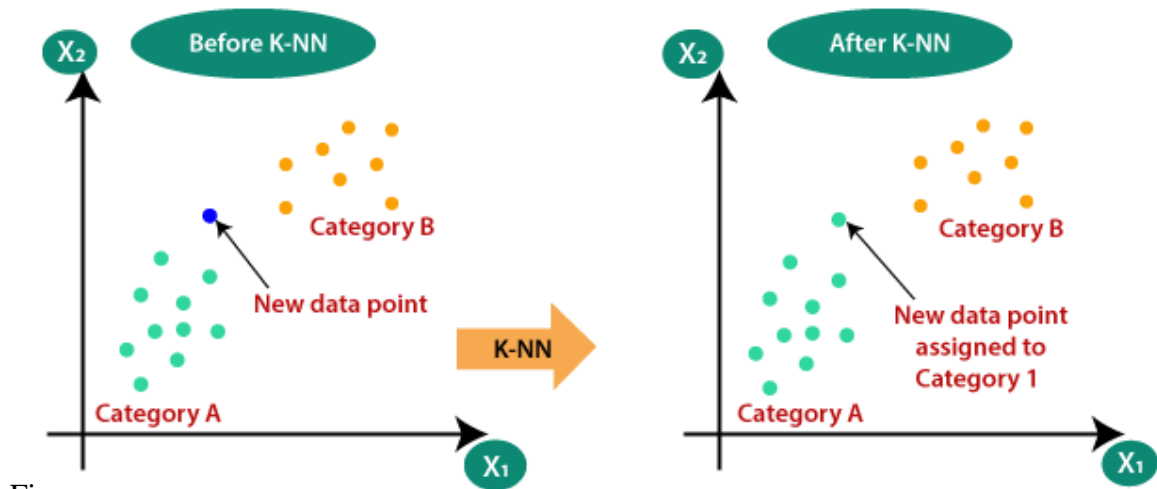
- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of K number of neighbors
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

6.3.2 ADVANTAGES OF KNN ALGORITHM:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

6.3.3 DISADVANTAGES OF KNN ALGORITHM:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.



Figure

6.3: KNN Algorithm

CHAPTER 7

IMPLEMENTATION

7.1 MODULES:

1) New Farmer Signup:

Using this module farmers can sign up with application

2) Farmer Login:

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

3) Admin Login:

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

CHAPTER 8

SOFTWARE ENVIRONMENT

8.1 WHAT IS PYTHON?

Below are some facts about Python:

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
- Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.
- The biggest strength of Python is huge collection of standard library which can be used for the following

8.1.1 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 purpose 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**. These 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.

8.1.2 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.

8.1.3 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 **Carbonnelle**.

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.

8.1.4 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."

8.2 WHAT IS MACHINE LEARNING?

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

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

8.2.1 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.

8.2.2 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.

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.

8.2.3 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.

8.2.4 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
- Recommendation of products to customer in online shopping

8.2.5 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!

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.

8.2.6 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.

8.2.7 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.

8.3 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 lists, 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 3.0:

- 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.

CHAPTER 9

MODULES

9.1 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.

9.2 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.

9.3 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.

9.4 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.

9.5 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.

CHAPTER 10

SOFTWARE INSTALLATION

10.1 INSTALLATION OF PYTHON STEP BY STEP:

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.

10.2 HOW TO INSTALL PYTHON IN 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.

10.3 DOWNLOAD THE CORRECT VERSION:

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>



Figure 10.3.1: Python.org

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

Step 2: Click on the Download Tab.



Figure 10.3.2: Python version

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 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Figure 10.3.3: Specific version

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		68111671e5b2db4aef7b9ab010f09be	23017663	SIG
XZ compressed source tarball	Source release		d33e4a9e6097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daf1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c56b2ac56cade6b47cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64/x64	9b003c3fd9ec0b9abe63184a40728a2	7504291	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64/x64	a702b4b0ad76debd83043a383e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64/x64	28c31c6088ed72a88e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3bd18841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc802942a54446a3d8451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

Figure 10.3.4: Executable version

- 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.

10.4 INSTALLATION OF PYTHON:

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

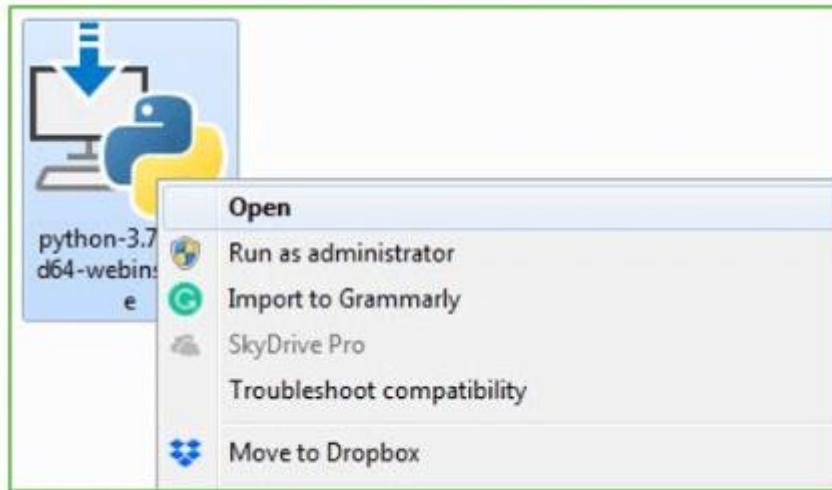


Figure 10.4.1: Python installation

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



Figure 10.4.2: Path specification

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



Figure 10.4.3: Setup successful

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.

10.5 VERIFY THE PYTHON INSTALLATIONS:

Step 1: Click on Start

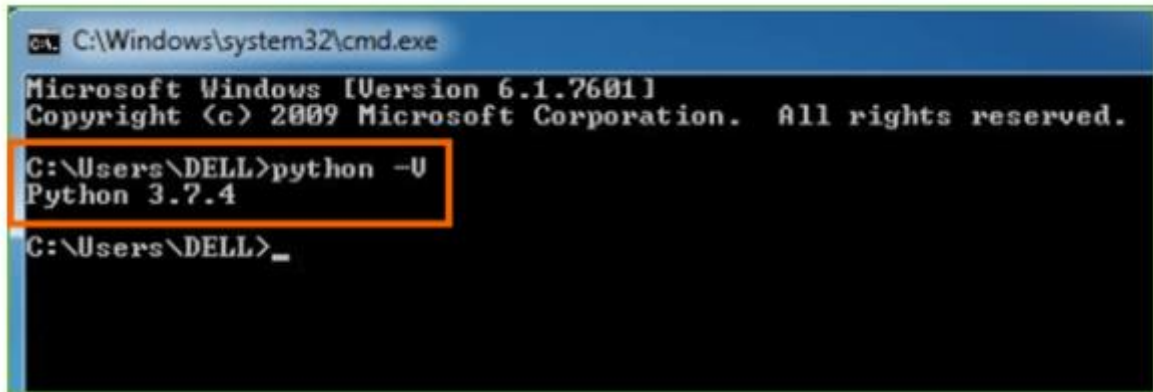
Step 2: In the Windows Run Command, type "cmd".



