**A Project-I Report**

**On**

**CROP PREDICTION SYSTEM**

*Submitted to*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU**

*In Partial Fulfillment of the Requirements for the Award of the Degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted By**

**M.BHAVYA SREE - (19699A0507)**

**P.JAHNAVI - (19699A0518)**

**K.KEERTHI - (19699A0521)**

**G.SAI VENKATA - (19699A0546)**

**NITHYANANDA REDDY**

**C.USHA - (19699A0557)**

**Under the Guidance of**

**Mrs. G. Vasundara Devi**

**Assistant Professor**

**Department of Computer Science & Engineering**



**MADANAPALLE INSTITUTE OF TECHNOLGY & SCIENCE**

**(UGC – AUTONOMOUS)**

**(Affiliated to JNTUA, Ananthapuramu)**

**Accredited by NBA, Approved by AICTE, New Delhi)**

**AN ISO 9001:2008 Certified Institution**

**P. B. No: 14, Angallu, Madanapalle – 517325 ,2019-2023**

**2009-2013**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that the project work entitled **“CROP PREDICTION SYSTEM”** is a bonafide work carried out by

**M.BHAVYA SREE - (19699A0507)**

**P.JAHNAVI - (19699A0518)**

**K.KEERTHI - (19699A0521)**

**G.SAI VENKATA - (19699A0546)**

**NITHYANANDA REDDY**

**C.USHA - (19699A0557)**

Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the stream of **Computer Science & Engineering** in **Madanapalle Institute of Technology and Science,** **Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2022-2023

Guide Head of the Department

**Mrs. G. Vasundara Devi  Dr.** **R. Kalpana**

**Assistant Professor, Professor and Head,**

**Department of CSE Department of CSE**

Submitted for the University examination held on:

**Internal Examiner External Examiner**

**Date:**  **Date:**

**ACKNOWLEDGEMENT**

We sincerely thank the **MANAGEMENT** of **Madanapalle Institute of Technology and Science** for providing excellent infrastructure and lab facilities that helped me to complete this project.

We sincerely thank **Dr. C. Yuvaraj, M.E., Ph.D., Principal** for guiding and providing facilities for the successful completion of our project at **Madanapalle Institute of Technology and Science,** Madanapalle.

We express our deep sense of gratitude to **Dr. R. Kalpana, M. Tech., Ph.D., Professor and Head of the Department of CSE** for her continuous support in making necessary arrangements for the successful completion of the Project.

We express my deep sense gratitude to **Dr. K P Manikandan, Ph.D , Project Coordinator and Mrs. M Bommy ,M.E., Project Co Coordinator** for their valuable guidance and encouragement that helped us to complete this project.

We express our deep gratitude to my guide **Mrs. G. Vasundara Devi, Assistant Professor, Department of CSE** for her guidance and encouragement that helped us to complete this project.

We also wish to place on record my gratefulness to other **Faculty of CSE Department** and also to our friends and our parents for their help and cooperation during our project work.

**DECLARATION**

We hereby declare that the results embodied in this project **“CROP PREDICTION SYSTEM”** by us under the guidance of **Mrs. G. Vasundara Devi,Assistant Professor, Dept. of CSE** in partial fulfillment of the award of **Bachelor of Technology** in **Computer Science & Engineering** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** and we have not submitted the same to any other University/institute for award of any other degree.

**Date :**

**Place :**

**PROJECTASSOCIATES**

**M.BHAVYA SREE**

**P.JAHNAVI**

**K.KEERTHI**

**G.SAI VENKATA NITHYANANDA REDDY**

**C.USHA**

I certify that above statement made by the students is correct to the best of my knowledge.

**Date : Guide**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPIC** | **PAGE NO.** |
| **1.** | **INTRODUCTION** |  |
|  | 1.1 Motivation | 7 |
|  | 1.2 Problem Definition | 7 |
|  | 1.3 Objective of the Project | 7 |
|  | 1.4 Limitations of Project | 7 |
|  | 1.5 Organization of Documentation | 8 |
| **2.** | **LITERATURE SURVEY** | **9** |
|  | 2.1 Introduction | 9 |
|  | 2.2 Existing System | 9 |
|  | 2.3 Disadvantages of Existing System | 9 |
|  | 2.4 Proposed System | 9 |
|  | 2.5 Advantages over Existing System | 9 |
| **3.** | **ANALYSIS** | **10** |
|  | 3.1Introduction | 10 |
|  | 3.2 Software Requirement Specification | 11 |
|  | 3.3 Content diagram of Project | 12 |
| **4.** | **DESIGN** | **13** |
|  | 4.1 Introduction | 13 |
|  | 4.2 ER/UML Diagrams | 13 |
|  | 4.3 Module Design and Organization | 16 |
|  | 4.4 Conclusion | 18 |
| **5.** | **IMPLEMENTATION AND RESULTS** | **19** |
|  | 5.1 Introduction | 19 |
|  | 5.2 Implementation of key functions | 19 |
|  | 5.3 Method of Implementation | 19 |
|  | 5.4 Output Screens and Result Analysis | 23 |
|  | 5.5 Conclusion | 23 |
|  |  |  |
| **6.** | **TESTING AND VALIDATION** | **37** |
|  | 6.1 Introduction | 37 |
|  | 6.2 Design of Test cases and Scenarios | 37 |
|  | 6.3 Validation | 38 |
|  | 6.4 Conclusion | 38 |
| **7.** | **CONCLUSION** | **40** |
|  | 7.1 Conclusion | 40 |
| **8.** | **REFERENCES** | **41** |

**LIST OF FIGURES**

**Figure Title Page**

**3.3 Content Diagram**

**ABSTRACT**

The project aims to discover the best model for crop prediction, which can help farmers to decide the suitable crop to grow based on the nutrients present in the soil. As India is one of the top Agricultural producing Countries in the World and we are mostly depend on a cultivation which is depending on the soil. There are 3 most important nutrients in any soil, it’s known as the primary macronutrients: Nitrogen (N), Phosphorus (P), and Potassium(K). Each of the primary nutrients is very essential in plant nutrition, serving a critical role in the growth and reproduction of the plant. In this project, we will use K-Nearest Neighbour algorithm. Here the N:P:K values plays a vital role to give the prediction of suitable crops for that particular soil.Machine learning is an important decision support tool for crop prediction, including supporting decisions on what crop to grow in the soil.

**CHAPTER 1**

**INTRODUCTION**

For most developing countries, agriculture is their primary source of revenue. Modern agriculture is a constantly growing approach for agricultural advances and farming techniques. It becomes challenging for the farmers to satisfy our planet's evolving requirements and the expectations of merchants, customers, etc. Some of the challenges the farmers face are-(i) Nutrient deficiency in the soil, caused by a shortage of crucial minerals such as potassium, nitrogen, and phosphorus can result in reduced crop growth. (ii) Farmers make a mistake by cultivating the same crops year after year without experimenting with different varieties. Agriculture is the most important supply of Indian Economy. For the better crop yield, the farmers always require a correct crop that can give good yield in that particular soil. It is predicting the future possible crops that can be sowed in soil with its respective NPK percentages

* 1. **MOTIVATION**

As we are seeing many farmers committing Suicides these days by cultivating crops without knowing the suitability of the crop. By considering this scenario, We worked on project which suggest the suitable crop for the soil. Farmers are migrating to cities after facing many problems. Proper Utilization of the available resources are getting reduced and Biggest problems faced by farmers is due to lack of availability of Manures ,fertilizers , biocides ,seeds , machineries not in terms of raising crops.

* 1. **PROBLEM DEFINITION**

Without knowing the N,P,K values of particular crop that grows in the soil, Farmer grow the crop which results in decrease of crop production and crop yield. So in this project, We will find the proper N,P,K values of the soil .Based on that values we suggest the crop for that particular soil ,So that the crop yield and crop production increases

* 1. **OBJECTIVE OF THE PROJECT**

➢ The primary goal of this study is to examine, evaluate and identify the suitable crop using trained modelspresent in the dataset.

