CROP YIELD RECOMMENDATION AND PREDICTION

A

MAJOR PROJECT REPORT

Submitted in the partial Fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By

1. B.Swathi	**20R21A0409**
2. K.Komali	**20R21A0430**
3.S.Mahesh kumar	**19R21A0447**

UNDER THE GUIDANCE OF

Mr.G.KAUSHIK
Assistant professor

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS

MLR Institute of Technology

(Autonomous)

(Affiliated to JNTUH, Hyderabad)
Dundigal, Hyderabad-500043
2020-2024

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS **MLR Institute of Technology**

(Autonomous)

(Affiliated to JNTUH, Hyderabad)

Dundigal, Hyderabad-500043

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project entitled *"AUTOMATIC STREETLIGHT AND FAULT DETECTION FOR SMART CITIES" is the bonafied work done* by B.Swathi (20R21A0409), K.Komali (20R21A0430),S.Maheshkumar (19R21A0447) in partial fulfillment of the requirement for the award of the degree of B. Tech in Electronics and Communication Engineering, during the academic year 2023-24.

Internal Guide
Mr.G.KAUSHIK

Head of the Department
Dr. S. V. S. PRASAD

External Examiner

ACKNOWLEDGEMENT

We express our profound thanks to the management of **MLR Institute of Technology**, Dundigal, Hyderabad, for supporting us to complete this project.

We take immense pleasure in expressing our sincere thanks to **Dr. K. Srinivasa Rao**, Principal, MLR Institute of Technology, for his kind support and encouragement.

We are very much grateful to **Dr S.V.S. Prasad**, Professor & Head of the Department, MLR Institute of Technology, for encouraging us with his valuable suggestions.

We are very much grateful to **Mr.G.Kaushik** assistant professor for his unflinching cooperation throughout the project.

We would like to express our sincere thanks to the teaching and non-teachingfaculty members of ECE Dept., MLR Institute of Technology, who extended their help to us in making our project work successful.

Project associates:

B.Swathi	20R21A0409
K.Komali	20R21A0430
S.Maheshkumar	19R21A0447

ABSTRACT

In many economies, including India and specifically Tamil Nadu, agriculture forms the backbone, supporting livelihoods and contributing significantly to the nation's GDP. However, the younger generation entering the farming sector often faces the challenge of making informed decisions on crop selection and anticipating yields. This critical issue is addressed through a novel approach deploying classification and regression algorithms to recommend crop types and predict yields.

The proposed system utilizes supervised machine learning techniques, encompassing comprehensive dataset analysis with variable identification, uni-variate, bi-variate, and multi-variate analyses, and missing value treatments. A thorough comparison of machine learning algorithms, including Random Forest, Linear Regression, KNN, XGBoost Classifier, Deep Q Network, and RNN, revealed superior accuracy in predicting optimal harvests.

The results showcase the efficacy of the proposed machine learning and deep learning algorithms, providing a holistic assessment through metrics such as entropy calculation, precision, recall, F1 score, sensitivity, specificity, and entropy. By accurately projecting yields for a wide array of crops cultivated in Tamil Nadu, the proposed system alleviates the burden on farmers, enabling them to navigate the complexities of agriculture with confidence. This innovative solution not only aids in reducing losses and managing price fluctuations but also empowers the agricultural community, especially the new generation, to make informed decisions, ultimately contributing to the sustainable growth of the agricultural sector.

TABLE OF CONTENTS

INTRODUCTION

1. INTRODUCTION

In the agrarian landscapes of economies like India, and specifically in regions such as Tamil Nadu, agriculture stands as the bedrock of livelihoods, playing a pivotal role in shaping the nation's economic landscape. However, a pressing challenge faced by the younger generation entering the farming sector revolves around the need for informed decision-making regarding crop selection and yield anticipation. Addressing this critical issue, a groundbreaking approach emerges, leveraging the power of classification and regression algorithms to recommend optimal crop types and predict yields with precision.

At the heart of this innovative solution lies a sophisticated system that embraces supervised machine learning techniques. This encompasses a meticulous process of dataset analysis, ranging from variable identification to uni-variate, bi-variate, and multi-variate analyses, coupled with robust missing value treatments. A comprehensive evaluation of various machine learning algorithms, including Random Forest, Linear Regression, KNN, XGBoost Classifier, Deep Q Network, and RNN, unfolds, ultimately revealing superior accuracy in predicting optimal harvests.

The results of this groundbreaking initiative underscore the efficacy of the proposed machine learning and deep learning algorithms. The system provides a holistic assessment through an array of metrics, such as entropy calculation, precision, recall, F1 score, sensitivity, specificity, and entropy. By adeptly projecting yields for a diverse range of crops cultivated in Tamil Nadu, the proposed system not only alleviates the burden on farmers but also empowers them,

particularly the newer generation, to navigate the complexities of agriculture with confidence.

In essence, this innovative solution goes beyond mitigating losses and managing price fluctuations. It serves as a beacon for the agricultural community, offering a tool that enables farmers to make well-informed decisions. This, in turn, contributes significantly to the sustainable growth of the agricultural sector. As a transformative force, the integration of machine learning and deep learning algorithms emerges as a cornerstone in fostering a resilient and empowered agricultural landscape, ensuring a brighter and more sustainable future for the sector.

1.1 OBJECTIVE

The objective of this groundbreaking initiative is to empower the younger generation in the agrarian landscapes of Tamil Nadu, India, by leveraging supervised machine learning techniques. The aim is to develop a sophisticated system that recommends optimal crop types and accurately predicts yields. Through comprehensive dataset analysis and algorithm evaluation, the objective is to provide farmers with a holistic assessment tool, utilizing metrics such as entropy, precision, recall, and F1 score. This innovative solution seeks to enable well-informed decision-making, mitigating losses, and contributing to the sustainable growth of the agricultural sector.

1.2 PROBLEM STATEMENT

In the agrarian regions of Tamil Nadu, India, the younger generation entering the farming sector faces a critical challenge in making informed decisions regarding crop selection and yield anticipation. The lack of precise tools often leads to uncertainties, impacting livelihoods. This initiative addresses the pressing need for

a solution by proposing a system based on supervised machine learning techniques. The problem lies in the absence of a reliable method to recommend optimal crop types and predict yields accurately. The proposed solution aims to bridge this gap, providing a comprehensive tool that empowers farmers, particularly the newer generation, to navigate the complexities of agriculture with confidence, ultimately contributing to the sector's sustainable growth.

1.3 SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Platform – In computing, a platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer’s architecture, operating system, or programming languages and their runtime libraries.

Operating system is one of the first requirements mentioned when defining system requirements (software). Software may not be compatible with different versions of same line of operating systems, although some measure of backward compatibility is often maintained. For example, most software designed for Microsoft Windows XP does not run on Microsoft Windows 98, although the converse is not always true. Similarly, software designed using newer features of Linux Kernel v2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v2.4.

APIs and drivers – Software making extensive use of special hardware devices, like high-end display adapters, needs special API or newer device drivers. A good example is DirectX, which is a collection of APIs for handling tasks related to multimedia, especially game programming, on Microsoft platforms.

Web browser – Most web applications and software depending heavily on Internet technologies make use of the default browser installed on system. Microsoft Internet Explorer is a frequent choice of software running on Microsoft Windows, which makes use of ActiveX controls, despite their vulnerabilities.

1. **Anaconda**

1.2 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Architecture – All computer operating systems are designed for a particular computer architecture. Most software applications are limited to particular operating systems running on particular architectures. Although architecture-independent operating systems and applications exist, most need to be recompiled

to run on a new architecture. See also a list of common operating systems and their supporting architectures.

Processing power – The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on x86 architecture define processing power as the model and the clock speed of the CPU. Many other features of a CPU that influence its speed and power, like bus speed, cache, and MIPS are often ignored. This definition of power is often erroneous, as AMD Athlon and Intel Pentium CPUs at similar clock speed often have different throughput speeds. Intel Pentium CPUs have enjoyed a considerable degree of popularity, and are often mentioned in this category.

Memory – All software, when run, resides in the random access memory (RAM) of a computer. Memory requirements are defined after considering demands of the application, operating system, supporting software and files, and other running processes. Optimal performance of other unrelated software running on a multi-tasking computer system is also considered when defining this requirement.

Secondary storage – Hard-disk requirements vary, depending on the size of software installation, temporary files created and maintained while installing or running the software, and possible use of swap space (if RAM is insufficient).

Display adapter – Software requiring a better than average computer graphics display, like graphics editors and high-end games, often define high-end display adapters in the system requirements.

Peripherals – Some software applications need to make extensive and/or special use of some peripherals, demanding the higher performance or functionality of

such peripherals. Such peripherals include CD-ROM drives, keyboards, pointing devices, network devices, etc.

1) Operating System : Windows Only

2) Processor : i5 and above

3) Ram : 8gb and above

4) Hard Disk : 25 GB in local drive

FEASIBILITY STUDY

2. 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

2.1 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.

2.2 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.

2.3 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.

LITERATURE SURVEY

3. LITERATURE SURVEY

3.1 Supervised Machine learning Approach for Crop Yield Prediction in Agriculture Sector:

https://www.researchgate.net/publication/342860043_Supervised_Machine_learning_Approach_for_Crop_Yield_Prediction_in_Agriculture_Sector

ABSTRACT: Machine learning (ML) is a crucial perspective for acquiring real-world and operative solution for crop yield issue. From a given set of predictors, ML can predict a target/outcome by using Supervised Learning. To get the desired outputs need to generate a suitable function by set of some variables which will map the input variable to the aim output. Crop yield prediction incorporates forecasting the yield of the crop from past historical data which includes factors such as temperature, humidity, ph, rainfall, crop name. It gives us an idea for the finest predicted crop which will be cultivate in the field weather conditions. These predictions can be done by a machine learning algorithm called Random Forest. It will attain the crop prediction with best accurate value. The algorithm random forest is used to give the best crop yield model by considering least number of models. It is very useful to predict the yield of the crop in agriculture sector

3.2 Prediction of Land Suitability for Crop Cultivation Based on Soil and Environmental Characteristics Using Modified Recursive Feature Elimination Technique With Various Classifiers:

https://www.researchgate.net/publication/354998192_Prediction_of_Land_Suitability_for_Crop_Cultivation_Based_on_Soil_and_Environmental_Characteristics_Using_Modified_Recursive_Feature_Elimination_Technique_With_Various_Classifiers

ABSTRACT: Crop cultivation prediction is an integral part of agriculture and is primarily based on factors such as soil, environmental features like rainfall and temperature, and the quantum of fertilizer used, particularly nitrogen and phosphorus. These factors, however, vary from region to region: consequently, farmers are unable to cultivate similar crops in every region. This is where machine learning (ML) techniques step in to help find the most suitable crops for a particular region, thus assisting farmers a great deal in crop prediction. The feature selection (FS) facet of ML is a major component in the selection of key features for a particular region and keeps the crop prediction process constantly upgraded. This work proposes a novel FS approach called modified recursive feature elimination (MRFE) to select appropriate features from a data set for crop prediction. The proposed MRFE technique selects and ranks salient features using a ranking method. The experimental results show that the MRFE method selects the most accurate features, while the bagging technique helps accurately predict a suitable crop. The performance of proposed MRFE technique is evaluated by various metrics such as accuracy (ACC), precision, recall, specificity, F1 score, area under the curve, mean absolute error, and log loss. From the performance analysis, it is justified that the MRFE technique performs well with 95% ACC than other FS methods.

3.3 **Ascertaining the Fluctuation of Rice Price in Bangladesh Using Machine Learning Approach**:

https://ieeexplore.ieee.org/document/9225468

ABSTRACT: Rice is the most grown crop in Bangladesh. It is consumed as the main food course in Bangladesh. The price of rice makes a difference in whether people will eat or starve. To know what's going to happen in the rice market using

pen and paper is a far cry as well as time-consuming. Machine Learning (ML) provides the facilities to predict the price of any products to prevent a future collapse in the market. The goal of this paper is to predict the price of rice using Machine learning approach. Data collected from the Ministry of Agriculture website, Bangladesh was used to predict the price. Several machine learning algorithms were used to make this prediction i.e. Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes, Decision Tree and Random Forest. All these algorithms are analyzed to find out which algorithm provides the best performance. Now, we can predict the price of rice, whether it is reasonable, low, or high based on the results achieved by the mentioned algorithms.

3.4 Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique:

https://www.researchgate.net/publication/301671246_Crop_Selection_Method_to_Maximize_Crop_Yield_Rate_using_Machine_Learning_Technique

ABSTRACT: Agriculture planning plays a significant role in economic growth and food security of agro-based country. Se- lection of crop(s) is an important issue for agriculture planning. It depends on various parameters such as production rate, market price and government policies. Many researchers studied prediction of yield rate of crop, prediction of weather, soil classification and crop classification for agriculture planning using statistics methods or machine learning techniques. If there is more than one option to plant a crop at a time using limited land resource, then selection of crop is a puzzle. This paper proposed a method named Crop Selection Method (CSM) to solve crop selection problem, and maximize net yield rate of crop over season and subsequently achieves maximum economic growth of the country. The proposed method may improve net yield rate of crops.

3.5 Crop Recommender System Using Machine Learning Approach:

https://jpinfotech.org/crop-recommender-system-using-machine-learning-approach/

ABSTRACT: Agriculture and its allied sectors are undoubtedly the largest providers of livelihoods in rural India. The agriculture sector is also a significant contributor factor to the country's Gross Domestic Product (GDP). Blessing to the country is the overwhelming size of the agricultural sector. However, regrettable is the yield per hectare of crops in comparison to international standards. This is one of the possible causes for a higher suicide rate among marginal farmers in India. This paper proposes a viable and user-friendly yield prediction system for the farmers. The proposed system provides connectivity to farmers via a mobile application. GPS helps to identify the user location. The user provides the area & soil type as input. Machine learning algorithms allow choosing the most profitable crop list or predicting the crop yield for a user-selected crop. To predict the crop yield, selected Machine Learning algorithms such as Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest (RF), Multivariate Linear Regression (MLR), and K-Nearest Neighbour (KNN) are used. Among them, the Random Forest showed the best results with 95% accuracy. Additionally, the system also suggests the best time to use the fertilizers to boost up the yield.

SYSTEM ANALYSIS

4. SYSTEM ANALYSIS

4.1 EXISTING SYSTEM:

In existing work a systematic literature review (SLR) was conducted to analyze 567 studies on machine learning applications for crop yield prediction. From the selected 50 studies, key features like temperature, rainfall, and soil type were identified, with Artificial Neural Networks being the predominant algorithm. A subsequent search for deep learning-based studies yielded 30 papers, revealing Convolutional Neural Networks (CNN) as the most employed algorithm, alongside notable use of Long-Short Term Memory (LSTM) and Deep Neural Networks (DNN). This comprehensive analysis informs decision-making in agriculture, guiding crop selection and management strategies during the growing season. Future research could explore the integration of these findings for enhanced precision in crop yield predictions.