Figure 10.5.1: 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>_
```

Figure 10.5.2: Python verification

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.

10.6 CHECK HOW THE PYTHON IDLE WORKS:

Step 1: Click on Start

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

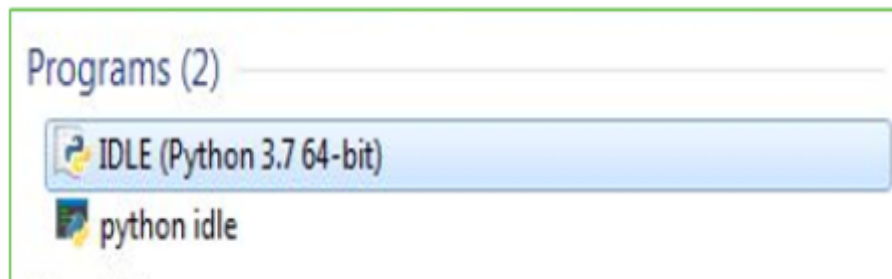


Figure 10.6.1: Python IDLE

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**

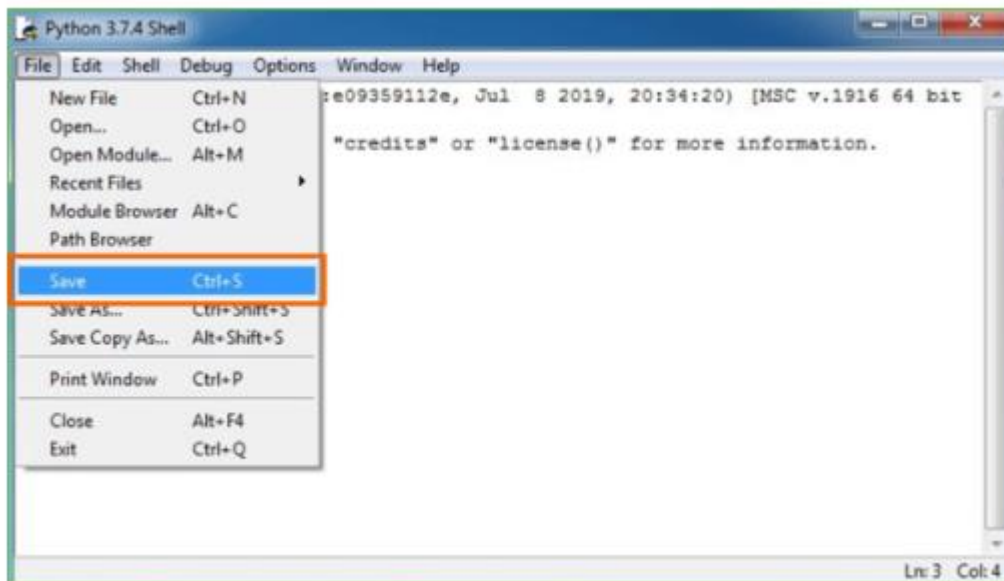


Figure 10.6.2: Python Shell

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**

CHAPTER 11

SYSTEM TEST

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.

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

- **Functional test:**

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

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked.

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

- **System Test:**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

- **White Box Testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

- **Black Box Testing:**

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

- **Unit Testing:**

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

11.2 TEST STRATEGY AND APPROACH:

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

Test objectives:

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

Features to be tested:

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

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:

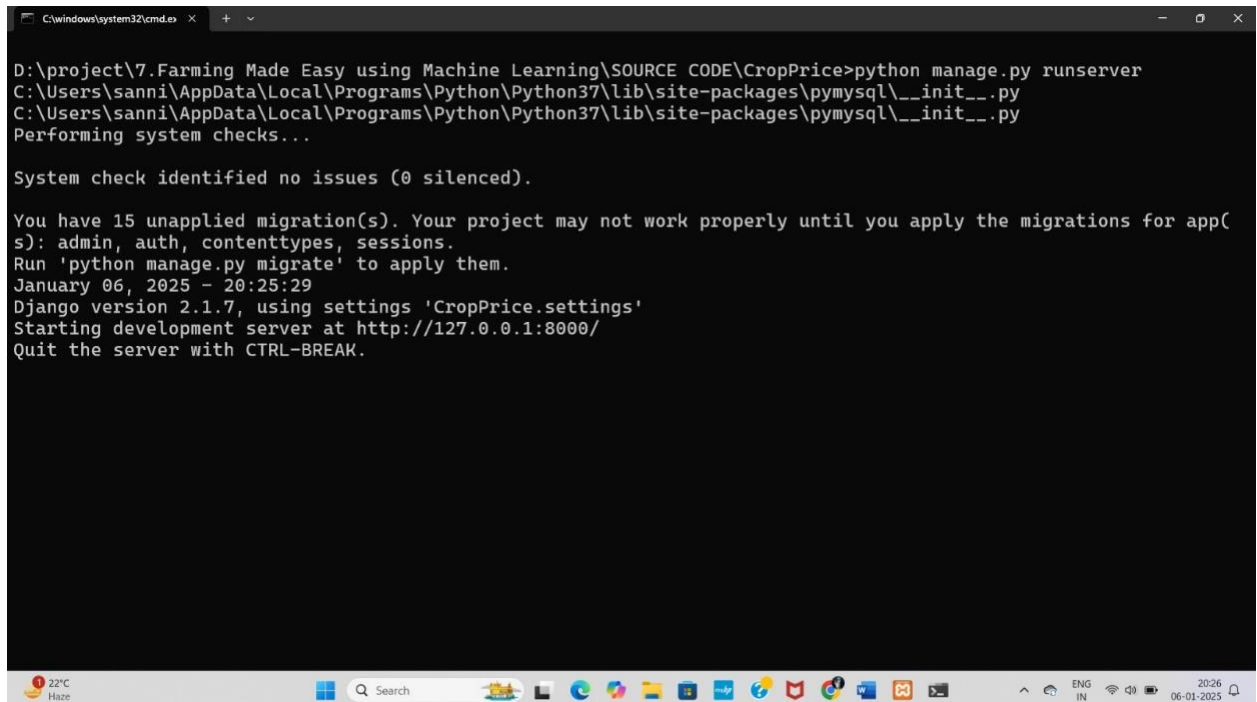
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

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

CHAPTER 12

OUTPUT SCREENS