➢ And also increase the accuracy in detecting the prediction of crop by using KNN Algorithm.

* 1. **LIMITATIONS OF PROJECT**

➢ lack of practical knowledge of farmers

➢ We will detect only the nitrogen, potassium and phosphate present in the soil. Parameters like humidity, pressure, pH are not identified.

**1.5** **ORGANISATION OF THE DOCUMENT**

* Chapter 2 is Literature Survey which emphasizes the related works and their disadvantages.
* Chapter 3 deals with Requirements and Analysis.
* Chapter 4 deals with the design
* Chapter 5 deals with the Implementation and result of the Project
* Chapter 6 deals with Testing and Validation
* Chapter 7 deals with the Conclusion and Future work

**CHAPTER 2**

**LITERATURE REVIEW**

* 1. **EXISTING SYSTEM**

In Existing system, the crop prediction is done based on PH values of soil and uses CNN algorithm approaches to build the model. There will be change in the accurate prediction if we use PH values

* 1. **DRAWBACKS OF EXISTING SYSTEM**
* There will be change in PH values due to Rainfall. So, The prediction may get wrong.
* It analyse and detect the soil PH through soil image if the image quality is poor then there will be problem in recognizing of soil PH value.
  1. **PROPOSED SYSTEM**

The proposed system uses N,P,K values to predict the suitability of the crop for the soil. Machine learning is the latest technology which python programming language gives advantage in using various algorithms for crop yield prediction based on the input data set. In this process KNN classification algorithm is used for prediction. In this project testing,training is performed on given text dataset which includes N,P,K values as features and type of crop as labels.

* 1. **ADVANTAGES OF PROPOSED SYSTEM**
* Crop yield prediction is performed based on textual dataset and any user can check type of crop best suits for conditions and get crop suggestions.
* Quick Calculation time
* Versatile-useful for classification and Regression
* Low chance for getting fake

**CHAPTER 3**

**REQUIREMENTS AND ANALYSIS**

**3.1 INTRODUCTION**

Agriculture is considered as import field all over the world where there are many challenges in solving problems in the process of estimating crops based on the conditions. This has become a challenge for developing countries. Using latest technologies many companies are using IOT based services and Mechanical technology to reduce manual work. These methods are mostly useful in the case on reducing manual work but not in prediction process. In this project crop yield prediction using Machine learning latest ML technology and KNN classification algorithm is used for prediction crop yield based on soil and temperature factors. Dataset is prepared with various soil conditions as features and labels for predicting type of each label is related to certain crop. In prediction process user can give input as soil features and result will be type of crop suitable for specific conditions and application also helps in suggesting best crops.

**3.2 Requirement Specification**

**3.2.1 HARDWARE DESCRIPTION**

* Processor - Intel 486/Pentium processor or better
* Processor Speed - 500 MHz or above
* Hard Disk - 20GB(approx)
* RAM - 64MB or above

**3.2.2 SOFTWARE DESCRIPTION**

* + Operating System : Windows 8 and above
  + Language : Python
  + Libraries : NumPy
  + Platform : Google COLAB

**3.2.3 LANGUAGE SPECIFICATION**

Python is a strong and easy-to-learn programming language. It has efficient high-level data structures and an object-oriented programming technique that is simple but effective. Python's beautiful syntax and dynamic typing, as well as its interpreted nature, make it an excellent language for scripting and quick application development across a wide range of platforms. The Python interpreter and comprehensive standard library are freely accessible for all major platforms in source or binary form from the Py Site, https://www.python.org/, and may be made available. Its same site also has links to and releases of a variety of free Python modules, scripts, and tools, as well as extra documentation. New functions and data types written in C or C++ can simply be added to the Python interpreter (or other languages callable from C). Python can also be used as an emphasize for programmes that can be customized. Python is a scripting language that is high-level, interpreted, interactive, and object-oriented. Python is intended to be a very understandable language. It typically uses English terms instead of punctuation, and has fewer syntactical structures than other languages.

**3.2.4 Google Colab**

Google is quite aggressive in AI research. Over many years, Google developed AI framework called TensorFlow and a development tool called Colaboratory. Today TensorFlow is open-sourced and since 2017, Google made Colaboratory free for public use. Colaboratory is now known as Google Colab or simply Colab.

Another attractive feature that Google offers to the developers is the use of GPU. Colab supports GPU and it is totally free. The reasons for making it free for public could be to make its software a standard in the academics for teaching machine learning and data science. It may also have a long term perspective of building a customer base for Google Cloud APIs which are sold per-use basis.

Irrespective of the reasons, the introduction of Colab has eased the learning and development of machine learning applications.

**What Colab Offers You?**

As a programmer, you can perform the following using Google Colab.

Write and execute code in Python

Document your code that supports mathematical equations

Create/Upload/Share notebooks

Import/Save notebooks from/to Google Drive

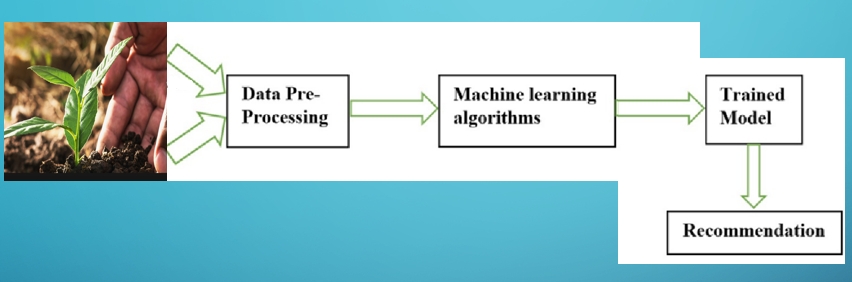
Import/Publish notebooks from GitHub

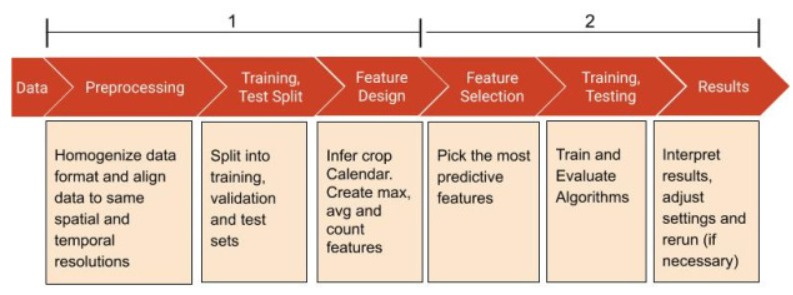
Import external datasets e.g. from Kaggle