4.1.1 DISADVANTAGES OF EXISTING SYSTEM:

1. Limited diversity in algorithms may hinder innovation and robustness.
2. Overemphasis on key features may overlook emerging influential factors.
3. Reliance on historical data may lead to outdated predictions.
4. Complexity of deep learning models may hinder interpretability and adoption.
5. Neglect of socio-economic factors may limit holistic agricultural decision-making.

4.2 Proposed System:

The proposed system for Tamil Nadu's agriculture employs advanced supervised machine learning techniques, featuring comprehensive dataset analysis, variable identification, and missing value treatments. Leveraging classification and regression algorithms, including Random Forest, Linear Regression, KNN, XGBoost Classifier, Deep Q Network, and RNN, the system recommends crop types and predicts yields with superior accuracy. Through thorough metric assessments like entropy, precision, recall, F1 score, sensitivity, and specificity, the system provides a holistic evaluation. This innovative solution empowers the younger generation in farming, enabling informed decisions, reducing losses, managing price fluctuations, and fostering sustainable growth in Tamil Nadu's agricultural sector.

4.2.1 Advantages of proposed system:

1. Enhanced decision-making for farmers with accurate crop recommendations and yield predictions.
2. Efficient resource utilization through advanced data analysis and variable identification.
3. Minimized losses by managing price fluctuations and optimizing agricultural practices.
4. Empowering the younger generation in farming with cutting-edge technology.
5. Sustainable growth fostered in Tamil Nadu's agricultural sector through informed decisions.

4.3 FUNCTIONAL REQUIREMENTS

1. Data Collection
2. Data Pre-processing

3. Training and Testing
4. Modiling
5. Predicting

4.4 NON FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *"how fast does the website load?"* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non-functional Requirements allow you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users is > 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

SYSTEM DESIGN

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE:

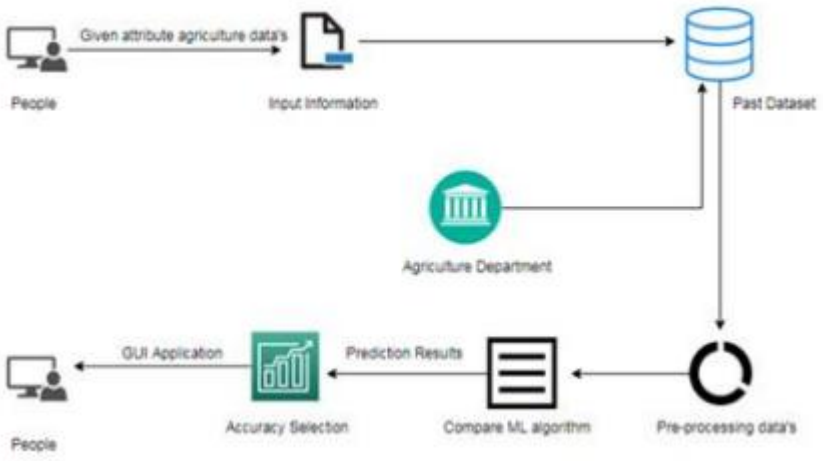
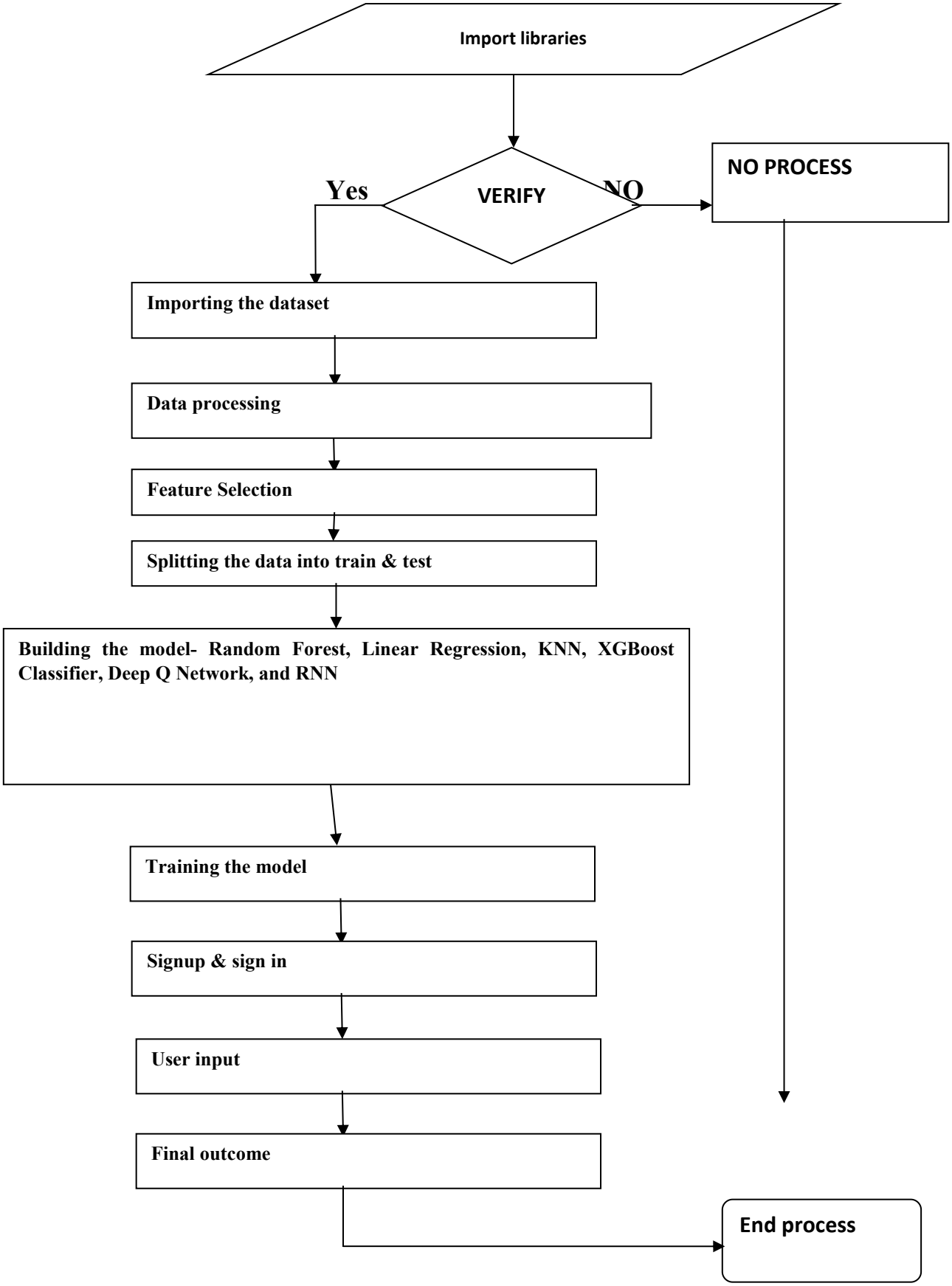


Fig.5.1.1 System architecture

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.


Import libraries
VERIFY
Yes
NO
NO PROCESS
Importing the dataset
Data processing
Feature Selection
Splitting the data into train & test
Building the model- Random Forest, Linear Regression, KNN, XGBoost Classifier, Deep Q Network, and RNN
Training the model
Signup & sign in
User input
Final outcome
End process

5.2 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:

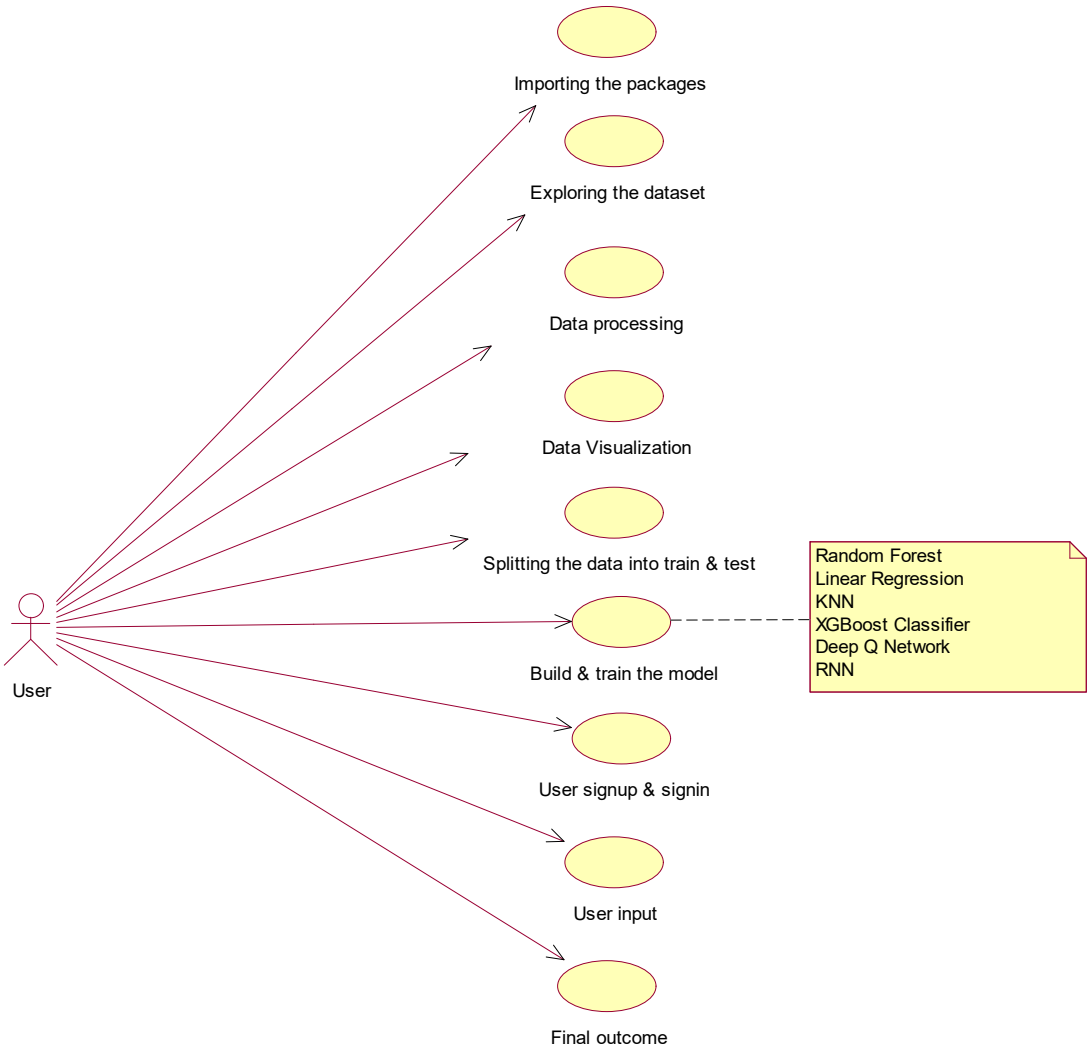
The Primary goals in the design of the UML are as follows:

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.

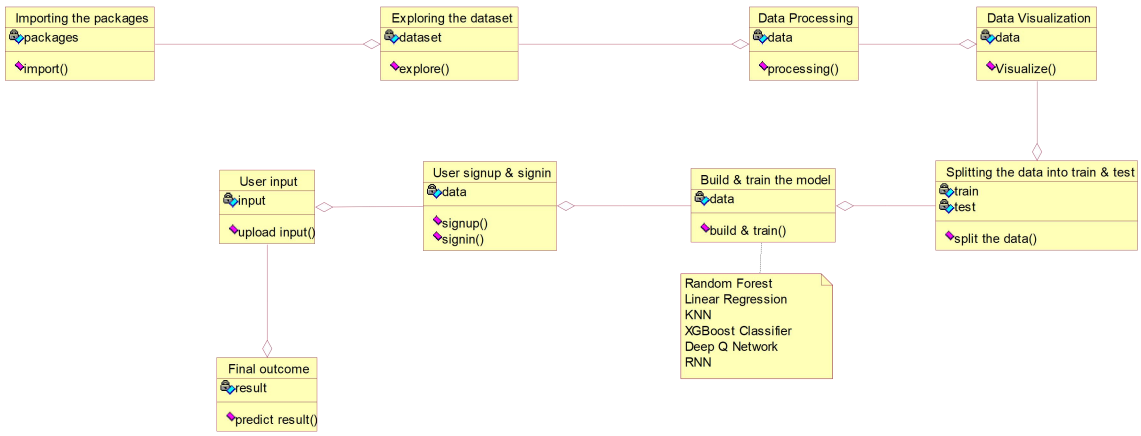
Use case diagram:

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.