```
C:\windows\system32\cmd.exe x + v

D:\project\7.Farming Made Easy using Machine Learning\SOURCE CODE\CropPrice>python manage.py runserver
C:\Users\sanni\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
C:\Users\sanni\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 06, 2025 - 20:25:29
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.
```

Figure 12.1: In above screen server started and now open browser and enter URL



Figure 12.2: In above screen click on 'Admin Login' link to get below login screen

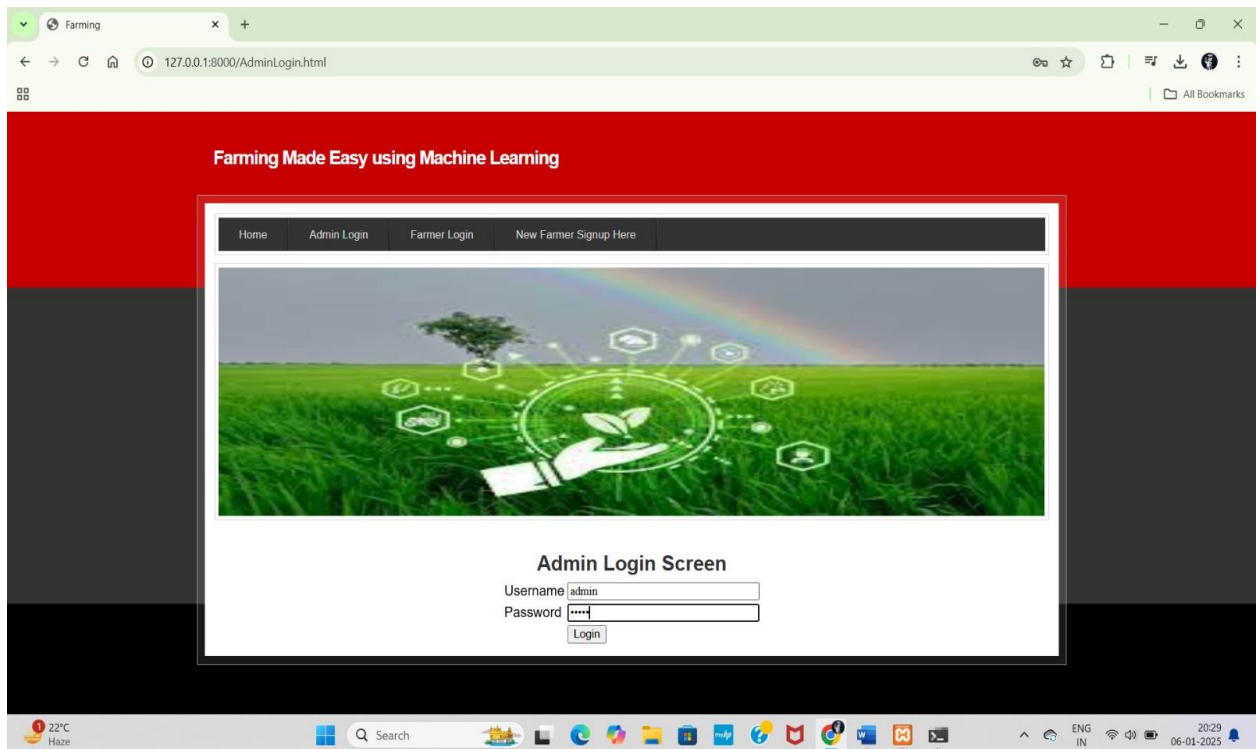


Figure 12.3: In above screen admin is login and after login will get below screen

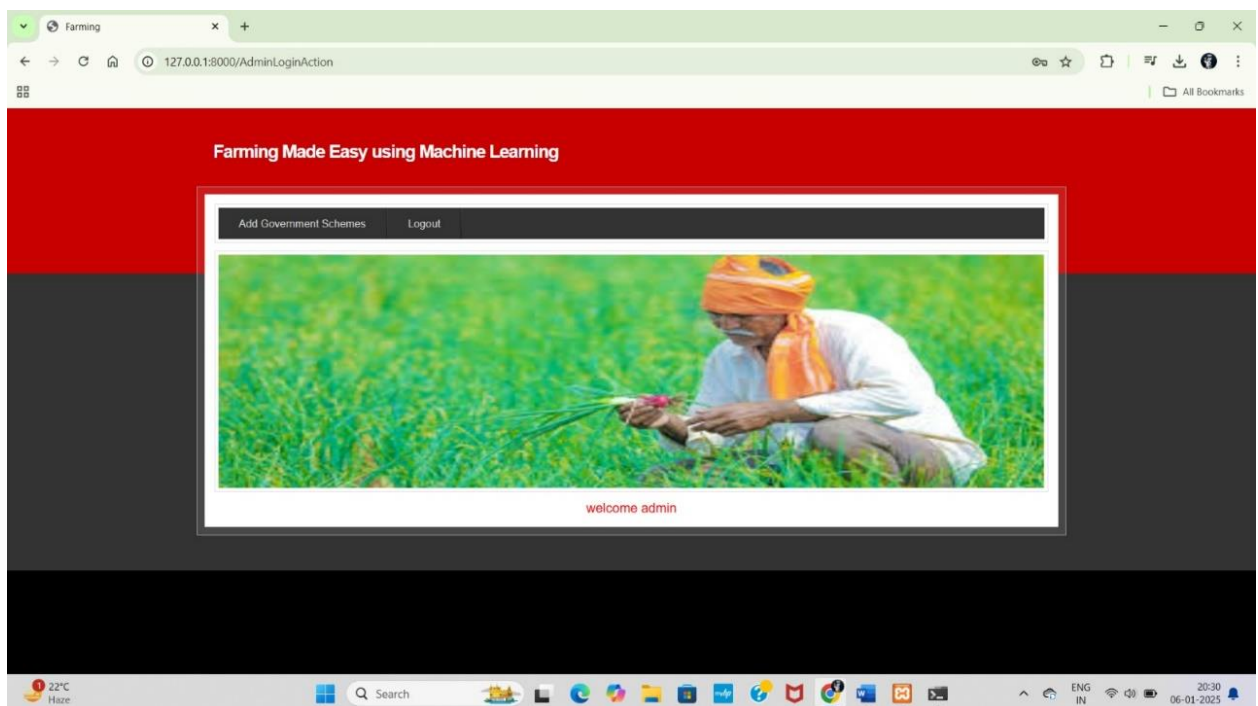


Figure 12.4: In above screen click on 'Add Government Schemes' link to add new schemes

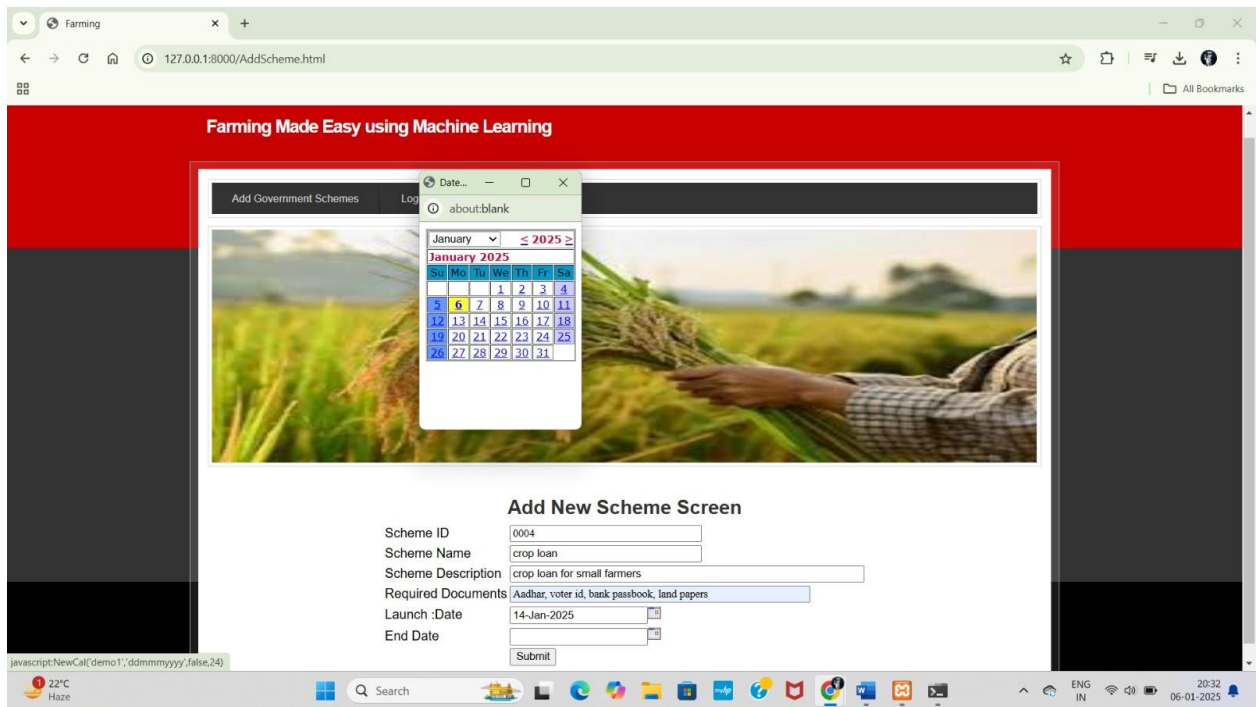


Figure 12.5: In above screen admin will add schemes details with start and end date and then click on ‘**Submit**’ button to save schemes details

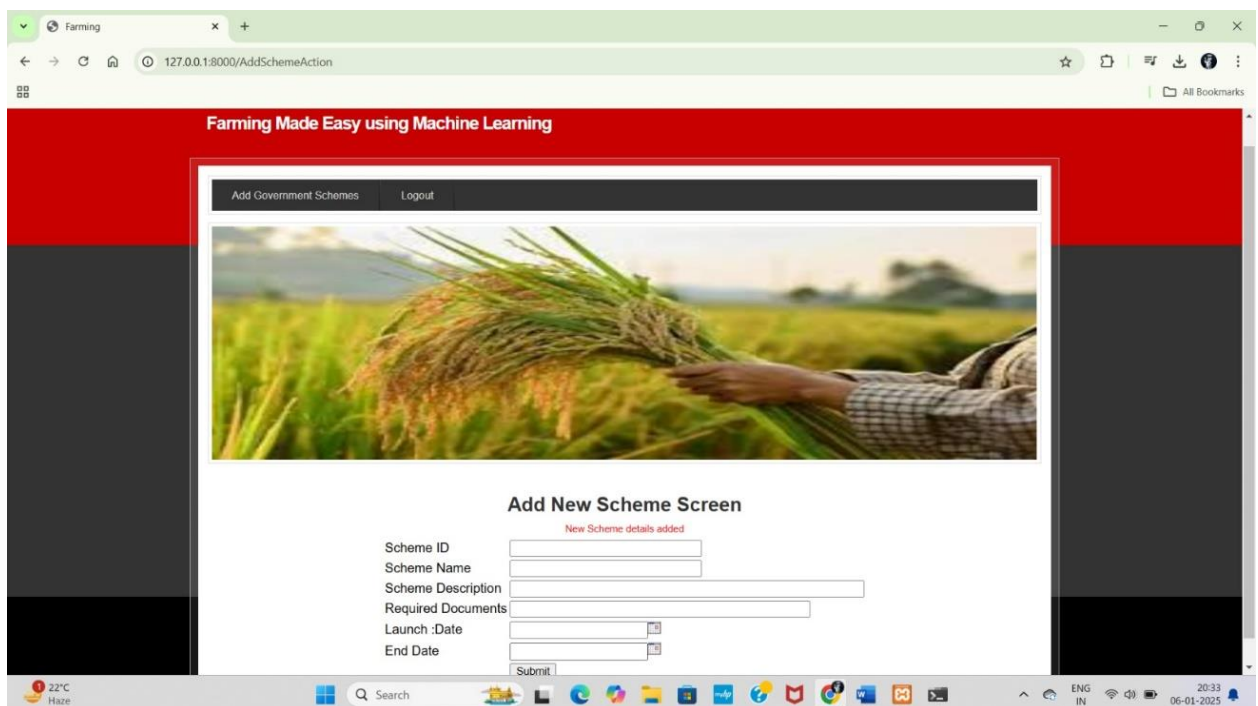


Figure 12.6: In above screen in red colour text we can see scheme details added and now logout and signup new farmer

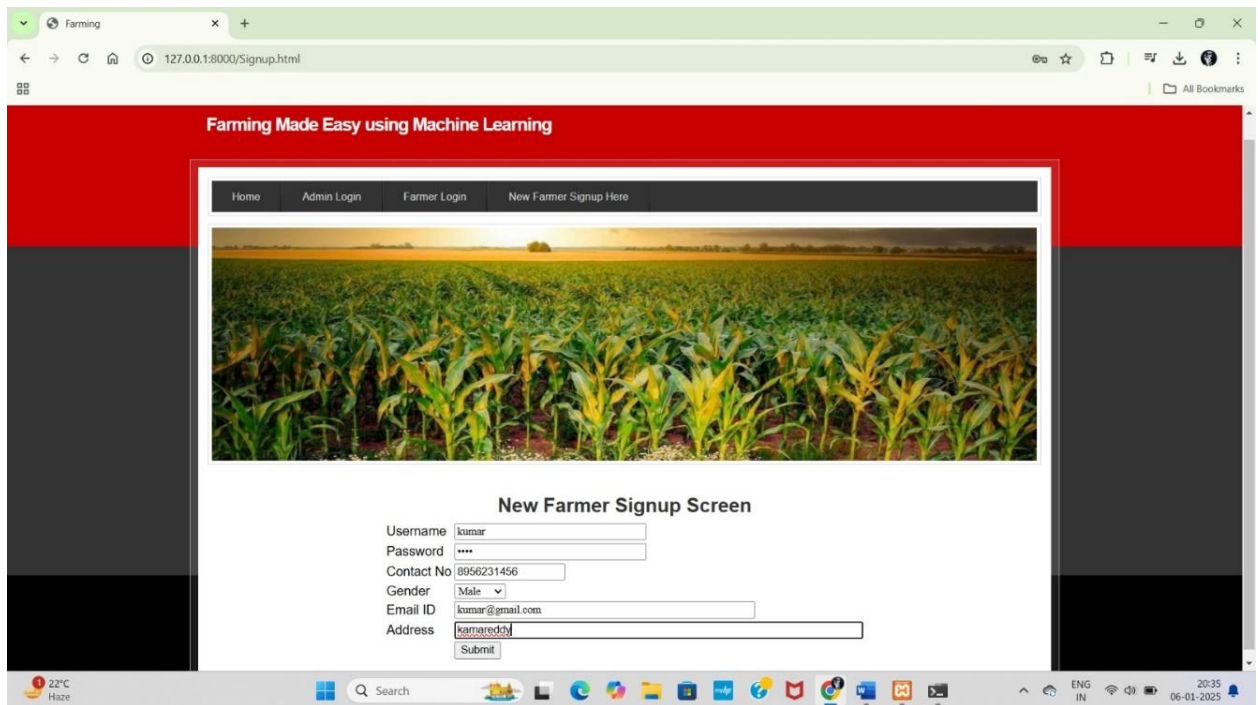


Figure 12.7: In above screen farmer is signup and click on ‘Submit’ button to complete signup process

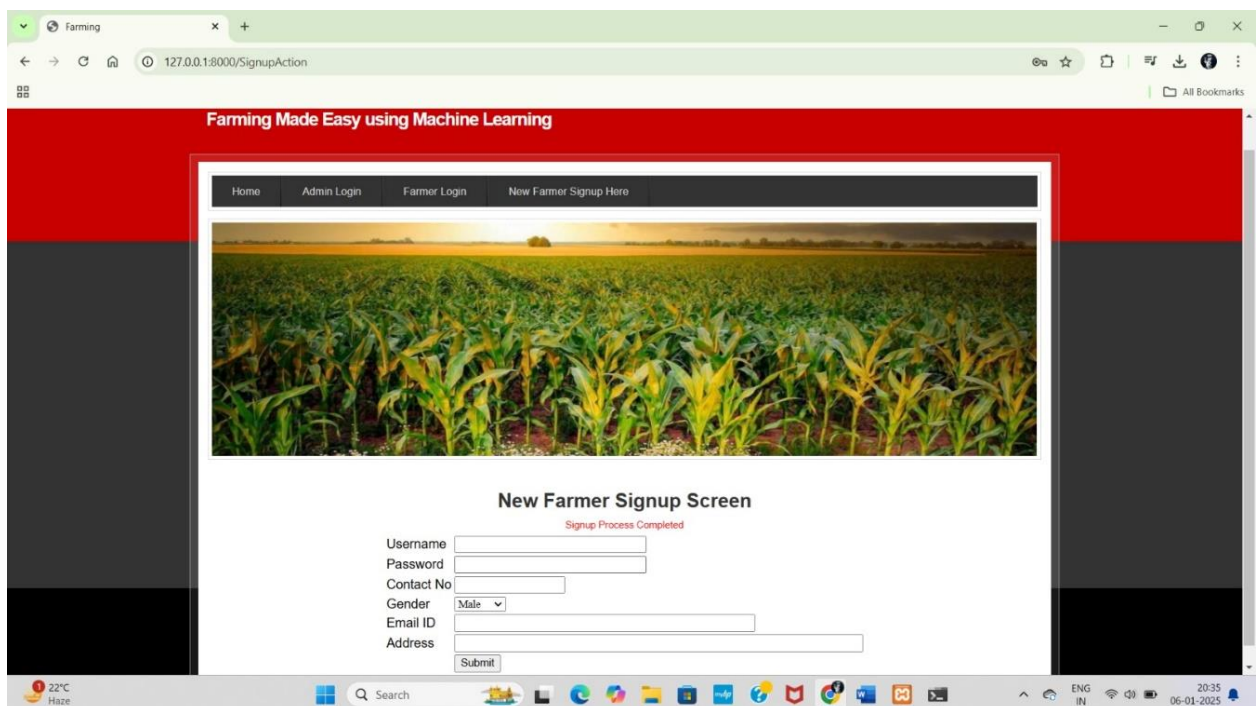