Integrate PyTorch, TensorFlow, Keras, OpenCV

Free Cloud service with free GPU

**3.3 Content diagram of Project**





**CHAPTER 4**

**DESIGN**

**4.1 INTRODUCTION**

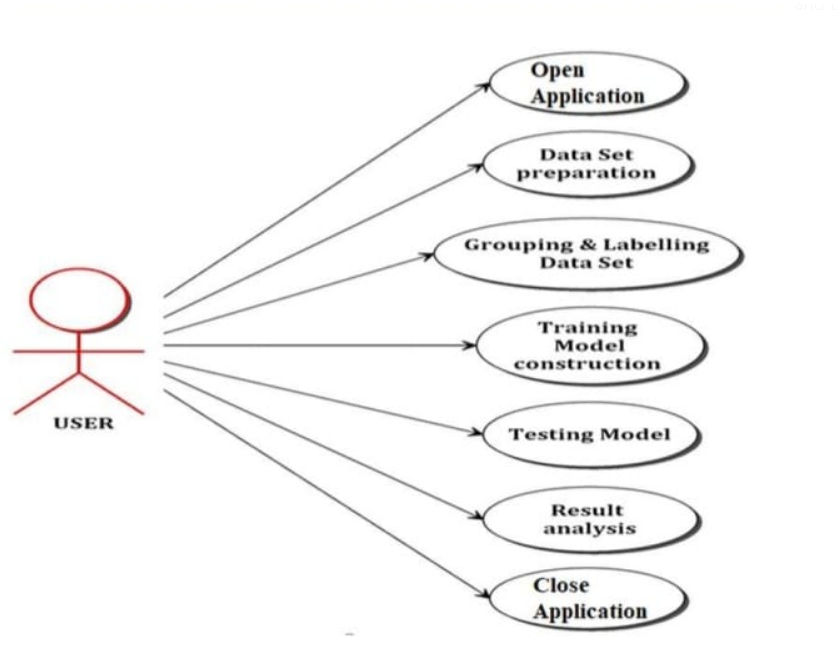
Designing any project requires the implementation of various architectural diagrams. Specifically, UML diagrams are mainly used to design and analyze projects before prototyping and production. Design is an early phase of a project where the project's key features, structure, criteria for success, and major deliverables are planned out.

**4.2 UML DIAGRAMS**

The UML diagrams help us to understand and design the application easily. UML is a model-based language for visualising, describing, building, and documenting softwareintensive systems. UML provides a standard method for writing a system model that includes conceptual notions.

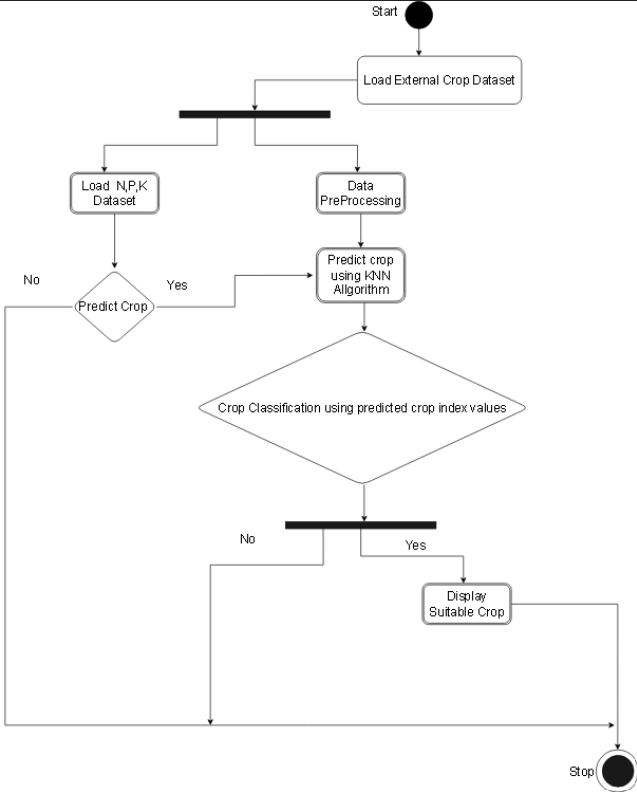
**4.2.1 USE CASE DIAGRAM**

At its most basic level, a use-case diagram depicts the parameters of a use-case and represents a user's interactions. A use case diagram can depict the various types of system users and how they communicate with it. This style of diagram is frequently used in connection with a textual use-case, and it is frequently accompanied by other diagrams.



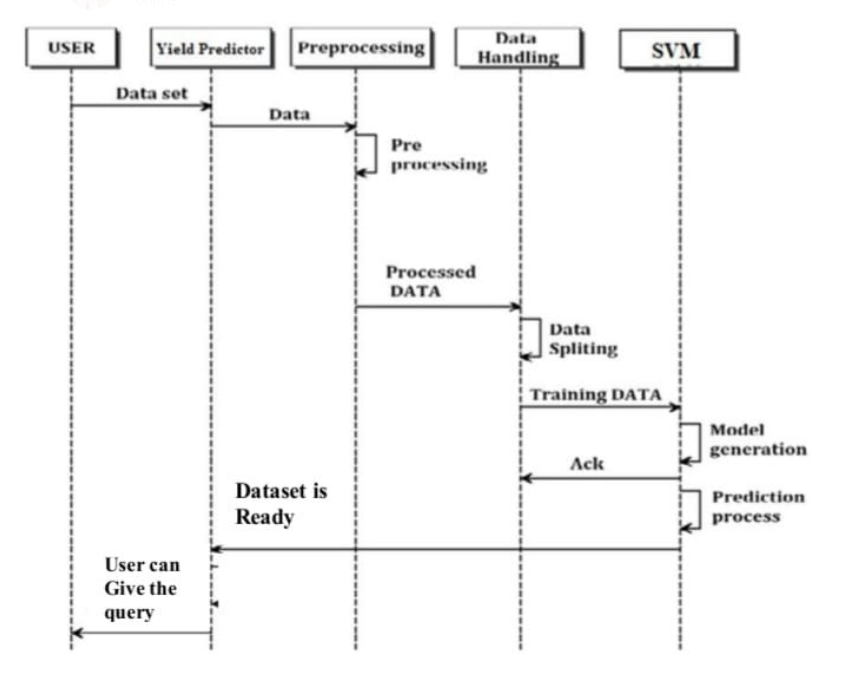
**4.2.2 ACTIVITY DIAGRAM**

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinate to represent business workflows

****

**4.2.2 SEQUENCE DIAGRAM**

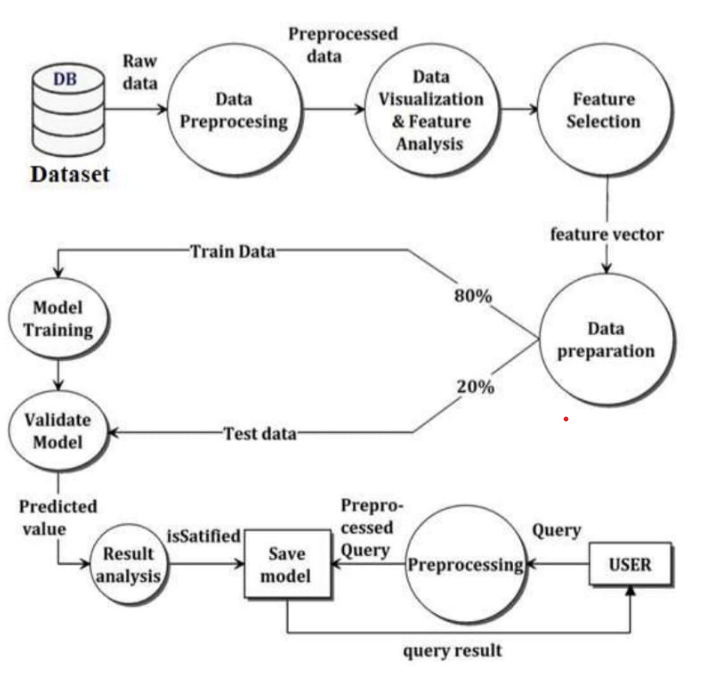
A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects. For example, lifelines in a sequence diagram for a banking scenario can represent a customer, bank teller, or bank manager. The communication between the customer, teller, and manager are represented by messages passed between them. The sequence diagram shows the objects and the messages between the objects

****

**4.3 Module Design and Organization**

**4.3.1:Model design:**

Modular design, or modularity in design, is a design principle that subdivides a system into smaller parts called modules (such as modular process skids), which can be independently created, modified, replaced, or exchanged with other modules or between different systems.