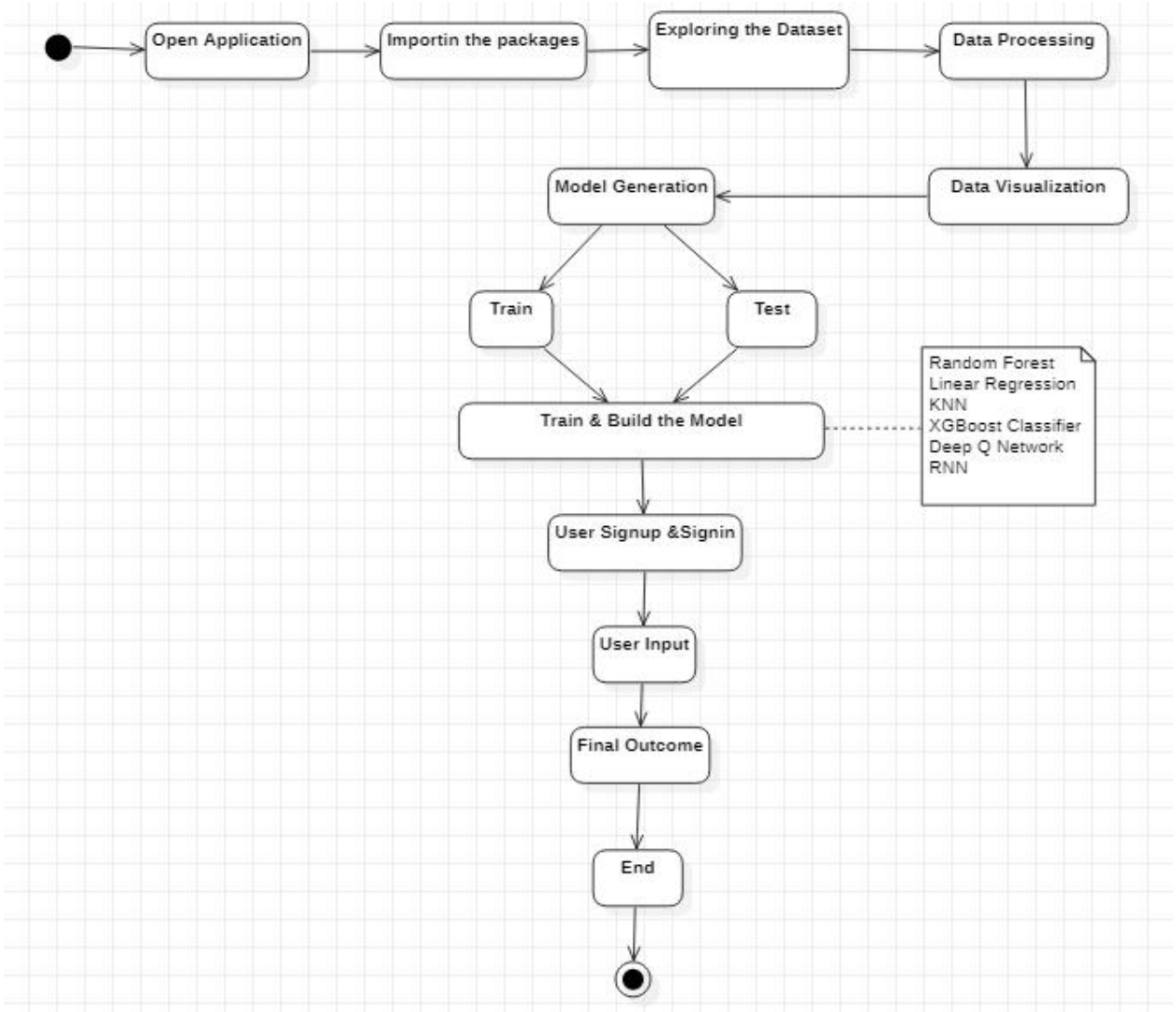
Class diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



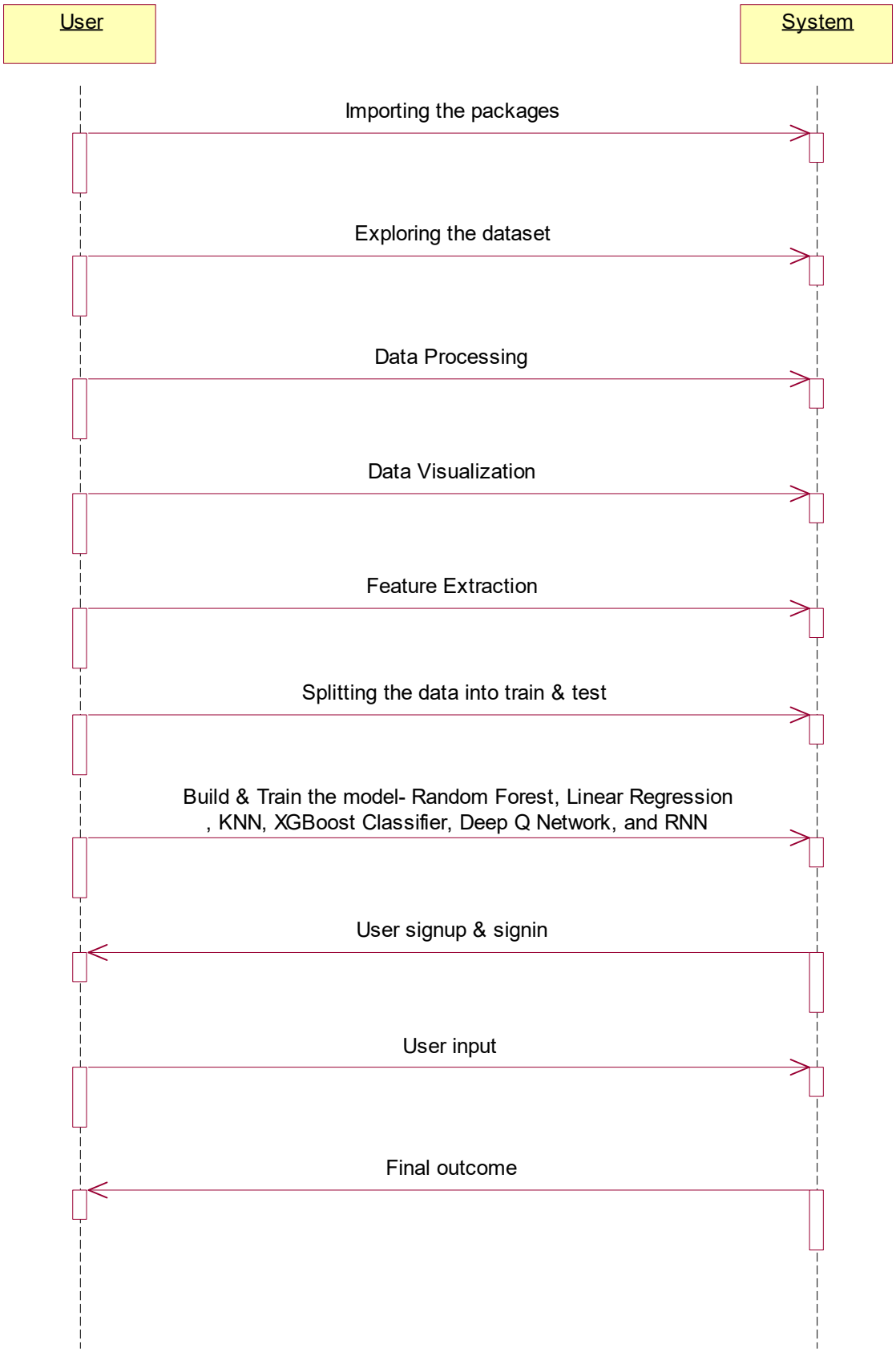
Activity diagram:

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.



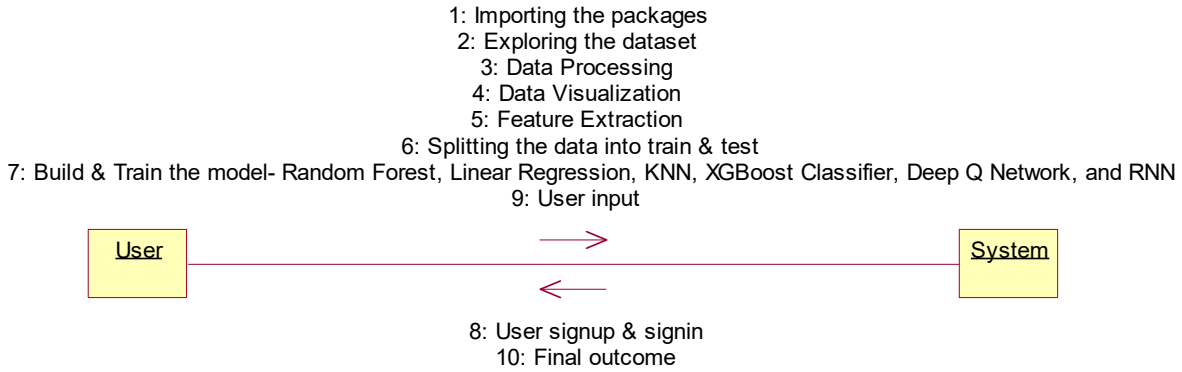
Sequence diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".


User
System
Importing the packages
Exploring the dataset
Data Processing
Data Visualization
Feature Extraction
Splitting the data into train & test
Build & Train the model- Random Forest, Linear Regression
, KNN, XGBoost Classifier, Deep Q Network, and RNN
User signup & signin
User input
Final outcome

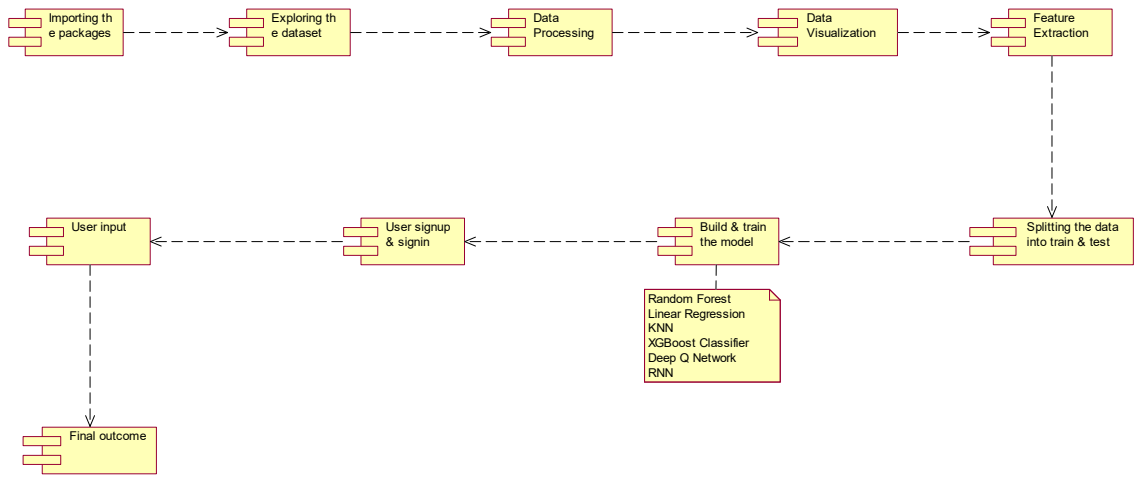
Collaboration diagram:

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.



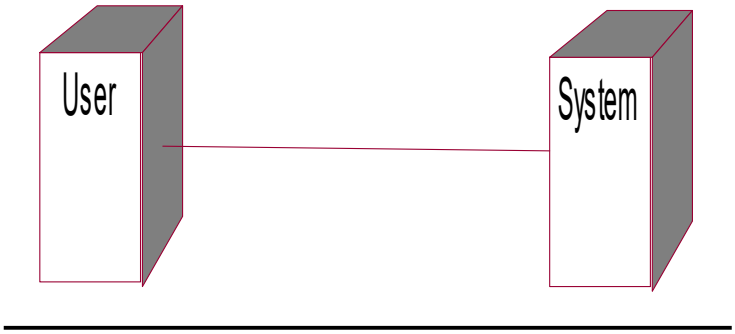
Component diagram:

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.



Deployment diagram:

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.



IMPLEMENTATION

6. IMPLEMENTATION

MODULES:

- Data exploration: using this module we will load data into system
- Processing: Using the module we will read data for processing
- Splitting data into train & test: using this module data will be divided into train & test
- Model generation: Model building
    - Random Forest, Linear Regression, KNN, XGBoost Classifier, Deep Q Network, and RNN
- User signup & login: Using this module will get registration and login
- User input: Using this module will give input for prediction
- Prediction: final predicted displayed

Algorithms:

Random Forest:

Random Forest is an ensemble learning algorithm used for both classification and regression tasks. It builds multiple decision trees and merges their predictions to improve accuracy and reduce over fitting. Each tree is trained on a random subset of the data, and the final prediction is based on a majority vote or average of individual tree predictions. This approach enhances the model's robustness and generalization performance.

Linear Regression:

Linear Regression is a fundamental statistical method used for modeling the relationship between a dependent variable and one or more independent variables. The model assumes a linear relationship, where changes in the independent variables lead to proportional changes in the dependent variable. The algorithm estimates the coefficients that define this linear relationship, allowing predictions of the dependent variable based on new input data. Linear Regression is widely employed in various fields due to its simplicity and interpretability.

K-Nearest Neighbors (KNN):

K-Nearest Neighbors is a non-parametric and instance-based learning algorithm used for both classification and regression tasks. It makes predictions by identifying the majority class or averaging values of the k-nearest data points in the feature space. KNN is simple to understand and implement but can be sensitive to irrelevant or redundant features.

XGBoost Classifier:

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm known for its high performance in classification and regression tasks. It is an ensemble model that combines the predictions of multiple weak learners, typically decision trees, in an additive manner. XGBoost incorporates regularization techniques, tree pruning, and parallel processing to optimize model accuracy and efficiency.

Deep Q Network (DQN):

Deep Q Network is a reinforcement learning algorithm that combines deep neural networks with Q-learning. It is used for training agents to make decisions in environments with discrete action spaces. DQN employs a neural network to

approximate the Q-function, representing the expected future rewards for taking a particular action in a given state. This enables the agent to learn optimal policies through exploration and exploitation strategies.

Recurrent Neural Network (RNN):

Recurrent Neural Network is a type of neural network designed for sequential data processing. Unlike traditional feedforward neural networks, RNNs have connections that form directed cycles, allowing them to capture temporal dependencies in the data. RNNs are commonly used in natural language processing, speech recognition, and time series prediction tasks. However, they face challenges such as vanishing and exploding gradients, which led to the development of more advanced variants like Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU).

6.2 SAMPLE CODE:

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score
from flask import Flask, request, jsonify, session, redirect, url_for


# Data Exploration & Processing Module
def load_data(file_path):
```

```
    return pd.read_csv(file_path)

# Splitting data into train & test Module
def split_data(data, target_column):
    X = data.drop(columns=[target_column])
    y = data[target_column]
    return train_test_split(X, y, test_size=0.2, random_state=42)

# Model Generation Module
def build_model(model_type):
    if model_type == 'RandomForest':
        return RandomForestClassifier()
    elif model_type == 'LinearRegression':
        return LinearRegression()
    elif model_type == 'KNN':
        return KNeighborsClassifier()
    elif model_type == 'XGBoost':
        return XGBClassifier()
    # Add other models like Deep Q Network and RNN as needed

# User Signup & Login Module (using Flask for simplicity)
app = Flask(__name__)
app.secret_key = 'your_secret_key'

@app.route('/signup', methods=['POST'])
def signup():
    # Handle user registration logic
```

```
    # ...

@app.route('/login', methods=['POST'])
def login():
    # Handle user login logic
    # ...