Figure 12.8: In above screen signup is completed and now click on ‘Farmer Login’ link to get below screen

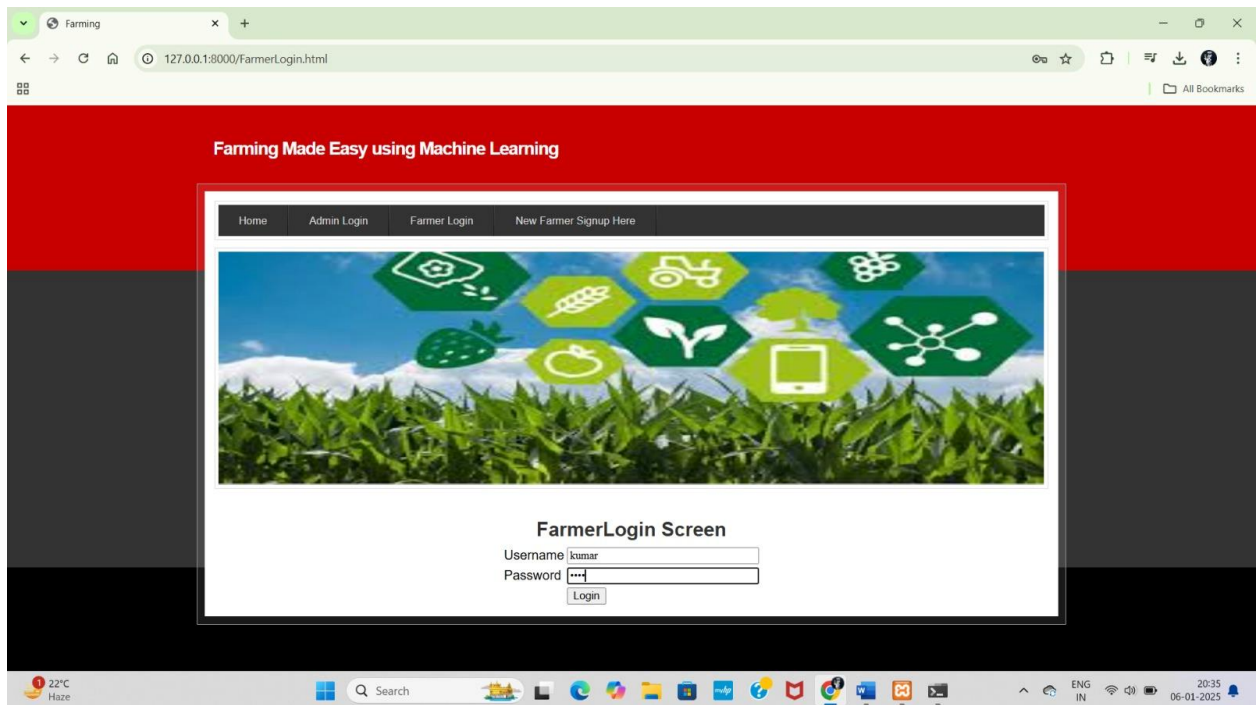


Figure 12.9: In above screen farmer is login and click on ‘**Login**’ button to get below screen

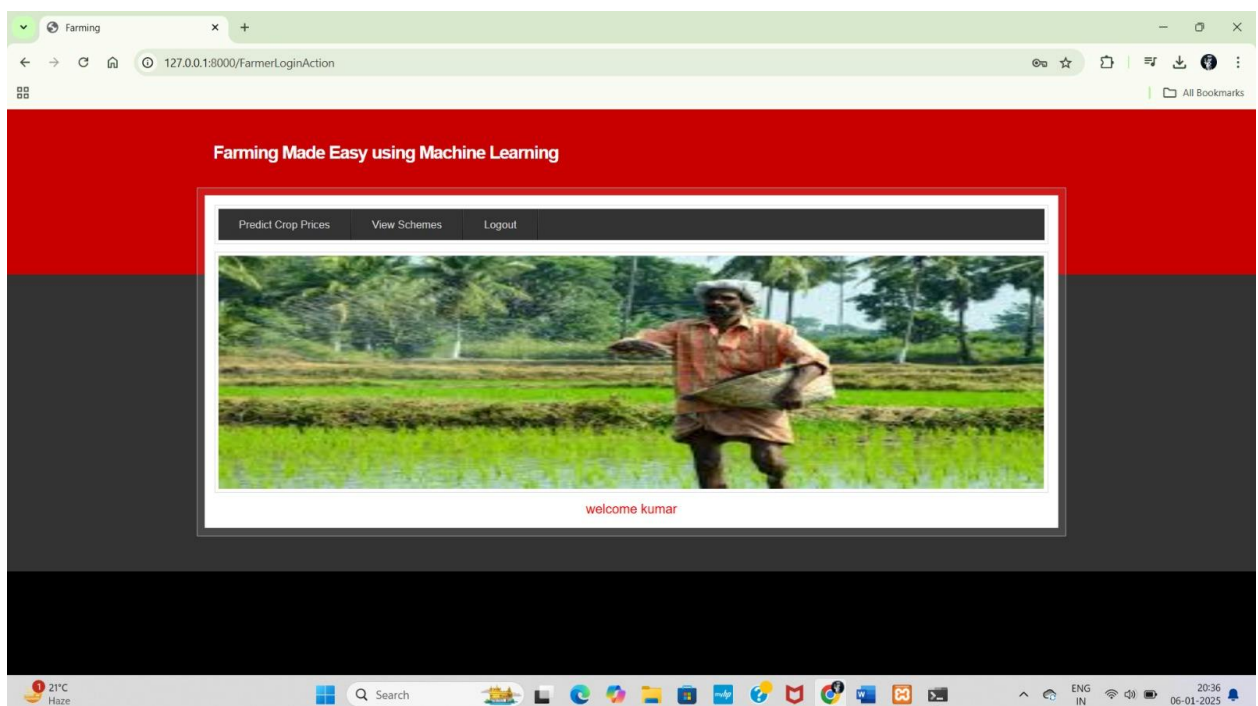


Figure 12.10: In above screen farmer can click on ‘**View Schemes**’ link to get all schemes details

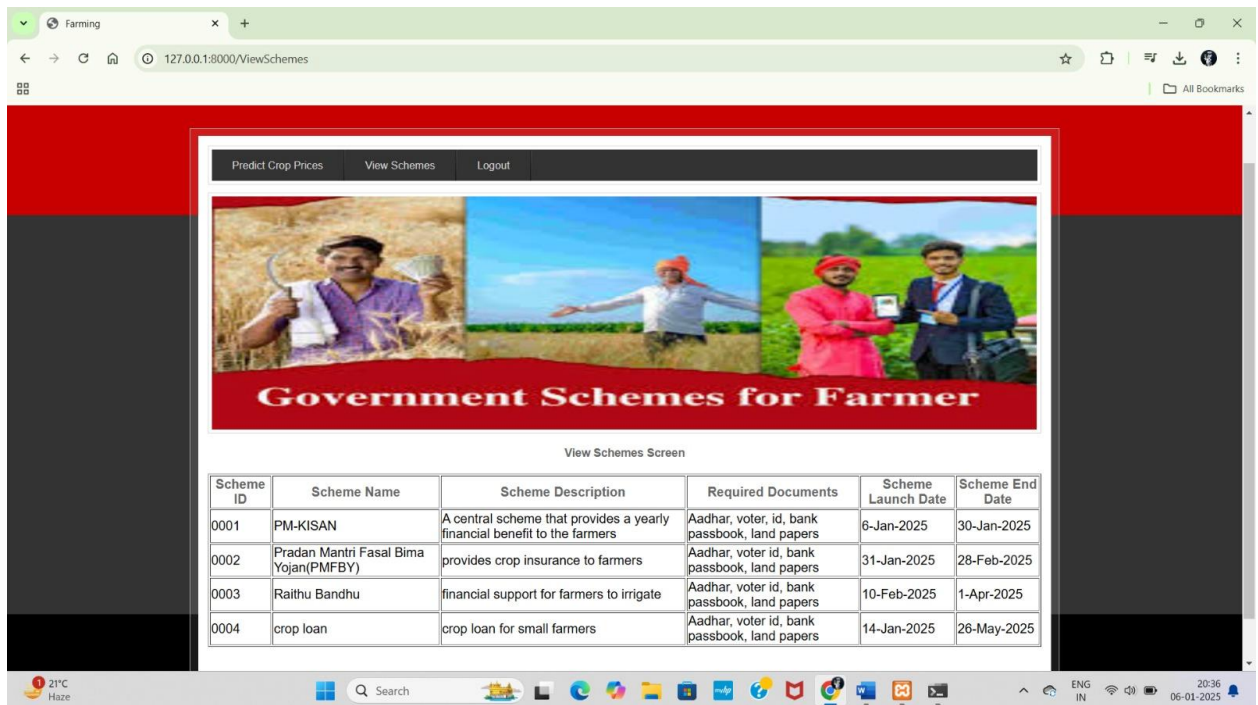


Figure 12.11: In above screen all schemes details can be viewed by farmer and now click on **‘Predict Crop Prices’** link to get below screen

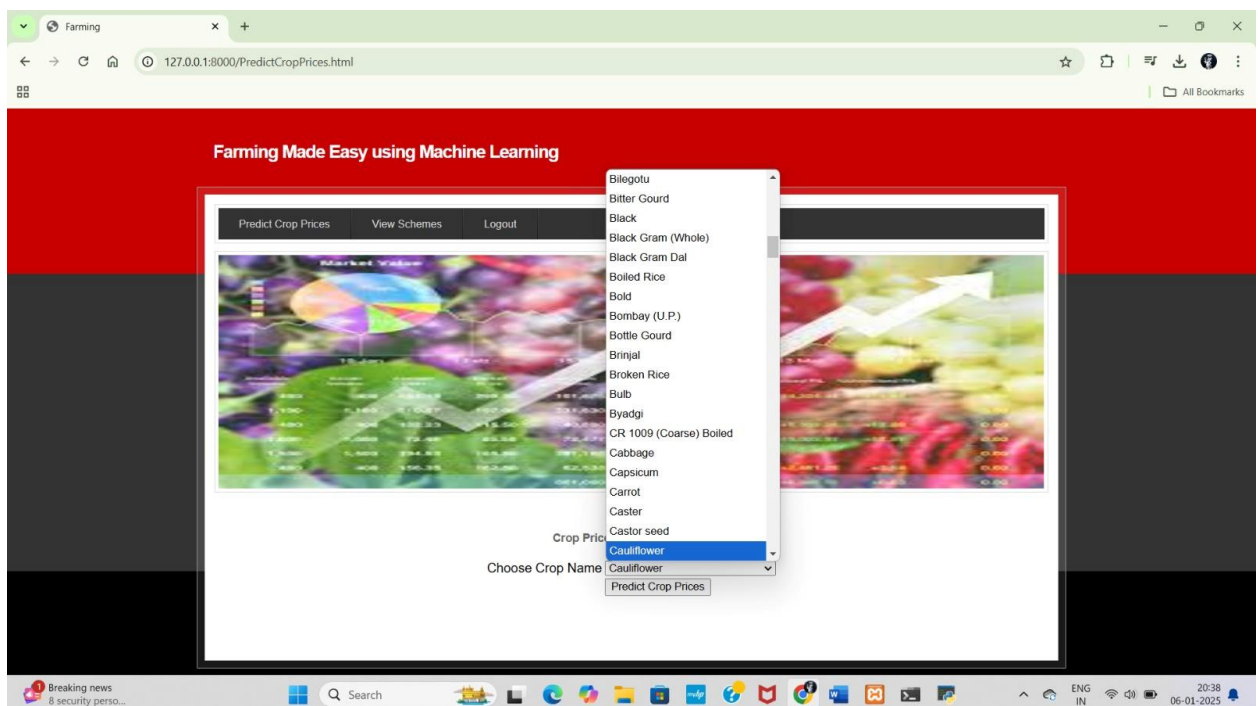


Figure 12.12: In above screen farmer can select desired crop and then click on **‘Predict Crop Prices’** link to get below prediction

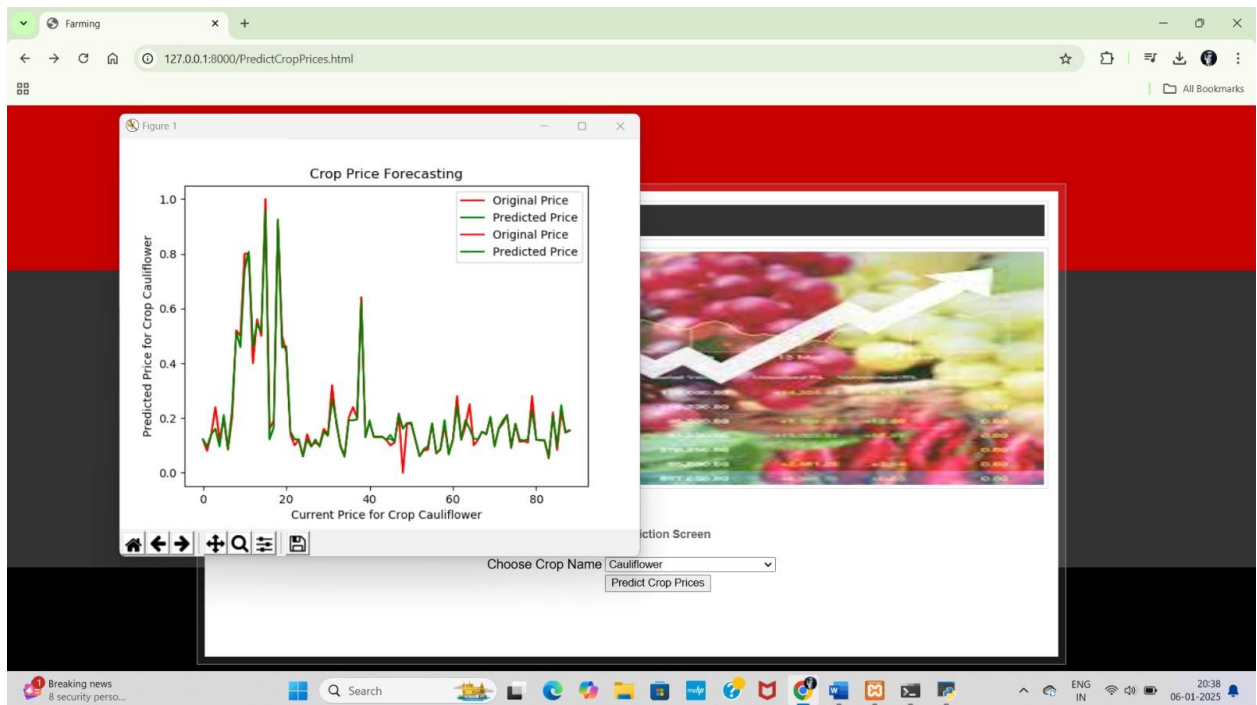


Figure 12.13: In above graph red line represents Original prices and green line represents predicted prices and by seeing above graph farmer can understand what is current price and what will be future price and now close above graph to view predicted values

District Market	Crop Name	Original Price	Predicted Price
Ambala	Cauliflower	600.0	607.6847763347764
Ambala	Cauliflower	400.0	474.98908806363914
Faridabad	Cauliflower	700.0	703.2166666666667
Karnal	Cauliflower	1200.0	873.4678571428572