**4.3.2:Module output:**

**Module A**

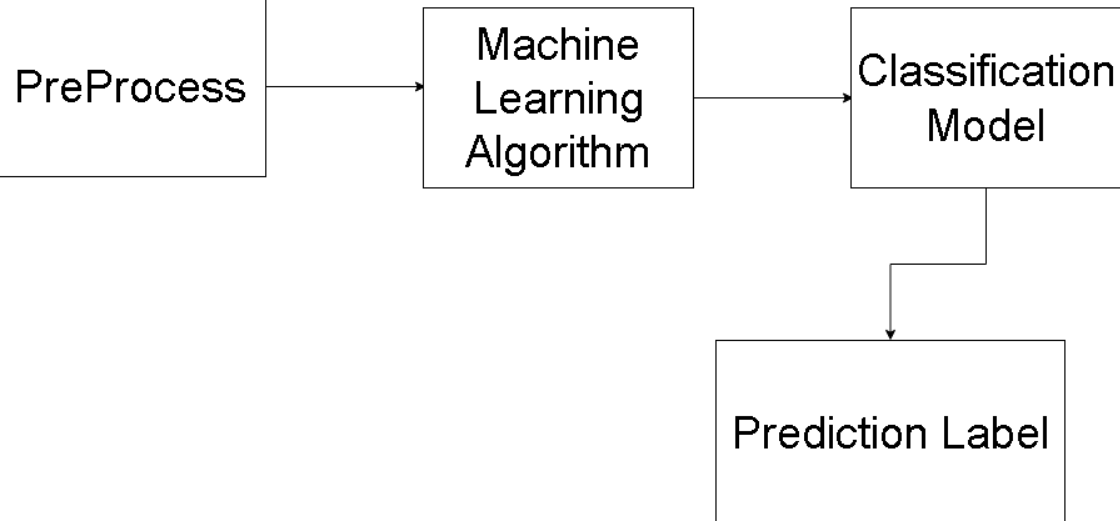
**Data Collection and PreProcessing**

* Data collection is the most efficient method for collecting and measure the data from different resources.
* To get an approximate dataset for the system. This dataset must contain the following attribute NPK values, those parameters will consider for crop prediction.
* After collecting datasets from various resources. Dataset must be preprocessing before training to the model. The data preprocessing can be done by various stages, begins with reading the collected dataset.
* The process continues to data cleaning. In data cleaning the datasets contain some redundant attributes, those attributes are not considering for crop prediction.
* So, we have to drop unwanted attributes and datasets containing some missing values we need to drop these missing values or fill with unwanted nan values in order to get better accuracy.
* Then define the target for a model

**Module B**

**Training Model**

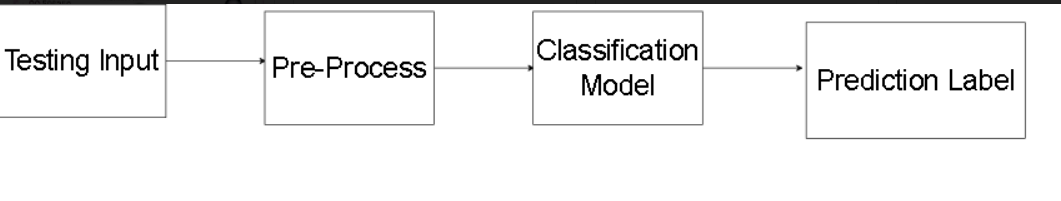
* Fetching the data from the Data Sets
* Provide Suitable N:P:K Values for various crops to the machine
* Training dataset to show which crop will suit the soil best using KNN Algorithm

****

**Module C**

**Testing Model**

* Here user can test and analyze the respective model by performing preprocessing over the input data.
* Mapping to user input using saved featured (based on training dataset). Then feed to saved model for prediction.
* The testing dataset is used to predict the crop to be raised, using the trained classifier
* Finally, a suitable crop is obtained by using KNN algorithm



**4.4 Conclusion**

The current system is capable of predicting the best suitable crop and sharing agricultural resources among the users.

The system can be improved by adding many datasets and as the user increases the efficiency of the system increases accordingly. Optimized hardware can provide strength in the realtime scenarios.

**CHAPTER-5**

|  |  |
| --- | --- |
|  | **IMPLEMENTATION AND RESULTS** |

**5.1 INTRODUCTION**

The work done to fulfill the standards of the scope of work is referred to as the implementation phase. The term "machine learning" refers to a method for recognizing patterns in data. During the implementation phase, the project team accomplished the tasks outlined in the plan and made any necessary revisions.

**5.2 IMPLEMENTATION OF KEY FUNCTIONS**

● read csv () from the pandas package is used to read the data set.

● seaborn library is used for visualizing the data

● Data cleaning functions for removing the anomalies

● The model was built using the KNN Algorithm.

**5.3 Method of Implementation**

**5.3.1 DATA PREPROCESSING**

Data preprocessing is the process of modifying or removing data before to its use in order to ensure or improve performance. The process of preparing raw data for use in a machine learning model is known as data preprocessing. It's the most crucial and first step in creating a machine learning model. When working on a machine learning project, we don't always have access to clean, prepared data. And, before doing any data-related activity, it is necessary to clean the data and format it. As a result, we apply the data preprocessing task for this.

**5.3.2 DATA SPLITTING**

The dataset is split into two parts: a Training data set and a Testing data set. In most circumstances, the data is divided into two sets, the training set and the testing set, using an 8:2 ratio. The training data set is used to build a model, whereas the test dataset is used to determine whether or not the model is right. Because the data contains 17880 job advertisements, 14304 are utilized to develop a model and 3576 are used to validate the model

**5.3.3 CLASSIFICATION MODEL**

KNN Classification Algorithm is applied on the Training set and based on the test result accuracy,it suggests whether crop suits or not.

* + 1. **IMPLEMENTED MACHINE LEARNING MODELS**

1. **K-Nearest Neighbour(KNN) Algorithm for Machine Learning**

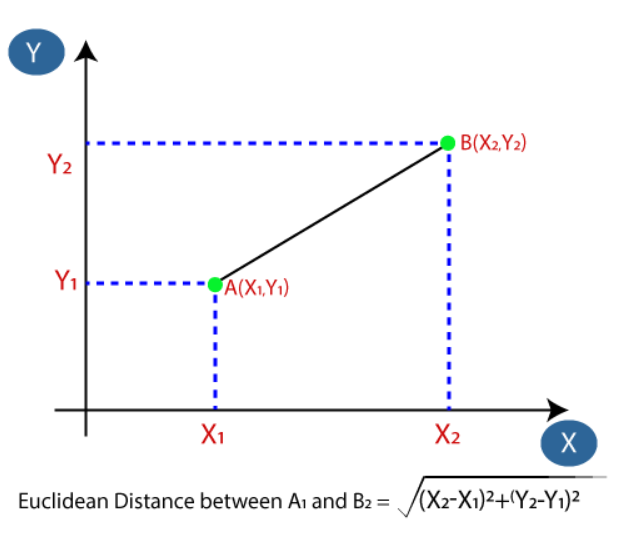
* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

**Why do we need a K-NN Algorithm?**

* Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

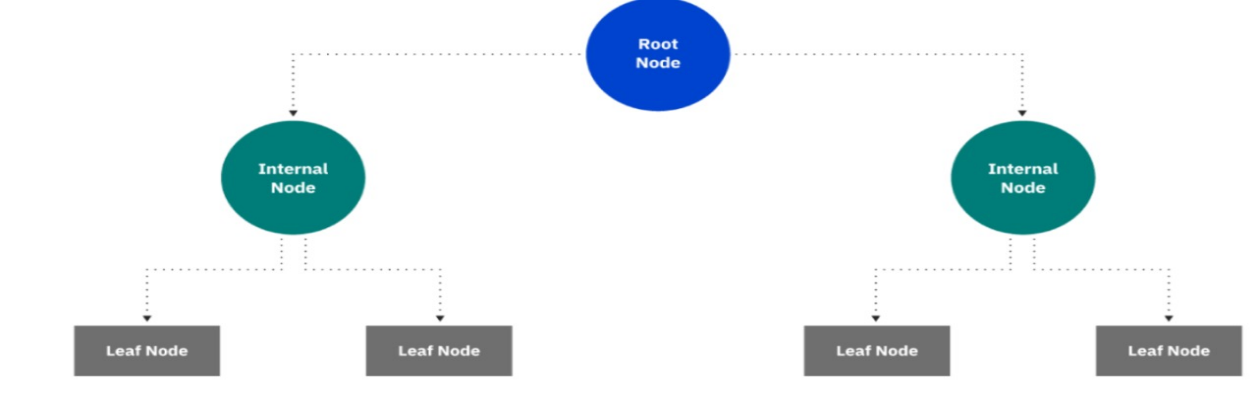


* We use Euclidean distance formula for finding k number of neighbours
* Firstly, we will choose the number of neighbors
* Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:

****

1. **Decision Tree (DT):**

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.



