### **CHAPTER 1**

### **INTRODUCTION**

**1.1 Overview**

Crop Recommendation is a critical task for farmers and decision-makers in the agricultural sector. The ability to accurately forecast suitable crops for cultivation based on environmental and soil parameters can significantly impact agricultural productivity and profitability. This project aims to address this challenge by developing a crop prediction model that leverages machine learning techniques to provide farmers with informed recommendations.

The primary objective of this project is to assist farmers in selecting the most suitable crops for cultivation by analyzing predicted rainfall, soil contents, and weather parameters. By considering these factors, the model can predict the crops that are likely to thrive under specific conditions. Moreover, the model provides additional insights such as the required fertilizers (Nitrogen, Phosphorus, and Potassium) in kilograms per hectare and the necessary seed quantity in kilograms per acre for the recommended crop.

#### **1.2 Purpose**

This project aims to empower farmers with accurate information for making crop selection decisions. By recommending crops suitable for prevailing environmental conditions, farmers can optimize their practices and improve yields. The system also provides market prices and approximate yields for the recommended crop, enabling farmers to consider profitability. The crop prediction model utilizes the Random Forest Classifier algorithm, known for its accuracy in handling complex datasets. By training the model on historical crop data and environmental parameters, it can learn patterns and correlations. The integration of the model into a user-friendly web application enhances its accessibility and usability, benefiting farmers in their decision-making process. In conclusion, this project develops a crop prediction model that leverages machine learning and historical data to provide accurate recommendations based on environmental parameters. The model's integration into a web application empowers farmers to make informed decisions about crop selection, optimizing their practices and increasing profitability.

**1.3 Objectives**

* Data set collection from various sources.
* Data parsing and cleansing technique is applied to make the raw data into processing data.
* The data collected is subject to machine learning system along with run time analysis makes an efficient crop value updation system.
* Usage of Ensemble of classifiers makes the model more robust and efficient.
* Ranking technique used in the project helps us to make efficient decisions.
* Creating a web application for user registrations and collection of data.
* The main objective is to obtain a better variety of crops that can be grown over the season. The proposed system would help to minimize the difficulties faced by farmers in choosing a crop and maximize the yield.
* The model predicts the crop yield by studying factors such as rainfall, temperature, area, season, soil type etc

**1.4 Scope of the project.**

**1.4.1. Soil Analysis**

* Soil Type Identification: ML models can analyze soil samples to determine the type of soil and its suitability for different crops.
* Nutrient Analysis: Using data from soil sensors, ML can predict nutrient deficiencies and recommend appropriate fertilizers.

**1.4.2. Climate and Weather Analysis**

* Weather Prediction: ML models can predict short-term and long-term weather patterns, helping farmers to plan planting and harvesting times.
* Climate Suitability: By analyzing historical climate data, ML can recommend crops that are best suited for the predicted climatic conditions.

**1.4.3. Crop Selection**

* Crop Suitability: Based on soil properties, weather conditions, and historical crop performance data, ML can recommend the best crops to plant.
* Yield Prediction: Predicting the potential yield of different crops based on current conditions and historical data.

**1.4.4. Pest and Disease Management**

* Pest Detection: ML models can detect pests through image recognition and recommend appropriate pest control measures.
* Disease Prediction: By analyzing environmental conditions and plant health data, ML can predict the likelihood of disease outbreaks and suggest preventive measures.

**1.4.5. Resource Management**

* Water Management: Recommending efficient irrigation schedules based on weather forecasts and soil moisture levels.
* Fertilizer Use: Optimizing the use of fertilizers by predicting crop nutrient needs at different growth stages.

**1.4.6. Economic Factors**

* Market Trends: Analyzing market trends and prices to recommend crops that will be economically profitable.
* Cost Analysis: Estimating the cost of production and potential revenue to help farmers make informed decisions.

**1.4.7. Precision Agriculture**

* Variable Rate Technology (VRT): Using ML to control the application rates of seeds, fertilizers, and pesticides based on the specific needs of different parts of a field.
* Drone and Satellite Imaging: Analyzing images captured by drones and satellites to monitor crop health and development.

**1.5 Implementation Steps:**

* Data Collection: Gather data from various sources such as soil sensors, weather stations, satellite imagery, and historical agricultural records.
* Data Preprocessing: Clean and preprocess the data to make it suitable for ML models. This includes handling missing values, normalizing data, and feature selection.
* Model Selection: Choose appropriate ML models (e.g., decision trees, random forests, neural networks) based on the problem requirements.
* Model Training: Train the models using historical data and validate their performance using test datasets.
* Model Deployment: Deploy the trained models into a production environment where they can make real-time recommendations.
* Monitoring and Maintenance: Continuously monitor the model performance and update them with new data to maintain accuracy.
  1. **Challenges:**
* Data Quality: Ensuring the availability of high-quality, reliable data.
* Model Interpretability: Making the recommendations of complex ML models understandable to farmers.
* Integration: Integrating ML solutions with existing agricultural practices and technologies.
  1. **Future Prospects:**
* Integration with IoT: Combining ML with Internet of Things (IoT) devices for real-time data collection and analysis.
* Advanced Analytics: Using advanced analytics like deep learning for more accurate predictions and recommendations.
* Personalized Farming: Developing personalized farming solutions tailored to individual farms' specific conditions.
  1. **Hardware/Software designing**

The hardware requirements for this project are minimal as it primarily involves software development.

The software requirements include:

* Python programming language
* Pandas library for data manipulation
* scikit-learn library for machine learning algorithms
* Flask framework for building the web application.

**CHAPTER 2**

**REQUIREMENT SPECIFICATION**

**2.1 INTRODUCTION**

A software requirements specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements, and may include a set of use cases that describe interactions the users will have with the software. A basic purpose of the SRS is to bridge this communication gap between client and the developer so they have a shared vision of the software being built. An SRS establishes the basis for agreement between the client and the supplier on what the software product will do. SRS provides a reference for validation of the final product. A high-quality SRS is a prerequisite to high-quality software and also reduces the development cost. The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete “Crop recommendation”by defining the problem statement in detail. The detailed requirements of “Crop Recommendation” are provided in this document.

**2.2 HARDWARE REQUIREMENTS**

* Operating System: Windows XP, 7 OR 8
* Processor: Intel Core Duo 2.0 GHz or more
* RAM: 1GB or more
* Hard Disc: 80GB or more
* Monitor: 15 inches CRT or LCD Monitor
* Keyboard: Normal or multimedia keyboard
* Mouse: Compatible Mouse

**2.3 SOFTWARE REQUIREMNTS**

* Notepad++ or Visual Studio Code.

**2.4 OTHER NON-FUNCTIONAL REQUIREMENTS**

**2.4.1 Performance**

* Response Time: The system should provide recommendations within an acceptable time frame to ensure a smooth user experience.
* Throughput : The system should handle multiple requests simultaneously without degradation in performance.

**2.4.2. Scalability**

* Horizontal and Vertical Scaling: The system should support scaling out (adding more machines) and scaling up (upgrading the existing machine's resources) to handle increasing loads.
* Elasticity: The system should automatically adjust its resources based on the current demand.

**2.4.3. Reliability**

* Availability: The system should have high uptime and be accessible whenever users need it.
* Fault Tolerance: The system should continue to operate properly in the event of a failure of some of its components.

**2.4 4. Usability**

* User Interface: The interface should be intuitive and easy to use for farmers and other end-users.
* Accessibility: The system should be accessible to users with disabilities, following accessibility guidelines.

**2.4.5. Security**

* Data Privacy: Ensure that user data is protected and used only for the intended purposes
* Authentication and Authorization: Implement robust authentication and authorization mechanisms to prevent unauthorized access.

**2.4.6. Maintainability**

* Code Quality: Ensure the code is well-documented, modular, and follows best practices to facilitate maintenance.
* Error Handling and Logging: Implement comprehensive error handling and logging to aid in troubleshooting and maintenance

**2.4.7. Interoperability**

* API Integration: The system should support integration with other systems and services via APIs.
* Data Standards: Use standardized data formats to ensure compatibility with other systems.

**CHAPTER 3**

**LITERATURE SURVEY**

**[1]Title:** A Review on Data Mining Techniques for Fertilizer Recommendation 2018. Authors : Jignasha M. Jethva, Nikhil Gondaliya, Vinita Shah To keep up nutrition levels in the soil in case of deficiency, fertilizers are added to soil. The standard issue existing among the Indian agriculturists choose approximate amount of fertilizers and add them manually. Excess or deficient extension of fertilizers can harm the plants life and reduce the yield. This paper gives overview of various data mining frameworks used on cultivating soil dataset for fertilizer recommendation.

**[2]Title:** A Survey on Data Mining Techniques in Agriculture, 2015. Authors : M.C.S.Geetha Agriculture is the most critical application area especially in the developing nations like India .Use of information technology in agriculture can change the situation of decision making and farmers can yield in better way.. This paper integrates the work of several authors in a single place so it is valuable for specialists to get data of current situation of data mining systems and applications in context to farming field.

**[3]Title :** AgroNutri Android Application,2016. Authors : S. Srija, R. Geetha Chanda, S.Lavanya, Dr. M. Kalpana Ph.D This paper communicates the idea regarding the making of AgroNutri an android application that helps in conveying the harvest particular fertilizer amount to be applied. The idea is to calculate the measure of NPK composts to be applied depend on the blanked proposal of the crop of interest. This application works depend on the product chosen by the farmer and that is taken as input, thus providing the farmers. The future scope of the AgroNutri is that GPRS can be included so that according to location nutrients are suggested. 9

**[4]Title:** Machine Learning: Applications in Indian Agriculture, 2016. Authors: Karandeep Kaur Agriculture is a field that has been lacking from adaption of technologies and their advancements. Indian agriculturists should be up to the mark with the universal procedures. Machine learning is a native concept that can be applied to every field on all inputs and outputs. It has effectively settled its ability over ordinary calculations of software engineering and measurements. Machine learning calculations have improved the exactness of artificial intelligence machines including sensor based frameworks utilized in accuracy farming. This paper has evaluated the different uses of machine learning in the farming area. It additionally gives a knowledge into the inconveniences looked by Indian farmers and how they can be resolved using these procedures.

**[5]Title:** Impacts of population growth, economic development, and technical change on global food production and consumption, 2011. Author: Uwe A. Schneider a,⇑, Petr Havlik b, Erwin Schmid c, Hugo Valin b, Aline Mosnier b,c, Michael Obersteiner b, Hannes Bottcher b, Rastislav Skalsky´ d, Juraj Balkovicˇ d, Timm Sauer a, Steffen Fritz b Throughout the following decades humanity will request more food from less land and water assets. This investigation evaluates the food production effects of four elective advancement situations from the Millennium Ecosystem Assessment and the Special Report on Emission Scenarios. partialy and jointly considered are land and water supply impacts from population development, and specialized change, and forests and agriculture demand request shifts from population development and economic improvement. The income impacts on nourishment request are registered with dynamic flexibilities. Worldwide farming area increments by up to 14% somewhere in the range of 2010 and 2030.Deforestation restrictions strongly impact the price of land and water resources but have little consequences for the global level of food production and food prices. While projected income changes have the highest partial impact on per capita food consumption levels, population growth leads to the highest increase in total food production. The impact of technical change is amplified or mitigated by adaptations of land management intensities 10

**[6]Title:** A Smart Agricultural Model by Integrating Iot, Mobile and Cloud-based Big Data Analytics, 2017. Authors: S.Rajeswari, K.Suthendran, K.Rajkumar. In the cultivating field, the system models play a significant role to the enhancement of the agro-normal and money related conditions. In the proportions of benefits of the field and farm examinations to give the information and to recognize fitting and fruitful organization practices. It can recognize the organization to arrive managers and transversely over reality as long as the required soil, the board, environment, and money related information. Decision Support Systems (DSSs) use to make the information for the vermin the board, develop the officials. These systems are not using the impelled strategies to process the data. Thusly, use the adroit system thoughts to take the decisions for the issue. It expects a crucial activity in the comprehension of agronomic results, and their use as decision sincerely steady systems for farmers is extending.