Kurukshetra	Cauliflower	600.0	474.98908806363914
Mahendragarh-Narnaul	Cauliflower	1000.0	1052.8750000000002
Yamuna Nagar	Cauliflower	420.0	426.11068907373954
Lohardaga	Cauliflower	1000.0	1148.9633333333332
Ernakulam	Cauliflower	2600.0	2575.35000000000017
Ernakulam	Cauliflower	2500.0	2296.1670634920633
Ernakulam	Cauliflower	4000.0	3705.74999999999945
Kollam	Cauliflower	4000.0	4013.4999999999994
Kottayam	Cauliflower	2000.0	2296.1670634920633
Kottayam	Cauliflower	2800.0	2725.60000000000035
Palakad	Cauliflower	2500.0	2575.35000000000017
Thiruvananthapuram	Cauliflower	5000.0	4699.5
Dewas	Cauliflower	800.0	607.6847763347764
Delhi	Cauliflower	950.0	799.9723665223656
Bargarh	Cauliflower	4600.0	4531.5000000000003
Dhenkanal	Cauliflower	2500.0	2296.1670634920633
Jajpur	Cauliflower	2200.0	2296.1670634920633
Amritsar	Cauliflower	700.0	749.8047619047624
Amritsar	Cauliflower	500.0	607.6847763347764
Barnala	Cauliflower	600.0	607.6847763347764
Fatehgarh	Cauliflower	300.0	301.95384615384575
Fazilka	Cauliflower	700.0	602.3846807359306
Gurdasur	Cauliflower	500.0	474.98908806363914

Figure 12.14: 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

Firozabad	Cauliflower	600.0	607.6847763347764
Gautam Budh Nagar	Cauliflower	749.9999999999999	748.5666666666667
Ghaziabad	Cauliflower	700.0	703.2166666666667
Ghaziabad	Cauliflower	1000.0	1018.1785714285705