**C)Gradient Boosting Classifier:**

Gradient boosting is a method standing out for its prediction speed and accuracy, particularly with large and complex datasets. From Kaggle competitions to machine learning solutions for business, this algorithm has produced the best results. We already know that errors play a major role in any machine learning algorithm. There are mainly two types of error, bias error and variance error. Gradient boost algorithm helps us minimize bias error of the model

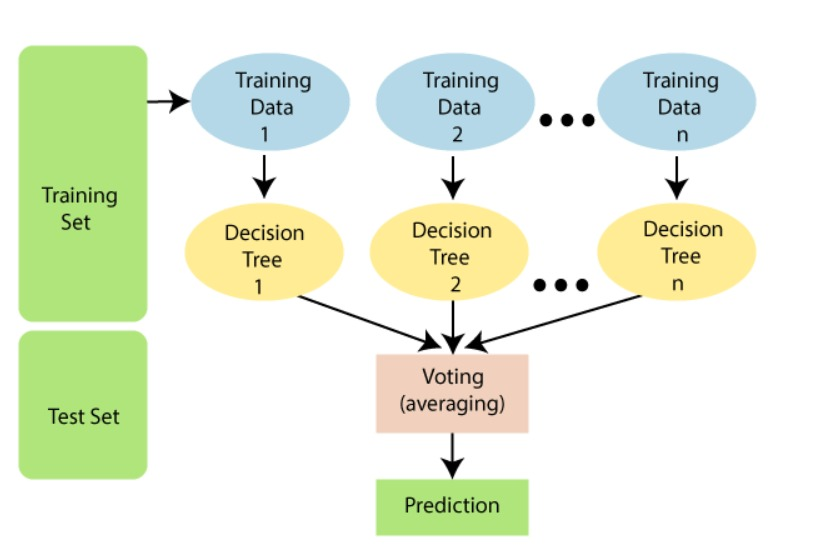
**D)Random Forest Classifier:**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

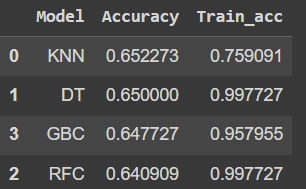
As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm



**ACCURACY MEASURED BY DIFFERENT MODELS**



The accuracy compared to the other models the KNN model will give the more accuracy

**5.4 Output Screens and Result Analysis**

**5.4.1.Source Code:**

import numpy as np

import pandas as pd

# for data visualizations

import matplotlib.pyplot as plt

import seaborn as sns

plt.style.use('fivethirtyeight')

# for interactivity

import ipywidgets

from ipywidgets import interact

import os

data = pd.read\_csv("sample1.csv")

# lets check teh shape of the dataset

print("Shape of the Dataset :", data.shape)

data.isnull().sum()

data['label'].value\_counts()

print("Average Ratio of Nitrogen in the Soil : {0:.2f}".format(data['N'].mean()))

print("Average Ratio of Phosphorous in the Soil : {0:.2f}".format(data['P'].mean()))

print("Average Ratio of Potassium in the Soil : {0:.2f}".format(data['K'].mean()))

@interact

def summary(crops = list(data['label'].value\_counts().index)):

    x = data[data['label'] == crops]

    print("---------------------------------------------")

    print("Statistics for Nitrogen")

    print("Minimum Nitrigen required :", x['N'].min())

    print("Average Nitrogen required :", x['N'].mean())

    print("Maximum Nitrogen required :", x['N'].max())

    print("---------------------------------------------")

    print("Statistics for Phosphorous")

    print("Minimum Phosphorous required :", x['P'].min())

    print("Average Phosphorous required :", x['P'].mean())

    print("Maximum Phosphorous required :", x['P'].max())

    print("---------------------------------------------")

    print("Statistics for Potassium")

    print("Minimum Potassium required :", x['K'].min())

    print("Average Potassium required :", x['K'].mean())

    print("Maximum Potassium required :", x['K'].max())

    print("---------------------------------------------")

## Lets compare the Average Requirement for each crops with average conditions

@interact

def compare(conditions = ['N','P','K']):

    print("Average Value for", conditions,"is {0:.2f}".format(data[conditions].mean()))

    print("----------------------------------------------")

    print("Rice : {0:.2f}".format(data[(data['label'] == 'rice')][conditions].mean()))

    print("Black Grams : {0:.2f}".format(data[data['label'] == 'blackgram'][conditions].mean()))

    print("Banana : {0:.2f}".format(data[(data['label'] == 'banana')][conditions].mean()))

    print("Jute : {0:.2f}".format(data[data['label'] == 'jute'][conditions].mean()))

    print("Coconut : {0:.2f}".format(data[(data['label'] == 'coconut')][conditions].mean()))

    print("Apple : {0:.2f}".format(data[data['label'] == 'apple'][conditions].mean()))

    print("Papaya : {0:.2f}".format(data[(data['label'] == 'papaya')][conditions].mean()))

    print("Muskmelon : {0:.2f}".format(data[data['label'] == 'muskmelon'][conditions].mean()))

    print("Grapes : {0:.2f}".format(data[(data['label'] == 'grapes')][conditions].mean()))

    print("Watermelon : {0:.2f}".format(data[data['label'] == 'watermelon'][conditions].mean()))

    print("Kidney Beans: {0:.2f}".format(data[(data['label'] == 'kidneybeans')][conditions].mean()))

    print("Mung Beans : {0:.2f}".format(data[data['label'] == 'mungbean'][conditions].mean()))

    print("Oranges : {0:.2f}".format(data[(data['label'] == 'orange')][conditions].mean()))

    print("Chick Peas : {0:.2f}".format(data[data['label'] == 'chickpea'][conditions].mean()))

    print("Lentils : {0:.2f}".format(data[(data['label'] == 'lentil')][conditions].mean()))

    print("Cotton : {0:.2f}".format(data[data['label'] == 'cotton'][conditions].mean()))

    print("Maize : {0:.2f}".format(data[(data['label'] == 'maize')][conditions].mean()))

    print("Moth Beans : {0:.2f}".format(data[data['label'] == 'mothbeans'][conditions].mean()))

    print("Pigeon Peas : {0:.2f}".format(data[(data['label'] == 'pigeonpeas')][conditions].mean()))

    print("Mango : {0:.2f}".format(data[data['label'] == 'mango'][conditions].mean()))

    print("Pomegranate : {0:.2f}".format(data[(data['label'] == 'pomegranate')][conditions].mean()))

    print("Coffee : {0:.2f}".format(data[data['label'] == 'coffee'][conditions].mean()))