#### **3.1 Existing problem**

Crop prediction are crucial for agricultural decision-making. Traditional approaches to crop prediction, relying on manual analysis of historical data, have limitations in terms of accuracy and efficiency. To overcome these challenges, this project proposes a machine learning-based approach. By leveraging historical data and advanced algorithms, the model aims to provide more accurate and efficient predictions for optimal crop selection. The integration of this model into a user-friendly web application enhances accessibility and usability, allowing farmers to input parameters and receive real-time crop recommendations. The system also provides information on fertilizers, seed quantities, market prices, and estimated yields. Future enhancements could involve incorporating real-time weather data, expanding the dataset to include more crops, and integrating additional features like pest and disease prediction.

In conclusion, the development of a machine learning-based crop prediction model offers a promising solution for improving traditional approaches. By leveraging historical data, advanced algorithms, and user-friendly interfaces, the model empowers decision-makers and farmers to optimize crop selection and enhance agricultural productivity. Continual refinement and expansion of the model hold significant potential for revolutionizing crop prediction and promoting sustainable agricultural practices.

#### **3.2 Proposed solution**

This project proposes a machine learning-based approach for crop prediction by leveraging historical crop data and corresponding environmental parameters. The Random Forest Classifier algorithm is chosen as the foundation for the crop prediction model due to its suitability for classification tasks. By constructing multiple decision trees and combining their predictions, the Random Forest algorithm generates accurate predictions for crop selection based on input parameters.

The Random Forest Classifier algorithm effectively analyzes the relationships between environmental parameters such as rainfall, soil contents, temperature, humidity, and pH level. These factors significantly influence crop growth and yield. By considering these parameters, the model provides recommendations for the most suitable crop for cultivation.

The machine learning-based approach allows the model to handle complex patterns and interactions between environmental parameters and crop selection. By training the model on a comprehensive dataset of historical crop data, it learns from past experiences to make informed predictions.

The Random Forest Classifier algorithm offers advantages such as high accuracy, robustness, and built-in feature selection. It is capable of handling both numerical and categorical features and is less prone to overfitting due to the combination of multiple decision trees.

Overall, the proposed machine learning-based approach using the Random Forest Classifier algorithm provides a reliable solution for crop prediction. By considering a wide range of environmental parameters and historical crop data, the model assists farmers in making informed decisions, leading to increased agricultural productivity and profitability.

**CHAPTER 4**

**SYSTEM ANALYSIS AND DSEIGN**

**4.1 Introduction:**

* **System Analysis:** The system analysis approach emphasises a closed look on all parts of the system. The analyst must consider all the system elements, their inputs, outputs, control, feedback and the environment when the system is being constructed.
* **System Design**:The goal of system design phase is to produce a model or representation of the system, which can be used to build the system. Here the emphasis is on translating the requirements of the system into design specification.

1. **Applicable Documents:**

The document used in system design is Software Requirement Specification Document.

1. **Functional Decomposition:**

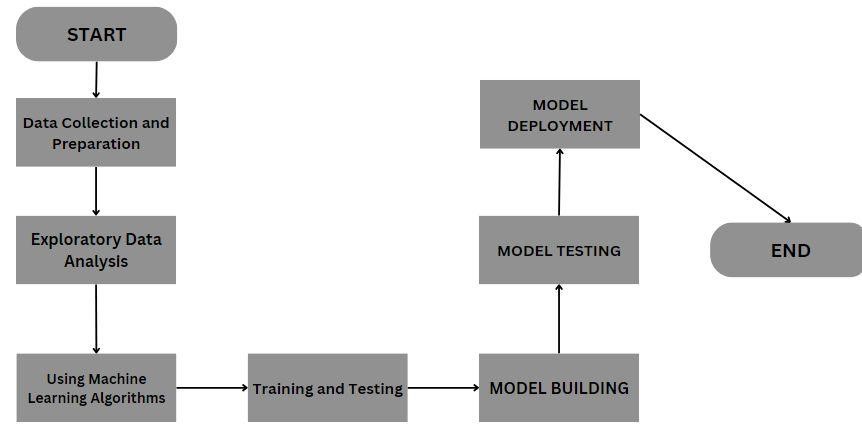
The system can be decomposed into functional components as follows. The Components –

* Data Collection: Gather data from various sources such as soil sensors, weather stations, satellite imagery, and historical agricultural records.
* Data Preprocessing: Clean and preprocess the data to make it suitable for ML models. This includes handling missing values, normalizing data, and feature selection.
* Model Selection: Choose appropriate ML models (e.g., decision trees, random forests, neural networks) based on the problem requirements.
* Model Training: Train the models using historical data and validate their performance using test datasets.
* Model Deployment: Deploy the trained models into a production environment where they can make real-time recommendations.
* Monitoring and Maintenance: Continuously monitor the model performance and update them with new data to maintain accuracy.

**4.2 . Program Description:**

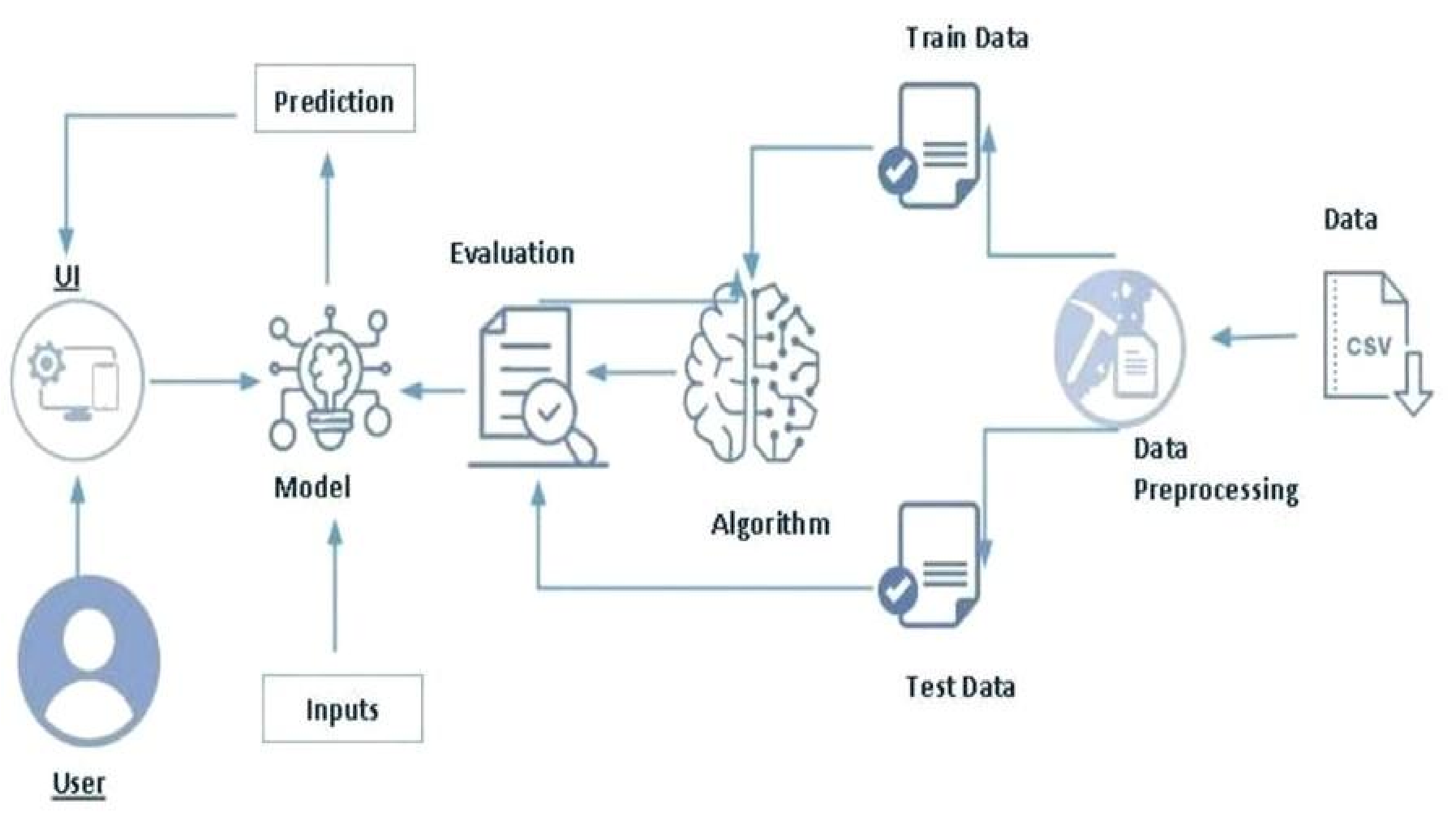
**4.2.1. Context Flow Diagram:**

The flowchart illustrates the control flow of the crop prediction model. It outlines the sequence of steps involved, from receiving input data to generating crop recommendations and displaying the relevant information to the user.



#### **4.2.2 Block diagram**

The block diagram provides an overview of the project architecture and its components. It illustrates the flow of data and the interactions between different modules, such as data preprocessing, model training, and prediction.



**4.3 Data Collection**

Dataset(Crop Recommendation) consists of 8 variables where 7 variables are considered for predicting output variable. The details of Variable is given Below.

1. %N (Nitrogen) : Nitrogen content in the soil. Nitrogen is really important for plant growth (structure), plant food processing (metabolism), and the creation of chlorophyll. Without enough nitrogen in the plant, the plant cannot grow taller, or produce enough food (usually yellow).
2. %P (Phosphorus) : Phosphorus content in the soil. Phosphorus primary role in a plant is to store and transfer energy produced by photosynthesis for use in growth and reproductive processes. Soil P cycles in a variety forms in the soil
3. %K (Potassium) : Potassium content in the soil. Potassium is an essential nutrient for plant growth.
4. Temperature : Temperature in degree celsius. High temperatures affect plant growth in numerous ways. The most obvious are the effects of heat on photosynthesis, in which plants use carbon dioxide to produce oxygen, and respiration, an opposite process in which plants use oxygen to produce carbon dioxide.
5. Humidity : Relative humidity in %. When conditions are too humid, it may promote the growth of mold and bacteria that cause plants to die and crops to fail, as well as conditions like root or crown rot. Humid conditions also invite the presence of pests, such as fungus gnats, whose larvae feed on plant roots and thrive in moist soil.
6. PH : ph value of the soil. Plant nutrients leach from the soil much faster at pH values below 5.5 than from soils within the 5.5 to 7.0 range. In some mineral soils aluminum can be dissolved at pH levels below 5.0 becoming toxic to plant growth. Soil pH may also affect the availability of plant nutrients.
7. Rainfall : Rainfall in mm. Plants use the moisture in the soil to replenish the water lost through transpiration. If there is no water in the soil, the leaves will wilt. As more water is lost, the plant will fail and eventually die. Rainwater builds up the moisture levels in the soil and assures a healthy plant. Finally,
8. Label : This is the output variable which contains 22 unique values i.e., 22 different cropsandtheyare['Apple','Banana','blackgram','chickpea','coconut','coffee','cotton','grapes','jute','kidneybeans','lentil','maize','mango','mothbeans','mung bean','muskmelon','orange','papaya','pigeonpeas','pomegranate','Rice','Watermelon']