Jyotiba Phule Nagar	Cauliflower	500.0	474.98908806363914
Lakhimpur	Cauliflower	800.0	818.0958333333331
Lakhimpur	Cauliflower	900.0000000000001	916.9649999999992
Maharajganj	Cauliflower	1050.0	1052.8750000000002
Mainpuri	Cauliflower	450.00000000000006	474.98908806363914
Mainpuri	Cauliflower	890.0	894.6666666666661
Mathura	Cauliflower	565.0	602.3846807359306
Meerut	Cauliflower	575.0	583.0566666666667
Muzaffarnagar	Cauliflower	550.0	607.6847763347764
Muzaffarnagar	Cauliflower	1400.0	1148.9633333333327
Pilibhit	Cauliflower	600.0	599.1649999999995
Saharanpur	Cauliflower	600.0	590.9773809523805
Saharanpur	Cauliflower	580.0	590.9773809523805
Saharanpur	Cauliflower	260.0	275.8538461538458
Shahjahanpur	Cauliflower	1100.0	1052.8750000000002
Shahjahanpur	Cauliflower	415.0	474.98908806363914
Siddharth Nagar	Cauliflower	1100.0	1241.6000000000001
Sitapur	Cauliflower	749.9999999999999	731.0833333333344
Unnao	Cauliflower	775.0	764.7166666666666
Random Forest Accuracy		0.999172994725345	
Decision Tree Accuracy		0.9993231514999108	
KNN Accuracy		0.9988357462921348	

Figure 12.15: Crop price prediction list

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

CHAPTER 13

CONCLUSION