# User Input & Prediction Module
@app.route('/predict', methods=['POST'])
def predict():
    if 'user_id' not in session:
        return redirect(url_for('login'))

    # Get user input for prediction
    user_input = request.json
    # ...

    # Load data, split, and train model
    data = load_data('your_data.csv')
    X_train, X_test, y_train, y_test = split_data(data, 'target_column')
    model_type = 'RandomForest'  # Choose the appropriate model type
    model = build_model(model_type)
    model.fit(X_train, y_train)

    # Make predictions
    predictions = model.predict(X_test)
```

```
    # Display final predictions
    return jsonify({'predictions': predictions.tolist()})

if __name__ == '__main__':
    app.run(debug=True)
```

SOFTWARE ENVIRONMENT

7. SOFTWARE ENVIRONMENT

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.

Challenges in Machines Learning:-

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

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

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

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

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

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

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

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

DEEP LEARNING

Deep learning is a branch of machine learning which is based on artificial neural networks. It is capable of learning complex patterns and relationships within data. In deep learning, we don’t need to explicitly program everything. It has become increasingly popular in recent years due to the advances in processing power and the availability of large datasets. Because it is based on

artificial neural networks (ANNs) also known as deep neural networks (DNNs). These neural networks are inspired by the structure and function of the human brain's biological neurons, and they are designed to learn from large amounts of data.

What is Anaconda for Python?

Anaconda Python is a free, open-source platform that allows you to write and execute code in the programming language Python. It is by continuum.io, a company that specializes in Python development. The Anaconda platform is the most popular way to learn and use Python for scientific computing, data science, and machine learning. It is used by over thirty million people worldwide and is available for Windows, macOS, and Linux.

People like using Anaconda Python because it simplifies package deployment and management. It also comes with a large number of libraries/packages that you can use for your projects. Since Anaconda Python is free and open-source, anyone can contribute to its development.

What is Anaconda for Python?

Anaconda software helps you create an environment for many different versions of Python and package versions. Anaconda is also used to install, remove, and upgrade packages in your project environments. Furthermore, you may use Anaconda to deploy any required project with a few mouse clicks. This is why it is perfect for beginners who want to learn Python.

Now that you know what Anaconda Python is, let's look at how to install it.

How to install Anaconda for Python?

To install Anaconda, just head to the Anaconda Documentation website and follow the instructions to download the installer for your operating system. Once the installer successfully downloads, double-click on it to start the installation process.

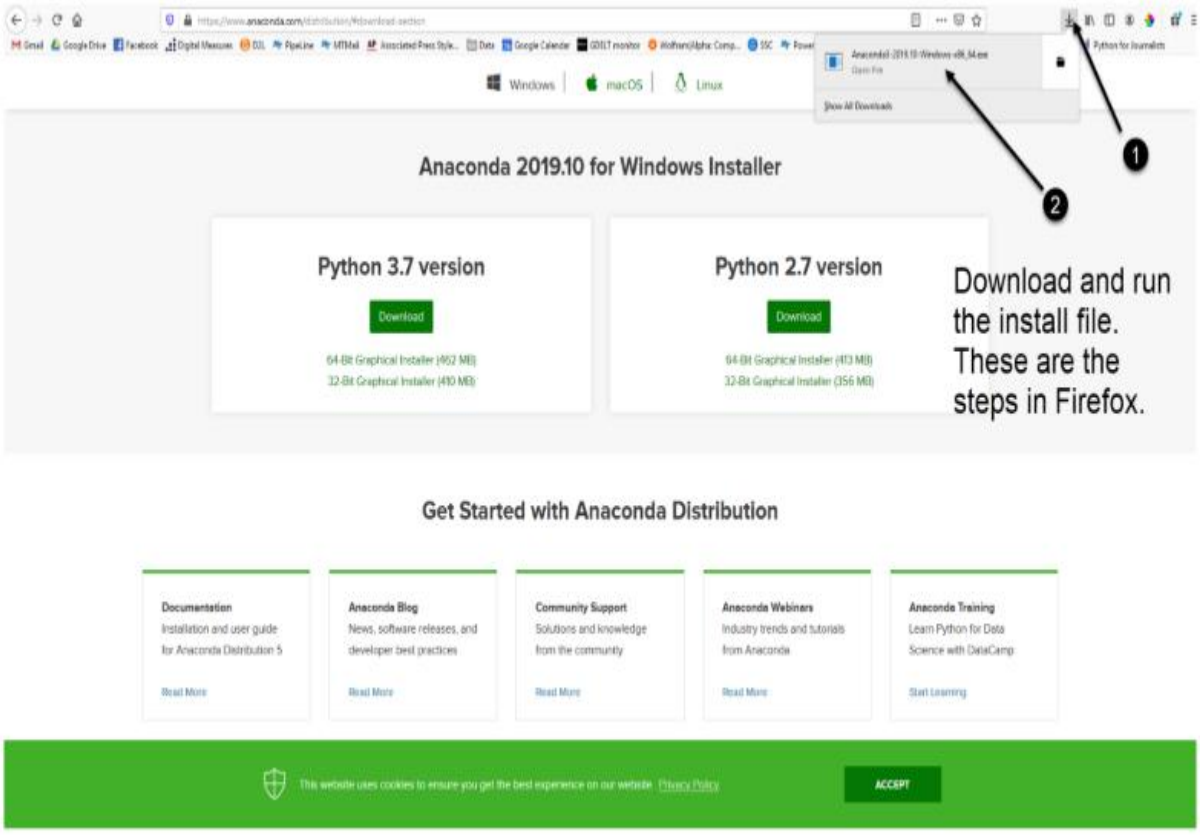