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 Source Code**

**Crop recommendation.py**

import pandas as pdimport numpy as npfrom sklearn.metrics import classification\_reportfrom sklearn import metricsimport warningswarnings.filterwarnings('ignore')df = pd.read\_csv("crop\_recommendation.csv")features = df[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]target = df['label']# Splitting into train and test datafrom sklearn.model\_selection import train\_test\_splitXtrain, Xtest, Ytrain, Ytest = train\_test\_split(features,target,test\_size = 0.2,random\_state =2)##random forest algorith from sklearn.ensemble import RandomForestClassifierRF = RandomForestClassifier(n\_estimators=20, random\_state=0)RF.fit(Xtrain,Ytrain)predicted\_values = RF.predict(Xtest)x = metrics.accuracy\_score(Ytest, predicted\_values)print("RF's Accuracy is: {:.2f} ".format(x\*100))print(classification\_report(Ytest,predicted\_values))

from tkinter import \*from PIL import Image,ImageTkglobal rootroot = Tk()root.title('Crop Recommendation window')root.geometry('1500x750')img=Image.open("a.jpg")img=img.resize((1500,750))bg=ImageTk.PhotoImage(img)lbl=Label(root,image=bg)lbl.place(x=0,y=0)label=Label(root,text='CropRecommendation System',font=('arial',24,'bold'),bd=20,background="#CDD954")label.place(x=300,y=10)label\_1 = Label(root, text ='nitrogen',font=("Helvetica", 18),background="#CDD954")label\_1.place(x=150,y=100) Entry\_1= Entry(root,font=("Helvetica", 18),justify=CENTER)Entry\_1.place(x=450,y=100)label\_2 = Label(root, text ='phosporus',font=("Helvetica", 16),background="#CDD954")label\_2.place(x=150,y=160) Entry\_2 = Entry(root,font=("Helvetica", 18),justify=CENTER)Entry\_2.place(x=450,y=160) label\_3 = Label(root, text ='pottasium',font=("Helvetica", 18),background="#CDD954")label\_3.place(x=150,y=220) Entry\_3 = Entry(root,font=("Helvetica", 18),justify=CENTER)Entry\_3.place(x=450,y=220)label\_4 = Label(root, text ='temperature',font=("Helvetica", 18),background="#CDD954")label\_4.place(x=150,y=280) Entry\_4= Entry(root,font=("Helvetica", 18),justify=CENTER)Entry\_4.place(x=450,y=280)label\_5 = Label(root, text ='humidity',font=("Helvetica", 18),background="#CDD954")label\_5.place(x=150,y=340) Entry\_5 = Entry(root,font=("Helvetica", 18),justify=CENTER)Entry\_5.place(x=450,y=340)label\_6 = Label(root, text ='ph',font=("Helvetica", 18),background="#CDD954")label\_6.place(x=150,y=400) Entry\_6 = Entry(root,font=("Helvetica", 18),justify=CENTER)Entry\_6.place(x=450,y=400)label\_7 = Label(root, text ='rainfall',font=("Helvetica", 18),background="#CDD954")label\_7.place(x=150,y=460) Entry\_7 = Entry(root,font=("Helvetica", 18),justify=CENTER)Entry\_7.place(x=450,y=460)def acc(): image = Image.open("result.jpg") image = image.resize((300, 300), Image.ANTIALIAS) img = ImageTk.PhotoImage(image) global panel1 panel1 = Button(root10, image=img,command=close\_acc) panel1.image = img panel1.place(x=575,y=220)def clear\_out(): out\_img.destroy() output.configure(text="") Entry\_1.delete(0,END) Entry\_2.delete(0,END) Entry\_3.delete(0,END) Entry\_4.delete(0,END) Entry\_5.delete(0,END) Entry\_6.delete(0,END) Entry\_7.delete(0,END) def predict(): N = Entry\_1.get() P = Entry\_2.get() K = Entry\_3.get() temperature =Entry\_4.get() humidity =Entry\_5.get() ph =Entry\_6.get() rainfall = Entry\_7.get() out = RF.predict([[float(N), float(P), float(K), float(temperature), float(humidity), float(ph), float(rainfall)]]) print(out[0]) output.configure(text=" you want to grow "+str(out[0])) res\_img=Image.open("result\\"+str(out[0])+".jpg") res\_img=res\_img.resize((300,300),Image.ANTIALIAS) bgg=ImageTk.PhotoImage(res\_img)## out\_img.configure(image=bgg) global out\_img out\_img = Button(root,image=bgg,command=clear\_out) out\_img.image=bgg out\_img.place(x=800,y=300) b1 = Button(root, text = 'predict',font=("Helvetica", 18),background="#CDD954",command = predict)b1.place(x=150,y=550) output = Label(root,font=("Helvetica", 18),justify=CENTER)output.place(x=450,y=550) root.mainloop()

**test.py**

from tkinter import \*from PIL import Image,ImageTkroot = Tk()root.title('Crop Recommendation window')root.geometry('1500x750')img=Image.open("a.jpg")img=img.resize((1500,750))bg=ImageTk.PhotoImage(img)lbl=Label(root,image=bg)lbl.place(x=0,y=0)out=['rice']res\_img=Image.open("result\\"+str(out[0])+".jpg")res\_img=res\_img.resize((300,300))bgg=ImageTk.PhotoImage(res\_img)## out\_img.configure(image=bgg)out\_img = Label(root,image=bgg)out\_img.place(x=900,y=500)root.mainloop()

**Crop\_Recommendation\_Model.ipynb**

# Importing libraries

from \_\_future\_\_ import print\_function

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import classification\_report

from sklearn import metrics

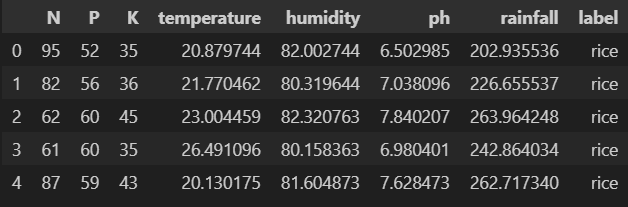
from sklearn import tree

import warnings

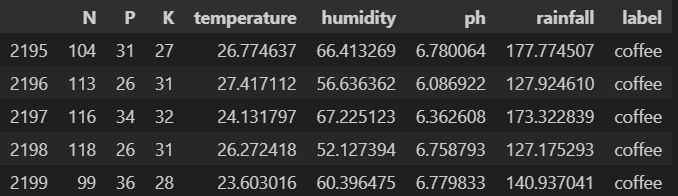
warnings.filterwarnings('ignore')

df = pd.read\_csv("crop\_recommendation.csv")

df.head()

****

df.tail()

****

df.size

17600

df.shape

(2200, 8)

df.columns

Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label'], dtype='object')

df['label'].unique()

array(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',

'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',

'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',

'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],

dtype=object)

df.dtypes

**N int64**

**P int64**

**K int64**

**temperature float64**

**humidity float64**

**ph float64**

**rainfall float64**

**label object**

**dtype: object**

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

**label**

**rice 100**

**maize 100**

**jute 100**

**cotton 100**

**coconut 100**

**papaya 100**

**orange 100**

**apple 100**

**muskmelon 100**

**watermelon 100**

**grapes 100**

**mango 100**

**banana 100**

**pomegranate 100**

**lentil 100**

**blackgram 100**

**mungbean 100**

**mothbeans 100**

**pigeonpeas 100**

**kidneybeans 100**

**chickpea 100**

**coffee 100**

**Name: count, dtype: int64**

import seaborn as sns

import pandas as pd

import numpy as np

# Sample DataFrame for illustration (replace with your actual DataFrame)

data = {

**'rice': ['a', 'b', 'c'],**

**'wheat': [1, 2, 3],**

**'corn': [4, 5, 6]**

}

df = pd.DataFrame(data)

# Select only numeric columns

numeric\_df = df.select\_dtypes(include=[np.number])

# Optionally handle missing values (e.g., drop or fill)

# numeric\_df = numeric\_df.dropna()  # or numeric\_df = numeric\_df.fillna(value)

sns.set(rc={'figure.figsize': (20, 15)})

sns.heatmap(numeric\_df.corr(), annot=True)



**Seperating features and target label**

# Print the column names of the DataFrame

print(df.columns)

# Check if the specified columns exist in the DataFrame

expected\_columns = ['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall']

missing\_columns = [col for col in expected\_columns if col not in df.columns]

if missing\_columns:

    print(f"The following columns are missing from the DataFrame: {missing\_columns}")

else:

    features = df[expected\_columns]

    target = df['label']

    print("Selected features and target successfully.")

Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label'], dtype='object')

Selected features and target successfully.

# Initialzing empty lists to append all model's name and corresponding name

acc = []

model = []

# Splitting into train and test data

import pandas as pd

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

# Load your dataset into DataFrame `df`

df = pd.read\_csv('crop\_recommendation.csv')  # or however you load your DataFrame

# Define features and target

features = df.drop(columns=['label'])

target = df['label']

# Splitting into train and test data

Xtrain, Xtest, Ytrain, Ytest = train\_test\_split(features, target, test\_size=0.2, random\_state=2)

print("Training features shape:", Xtrain.shape)

print("Test features shape:", Xtest.shape)

print("Training target shape:", Ytrain.shape)

print("Test target shape:", Ytest.shape)

Training features shape: (1760, 7)

Test features shape: (440, 7)

Training target shape: (1760,)

Test target shape: (440,)

# Decision Tree

from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random\_state=2,max\_depth=5)

DecisionTree.fit(Xtrain,Ytrain)

predicted\_values = DecisionTree.predict(Xtest)

x = metrics.accuracy\_score(Ytest, predicted\_values)

acc.append(x)

model.append('Decision Tree')

print("DecisionTrees's Accuracy is: ", x\*100

print(classification\_report(Ytest,predicted\_values))

Naive Bayes's Accuracy is: 0.9886363636363636

precision recall f1-score support

**apple 1.00 1.00 1.00 13**

**banana 1.00 1.00 1.00 17**

**blackgram 0.94 1.00 0.97 16**

**chickpea 1.00 1.00 1.00 21**

**coconut 1.00 1.00 1.00 21**

**coffee 1.00 1.00 1.00 22**

**cotton 1.00 1.00 1.00 20**

**grapes 1.00 1.00 1.00 18**

**jute 0.88 1.00 0.93 28**

**kidneybeans 1.00 1.00 1.00 14**

**lentil 1.00 1.00 1.00 23**

**maize 1.00 1.00 1.00 21**

**mango 1.00 1.00 1.00 26**

**mothbeans 1.00 0.95 0.97 19**

**mungbean 1.00 1.00 1.00 24**

**muskmelon 1.00 1.00 1.00 23**

**orange 1.00 1.00 1.00 29**

**papaya 1.00 1.00 1.00 19**

**pigeonpeas 1.00 1.00 1.00 18**

**pomegranate 1.00 1.00 1.00 17**

**rice 1.00 0.75 0.86 16**

**watermelon 1.00 1.00 1.00 15**

**...**

**accuracy 0.99 440**

**macro avg 0.99 0.99 0.99 440**

**weighted avg 0.99 0.99 0.99 440**

from sklearn.model\_selection import cross\_val\_score

# Cross validation score (Decision Tree)

score = cross\_val\_score(DecisionTree, features, target,cv=5)

score

**array([0.99772727, 0.99772727, 0.99318182, 0.99090909, 0.99090909])**

### Saving trained Decision Tree model

import pickle

# Dump the trained Naive Bayes classifier with Pickle

DT\_pkl\_filename = 'DecisionTree.pkl'

# Open the file to save as pkl file

DT\_Model\_pkl = open(DT\_pkl\_filename, 'wb')

pickle.dump(DecisionTree, DT\_Model\_pkl)

# Close the pickle instances

DT\_Model\_pkl.close()

# Support Vector Machine (SVM)

from sklearn.svm import SVC

# data normalization with sklearn

from sklearn.preprocessing import MinMaxScaler

# fit scaler on training data

norm = MinMaxScaler().fit(Xtrain)

X\_train\_norm = norm.transform(Xtrain)

# transform testing dataabs

X\_test\_norm = norm.transform(Xtest)

SVM = SVC(kernel='poly', degree=3, C=1)

SVM.fit(X\_train\_norm,Ytrain)

predicted\_values = SVM.predict(X\_test\_norm)

x = metrics.accuracy\_score(Ytest, predicted\_values)

acc.append(x)

model.append('SVM')

print("SVM's Accuracy is: ", x)

print(classification\_report(Ytest,predicted\_values))

# Cross validation score (SVM)

score = cross\_val\_score(SVM,features,target,cv=5)

score

#Saving trained SVM model

import pickle

# Dump the trained SVM classifier with Pickle

SVM\_pkl\_filename = 'SVMClassifier.pkl'

# Open the file to save as pkl file

SVM\_Model\_pkl = open(SVM\_pkl\_filename, 'wb')

pickle.dump(SVM, SVM\_Model\_pkl)

# Close the pickle instances

SVM\_Model\_pkl.close()