This project is undertaken using machine learning and evaluates the performance by using KNN, and Decision Tree algorithms. In our proposed model among all the three algorithm Random Forest 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%.

The **Crop Price Prediction** project demonstrates the potential of using machine learning techniques to analyze historical data and forecast crop prices effectively. By leveraging key features such as weather conditions, market demand, production trends, and government policies, the model provides valuable insights to farmers, traders, and policymakers.

Our results show that predictive models, when trained on high-quality and diverse datasets, can significantly reduce the uncertainty associated with agricultural markets. This can empower farmers to make informed decisions about planting, harvesting, and selling crops, leading to improved profitability and reduced losses.

However, the project also highlights certain limitations, including data availability, regional variability, and the impact of unpredictable external factors such as natural disasters or sudden policy changes. Future work could focus on integrating real-time data, incorporating satellite imagery, and exploring advanced models like deep learning to improve prediction accuracy further.

Overall, the project underscores the importance of technology in modernizing agriculture and creating a more sustainable and equitable system for all stakeholders in the agricultural value chain.

CHAPTER 14

REFERENCES

- 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
- Nishiba Kabeer, Dr.Loganathan.D, Cowsalya.T, Prediction of Crop Yield Using Data Mining, International Journal of Computer Science and Network, June 2019
- J. Vijayalakshmi, K. PandiMeena, Agriculture TalkBot Using AI, International Journal of Recent Technology and Engineering (IJRTE), July 2019
- 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
- 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.
- Nguyen, Huy Vuong, et al. "A smart system for short-term price prediction using time series models." Computers & Electrical Engineering 76 (2019)
- 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
- Rohith R, Vishnu R, Kishore A, Deeban Chakkarawarthy, Crop Price Prediction and Forecasting System using Supervised Machine Learning Algorithms, International Journal of Advanced Research in Computer and Communication Engineering, March 2020
- 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.
- 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.
- 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
- Rubhi gupta, Review on weather prediction using machine learning, International Journal of Engineering Development and Research, 2020