# lets make this funtion more Intuitive

@interact

def compare(conditions = ['N','P','K']):

    print("Crops which require greater than average", conditions,'\n')

    print(data[data[conditions] > data[conditions].mean()]['label'].unique())

    print("----------------------------------------------")

    print("Crops which require less than average", conditions,'\n')

    print(data[data[conditions] <= data[conditions].mean()]['label'].unique())

plt.rcParams['figure.figsize'] = (15, 7)

plt.subplot(2, 4, 1)

sns.distplot(data['N'], color = 'grey')

plt.xlabel('Ratio of Nitrogen', fontsize = 12)

plt.grid()

plt.subplot(2, 4, 2)

sns.distplot(data['P'], color = 'blue')

plt.xlabel('Ratio of Phosphorous', fontsize = 12)

plt.grid()

plt.subplot(2, 4, 3)

sns.distplot(data['K'], color ='green')

plt.xlabel('Ratio of Potassium', fontsize = 12)

plt.grid()

plt.suptitle('Distribution for Agricultural Conditions', fontsize = 20)

plt.show()

### Data Visualizations

plt.rcParams['figure.figsize'] = (15, 8)

plt.subplot(2, 4, 1)

sns.barplot(data['N'], data['label'])

plt.ylabel(' ')

plt.xlabel('Ratio of Nitrogen', fontsize = 10)

plt.yticks(fontsize = 10)

plt.subplot(2, 4, 2)

sns.barplot(data['P'], data['label'])

plt.ylabel(' ')

plt.xlabel('Ratio of Phosphorous', fontsize = 10)

plt.yticks(fontsize = 10)

plt.subplot(2, 4, 3)

sns.barplot(data['K'], data['label'])

plt.ylabel(' ')

plt.xlabel('Ratio of Potassium', fontsize = 10)

plt.yticks(fontsize = 10)

plt.suptitle('Visualizing the Impact of Different Conditions on Crops', fontsize = 15)

plt.show()

# lets split the Dataset for Predictive Modelling

y = data['label']

x = data.drop(['label'], axis = 1)

print("Shape of x:", x.shape)

print("Shape of y:", y.shape)

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

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2, random\_state = 0)

print("The Shape of x train:", x\_train.shape)

print("The Shape of x test:", x\_test.shape)

print("The Shape of y train:", y\_train.shape)

print("The Shape of y test:", y\_test.shape)

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import BaggingClassifier

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.ensemble import AdaBoostClassifier

from xgboost import XGBClassifier

import xgboost as xgb

from sklearn.metrics import classification\_report

from sklearn.metrics import accuracy\_score,confusion\_matrix,roc\_auc\_score

from mlxtend.plotting import plot\_confusion\_matrix

def evaluator(y\_test, y\_pred):

    # Accuracy:

    print('Accuracy is: ', accuracy\_score(y\_test,y\_pred))

    print('')

    # Classification Report:

    print('Classification Report: \n',classification\_report(y\_test,y\_pred))

    print('Confusion Matrix: \n\n')

    plt.style.use("ggplot")

    cm = confusion\_matrix(y\_test,y\_pred)

    plot\_confusion\_matrix(conf\_mat = cm,figsize=(10,10),show\_normed=True)

    plt.title('Confusion Matrix for Logistic Regression', fontsize = 15)

    plt.show()

model\_accuracy = pd.DataFrame(columns=['Model','Accuracy'])

models = {

          "KNN" : KNeighborsClassifier(),

          "DT" : DecisionTreeClassifier(),

          'RFC' : RandomForestClassifier(),

          'GBC' : GradientBoostingClassifier(),

          }

for test, clf in models.items():

    clf.fit(x\_train, y\_train)

    y\_pred = clf.predict(x\_test)

    acc = accuracy\_score(y\_test,y\_pred)

    train\_pred = clf.predict(x\_train)

    train\_acc = accuracy\_score(y\_train, train\_pred)

    print("\n", test + ' scores')

    print(acc)

    print(classification\_report(y\_test,y\_pred))

    print(confusion\_matrix(y\_test,y\_pred))

    print('\*' \* 100,"\n")

    model\_accuracy = model\_accuracy.append({'Model': test, 'Accuracy': acc, 'Train\_acc': train\_acc}, ignore\_index=True)

model\_accuracy.sort\_values(ascending=False, by = 'Accuracy')

from sklearn.neighbors import KNeighborsClassifier

kn\_classifier = KNeighborsClassifier()

kn\_classifier.fit(x\_train,y\_train)

pred\_kn = kn\_classifier.predict(x\_test)

evaluator(y\_test, pred\_kn)

data.head()

with open('npk.txt') as f:

  datafile = f.readlines()

  npk=[]

for line in datafile:

  if 'n=' in line:

    n=str(line)

    npk.insert(0,line)

  if 'p=' in line:

    npk.insert(1,line)

    p=str(line)

  if 'k=' in line:

    npk.insert(2,line)

    k=str(line)

inputdata = list(map(lambda x: x.replace('n=','').replace('p=','').replace('k=','').replace('\n',''),npk))

print(inputdata)

inarr=[]

print(type(inputdata))

for i in inputdata:

  inarr.append(float(i))

print(inarr)

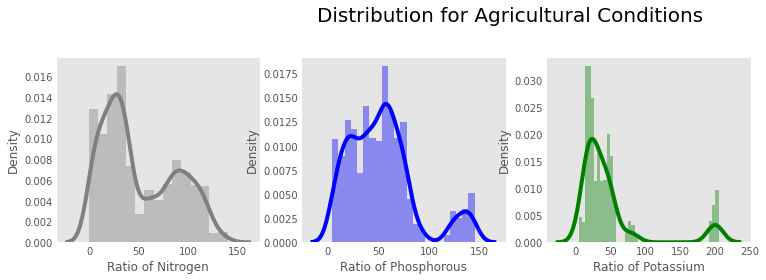
prediction = kn\_classifier.predict((np.array([[inarr[0],

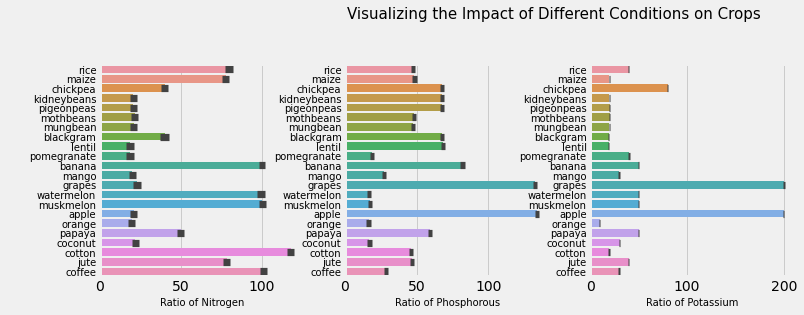
                                       inarr[1],

                                       inarr[2]]])))

print("The Suggested Crop for Given NPK VALUES is :", prediction)