**SVM's Accuracy is: 0.9818181818181818**

**precision recall f1-score support**

**apple 1.00 1.00 1.00 13**

**banana 1.00 1.00 1.00 17**

**blackgram 0.94 1.00 0.97 16**

**chickpea 1.00 1.00 1.00 21**

**coconut 1.00 1.00 1.00 21**

**coffee 1.00 1.00 1.00 22**

**cotton 1.00 1.00 1.00 20**

**grapes 1.00 1.00 1.00 18**

**jute 0.93 0.89 0.91 28**

**kidneybeans 1.00 1.00 1.00 14**

**lentil 1.00 1.00 1.00 23**

**maize 1.00 1.00 1.00 21**

**mango 0.93 1.00 0.96 26**

**mothbeans 1.00 0.84 0.91 19**

**mungbean 1.00 1.00 1.00 24**

**muskmelon 1.00 1.00 1.00 23**

**orange 1.00 1.00 1.00 29**

**papaya 1.00 1.00 1.00 19**

**pigeonpeas 1.00 1.00 1.00 18**

**pomegranate 1.00 1.00 1.00 17**

**rice 0.82 0.88 0.85 16**

**watermelon 1.00 1.00 1.00 15**

**accuracy 0.98 440**

**macro avg 0.98 0.98 0.98 440**

**weighted avg 0.98 0.98 0.98 440**

import pandas as pd

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.svm import SVC

# Load your dataset into DataFrame `df`

# Assuming you have a CSV file named 'your\_dataset.csv'

df = pd.read\_csv('crop\_recommendation.csv')

# Define features and target

features = df.drop(columns=['label'])

target = df['label']

# Define the SVM model

SVM = SVC()

# Perform cross-validation

score = cross\_val\_score(SVM, features, target, cv=5)

print("Cross-validation scores:", score)

print("Mean cross-validation score:", score.mean())

**Cross-validation scores: [0.98409091 0.97727273 0.96818182 0.97727273 0.97727273]**

**Mean cross-validation score: 0.9768181818181819**

## Accuracy Comparison

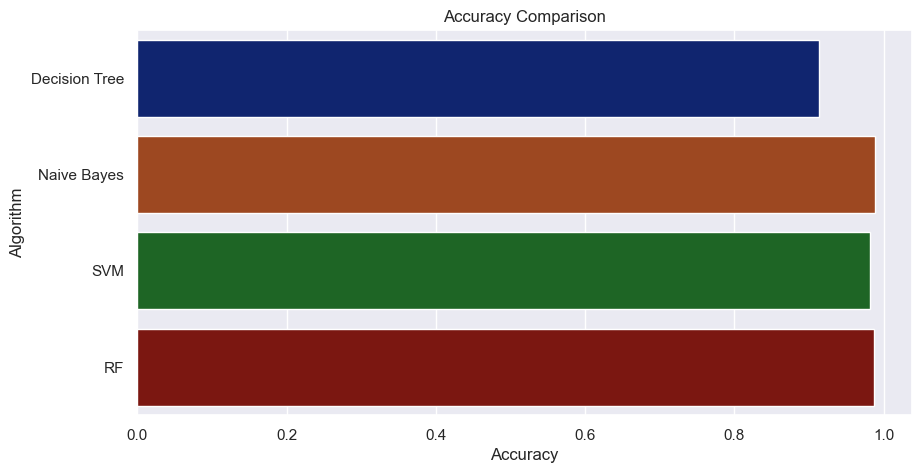
plt.figure(figsize=[10,5],dpi = 100)

plt.title('Accuracy Comparison')

plt.xlabel('Accuracy')

plt.ylabel('Algorithm')

sns.barplot(x = acc,y = model,palette='dark')



accuracy\_models = dict(zip(model, acc))

for k, v in accuracy\_models.items():

    print (k, '-->', v)

**SVM --> 0.9818181818181818**

## Making a prediction

data = np.array([[104,18, 30, 23.603016, 60.3, 6.7, 140.91]])

prediction = RF.predict(data)

print(prediction)

**['coffee']**

data = np.array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])

prediction = RF.predict(data)

print(prediction)

**['coffee']**

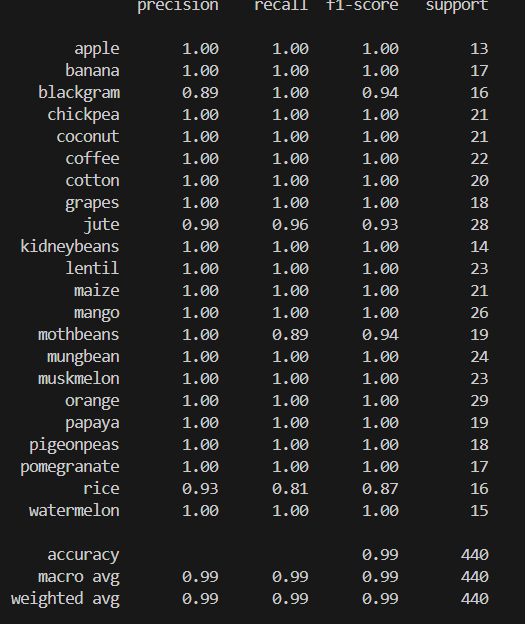
data = np.array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])

prediction = RF.predict(data)

print(prediction)