Follow the prompts and agree to the terms and conditions. When you are asked if you want to "add Anaconda to my PATH environment variable," make sure that you select "yes." This will ensure that Anaconda is added to your system's PATH, which is a list of directories that your operating system uses to find the files it needs.

Once the installation is complete, you will be asked if you want to "enable Anaconda as my default Python." We recommend selecting "yes" to use Anaconda as your default Python interpreter.

Python Anaconda Installation

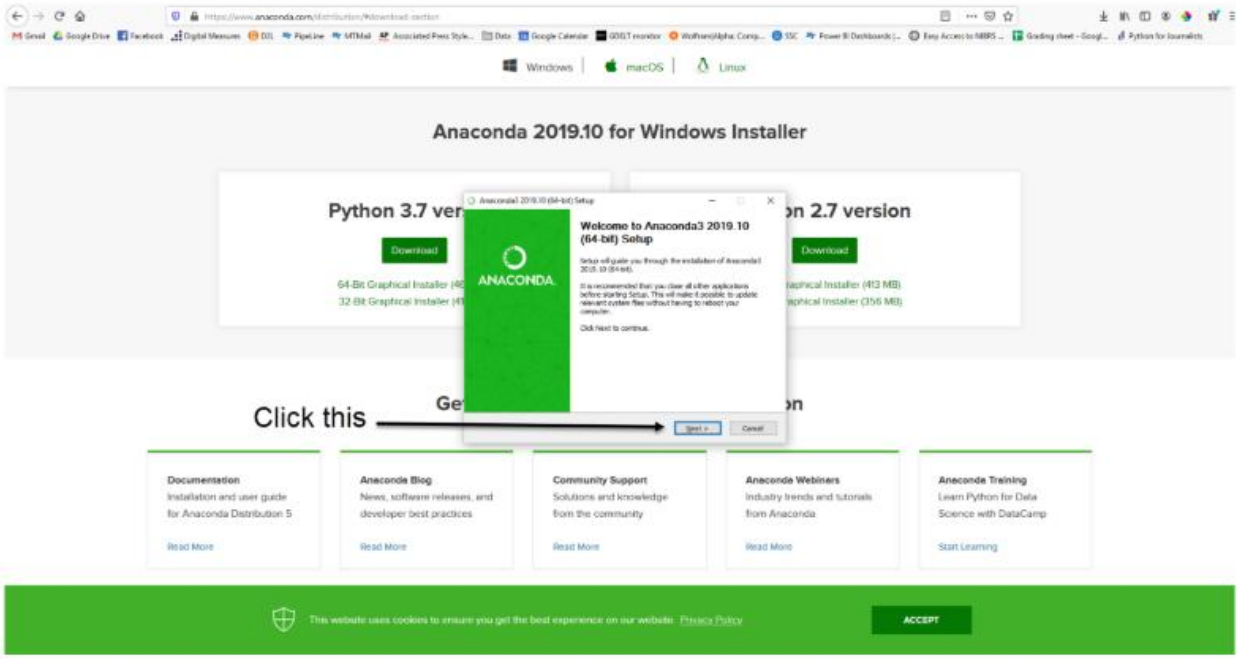
Next in the Python anaconda tutorial is its installation. The latest version of Anaconda at the time of writing is 2019.10. Follow these steps to download and install Anaconda on your machine:

1. Go to this link and download Anaconda for Windows, Mac, or Linux: – Download anaconda

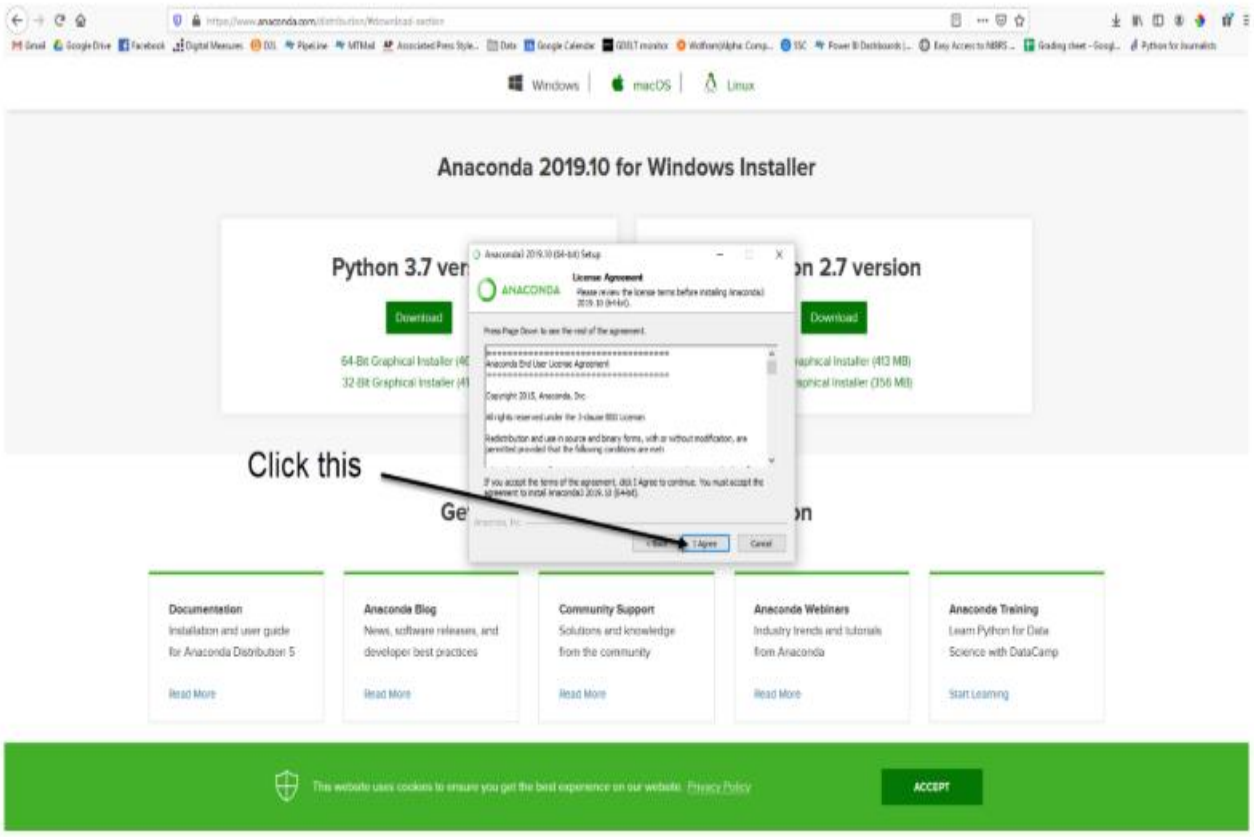


You can download the installer for Python 3.7 or for Python 2.7 (at the time of writing). And you can download it for a 32-bit or 64-bit machine.

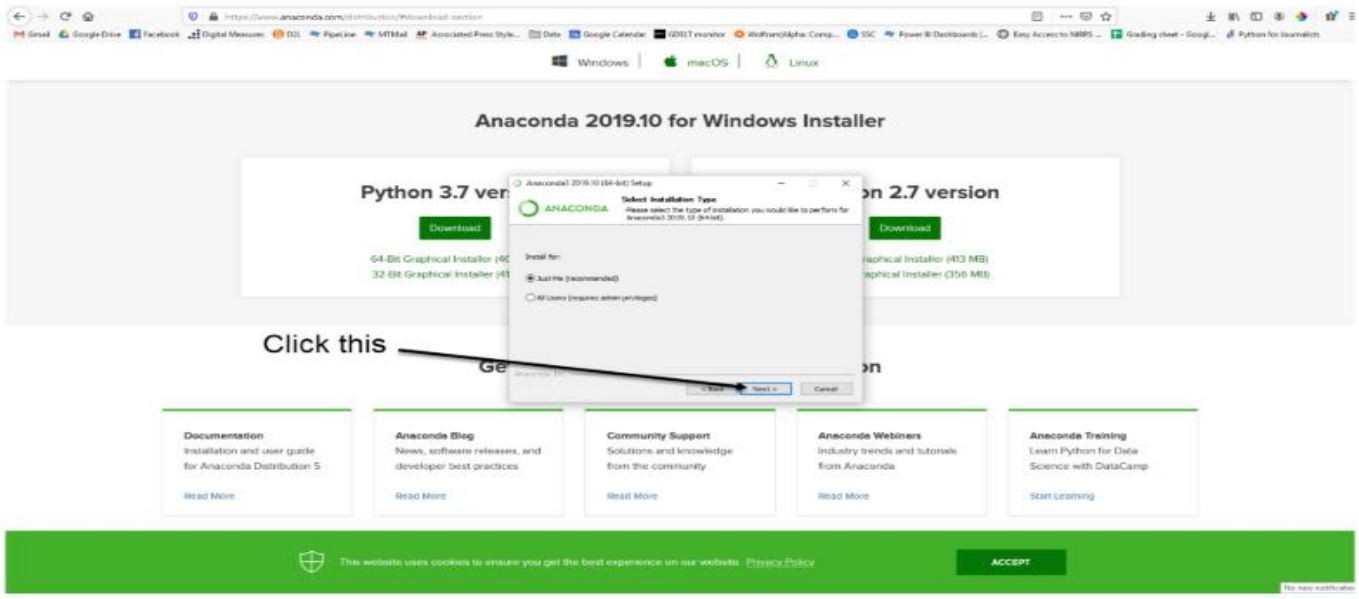
2. Click on the downloaded .exe to open it. This is the Anaconda setup. Click next.



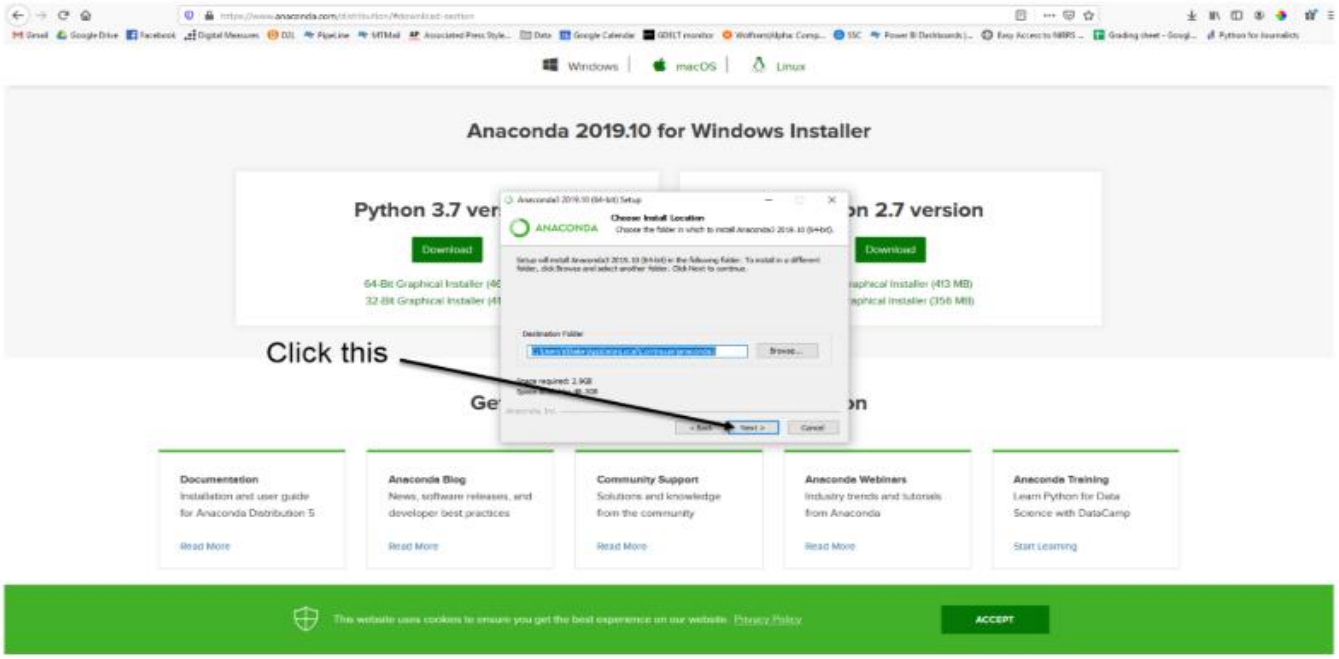
3. Now, you'll see the license agreement. Click on 'I Agree'.



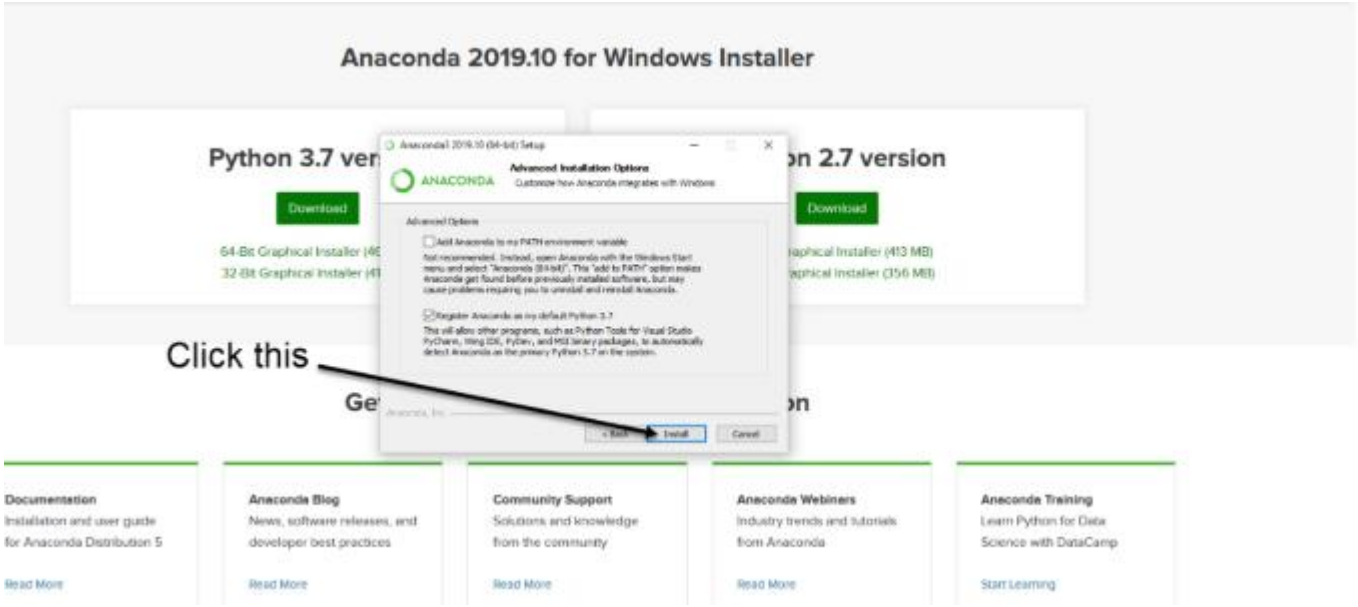
4. You can install it for all users or just for yourself. If you want to install it for all users, you need administrator privileges.



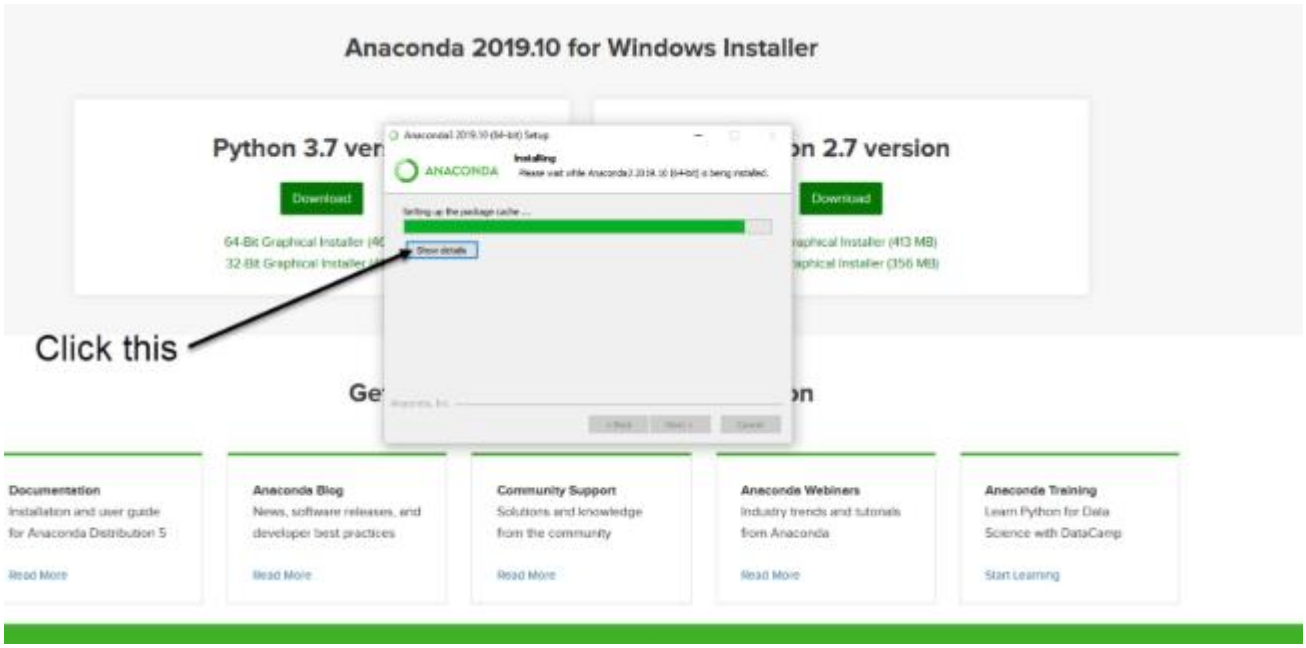
5. Choose where you want to install it. Here, you can see the available space and how much you need.



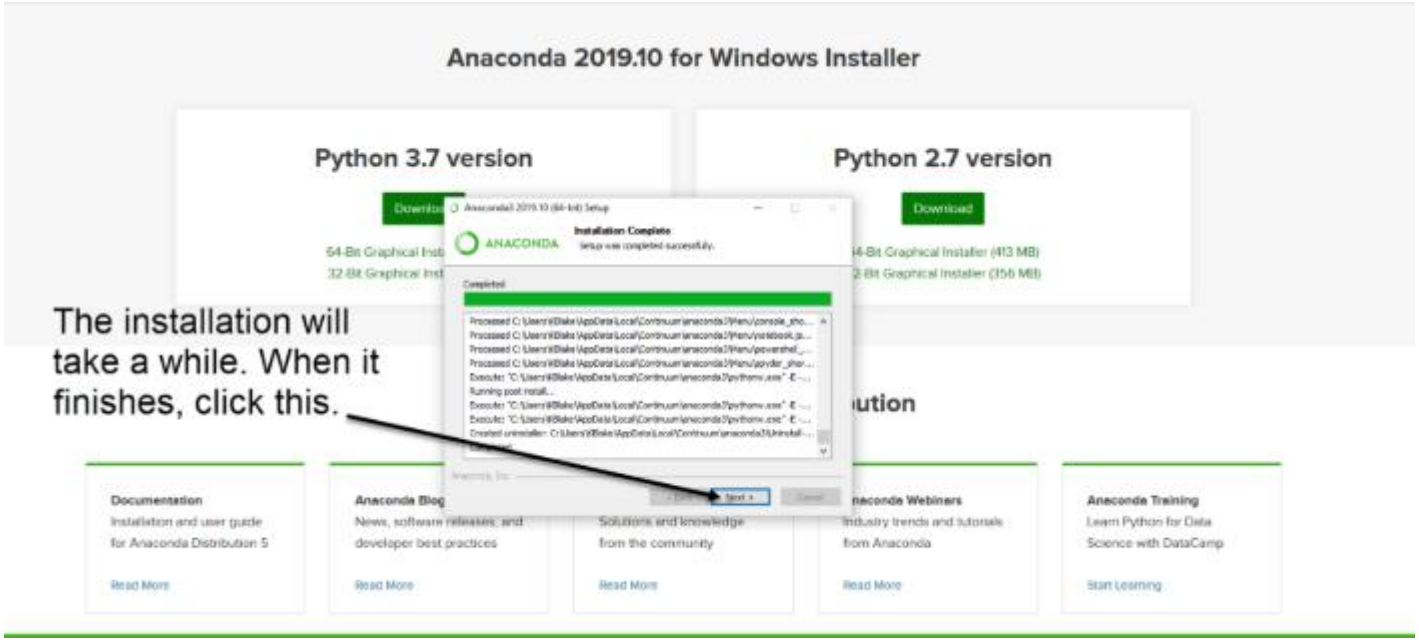
6. Now, you'll get some advanced options. You can add Anaconda to your system's PATH environment variable, and register it as the primary system Python 3.7. If you add it to PATH, it will be found before any other installation. Click on 'Install'.



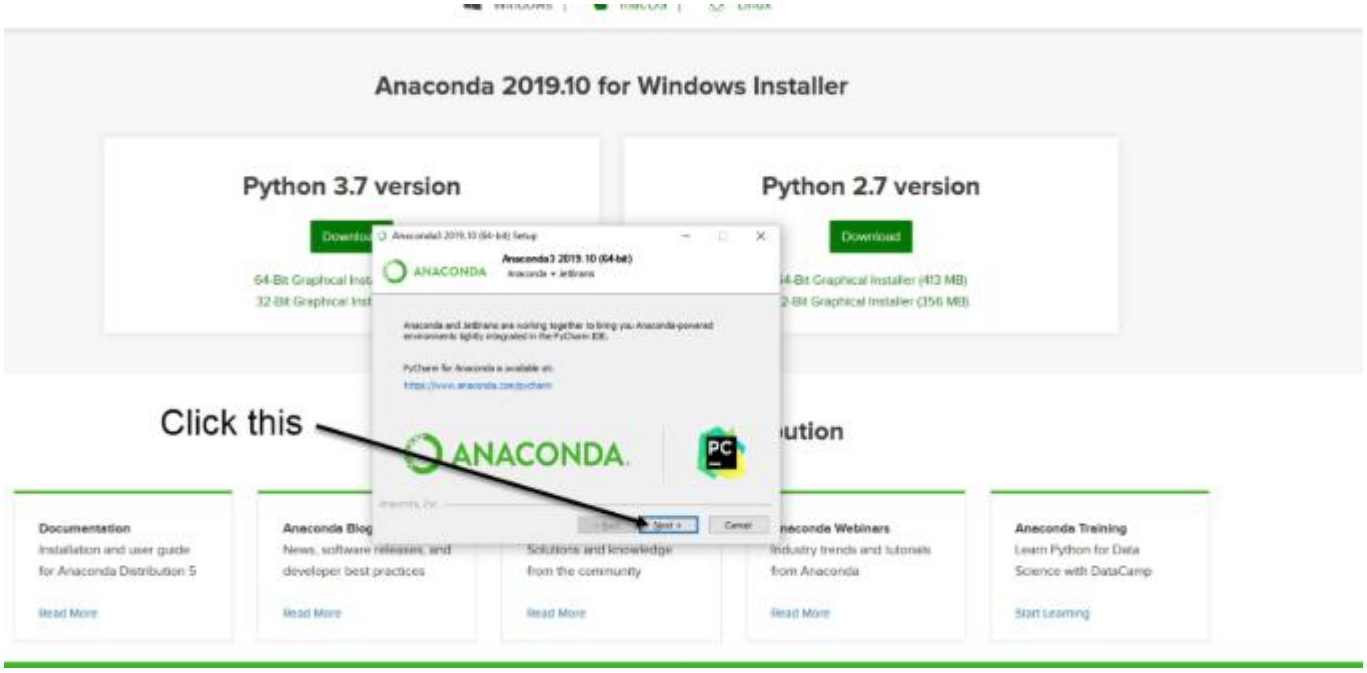
7. It will unpack some packages and extract some files on your machine. This will take a few minutes.



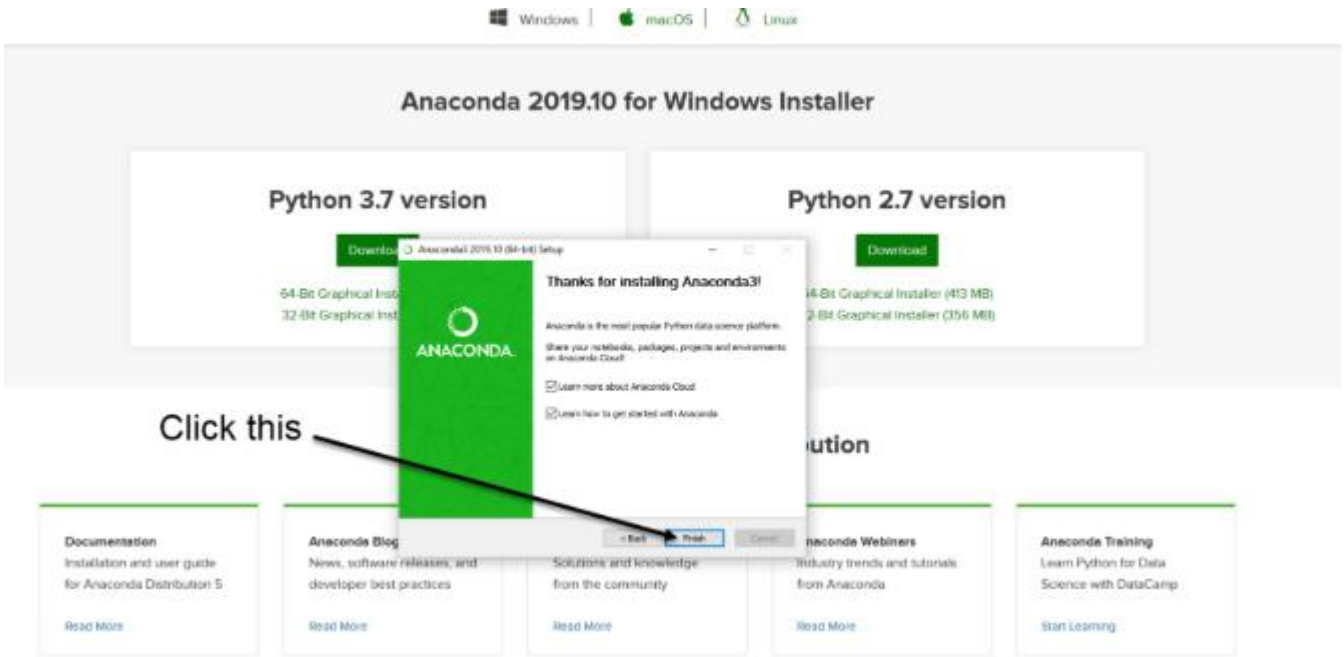
8. The installation is complete. Click Next.



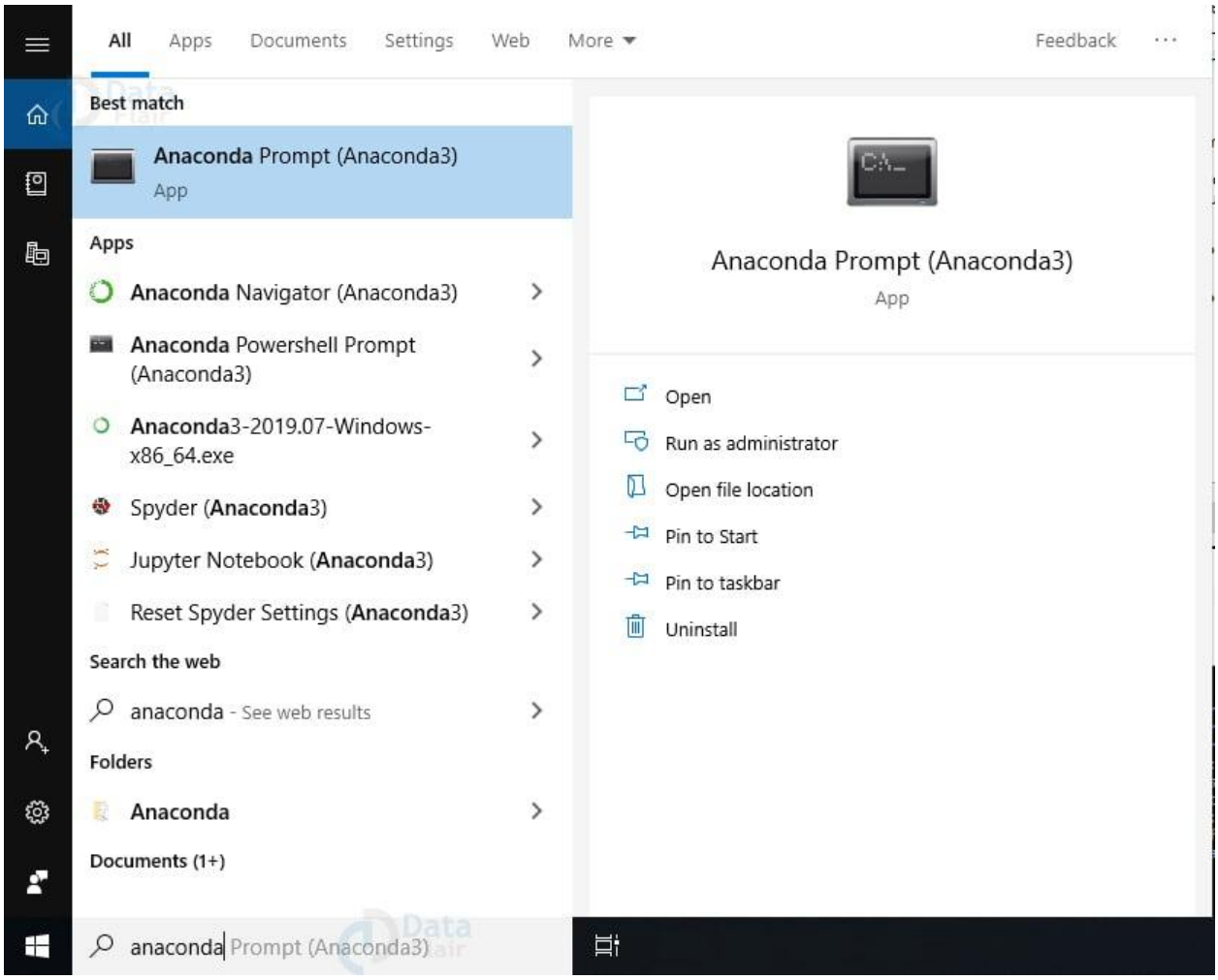
9. This screen will inform you about PyCharm. Click Next.



10. The installation is complete. You can choose to get more information about Anaconda cloud and how to get started with Anaconda. Click Finish.



11. If you search for Anaconda now, you will see the following options:



PYTHON LANGUAGE:

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form

without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python is a dynamic, high-level, free open source, and interpreted programming language. It supports object-oriented programming as well as procedural-oriented programming. In Python, we don’t need to declare the type of variable because it is a dynamically typed language. For example, x = 10 Here, x can be anything such as String, int, etc.

Features in Python:

There are many features in Python, some of which are discussed below as follows:

1. Free and Open Source

Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword. Download Python Since it is open-source, this means that

source code is also available to the public. So you can download it, use it as well as share it.

2. Easy to code

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, JavaScript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language.

3. Easy to Read

As you will see, learning Python is quite simple. As was already established, Python's syntax is really straightforward. The code block is defined by the indentations rather than by semicolons or brackets.

4. Object-Oriented Language