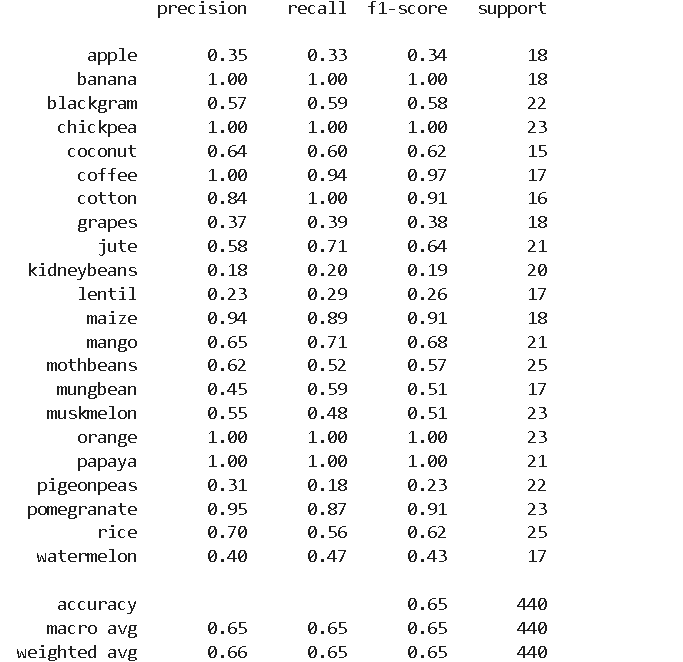
**5.4.2:Graphs:**

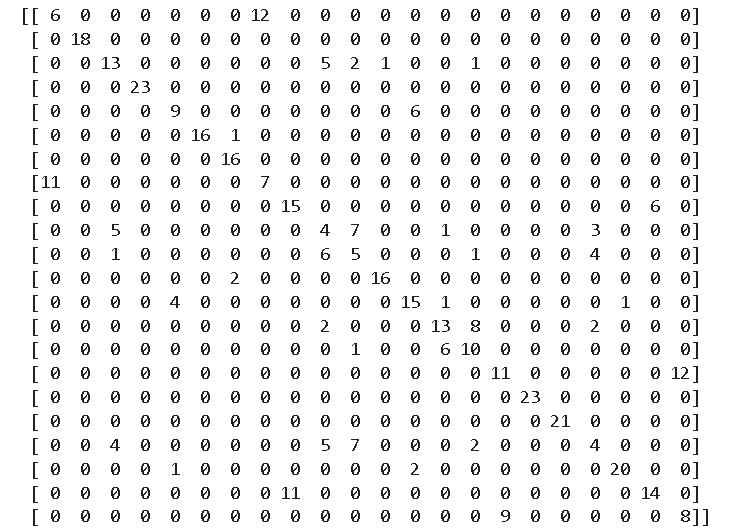
****

****

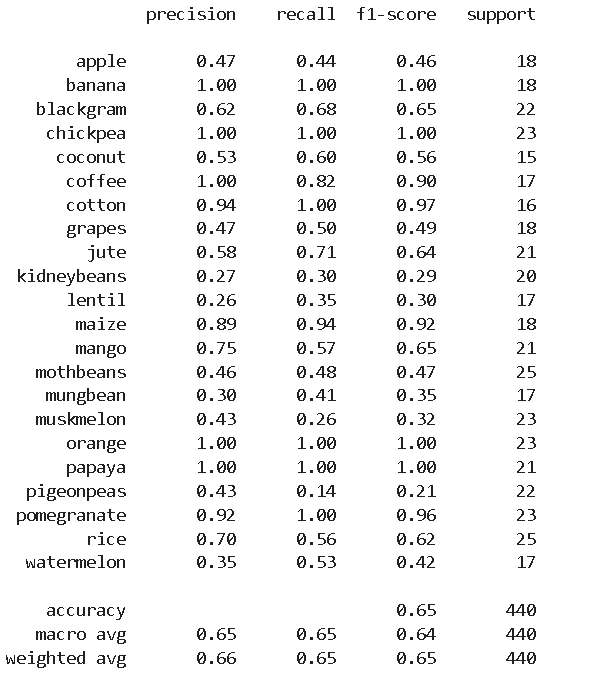
**CLASSIFICATION REPORT AND CONFUSION MATRIX**