**['coffee']**

**f:/CROP\_RECOMMENDATION/CropRecommend.py**

**RF's Accuracy is:**

#### **Data Set**

Crop\_recommendation.csv

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **N** | **P** | **K** | **temperature** | **humidity** | **ph** | **rainfall** | **label** |
| 95 | 52 | 35 | 20.87974 | 82.00274 | 6.502985 | 202.9355 | rice |
| 82 | 56 | 36 | 21.77046 | 80.31964 | 7.038096 | 226.6555 | rice |
| 62 | 60 | 45 | 23.00446 | 82.32076 | 7.840207 | 263.9642 | rice |
| 61 | 60 | 35 | 26.4911 | 80.15836 | 6.980401 | 242.864 | rice |
| 87 | 59 | 43 | 20.13017 | 81.60487 | 7.628473 | 262.7173 | rice |
| 78 | 36 | 45 | 23.05805 | 83.37012 | 7.073454 | 251.055 | rice |
| 100 | 44 | 43 | 22.70884 | 82.63941 | 5.700806 | 271.3249 | rice |
| 60 | 47 | 38 | 20.27774 | 82.89409 | 5.718627 | 241.9742 | rice |
| 71 | 38 | 36 | 24.51588 | 83.53522 | 6.685346 | 230.4462 | rice |
| 70 | 47 | 43 | 23.22397 | 83.03323 | 6.336254 | 221.2092 | rice |
| 61 | 44 | 40 | 26.52724 | 81.41754 | 5.386168 | 264.6149 | rice |
| 99 | 50 | 39 | 23.97898 | 81.45062 | 7.502834 | 250.0832 | rice |
| 96 | 48 | 44 | 26.8008 | 80.88685 | 5.108682 | 284.4365 | rice |
| 94 | 54 | 42 | 24.01498 | 82.05687 | 6.984354 | 185.2773 | rice |
| 69 | 57 | 36 | 25.66585 | 80.66385 | 6.94802 | 209.587 | rice |
| 73 | 37 | 41 | 24.28209 | 80.30026 | 7.042299 | 231.0863 | rice |
| 99 | 42 | 36 | 21.58712 | 82.78837 | 6.249051 | 276.6552 | rice |
| 60 | 51 | 38 | 23.79392 | 80.41818 | 6.97086 | 206.2612 | rice |
| 61 | 49 | 40 | 21.86525 | 80.1923 | 5.953933 | 224.555 | rice |
| 85 | 52 | 43 | 23.57944 | 83.5876 | 5.853932 | 291.2987 | rice |
| 73 | 39 | 36 | 21.32504 | 80.47476 | 6.442475 | 185.4975 | rice |
| 83 | 49 | 40 | 25.15746 | 83.11713 | 5.070176 | 231.3843 | rice |
| 86 | 46 | 36 | 21.94767 | 80.97384 | 6.012633 | 213.3561 | rice |
| 97 | 53 | 39 | 21.05254 | 82.6784 | 6.254028 | 233.1076 | rice |
| 81 | 47 | 40 | 23.48381 | 81.33265 | 7.375483 | 224.0581 | rice |
| 92 | 46 | 40 | 25.07564 | 80.52389 | 7.778915 | 257.0039 | rice |
| 64 | 36 | 44 | 26.35927 | 84.04404 | 6.2865 | 271.3586 | rice |
| 71 | 37 | 37 | 24.52923 | 80.54499 | 7.07096 | 260.2634 | rice |
| 70 | 47 | 39 | 20.77576 | 84.49774 | 6.244841 | 240.0811 | rice |
| 66 | 39 | 42 | 22.30157 | 80.64416 | 6.043305 | 197.9791 | rice |
| 82 | 59 | 40 | 21.44654 | 84.94376 | 5.824709 | 272.2017 | rice |
| 72 | 41 | 38 | 22.17932 | 80.33127 | 6.357389 | 200.0883 | rice |
| 66 | 35 | 40 | 24.52784 | 82.73686 | 6.364135 | 224.6757 | rice |
| 82 | 38 | 43 | 20.26708 | 81.63895 | 5.014507 | 270.4417 | rice |
| 61 | 39 | 42 | 25.73543 | 83.88266 | 6.149411 | 233.1321 | rice |
| 86 | 43 | 35 | 26.79534 | 82.14809 | 5.950661 | 193.3474 | rice |
| 64 | 40 | 39 | 26.75754 | 81.17734 | 5.96037 | 272.2999 | rice |
| 88 | 49 | 44 | 23.8633 | 83.15251 | 5.561399 | 285.2494 | rice |
| 62 | 50 | 44 | 21.01945 | 82.95222 | 7.416245 | 298.4018 | rice |
| 82 | 39 | 22 | 22.6136 | 63.69071 | 5.749914 | 87.75954 | maize |
| 71 | 43 | 25 | 26.10018 | 71.57477 | 6.931757 | 102.2662 | maize |
| 88 | 37 | 17 | 23.55882 | 71.59351 | 6.657965 | 66.71995 | maize |
| 79 | 42 | 19 | 19.97216 | 57.68273 | 6.596061 | 60.65171 | maize |
| 67 | 39 | 19 | 18.47891 | 62.69504 | 5.970458 | 65.43835 | maize |
| 66 | 38 | 25 | 21.77689 | 57.80841 | 6.158831 | 102.0862 | maize |
| 65 | 41 | 17 | 25.62172 | 66.50415 | 6.047907 | 105.4655 | maize |
| 84 | 35 | 17 | 25.19192 | 66.69029 | 5.913665 | 78.0664 | maize |
| 83 | 54 | 25 | 20.41683 | 62.55425 | 5.855442 | 65.27798 | maize |
| 79 | 51 | 20 | 24.92162 | 66.78627 | 5.750255 | 109.2162 | maize |
| 89 | 47 | 25 | 23.31689 | 73.45415 | 5.852607 | 94.29713 | maize |
| 67 | 56 | 24 | 24.84017 | 68.35846 | 6.472523 | 74.05475 | maize |
| 84 | 57 | 20 | 22.27527 | 58.84016 | 6.967058 | 63.87021 | maize |
| 83 | 43 | 16 | 18.87751 | 65.76816 | 6.082974 | 94.76189 | maize |
| 76 | 46 | 19 | 25.19009 | 60.20017 | 5.919046 | 72.12376 | maize |
| 65 | 58 | 25 | 18.25405 | 55.2822 | 6.204748 | 63.72358 | maize |
| 63 | 42 | 17 | 24.61291 | 70.41624 | 6.600827 | 104.1626 | maize |
| 82 | 52 | 18 | 25.14206 | 65.26185 | 6.021902 | 76.68456 | maize |
| 66 | 60 | 25 | 23.09348 | 60.11594 | 6.03355 | 65.49731 | maize |
| 97 | 38 | 25 | 18.05034 | 62.89367 | 6.288868 | 84.23613 | maize |
| 75 | 42 | 15 | 24.93216 | 73.80435 | 6.550564 | 79.74079 | maize |
| 62 | 43 | 21 | 18.1471 | 71.09445 | 5.573286 | 88.07754 | maize |
| 76 | 54 | 16 | 18.28362 | 66.65953 | 6.829199 | 80.97573 | maize |
| 89 | 44 | 22 | 18.83344 | 58.75082 | 5.716223 | 79.75329 | maize |
| 73 | 36 | 20 | 25.71896 | 67.22191 | 5.549902 | 74.51491 | maize |
| 88 | 42 | 18 | 25.33798 | 68.49836 | 6.586245 | 96.4638 | maize |
| 67 | 53 | 25 | 23.89115 | 57.48776 | 5.893093 | 102.8302 | maize |
| 65 | 39 | 24 | 21.53574 | 71.50906 | 5.918264 | 102.4853 | maize |
| 60 | 51 | 24 | 23.08975 | 63.1046 | 5.588651 | 70.43474 | maize |
| 76 | 60 | 19 | 25.61707 | 63.47118 | 6.576418 | 108.8304 | maize |
| 94 | 50 | 25 | 21.44527 | 63.16216 | 6.178056 | 65.88951 | maize |
| 96 | 52 | 24 | 18.51817 | 55.53128 | 6.641906 | 90.98805 | maize |
| 95 | 54 | 22 | 22.53511 | 67.99257 | 6.48904 | 64.40866 | maize |
| 34 | 69 | 76 | 17.02498 | 16.98861 | 7.485996 | 88.55123 | chickpea |
| 58 | 71 | 79 | 19.02061 | 17.13159 | 6.920251 | 79.92698 | chickpea |
| 60 | 79 | 75 | 17.88776 | 15.4059 | 5.996932 | 68.54933 | chickpea |
| 43 | 78 | 76 | 18.86806 | 15.65809 | 6.391174 | 88.51049 | chickpea |
| 39 | 80 | 84 | 18.36953 | 19.56381 | 7.152811 | 79.26358 | chickpea |
| 33 | 79 | 75 | 20.45079 | 15.40312 | 5.988993 | 92.68374 | chickpea |
| 57 | 60 | 77 | 20.65432 | 16.60821 | 6.231049 | 74.66311 | chickpea |
| 43 | 60 | 81 | 17.33487 | 18.74927 | 7.550808 | 82.61735 | chickpea |
| 47 | 60 | 82 | 18.17912 | 18.90427 | 7.010571 | 81.84998 | chickpea |
| 34 | 62 | 81 | 18.01272 | 18.30968 | 8.753795 | 81.98569 | chickpea |
| 60 | 65 | 83 | 20.99374 | 19.3347 | 8.718193 | 93.5528 | chickpea |
| 45 | 58 | 83 | 19.46234 | 15.22539 | 7.976608 | 74.58565 | chickpea |
| 33 | 77 | 80 | 19.81345 | 14.69765 | 6.5155 | 78.96515 | chickpea |
| 55 | 62 | 76 | 18.97425 | 19.51612 | 8.490127 | 80.71087 | chickpea |
| 30 | 73 | 77 | 18.19737 | 14.71071 | 6.576416 | 70.18185 | chickpea |
| 53 | 56 | 84 | 18.72963 | 19.18197 | 6.481783 | 71.5801 | chickpea |
| 47 | 80 | 84 | 20.99502 | 19.86013 | 7.966605 | 73.50734 | chickpea |
| 58 | 56 | 82 | 20.28156 | 16.39535 | 8.140825 | 82.5234 | chickpea |
| 46 | 60 | 81 | 17.48043 | 15.75594 | 7.228963 | 66.96981 | chickpea |
| 58 | 55 | 83 | 20.88819 | 14.32314 | 6.492546 | 90.46228 | chickpea |
| 49 | 75 | 82 | 17.08499 | 16.14566 | 7.5286 | 71.31007 | chickpea |
| 41 | 76 | 78 | 20.09341 | 15.1128 | 7.701446 | 85.74905 | chickpea |
| 58 | 67 | 81 | 17.57212 | 14.99927 | 8.519976 | 89.31051 | chickpea |
| 54 | 61 | 75 | 19.12065 | 18.43476 | 6.620901 | 85.5295 | chickpea |
| 57 | 75 | 82 | 18.57666 | 19.22008 | 8.104396 | 72.9494 | chickpea |
| 53 | 69 | 78 | 17.16606 | 14.42458 | 6.204091 | 72.32668 | chickpea |
| 25 | 57 | 81 | 18.2872 | 16.67922 | 6.051091 | 74.87446 | chickpea |
| 37 | 74 | 77 | 19.03025 | 18.66726 | 7.690962 | 94.70992 | chickpea |
| 40 | 65 | 81 | 17.14186 | 17.06624 | 7.829211 | 83.74607 | chickpea |
| 26 | 72 | 85 | 17.47809 | 17.93254 | 6.7616 | 78.9206 | chickpea |
| 35 | 79 | 84 | 20.56602 | 14.25804 | 6.654425 | 83.75937 | chickpea |
| 20 | 62 | 78 | 17.22385 | 15.82069 | 6.129534 | 76.57581 | chickpea |
| 28 | 75 | 76 | 19.69142 | 19.44225 | 8.829273 | 91.76072 | chickpea |
| 33 | 65 | 19 | 17.13693 | 20.59542 | 5.685972 | 128.2569 | kidneybeans |
| 18 | 67 | 24 | 19.63474 | 18.90706 | 5.759237 | 106.3598 | kidneybeans |
| 3 | 66 | 24 | 22.9135 | 21.33953 | 5.873172 | 109.2256 | kidneybeans |
| 11 | 70 | 15 | 16.4334 | 24.24046 | 5.926677 | 140.3718 | kidneybeans |
| 23 | 76 | 22 | 22.13975 | 23.02251 | 5.955617 | 76.64128 | kidneybeans |
| 18 | 57 | 21 | 17.84807 | 18.77622 | 5.949949 | 143.0984 | kidneybeans |
| 3 | 74 | 21 | 19.88394 | 20.31564 | 5.789214 | 60.91975 | kidneybeans |
| 24 | 57 | 21 | 19.3251 | 23.33348 | 5.581022 | 104.7784 | kidneybeans |
| 16 | 62 | 22 | 18.4167 | 23.4283 | 5.689858 | 132.9801 | kidneybeans |
| 22 | 64 | 17 | 21.81168 | 23.20591 | 5.794159 | 130.0608 | kidneybeans |
| 29 | 80 | 15 | 19.72703 | 18.28173 | 5.74819 | 143.7631 | kidneybeans |
| 7 | 66 | 23 | 17.15433 | 19.87071 | 5.566523 | 87.9967 | kidneybeans |
| 28 | 67 | 21 | 19.62208 | 18.67171 | 5.80942 | 144.1567 | kidneybeans |
| 11 | 77 | 25 | 21.63149 | 21.1792 | 5.887263 | 134.365 | kidneybeans |
| 29 | 56 | 25 | 16.06523 | 18.7248 | 5.998125 | 88.06639 | kidneybeans |
| 36 | 78 | 15 | 20.61162 | 24.36314 | 5.792745 | 69.63834 | kidneybeans |