One of the key features of Python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, object encapsulation, etc.

5. GUI Programming Support

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

6. High-Level Language

Python is a high-level language. When we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory.

7. Extensible feature

Python is an Extensible language. We can write some Python code into C or C++ language and also we can compile that code in C/C++ language.

8. Easy to Debug

Excellent information for mistake tracing. You will be able to quickly identify and correct the majority of your program's issues once you understand how to interpret Python's error traces. Simply by glancing at the code, you can determine what it is designed to perform.

9. Python is a Portable language

Python language is also a portable language. For example, if we have Python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

10. Python is an integrated language

Python is also an integrated language because we can easily integrate Python with other languages like C, C++, etc.

11. Interpreted Language:

Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile Python code this makes it easier to debug our code. The source code of Python is converted into an immediate form called byte code.

12. Large Standard Library

Python has a large standard library that provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in Python such as regular expressions, unit-testing, web browsers, etc.

13. Dynamically Typed Language

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

14. Frontend and backend development

With a new project py script, you can run and write Python codes in HTML with the help of some simple tags <py-script>, <py-env>, etc. This will help you do frontend development work in Python like JavaScript. Backend is the strong forte of Python it's extensively used for this work cause of its frameworks like Django and Flask.

15. Allocating Memory Dynamically

In Python, the variable data type does not need to be specified. The memory is automatically allocated to a variable at runtime when it is given a value. Developers do not need to write int y = 18 if the integer value 15 is set to y. You may just type y=18.

LIBRARIES/PACKGES:-

Tensor flow

Tensor Flow 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 provide 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.

SYSTEM TESTING

8. SYSTEM TESTING

System testing, also referred to as system-level tests or system-integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application. System testing verifies that an application performs tasks as designed. This step, a kind of black box testing, focuses on the functionality of an application. System testing, for example, might check that every kind of user input produces the intended output across the application.

Phases of system testing:

A video tutorial about this test level. System testing examines every component of an application to make sure that they work as a complete and unified whole. A QA

team typically conducts system testing after it checks individual modules with functional or user-story testing and then each component through integration testing.

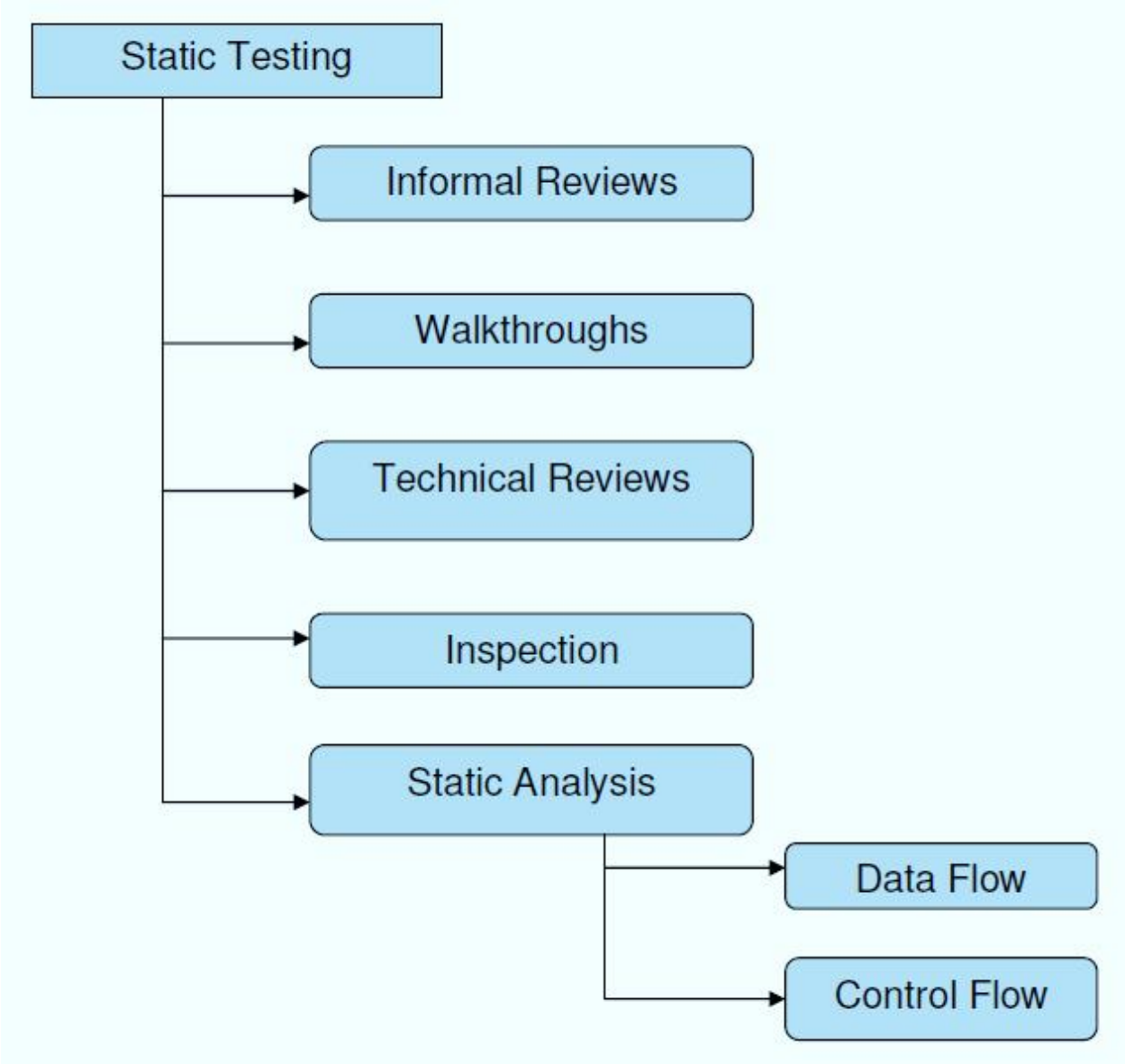
If a software build achieves the desired results in system testing, it gets a final check via acceptance testing before it goes to production, where users consume the software. An app-dev team logs all defects, and establishes what kinds and amount of defects are tolerable.

8.1Software Testing Strategies:

Optimization of the approach to testing in software engineering is the best way to make it effective. A software testing strategy defines what, when, and how to do whatever is necessary to make an end-product of high quality. Usually, the following software testing strategies and their combinations are used to achieve this major objective:

Static Testing:

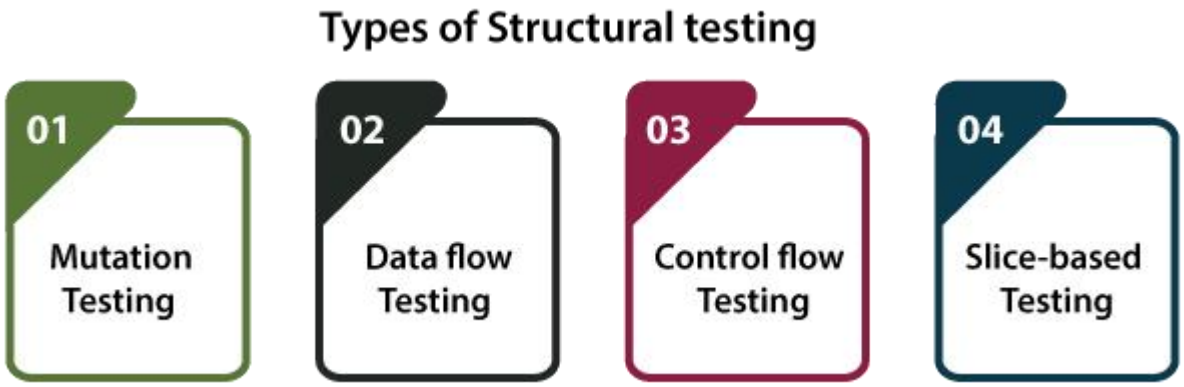
The early-stage testing strategy is static testing: it is performed without actually running the developing product. Basically, such desk-checking is required to detect bugs and issues that are present in the code itself. Such a check-up is important at the pre-deployment stage as it helps avoid problems caused by errors in the code and software structure deficits.



Structural Testing:

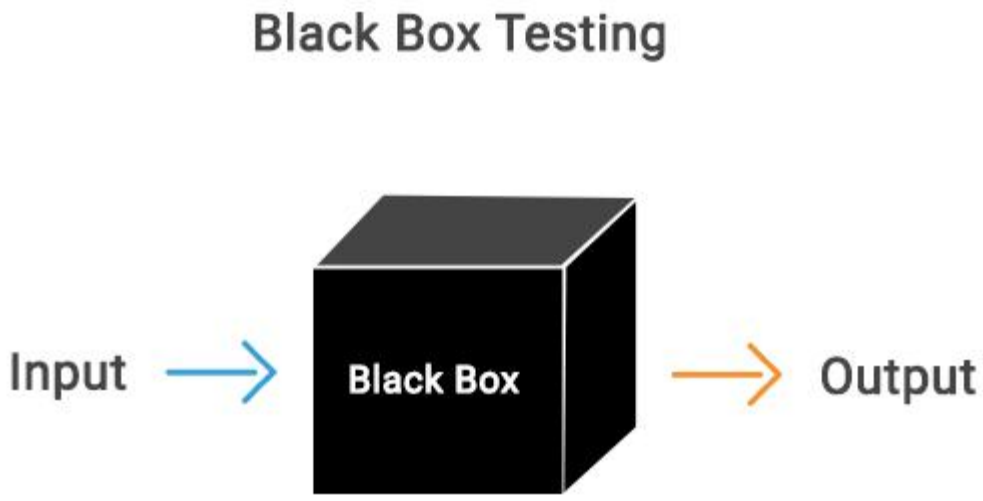
It is not possible to effectively test software without running it. Structural testing, also known as white-box testing, is required to detect and fix bugs and errors emerging during the pre-production stage of the software development process. At this stage, unit testing based on the software structure is performed using regression testing. In most cases, it is an automated process working within the test

automation framework to speed up the development process at this stage. Developers and QA engineers have full access to the software's structure and data flows (data flows testing), so they could track any changes (mutation testing) in the system's behavior by comparing the tests' outcomes with the results of previous iterations (control flow testing).



Behavioral Testing:

The final stage of testing focuses on the software's reactions to various activities rather than on the mechanisms behind these reactions. In other words, behavioral testing, also known as black-box testing, presupposes running numerous tests, mostly manual, to see the product from the user's point of view. QA engineers usually have some specific information about a business or other purposes of the software ('the black box') to run usability tests, for example, and react to bugs as regular users of the product will do. Behavioral testing also may include automation (regression tests) to eliminate human error if repetitive activities are required. For example, you may need to fill 100 registration forms on the website to see how the product copes with such an activity, so the automation of this test is preferable.



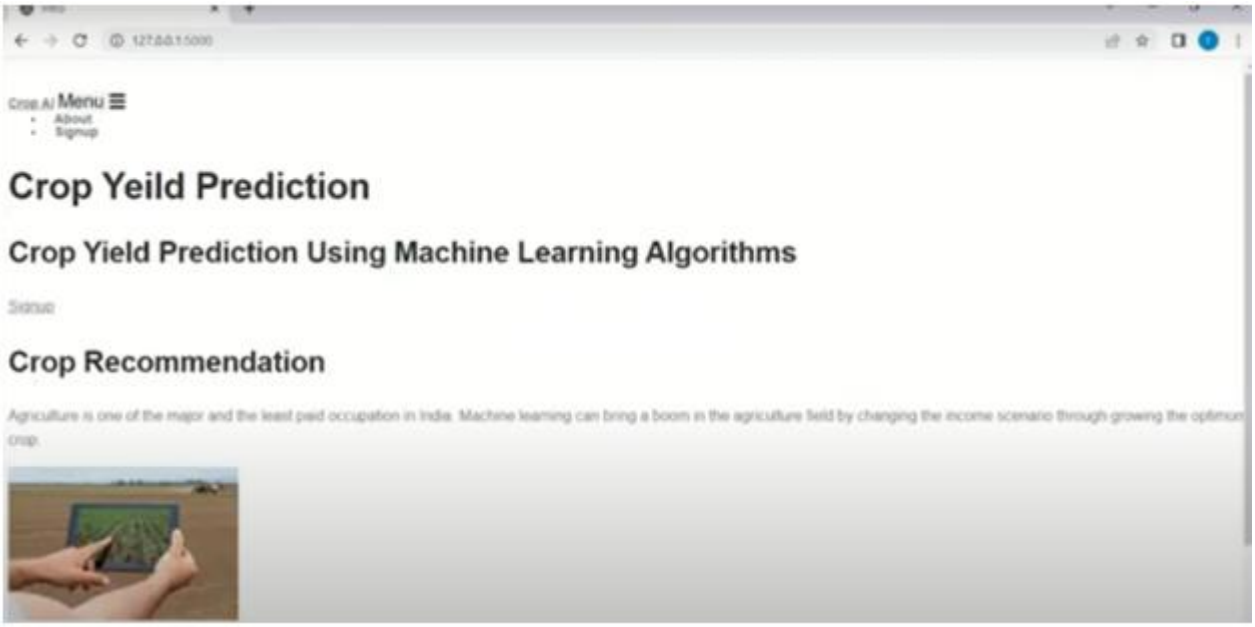
8.2 TEST CASES:

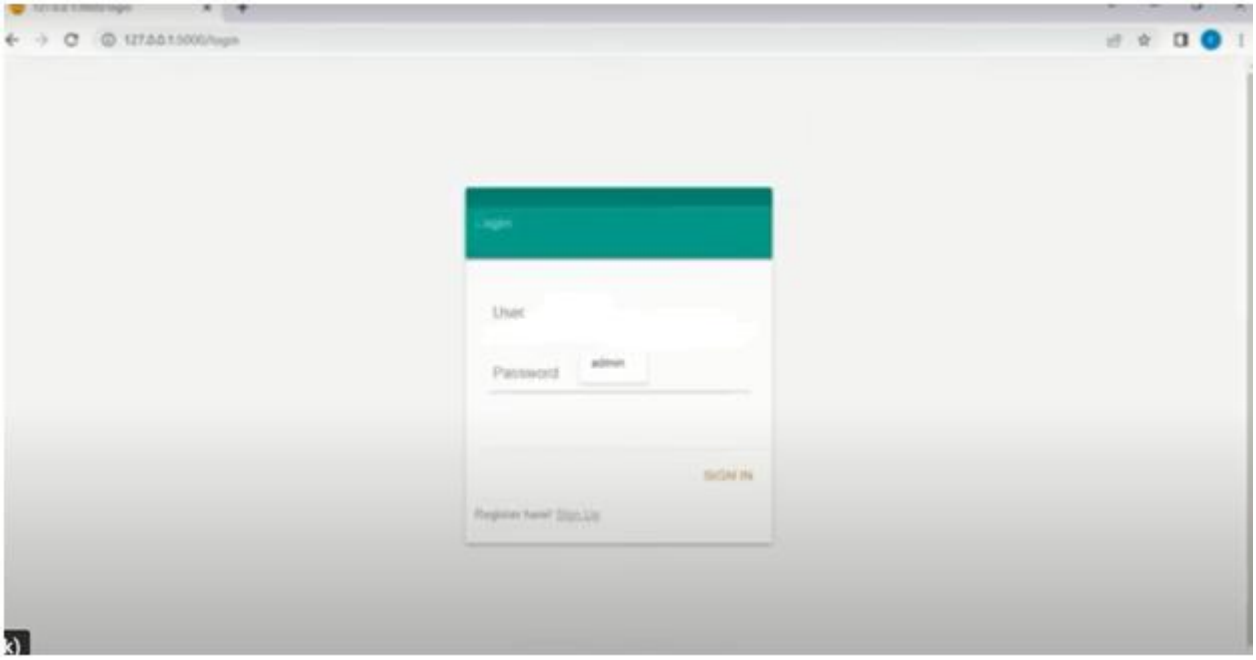
S.NO	INPUT	If available	If not available
1	User signup	User get registered into the application	There is no process
2	User sign in	User get login into the application	There is no process
3	Enter input for prediction	Prediction result displayed	There is no process

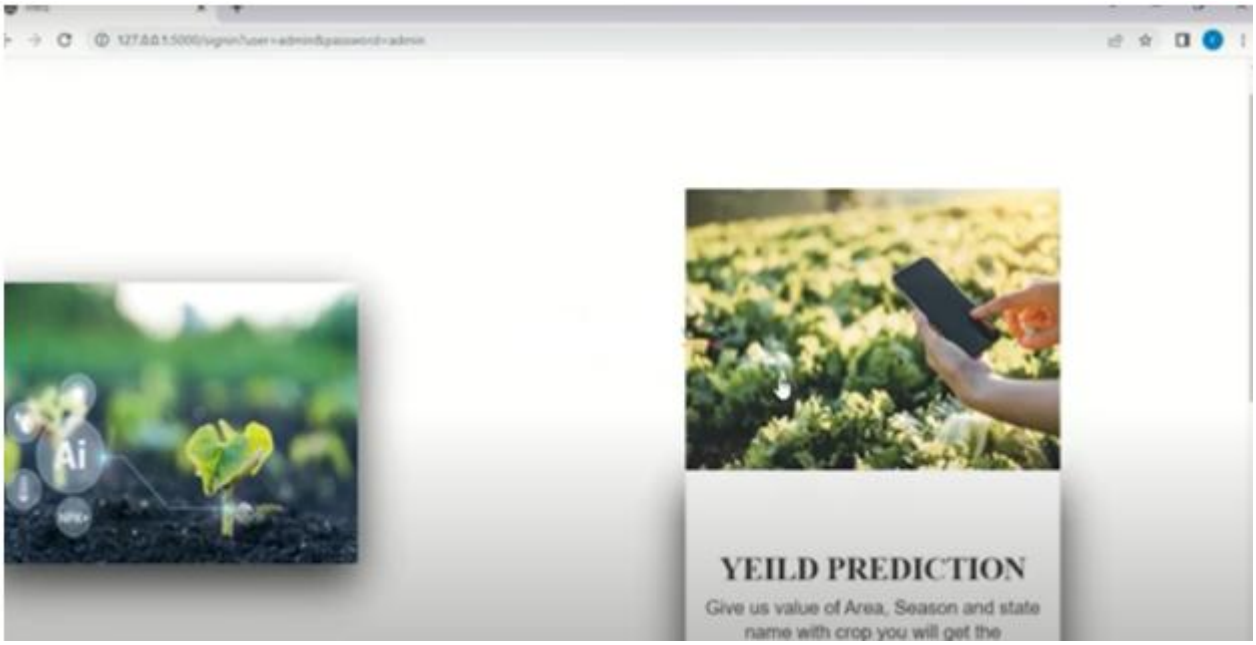
SCREENS

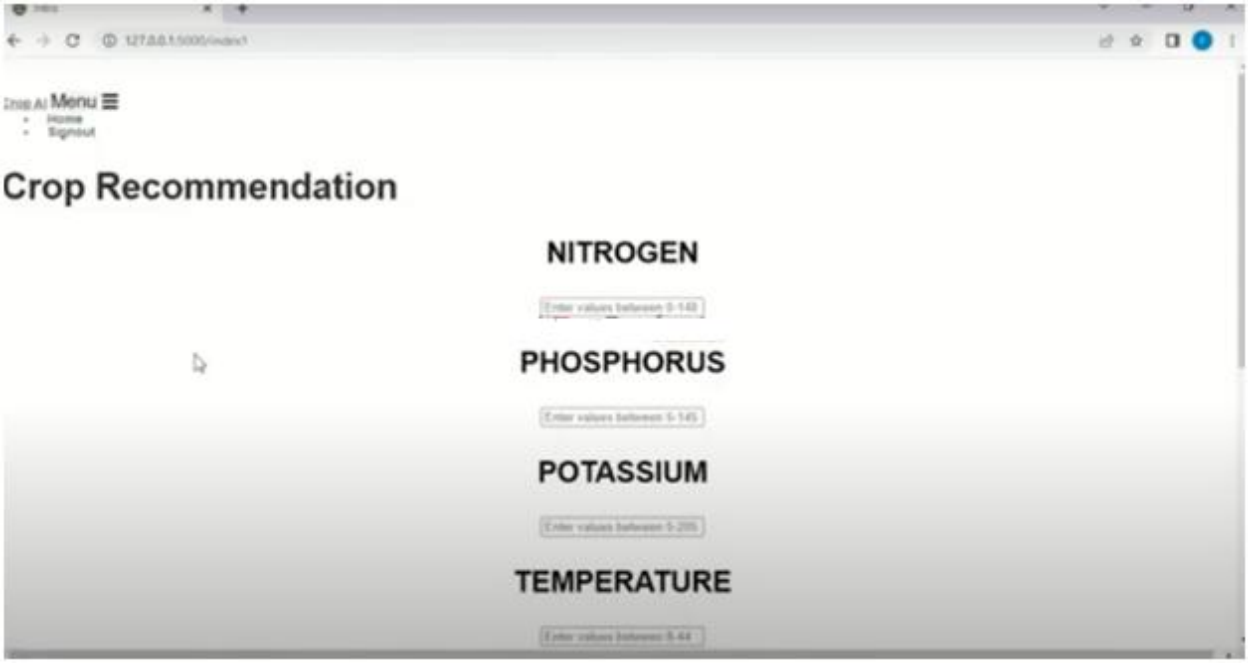
7. SCREENSHOTS

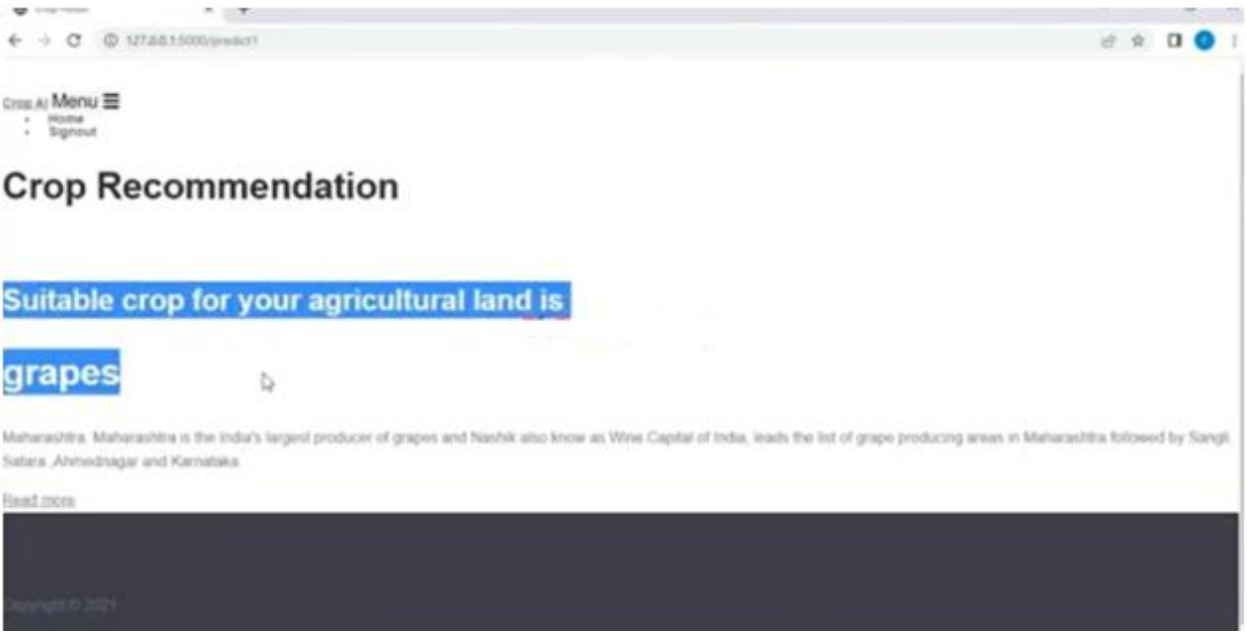
SCREENS:

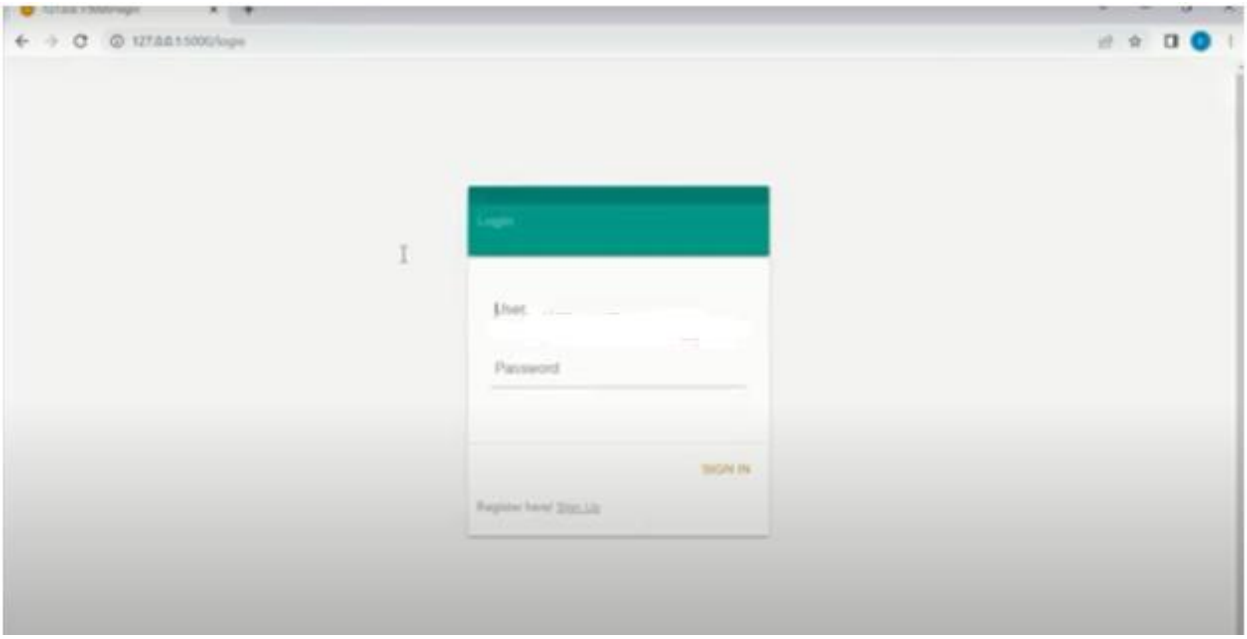

Crop AI Menu
About
Signup
Crop Yeild Prediction
Crop Yield Prediction Using Machine Learning Algorithms
Signup
Crop Recommendation
Agriculture is one of the major and the least paid occupation in India. Machine learning can bring a boom in the agriculture field by changing the income scenario through growing the optimum crop.

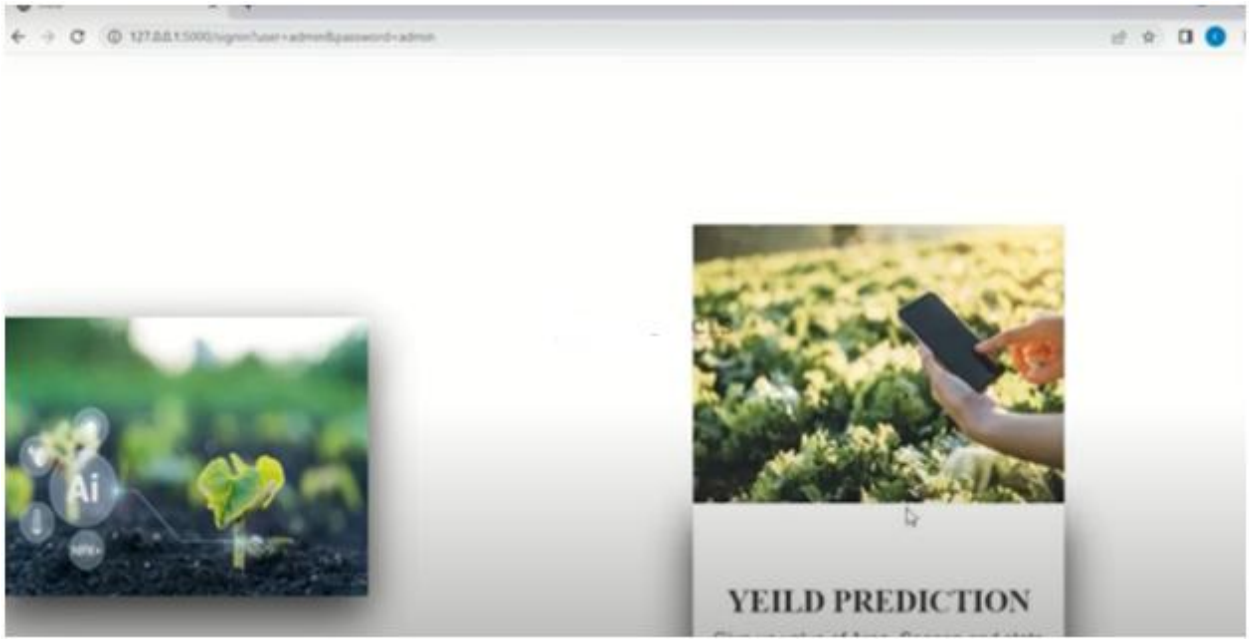

User
Password
admin
SIGN IN

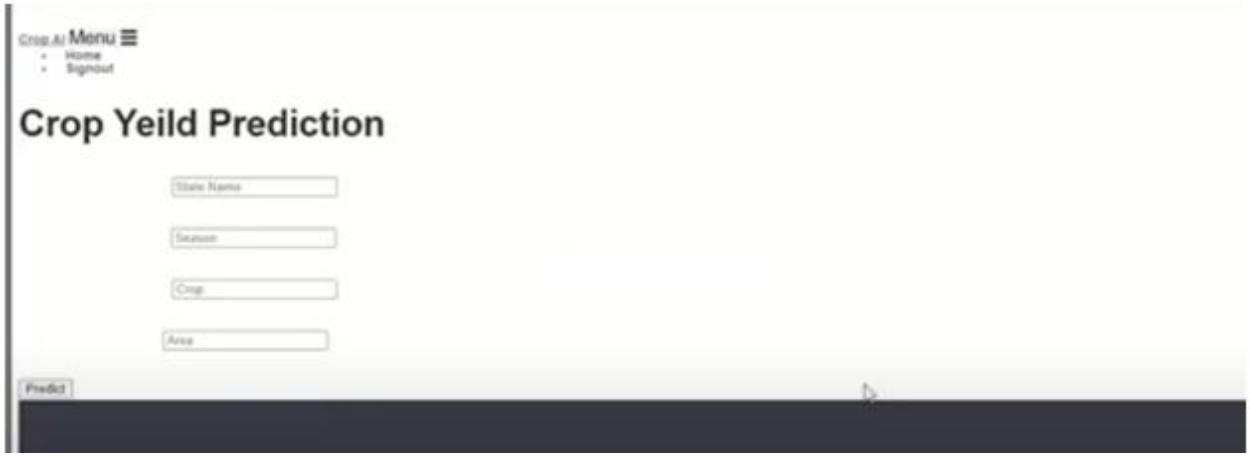

Ai
YEILD PREDICTION
Give us value of Area, Season and state
name with crop you will get the

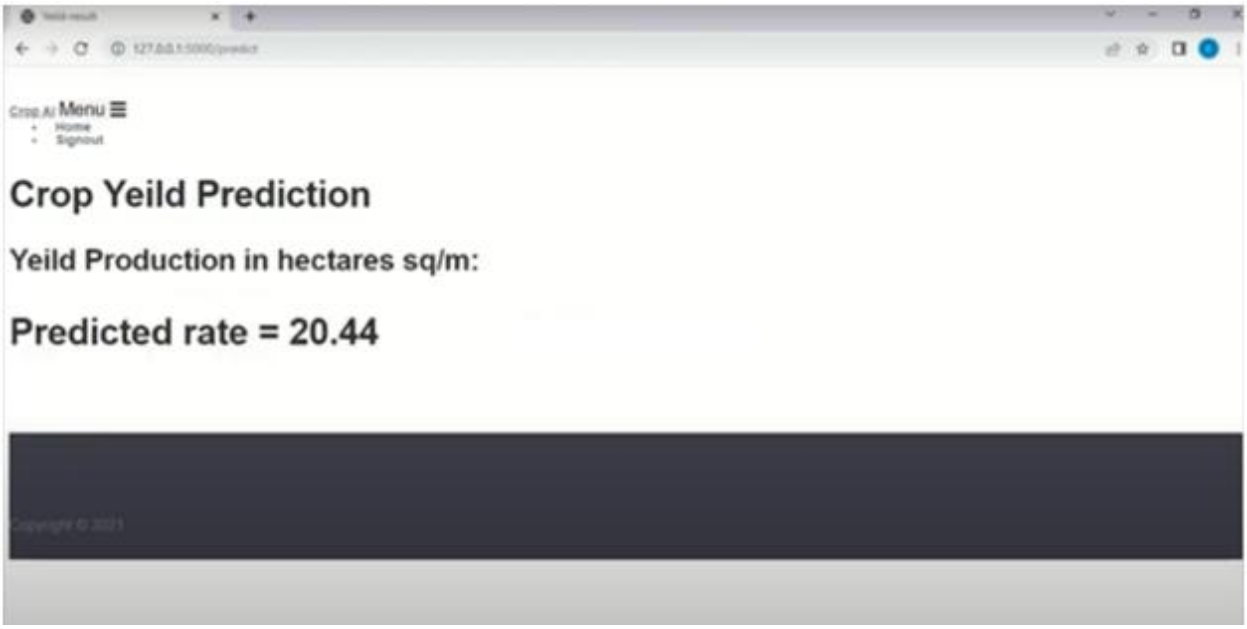