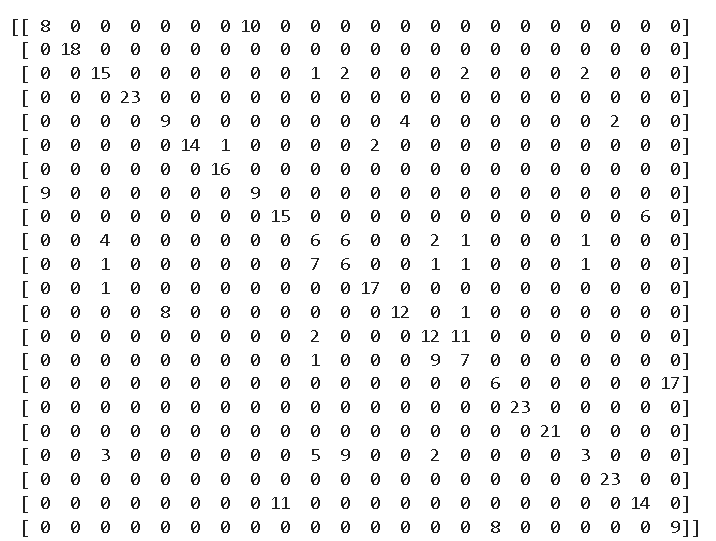
**KNN:** **0.6522727272727272**

****

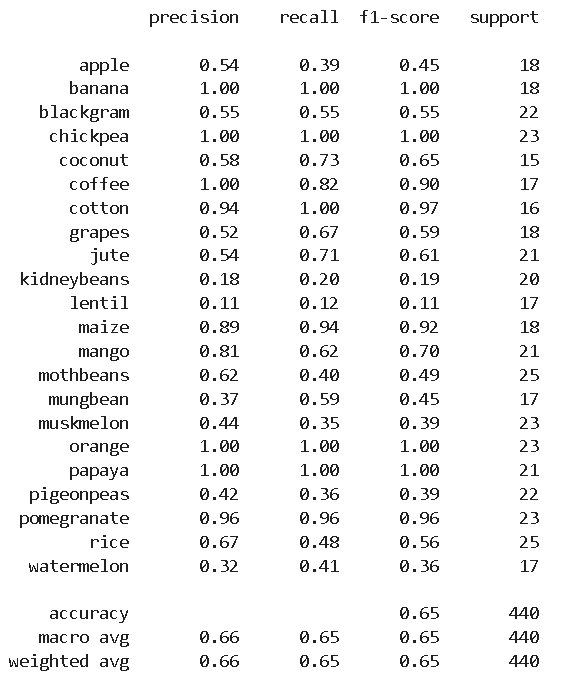
****

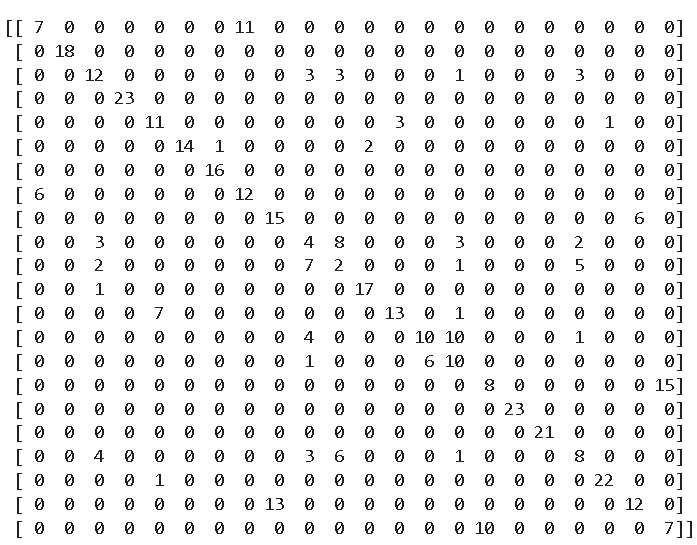
**Decision Tree:** **0.65**

****

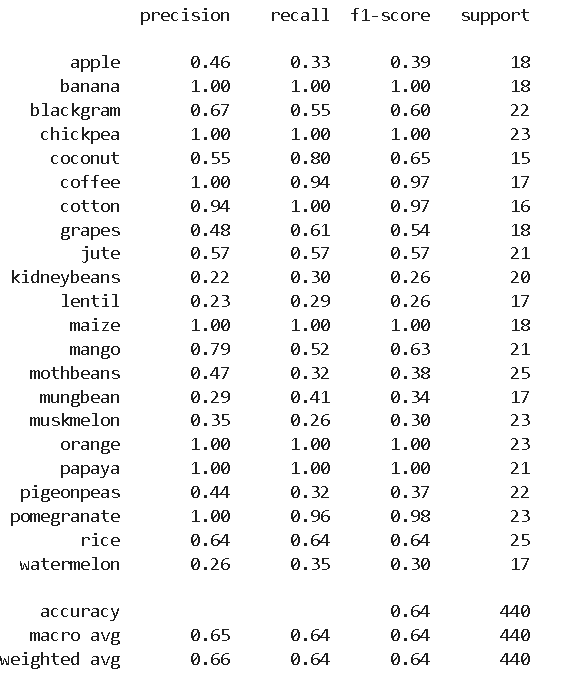
****

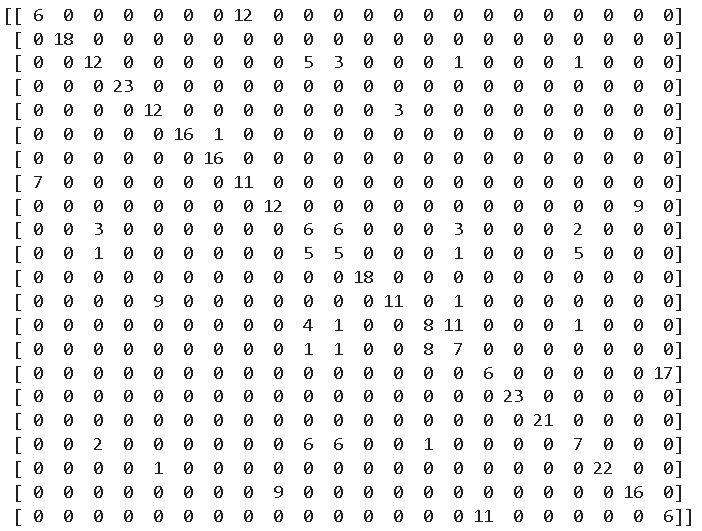
**Gradient Boosting Classifier:** **0.6477272727272727**

****

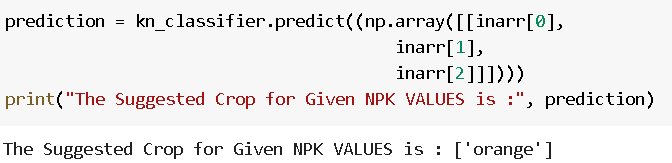
****

**Random Forest Classifier:** **0.6409090909090909**

****

****

**5.4.3:OUTPUT:**

****

**CHAPTER 6**

**TESTING AND VALIDATION**

**6.1 Introduction:**

Testing is a process, which reveals errors in the program. It is the most important quality metric used in software development. A series of testcases is used to run the application. The output of the program for the test cases is analyzed to see if the program is operating as expected .To ensure that the system does not have errors, the different levels of testing strategies are applied at differing phases of software development.

**6.2 TYPES OF TESTING**

**6.2.1 UNIT TESTING**

Unit testing entails creating test cases to ensure that the internal program logic is working properly and that program inputs result in legitimate outputs. Validation should be performed on all decision branches and internal code flow. It examines the application's individual software units. It's done after a unit is finished but before it's integrated. This structural testing relies on Knowledge of its construction and is invasive. Unit tests perform essential components and test a specific business process, application, and system configuration. Unit tests guarantee that each individual path of a business process follows the published specifications and has clearly defined inputs and outputs.

**6.2.2 FUNCTIONAL TESTING**

Functional tests demonstrate that the functions being tested are available in accordance with 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: interfacing systems or procedures must be invoked.

The organization and preparation of functional tests are focused on requirements, key functions, or exceptional test cases. In addition, systematic coverage about identifying Business process flows, data fields, predefined methods, and successive techniques must be considered for testing. Before functional testing is complete, additional tests are identified, and the practical value of current tests is determined.

**6.2.3 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event-driven and is more concerned with the primary outcome of screens or fields. Integration tests verify that, while individual components were satisfied, the combination of components is right and consistent, as demonstrated by successful unit testing. Integration testing is a type of testing that focuses on uncovering issues that occur from the integration of components.

**6.2.4 SYSTEM TESTING**

System testing guarantees that the complete integrated software system complies with the specifications. It checks a setup to ensure that the results are known and predictable. An example of system testing is the configuration-oriented system integration test. Process descriptions and flows are used to test systems, with an emphasis on pre-driven process connections and integration points.

**6.2.5 WHITE-BOX TESTING**

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

**6.2.6 BLACK BOX TESTING**

Black Box Testing tests the software without knowing the inner workings, structure, or language of the tested module. The majority of testing are black box tests. Other tests must be written using a definite source document, such as a 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 and responds to outputs without considering how the software works. 44

**6.3 DESIGN OF TEST CASES AND SCENARIOS**

Here the dataset is divided for both the testing and training the model by the criteria i.e., 70% will be used for training and 30% will be used to evaluate the model. We use the inbuilt models in python it train\_test\_split for this purpose which made our job much easier. 6.4 Validation: Test strategy and approach Field testing will be performed manually, and functional tests will be written in detail0.

**6.3.1 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

**6.3.2 Features to be tested**

• Verify that the entries are in the correct format

• No duplicate entries should be allowed

**6.3.3 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.

**CHAPTER 7**

**CONCLUSION**

Presently our farmers are not effectively using technology and analysis, so there may be a chance of wrong selection of crop for cultivation that will reduce their income. To reduce those type of loses we have developed a farmer friendly system with GUI, that will predict which would be the best suitable crop for particular land and this system will also provide information about required nutrients to add up, required seeds for cultivation, expected yield and market price. So, this makes the farmers to take right decision in selecting the crop for cultivation such that agricultural sector will be developed by innovative idea

**CHAPTER 8**

**REFERENCES**

* Shima Ramesh Maniyath,Mr. Ramachandra Hebbar, ”Soil Color Detection Using Knn Classifier”2018 International Conference on Design Innovations for 3Cs Compute Communicate Control**”**
* Rakesh Kumar, M.P. Singh, Prabhat Kumar and J.P. Singh (2015), ’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 (ICSTM)
* Dhanush Vishwakarma, Mahendra N,Ashwini, Manjuraju M.R,”Crop Prediction using Machine Learning”(2020)
* Rakesh Kumar, M.P. Singh, Prabhat Kumar and J.P. Singh,”Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique”,Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, T.N., India. 6 - 8 May 2015. pp.138-145
* Ramya M, Chetan Balaji, Girish L, “Environment Change Prediction to Adapt Climate-Smart Agriculture Using Big Data Analytics”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 5, May 2015.
* QiaoYing,ChenHao,ǁThe Design of smart cloud computing system, International Conference on Computational andInformation Sciences,IEEE,2011.
* M. GunasundariAnanthara, Dr. T. Arunkumar, Ms. R. Hemavathy “CRY – An improved Crop Yield Prediction model using Bee Hive Clustering Approach for Agricultural data sets”Proceedings of the 2013 International Conference onPattern Recognition, Informatics and Mobile Engineering, pp. 473 – 479, February 21-22, 2013.
* U.K. Shanwad, V.C. Patil and H. Honne Gowda “Precision Farming: Dreams and Realities for Indian Agriculture”, Map India, 2014.
* H. Guo, H. L. Viktor “Multirelational Cluster: AMultiple ViewApproach” Springer Knowledge Information System, Vol.17, Issue-3,pg.no-287–312, 2008.
* P. Victer Paul, A. Ramalingam, R. Baskaran, P. Dhavachelvan, K. Vivekanandan and R. Subramanian, “A new population seeding technique for permutation-coded Genetic Algorithm: Service transfer approach”, Journal of Computational Science, Elsevier, Issue 5, 2014, pp. 277–297. ISSN: 1877-7503.