| 33 | 65 | 22 | 21.42451 | 20.3966 | 5.91229 | 116.5207 | kidneybeans |
| 33 | 67 | 22 | 19.07096 | 21.21092 | 5.788387 | 86.21918 | kidneybeans |
| 6 | 66 | 25 | 21.18853 | 19.63439 | 5.728233 | 137.1949 | kidneybeans |
| 22 | 66 | 19 | 23.04291 | 22.42611 | 5.83394 | 108.3684 | kidneybeans |
| 2 | 70 | 25 | 20.65376 | 23.10539 | 5.967533 | 67.71769 | kidneybeans |
| 39 | 61 | 16 | 18.09551 | 18.29318 | 5.625096 | 144.7902 | kidneybeans |
| 13 | 62 | 18 | 18.23776 | 21.07643 | 5.515615 | 69.44952 | kidneybeans |
| 3 | 64 | 20 | 36.51268 | 57.92887 | 6.031608 | 122.654 | pigeonpeas |
| 10 | 69 | 22 | 36.89164 | 62.73178 | 5.269085 | 163.7267 | pigeonpeas |
| 33 | 76 | 21 | 29.23541 | 59.38968 | 5.985793 | 103.3302 | pigeonpeas |
| 15 | 61 | 25 | 27.33535 | 43.35796 | 6.091863 | 142.3304 | pigeonpeas |
| 30 | 60 | 20 | 21.06437 | 55.46986 | 5.624731 | 184.6227 | pigeonpeas |
| 38 | 80 | 25 | 30.33277 | 42.3525 | 6.446092 | 149.3 | pigeonpeas |
| 32 | 78 | 17 | 31.8013 | 45.03186 | 5.62349 | 147.0361 | pigeonpeas |
| 0 | 72 | 21 | 33.18184 | 38.23185 | 5.864623 | 198.8299 | pigeonpeas |
| 32 | 73 | 19 | 29.38539 | 63.47742 | 5.761703 | 90.05423 | pigeonpeas |
| 29 | 76 | 17 | 30.27375 | 67.38681 | 4.696519 | 127.7767 | pigeonpeas |
| 32 | 76 | 19 | 35.4579 | 68.75811 | 5.269504 | 108.6333 | pigeonpeas |
| 10 | 66 | 22 | 33.8002 | 40.03262 | 7.445445 | 176.6166 | pigeonpeas |
| 8 | 60 | 23 | 28.6918 | 49.47225 | 5.833032 | 96.36223 | pigeonpeas |
| 25 | 68 | 25 | 31.24022 | 56.67369 | 7.339321 | 122.0147 | pigeonpeas |
| 33 | 69 | 25 | 28.98039 | 57.23265 | 6.347929 | 120.7436 | pigeonpeas |
| 20 | 77 | 16 | 27.32199 | 34.13737 | 4.697751 | 96.51524 | pigeonpeas |
| 23 | 58 | 21 | 21.47608 | 38.80024 | 4.962661 | 180.3822 | pigeonpeas |
| 32 | 73 | 25 | 18.3191 | 34.69777 | 4.964888 | 107.4722 | pigeonpeas |
| 0 | 80 | 16 | 36.97794 | 37.73993 | 5.642813 | 161.4813 | pigeonpeas |
| 9 | 77 | 23 | 24.80468 | 40.12427 | 5.609396 | 121.5639 | pigeonpeas |
| 24 | 55 | 24 | 19.34794 | 55.96805 | 4.681576 | 194.5921 | pigeonpeas |
| 17 | 79 | 23 | 28.88302 | 50.12324 | 5.709512 | 179.2156 | pigeonpeas |
| 40 | 56 | 17 | 19.54314 | 47.19188 | 6.413544 | 192.4372 | pigeonpeas |
| 5 | 75 | 23 | 24.77419 | 50.54621 | 6.007508 | 114.2821 | pigeonpeas |
| 33 | 71 | 21 | 18.33125 | 38.40975 | 4.94637 | 139.6483 | pigeonpeas |
| 25 | 60 | 21 | 19.14729 | 45.37338 | 5.517208 | 132.7748 | pigeonpeas |
| 0 | 58 | 20 | 28.23416 | 49.44213 | 5.902103 | 186.5009 | pigeonpeas |
| 33 | 60 | 19 | 30.11812 | 34.13308 | 5.71989 | 157.0858 | pigeonpeas |
| 11 | 75 | 16 | 33.41269 | 35.4291 | 4.548202 | 139.6703 | pigeonpeas |
| 35 | 61 | 20 | 24.18553 | 46.68747 | 6.669529 | 177.3378 | pigeonpeas |
| 8 | 61 | 21 | 26.88631 | 41.69618 | 4.750929 | 94.46748 | pigeonpeas |
| 1 | 69 | 20 | 31.33171 | 57.97429 | 4.946264 | 161.782 | pigeonpeas |
| 35 | 79 | 16 | 33.30712 | 67.07781 | 5.266227 | 108.509 | pigeonpeas |
| 8 | 67 | 16 | 23.45379 | 46.48715 | 7.109598 | 150.8712 | pigeonpeas |
| 20 | 75 | 20 | 34.53824 | 39.04469 | 5.617008 | 168.5948 | pigeonpeas |
| 23 | 73 | 19 | 19.50112 | 34.51087 | 5.632353 | 197.3753 | pigeonpeas |
| 26 | 73 | 17 | 28.76795 | 37.57792 | 4.674942 | 91.72085 | pigeonpeas |
| 15 | 67 | 21 | 30.97759 | 32.24914 | 7.161798 | 180.7168 | pigeonpeas |
| 37 | 76 | 21 | 18.39759 | 36.82639 | 6.624966 | 93.12331 | pigeonpeas |
| 15 | 59 | 23 | 35.09357 | 30.98685 | 5.004075 | 116.9107 | pigeonpeas |
| 12 | 66 | 15 | 34.93174 | 30.40047 | 6.345806 | 159.265 | pigeonpeas |
| 35 | 57 | 15 | 29.50523 | 35.72032 | 6.216814 | 187.8962 | pigeonpeas |
| 6 | 63 | 18 | 34.51935 | 47.5298 | 5.921667 | 129.0065 | pigeonpeas |
| 3 | 69 | 22 | 28.43431 | 52.10011 | 6.012719 | 147.0415 | pigeonpeas |
| 34 | 72 | 15 | 18.41646 | 34.80541 | 4.684079 | 163.2747 | pigeonpeas |
| 28 | 56 | 19 | 31.07509 | 47.19848 | 7.07717 | 91.31256 | pigeonpeas |
| 24 | 72 | 21 | 20.89343 | 46.24857 | 6.208843 | 195.5698 | pigeonpeas |
| 35 | 63 | 18 | 30.90608 | 52.79913 | 7.051816 | 170.992 | pigeonpeas |
| 16 | 70 | 17 | 27.93483 | 66.45457 | 4.722222 | 145.3729 | pigeonpeas |
| 9 | 64 | 21 | 35.95177 | 36.52781 | 6.418063 | 136.0457 | pigeonpeas |
| 13 | 38 | 20 | 27.91095 | 64.70931 | 3.692864 | 32.67892 | mothbeans |
| 39 | 44 | 18 | 27.32221 | 51.27869 | 4.371746 | 36.50379 | mothbeans |
| 37 | 50 | 15 | 28.66024 | 59.31891 | 8.399136 | 36.9263 | mothbeans |
| 37 | 37 | 20 | 29.02955 | 61.09387 | 8.840656 | 72.98017 | mothbeans |
| 30 | 45 | 17 | 27.78032 | 54.6503 | 8.153023 | 32.05025 | mothbeans |
| 38 | 36 | 22 | 31.99929 | 54.10775 | 5.270749 | 71.62667 | mothbeans |
| 37 | 46 | 17 | 27.33581 | 55.27756 | 8.050304 | 73.44775 | mothbeans |
| 34 | 46 | 16 | 28.92953 | 53.57015 | 9.679241 | 66.35634 | mothbeans |
| 27 | 40 | 23 | 27.65472 | 58.59986 | 6.974978 | 36.94255 | mothbeans |
| 1 | 52 | 19 | 28.52397 | 55.77264 | 7.393899 | 61.32936 | mothbeans |
| 2 | 36 | 21 | 31.02216 | 45.89239 | 6.687275 | 53.56783 | mothbeans |
| 6 | 41 | 15 | 25.74095 | 45.38497 | 7.881186 | 67.43488 | mothbeans |
| 15 | 55 | 17 | 31.70447 | 56.8542 | 5.875334 | 44.94317 | mothbeans |
| 9 | 60 | 21 | 24.47876 | 58.51664 | 8.202706 | 34.96933 | mothbeans |
| 36 | 42 | 23 | 31.46511 | 51.79939 | 8.985348 | 74.44331 | mothbeans |
| 36 | 51 | 21 | 25.60973 | 50.73301 | 5.877075 | 53.3925 | mothbeans |
| 0 | 53 | 19 | 30.3026 | 47.18284 | 7.707595 | 68.0404 | mothbeans |
| 27 | 59 | 23 | 28.17489 | 43.66723 | 4.524172 | 45.78173 | mothbeans |
| 19 | 37 | 19 | 25.50635 | 44.83026 | 9.926212 | 74.32635 | mothbeans |
| 15 | 49 | 16 | 31.12897 | 43.58789 | 6.455593 | 32.76743 | mothbeans |
| 39 | 54 | 23 | 26.34043 | 55.5916 | 8.016211 | 35.10512 | mothbeans |
| 14 | 46 | 15 | 28.23495 | 61.56205 | 3.711059 | 72.66666 | mothbeans |
| 19 | 40 | 19 | 27.04453 | 49.3261 | 5.490911 | 48.25208 | mothbeans |
| 36 | 50 | 22 | 25.16125 | 55.25436 | 9.254089 | 40.89733 | mothbeans |
| 20 | 50 | 16 | 27.43329 | 87.80508 | 7.185301 | 54.73368 | mungbean |
| 31 | 41 | 24 | 28.33404 | 80.77276 | 7.034214 | 38.79764 | mungbean |
| 40 | 42 | 17 | 27.0147 | 84.34263 | 6.635969 | 55.29635 | mungbean |
| 14 | 48 | 19 | 28.17433 | 81.04555 | 6.828187 | 36.35721 | mungbean |
| 25 | 53 | 23 | 29.87888 | 87.32761 | 6.89078 | 44.75216 | mungbean |
| 31 | 43 | 17 | 29.89233 | 89.71503 | 7.165121 | 42.99499 | mungbean |
| 28 | 58 | 18 | 28.56212 | 83.24856 | 6.935804 | 56.48265 | mungbean |
| 0 | 51 | 15 | 27.53593 | 85.57019 | 7.196774 | 53.01899 | mungbean |
| 30 | 39 | 17 | 29.68362 | 87.93598 | 6.990095 | 41.8249 | mungbean |
| 25 | 44 | 16 | 28.14449 | 82.1193 | 7.064782 | 46.7569 | mungbean |
| 21 | 58 | 15 | 29.53038 | 86.73346 | 7.156563 | 59.87232 | mungbean |
| 26 | 47 | 17 | 27.88353 | 86.45148 | 6.364967 | 44.64407 | mungbean |
| 38 | 45 | 17 | 28.22471 | 82.35916 | 6.428054 | 44.01207 | mungbean |
| 35 | 47 | 18 | 29.00812 | 84.96089 | 6.664188 | 45.91011 | mungbean |
| 16 | 55 | 20 | 29.75539 | 86.45193 | 6.637677 | 37.54603 | mungbean |
| 18 | 53 | 20 | 29.78417 | 85.16907 | 6.793856 | 40.77873 | mungbean |
| 35 | 45 | 24 | 27.86401 | 80.45131 | 6.852885 | 42.83054 | mungbean |
| 10 | 42 | 22 | 27.11026 | 84.96772 | 7.121571 | 51.52617 | mungbean |
| 13 | 35 | 22 | 27.1103 | 83.64274 | 6.883308 | 49.11965 | mungbean |
| 24 | 42 | 15 | 28.95451 | 89.07866 | 6.421271 | 57.65901 | mungbean |
| 12 | 35 | 17 | 29.2178 | 87.93724 | 6.544502 | 43.13866 | mungbean |
| 26 | 60 | 17 | 28.74201 | 85.81676 | 6.452006 | 48.54599 | mungbean |
| 35 | 53 | 22 | 29.65021 | 80.29868 | 6.489259 | 56.76278 | mungbean |
| 22 | 56 | 16 | 27.23925 | 86.40424 | 6.713411 | 37.31237 | mungbean |
| 0 | 58 | 15 | 28.95172 | 81.67085 | 6.510841 | 56.51103 | mungbean |
| 24 | 49 | 17 | 28.18837 | 82.6063 | 6.28738 | 37.0111 | mungbean |
| 39 | 54 | 16 | 28.30041 | 86.20682 | 6.863086 | 50.47334 | mungbean |
| 35 | 46 | 18 | 27.89636 | 88.71782 | 6.784153 | 57.79863 | mungbean |
| 31 | 73 | 21 | 29.4844 | 63.19915 | 7.454532 | 71.89091 | blackgram |
| 55 | 72 | 16 | 26.73434 | 68.14 | 7.040056 | 67.15096 | blackgram |
| 28 | 75 | 22 | 26.27274 | 62.28815 | 7.418651 | 70.23208 | blackgram |
| 29 | 60 | 18 | 34.03679 | 67.21114 | 6.501869 | 73.23574 | blackgram |
| 32 | 66 | 24 | 28.03644 | 65.06602 | 6.814411 | 72.49508 | blackgram |
| 60 | 59 | 23 | 28.81461 | 65.33538 | 7.581443 | 62.26243 | blackgram |
| 57 | 76 | 17 | 34.03619 | 64.28791 | 7.741419 | 66.85511 | blackgram |
| 41 | 62 | 17 | 33.86429 | 61.57072 | 6.573532 | 68.022 | blackgram |
| 46 | 55 | 20 | 32.84213 | 68.68401 | 7.543804 | 73.67166 | blackgram |
| 34 | 74 | 25 | 27.10053 | 63.36086 | 6.540821 | 73.8495 | blackgram |
| 57 | 72 | 21 | 25.65843 | 61.18236 | 7.224059 | 69.28608 | blackgram |