Menu
Home
Signout
Crop Recommendation
NITROGEN
PHOSPHORUS
POTASSIUM
TEMPERATURE


Crop AI Menu
Home
Signout
Crop Recommendation
Suitable crop for your agricultural land is
grapes
Maharashtra. Maharashtra is the India's largest producer of grapes and Nashik also know as Wine Capital of India, leads the list of grape producing areas in Maharashtra followed by Sangli Satara ,Ahmednagar and Karnataka.
Read more


Login
User
Password
SIGN IN


YEILD PREDICTION


Crop.AI Menu
Home
Signout
Crop Yeild Prediction
Predict


Crop AI Menu
Home
Signout
Crop Yeild Prediction
Yeild Production in hectares sq/m:
Predicted rate = 20.44

CONCLUSION

10. CONCLUSION

In conclusion, the deployment of classification and regression algorithms in addressing the challenges faced by the younger generation entering the agriculture sector in Tamil Nadu represents a groundbreaking and impactful solution. The supervised machine learning techniques employed, encompassing comprehensive dataset analyses and algorithmic comparisons, have yielded promising results in recommending crop types and predicting yields.

The superiority of machine learning and deep learning algorithms, including Random Forest, Linear Regression, KNN, XGBoost Classifier, Deep Q Network, and RNN, in accurately projecting optimal harvests is evident from the rigorous evaluation metrics applied. By considering factors such as entropy calculation, precision, recall, F1 score, sensitivity, and specificity, the proposed system ensures a holistic assessment that empowers farmers with reliable insights.

This innovative approach not only contributes to the reduction of losses and effective management of price fluctuations but also plays a pivotal role in building confidence among the agricultural community, particularly the new generation of farmers. Enabling informed decision-making, the system serves as a catalyst for sustainable growth in the agricultural sector, ultimately strengthening the backbone of the economy. The positive impact on livelihoods, GDP contribution, and the overall resilience of the agricultural ecosystem in Tamil Nadu highlights the transformative potential of integrating machine learning into traditional farming practices.

BIBILOGRAPHY

11. REFERENCES

[1] Jeevan Nagendra Kumar Y, V Spandana, VS Vaishnavi, K Neha, VGRR Devi. Supervised Machine learning Approach for Crop Yield Prediction in Agriculture Sector. 5th International Conference on Communication and Electronics Systems. 2020;736-741

[2] Mariammal G, A Suruliandi, SP Raja, E Poongothai. Prediction of Land Suitability for Crop Cultivation Based on Soil and Environmental Characteristics Using Modified Recursive Feature Elimination Technique With Various Classifiers. IEEE Transactions on Computational Social Systems. 2021;8(5):1132-1142

[3] Mehedi Hasan Md, Muslima Tuz Zahara, Mahamudunnobi Sykot, Arafat Ullah Nur, Mohd Saifuzzaman, Rubaiya Hafiz. Ascertaining the Fluctuation of Rice Price in Bangladesh Using Machine Learning Approach. 11th International Conference on Computing, Communication and Networking Technologies. 2020;1-5

[4] Rakesh Kumar, MP Singh, Prabhat Kumar, JP Singh. Crop Selection Method to maximize crop yield rate using machine learning technique. International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials. 2015;138-145.

[5] Shilpa Mangesh Pande, Prem Kumar Ramesh, Anmol Anmol, BR Aishwarya, Karuna Rohilla, Kumar Shaurya. Crop Recommender System Using Machine Learning Approach. 5th International Conference on Computing Methodologies and Communication. 2021;1066-1071.