| 43 | 76 | 15 | 32.34744 | 66.61453 | 7.551364 | 64.55882 | blackgram |
| 54 | 61 | 18 | 29.58949 | 68.32177 | 6.928899 | 67.53021 | blackgram |
| 51 | 67 | 19 | 30.91219 | 68.79427 | 7.747775 | 66.63831 | blackgram |
| 38 | 62 | 17 | 28.12788 | 64.20978 | 6.706506 | 70.86341 | blackgram |
| 49 | 79 | 23 | 30.08545 | 69.34812 | 6.668239 | 67.13674 | blackgram |
| 47 | 72 | 25 | 31.74379 | 62.51008 | 7.332375 | 68.97098 | blackgram |
| 29 | 58 | 18 | 27.81327 | 62.5046 | 7.596802 | 69.75556 | blackgram |
| 28 | 56 | 15 | 32.88734 | 64.59457 | 7.706509 | 71.50569 | blackgram |
| 37 | 72 | 19 | 29.36359 | 64.98743 | 7.366543 | 61.91209 | blackgram |
| 39 | 58 | 19 | 27.82593 | 67.58619 | 6.919244 | 74.0123 | blackgram |
| 39 | 80 | 19 | 28.49539 | 60.44848 | 7.187722 | 74.9156 | blackgram |
| 60 | 67 | 23 | 27.74275 | 68.53997 | 7.075886 | 71.78615 | blackgram |
| 34 | 72 | 23 | 30.41588 | 67.66324 | 6.744412 | 63.02473 | blackgram |
| 20 | 67 | 21 | 27.32542 | 69.09048 | 6.726469 | 61.19251 | blackgram |
| 24 | 73 | 18 | 30.28497 | 61.69295 | 6.628265 | 65.6286 | blackgram |
| 46 | 59 | 19 | 28.56841 | 61.53279 | 7.127064 | 63.49726 | blackgram |
| 49 | 67 | 19 | 25.36586 | 66.63797 | 7.538631 | 65.81656 | blackgram |
| 58 | 60 | 19 | 32.74774 | 67.77955 | 7.453975 | 63.37784 | blackgram |
| 33 | 69 | 17 | 28.05154 | 63.49802 | 7.60411 | 43.35795 | lentil |
| 29 | 63 | 20 | 19.44084 | 63.27771 | 7.728832 | 46.8313 | lentil |
| 35 | 59 | 22 | 29.84823 | 60.63873 | 7.491217 | 46.80453 | lentil |
| 25 | 59 | 24 | 21.36384 | 69.92376 | 6.633865 | 46.63529 | lentil |
| 5 | 63 | 23 | 26.28664 | 68.51967 | 7.324863 | 46.13833 | lentil |
| 34 | 70 | 24 | 22.175 | 62.13874 | 6.410441 | 53.46623 | lentil |
| 12 | 74 | 24 | 26.57598 | 60.97877 | 7.83672 | 50.89111 | lentil |
| 24 | 57 | 17 | 26.58973 | 66.14008 | 6.139216 | 50.90994 | lentil |
| 28 | 61 | 19 | 19.13458 | 62.57527 | 6.590571 | 36.46947 | lentil |
| 26 | 76 | 15 | 28.75273 | 69.1564 | 7.28605 | 35.15426 | lentil |
| 17 | 68 | 20 | 25.78746 | 60.28163 | 6.058306 | 49.14337 | lentil |
| 33 | 69 | 23 | 23.89272 | 61.78779 | 6.658605 | 52.5573 | lentil |
| 9 | 62 | 16 | 28.67409 | 63.18833 | 7.299361 | 42.96019 | lentil |
| 27 | 61 | 21 | 28.42063 | 61.77336 | 7.815211 | 49.02367 | lentil |
| 7 | 56 | 21 | 21.35499 | 62.60136 | 5.925392 | 41.7822 | lentil |
| 18 | 62 | 25 | 21.12696 | 63.18739 | 6.403684 | 38.71834 | lentil |
| 25 | 66 | 15 | 24.02038 | 61.62313 | 7.397546 | 49.78103 | lentil |
| 6 | 73 | 15 | 25.40474 | 65.85675 | 7.722336 | 51.92057 | lentil |
| 22 | 78 | 20 | 29.03018 | 64.49167 | 7.475927 | 54.93938 | lentil |
| 20 | 66 | 24 | 20.21368 | 68.65258 | 6.88713 | 50.89733 | lentil |
| 12 | 79 | 22 | 29.19586 | 68.01966 | 7.441977 | 44.93262 | lentil |
| 33 | 60 | 25 | 18.29784 | 69.68976 | 7.62991 | 49.39111 | lentil |
| 13 | 68 | 24 | 27.41435 | 63.41786 | 7.336117 | 44.43178 | lentil |
| 24 | 76 | 18 | 24.84064 | 60.09117 | 6.750205 | 48.7779 | lentil |
| 7 | 68 | 19 | 29.94414 | 67.31323 | 7.52178 | 40.37114 | lentil |
| 36 | 66 | 22 | 25.8799 | 67.55109 | 6.347379 | 47.89645 | lentil |
| 19 | 67 | 17 | 20.04677 | 65.84395 | 7.135252 | 46.05333 | lentil |
| 1 | 71 | 17 | 22.99452 | 66.70897 | 7.670178 | 54.49044 | lentil |
| 13 | 72 | 19 | 25.13164 | 66.92642 | 7.399749 | 49.04016 | lentil |
| 19 | 69 | 24 | 28.49584 | 62.44616 | 7.841496 | 53.14531 | lentil |
| 36 | 76 | 16 | 18.28766 | 69.48515 | 6.254217 | 48.60449 | lentil |
| 6 | 60 | 15 | 24.38042 | 61.18458 | 6.868882 | 53.13947 | lentil |
| 30 | 65 | 15 | 21.31852 | 66.43935 | 7.320515 | 45.42617 | lentil |
| 10 | 65 | 25 | 18.54142 | 62.70638 | 6.296977 | 44.0782 | lentil |
| 37 | 75 | 23 | 25.28711 | 60.85994 | 7.241152 | 49.3737 | lentil |
| 36 | 63 | 18 | 25.4346 | 69.12613 | 7.685959 | 41.02683 | lentil |
| 26 | 62 | 15 | 28.83601 | 69.76113 | 6.89076 | 44.08563 | lentil |
| 32 | 66 | 22 | 27.3766 | 63.93928 | 6.155916 | 49.47372 | lentil |
| 21 | 70 | 22 | 28.31887 | 60.19461 | 6.167855 | 45.36521 | lentil |
| 34 | 62 | 15 | 27.48186 | 62.04815 | 6.86164 | 37.81124 | lentil |
| 28 | 78 | 19 | 18.28072 | 68.10365 | 6.978362 | 48.80253 | lentil |
| 23 | 57 | 23 | 27.61205 | 69.29786 | 7.04316 | 42.72374 | lentil |
| 31 | 76 | 24 | 23.43975 | 63.22012 | 5.942392 | 45.40277 | lentil |
| 17 | 66 | 18 | 20.95628 | 63.68129 | 7.239455 | 52.39881 | lentil |
| 34 | 68 | 19 | 23.79372 | 68.03209 | 6.516318 | 49.73922 | lentil |
| 9 | 68 | 22 | 22.63714 | 65.44545 | 6.233269 | 38.30411 | lentil |
| 9 | 73 | 21 | 21.53578 | 65.47228 | 7.505284 | 35.75108 | lentil |
| 26 | 76 | 16 | 29.87855 | 65.73085 | 6.950301 | 44.95655 | lentil |
| 34 | 61 | 23 | 25.26533 | 67.10005 | 6.958055 | 48.33941 | lentil |
| 36 | 80 | 23 | 25.17885 | 68.93307 | 6.548035 | 35.03485 | lentil |
| 27 | 65 | 20 | 24.12193 | 61.09534 | 6.461619 | 44.23629 | lentil |
| 12 | 10 | 40 | 24.55982 | 91.63536 | 5.922936 | 111.9685 | pomegranate |
| 15 | 17 | 35 | 19.6569 | 89.93701 | 5.93765 | 108.0459 | pomegranate |
| 38 | 18 | 42 | 18.7836 | 87.40248 | 6.804781 | 102.5185 | pomegranate |
| 22 | 11 | 37 | 24.14696 | 94.51107 | 6.424671 | 110.2317 | pomegranate |
| 32 | 24 | 35 | 22.44581 | 89.90147 | 6.738016 | 109.3906 | pomegranate |
| 36 | 7 | 45 | 24.96273 | 92.40501 | 6.497367 | 109.4169 | pomegranate |
| 25 | 22 | 35 | 22.55261 | 89.32595 | 6.327674 | 104.8956 | pomegranate |
| 8 | 15 | 36 | 22.77036 | 91.45499 | 6.361374 | 106.9659 | pomegranate |
| 28 | 12 | 38 | 19.2009 | 94.2766 | 6.923509 | 108.0424 | pomegranate |
| 34 | 16 | 39 | 23.12808 | 92.68328 | 6.630646 | 109.393 | pomegranate |
| 10 | 27 | 45 | 24.92639 | 85.19098 | 5.832526 | 104.7694 | pomegranate |
| 30 | 22 | 43 | 24.77464 | 85.63609 | 6.738994 | 105.7596 | pomegranate |
| 6 | 25 | 41 | 19.86712 | 86.3559 | 5.782436 | 108.3169 | pomegranate |
| 39 | 15 | 44 | 24.26601 | 93.79741 | 6.537043 | 104.5375 | pomegranate |
| 32 | 21 | 45 | 23.626 | 89.73267 | 6.145104 | 107.6837 | pomegranate |
| 35 | 28 | 42 | 19.67832 | 89.08936 | 6.890784 | 108.5469 | pomegranate |
| 20 | 6 | 45 | 22.36509 | 92.30882 | 7.175344 | 104.8216 | pomegranate |
| 3 | 30 | 43 | 21.57937 | 94.88268 | 5.938529 | 102.8593 | pomegranate |
| 18 | 26 | 38 | 18.26233 | 88.16779 | 5.70938 | 108.0757 | pomegranate |
| 12 | 30 | 36 | 23.71028 | 89.61794 | 6.1844 | 105.65 | pomegranate |
| 1 | 26 | 42 | 22.4872 | 89.92249 | 6.55351 | 111.6632 | pomegranate |
| 35 | 10 | 45 | 21.66025 | 94.79397 | 5.885638 | 112.435 | pomegranate |
| 37 | 8 | 45 | 20.13037 | 89.31505 | 6.143875 | 107.3417 | pomegranate |
| 28 | 8 | 40 | 18.41164 | 91.11927 | 6.101199 | 105.1835 | pomegranate |
| 19 | 29 | 38 | 19.68291 | 89.75273 | 6.594037 | 111.2819 | pomegranate |
| 27 | 11 | 43 | 23.20243 | 91.19443 | 6.859841 | 109.0946 | pomegranate |
| 15 | 23 | 39 | 18.92157 | 87.3129 | 6.568934 | 102.8013 | pomegranate |
| 24 | 6 | 35 | 22.10621 | 91.3404 | 6.769856 | 106.8705 | pomegranate |
| 12 | 29 | 36 | 18.47412 | 89.6892 | 7.130838 | 108.4759 | pomegranate |
| 39 | 30 | 38 | 19.81069 | 88.92944 | 5.740338 | 102.8601 | pomegranate |
| 9 | 9 | 39 | 24.48808 | 90.83687 | 5.843005 | 103.1969 | pomegranate |
| 30 | 19 | 43 | 18.75928 | 89.93458 | 6.648687 | 111.0197 | pomegranate |
| 34 | 12 | 43 | 19.54128 | 90.29752 | 6.902751 | 104.374 | pomegranate |
| 38 | 22 | 43 | 18.91251 | 87.74939 | 6.608024 | 111.2801 | pomegranate |
| 13 | 21 | 37 | 19.91331 | 94.95031 | 6.828522 | 104.0277 | pomegranate |
| 16 | 13 | 35 | 19.72621 | 89.64934 | 6.910375 | 108.2287 | pomegranate |
| 16 | 7 | 40 | 23.83186 | 87.84035 | 6.306606 | 111.2233 | pomegranate |
| 94 | 86 | 48 | 29.36792 | 76.249 | 6.149934 | 92.82841 | banana |
| 88 | 76 | 45 | 27.33369 | 83.67675 | 5.849076 | 101.0495 | banana |
| 117 | 71 | 46 | 27.40054 | 82.96221 | 6.2768 | 104.9378 | banana |
| 94 | 91 | 51 | 29.31591 | 80.11586 | 5.926825 | 90.10978 | banana |
| 92 | 86 | 50 | 26.05433 | 79.39655 | 5.519088 | 113.2297 | banana |
| 117 | 80 | 54 | 25.86632 | 84.42379 | 6.079179 | 114.5358 | banana |
| 101 | 91 | 55 | 27.00932 | 80.18547 | 6.134656 | 97.32532 | banana |
| 98 | 81 | 46 | 29.55055 | 78.06763 | 5.808498 | 99.34482 | banana |
| 97 | 87 | 55 | 26.28846 | 83.39004 | 5.891458 | 113.873 | banana |
| 87 | 84 | 46 | 29.16227 | 76.16152 | 5.816622 | 100.0076 | banana |
| 86 | 80 | 53 | 28.65004 | 82.68753 | 5.843163 | 98.75084 | banana |
| 109 | 92 | 55 | 29.07311 | 76.50045 | 6.376757 | 100.1693 | banana |
| 98 | 90 | 48 | 28.08166 | 75.2643 | 5.623616 | 118.2762 | banana |
| 95 | 87 | 45 | 27.19946 | 78.80861 | 5.915055 | 99.72431 | banana |
| 86 | 78 | 45 | 28.05484 | 78.04603 | 6.458715 | 108.3957 | banana |
| 95 | 75 | 46 | 25.14748 | 83.34688 | 5.565029 | 98.66679 | banana |
| 83 | 86 | 45 | 25.94019 | 78.34221 | 6.211833 | 119.848 | banana |
| 117 | 84 | 48 | 27.66753 | 79.68543 | 6.490074 | 108.6646 | banana |
| 120 | 83 | 49 | 25.56703 | 75.94068 | 5.590236 | 102.7868 | banana |
| 105 | 82 | 54 | 28.69562 | 82.54196 | 6.225225 | 116.1617 | banana |
| 108 | 88 | 54 | 26.33545 | 76.8532 | 6.190757 | 118.6858 | banana |
| 81 | 88 | 55 | 25.9373 | 78.89864 | 5.915569 | 98.21748 | banana |
| 38 | 33 | 31 | 29.7377 | 47.54885 | 5.954627 | 90.09587 | mango |
| 39 | 21 | 27 | 33.55696 | 53.7298 | 4.757115 | 98.67528 | mango |
| 24 | 17 | 26 | 27.00316 | 47.67525 | 5.699587 | 95.85118 | mango |
| 35 | 32 | 35 | 33.5615 | 45.53557 | 5.977414 | 95.70526 | mango |
| 22 | 31 | 33 | 35.89856 | 54.25964 | 6.430139 | 92.19722 | mango |
| 31 | 21 | 34 | 34.1772 | 50.62162 | 6.113935 | 98.00688 | mango |
| 39 | 23 | 30 | 30.01593 | 53.19212 | 5.074273 | 97.72843 | mango |
| 18 | 24 | 30 | 31.74592 | 45.16128 | 5.667508 | 93.75442 | mango |
| 35 | 40 | 33 | 35.9901 | 52.2278 | 5.978634 | 95.37135 | mango |
| 9 | 27 | 27 | 31.86641 | 52.19332 | 5.064613 | 98.46769 | mango |
| 32 | 30 | 34 | 27.75519 | 52.34606 | 4.772386 | 94.11213 | mango |
| 17 | 16 | 26 | 34.72413 | 51.42718 | 5.161149 | 97.31258 | mango |
| 27 | 19 | 32 | 27.53908 | 53.6355 | 6.797779 | 99.35408 | mango |
| 21 | 39 | 31 | 27.69638 | 48.56225 | 6.394743 | 89.85646 | mango |
| 28 | 29 | 32 | 27.25373 | 52.6632 | 5.566704 | 91.87312 | mango |
| 14 | 31 | 25 | 30.33724 | 48.88705 | 5.75505 | 94.42851 | mango |
| 32 | 22 | 29 | 31.85745 | 45.53106 | 5.417341 | 91.55846 | mango |
| 12 | 20 | 34 | 35.39986 | 49.45963 | 6.166174 | 97.41054 | mango |
| 21 | 15 | 28 | 29.80747 | 52.13798 | 5.191265 | 95.74606 | mango |
| 25 | 28 | 30 | 34.16439 | 54.16482 | 4.95474 | 98.33351 | mango |
| 20 | 19 | 30 | 28.9327 | 47.94054 | 5.664587 | 99.98342 | mango |
| 23 | 26 | 34 | 27.98393 | 53.33019 | 5.548585 | 99.61466 | mango |
| 3 | 37 | 31 | 31.20478 | 54.49961 | 6.804437 | 94.62955 | mango |
| 15 | 32 | 26 | 32.1341 | 50.52559 | 6.09787 | 98.63334 | mango |
| 36 | 15 | 26 | 28.91862 | 48.13975 | 5.075505 | 97.01332 | mango |
| 22 | 38 | 25 | 31.09779 | 47.41197 | 4.546466 | 90.28624 | mango |
| 13 | 19 | 29 | 34.73824 | 49.08864 | 5.855119 | 90.65022 | mango |
| 18 | 24 | 32 | 29.9808 | 49.48613 | 6.442393 | 91.82272 | mango |
| 3 | 28 | 29 | 33.80399 | 46.12866 | 4.507524 | 90.82549 | mango |
| 21 | 18 | 29 | 30.07203 | 50.96041 | 6.107296 | 92.0961 | mango |
| 7 | 26 | 34 | 27.92063 | 51.77966 | 6.475449 | 100.2586 | mango |
| 33 | 29 | 27 | 31.40949 | 49.21729 | 6.83298 | 92.99739 | mango |
| 3 | 38 | 25 | 35.78777 | 51.9419 | 5.395276 | 100.2161 | mango |
| 25 | 26 | 27 | 33.3614 | 45.02236 | 6.135269 | 98.81597 | mango |
| 0 | 24 | 25 | 35.96055 | 48.69678 | 4.555689 | 98.00644 | mango |
| 11 | 17 | 35 | 28.33333 | 51.39587 | 6.434198 | 91.67242 | mango |
| 2 | 34 | 35 | 32.27652 | 50.19369 | 5.316876 | 95.99487 | mango |
| 35 | 25 | 26 | 31.9949 | 50.84881 | 5.279389 | 97.38741 | mango |
| 32 | 26 | 29 | 27.58259 | 48.56916 | 6.720042 | 95.84456 | mango |
| 32 | 19 | 25 | 29.38472 | 45.88745 | 5.727423 | 100.8125 | mango |
| 18 | 26 | 35 | 32.38698 | 53.23282 | 4.691396 | 90.21633 | mango |
| 0 | 35 | 30 | 30.91471 | 49.92964 | 6.810186 | 90.14048 | mango |
| 14 | 24 | 28 | 35.37776 | 45.5811 | 6.454045 | 97.41586 | mango |
| 35 | 28 | 26 | 32.32362 | 52.58968 | 5.842764 | 93.36719 | mango |
| 30 | 27 | 26 | 28.22373 | 47.40519 | 5.024125 | 97.76832 | mango |
| 23 | 40 | 35 | 27.27433 | 47.16808 | 6.422711 | 95.25799 | mango |
| 30 | 29 | 26 | 27.10711 | 50.70881 | 4.94295 | 92.37239 | mango |
| 19 | 23 | 30 | 34.89227 | 48.75613 | 6.414527 | 91.63075 | mango |
| 13 | 27 | 29 | 33.74627 | 48.50388 | 6.777788 | 92.26439 | mango |
| 11 | 18 | 28 | 27.35111 | 54.43945 | 6.441328 | 96.27793 | mango |
| 3 | 145 | 203 | 22.75089 | 90.69489 | 5.521467 | 110.4318 | apple |
| 5 | 135 | 199 | 23.8494 | 94.34815 | 6.133221 | 114.0512 | apple |
| 19 | 124 | 202 | 22.60801 | 94.58901 | 6.22629 | 116.0397 | apple |
| 31 | 144 | 196 | 21.18667 | 91.13436 | 6.321152 | 122.2333 | apple |
| 37 | 131 | 196 | 23.41045 | 91.69913 | 5.587906 | 116.0778 | apple |
| 37 | 126 | 197 | 22.86007 | 93.1286 | 5.824152 | 117.7297 | apple |
| 5 | 135 | 196 | 22.48403 | 93.40819 | 5.77218 | 105.5474 | apple |
| 37 | 123 | 197 | 22.02775 | 92.96129 | 5.790993 | 121.1349 | apple |
| 37 | 138 | 200 | 21.91191 | 91.68748 | 6.499227 | 117.0761 | apple |
| 11 | 134 | 204 | 23.71059 | 93.27392 | 5.658474 | 112.6677 | apple |
| 26 | 127 | 204 | 21.37785 | 92.72044 | 5.573241 | 106.1417 | apple |
| 38 | 132 | 197 | 22.84853 | 94.3213 | 6.079497 | 123.5978 | apple |
| 22 | 138 | 200 | 23.10943 | 92.79631 | 6.38318 | 108.1838 | apple |
| 9 | 145 | 204 | 23.25231 | 94.54128 | 5.867421 | 105.3558 | apple |
| 29 | 136 | 202 | 23.67288 | 90.49356 | 5.708419 | 104.2298 | apple |
| 34 | 136 | 204 | 23.76882 | 90.5981 | 5.798351 | 102.2649 | apple |
| 1 | 138 | 195 | 23.34386 | 91.47685 | 6.281884 | 104.4268 | apple |
| 38 | 126 | 198 | 22.63946 | 90.18452 | 5.697946 | 108.3406 | apple |
| 17 | 129 | 200 | 22.45697 | 94.76285 | 5.605934 | 114.8408 | apple |
| 12 | 136 | 198 | 22.96388 | 93.58066 | 5.856481 | 104.6473 | apple |
| 17 | 130 | 196 | 21.07273 | 93.56586 | 6.041054 | 107.8737 | apple |
| 28 | 125 | 200 | 22.44075 | 92.70785 | 5.685062 | 121.4977 | apple |
| 35 | 127 | 203 | 22.71271 | 90.45262 | 5.669489 | 109.8853 | apple |
| 23 | 132 | 202 | 21.70417 | 93.44006 | 5.751707 | 115.1781 | apple |
| 2 | 124 | 197 | 22.43325 | 92.48668 | 5.800449 | 119.1025 | apple |
| 23 | 135 | 203 | 21.25941 | 92.84416 | 5.821348 | 109.0658 | apple |
| 16 | 129 | 200 | 22.92157 | 94.89613 | 6.280223 | 105.6942 | apple |
| 19 | 124 | 195 | 22.81213 | 91.51862 | 6.027314 | 107.8552 | apple |
| 15 | 141 | 203 | 21.12152 | 90.68788 | 5.636687 | 102.8017 | apple |
| 22 | 143 | 199 | 21.11479 | 90.31529 | 5.559364 | 104.5087 | apple |
| 7 | 140 | 196 | 23.59997 | 90.97598 | 5.596449 | 107.1728 | apple |
| 7 | 125 | 201 | 23.12653 | 94.71203 | 5.893493 | 108.6212 | apple |
| 28 | 124 | 204 | 22.98459 | 93.32045 | 5.875719 | 122.1952 | apple |
| 1 | 134 | 198 | 22.12659 | 90.97818 | 6.386021 | 104.5412 | apple |
| 27 | 130 | 202 | 22.5378 | 91.48136 | 5.71082 | 101.8475 | apple |
| 39 | 142 | 198 | 22.49095 | 91.70293 | 5.795986 | 124.3915 | apple |
| 40 | 125 | 196 | 22.35548 | 94.47812 | 6.046674 | 116.7366 | apple |
| 27 | 143 | 197 | 22.20701 | 93.50574 | 6.443383 | 120.1594 | apple |
| 4 | 132 | 196 | 22.44517 | 94.73764 | 5.617227 | 107.1843 | apple |
| 21 | 128 | 197 | 22.76643 | 92.12439 | 6.442289 | 120.436 | apple |
| 28 | 135 | 196 | 22.19109 | 90.02575 | 6.162034 | 112.3127 | apple |
| 40 | 137 | 197 | 23.61192 | 91.70294 | 5.812782 | 123.5901 | apple |
| 15 | 144 | 205 | 23.86087 | 94.92048 | 5.765015 | 105.0241 | apple |
| 11 | 141 | 195 | 21.42177 | 92.62665 | 6.184923 | 102.8046 | apple |
| 22 | 123 | 201 | 22.36629 | 90.78572 | 5.739652 | 124.9832 | apple |
| 9 | 132 | 197 | 23.99686 | 91.61002 | 5.824779 | 117.6103 | apple |
| 20 | 137 | 195 | 21.8013 | 92.73447 | 5.554824 | 120.0587 | apple |
| 32 | 141 | 203 | 23.80436 | 92.80442 | 6.024249 | 100.6193 | apple |
| 35 | 124 | 200 | 23.71475 | 91.53331 | 5.631333 | 121.8962 | apple |
| 11 | 130 | 196 | 23.34467 | 90.37981 | 5.811975 | 112.8954 | apple |
| 28 | 128 | 201 | 22.31254 | 90.03577 | 5.730557 | 113.0688 | apple |
| 11 | 129 | 202 | 23.50201 | 92.21084 | 5.669991 | 107.9869 | apple |
| 11 | 135 | 197 | 23.4626 | 91.45665 | 5.682751 | 111.7763 | apple |
| 2 | 121 | 195 | 23.06204 | 92.39544 | 6.245859 | 114.7399 | apple |
| 39 | 134 | 201 | 22.47421 | 91.2276 | 6.01737 | 124.218 | apple |
| 35 | 126 | 200 | 22.6978 | 92.82223 | 5.534567 | 105.0508 | apple |
| 35 | 144 | 198 | 23.66682 | 93.90191 | 5.952368 | 105.4005 | apple |
| 20 | 145 | 197 | 22.5005 | 92.45878 | 6.126437 | 100.9344 | apple |
| 35 | 137 | 196 | 23.83054 | 90.84422 | 6.406819 | 109.5967 | apple |
| 11 | 130 | 198 | 23.64142 | 93.74461 | 6.155939 | 116.6912 | apple |
| 39 | 141 | 196 | 21.41364 | 92.99125 | 5.878569 | 118.3979 | apple |
| 37 | 144 | 195 | 22.85267 | 94.57646 | 5.935336 | 117.5314 | apple |
| 29 | 140 | 195 | 22.98208 | 93.84505 | 5.971332 | 109.5852 | apple |
| 14 | 139 | 201 | 21.1991 | 90.80819 | 5.671306 | 103.6839 | apple |
| 8 | 140 | 195 | 23.87192 | 90.49939 | 5.882156 | 103.0548 | apple |
| 7 | 125 | 197 | 21.03653 | 94.3392 | 6.085519 | 114.7413 | apple |
| 36 | 124 | 198 | 22.61712 | 93.51978 | 5.904026 | 116.9257 | apple |
| 8 | 121 | 204 | 21.45279 | 90.74532 | 6.110219 | 116.7037 | apple |
| 39 | 133 | 200 | 22.81228 | 92.12992 | 6.212303 | 109.3384 | apple |
| 15 | 127 | 196 | 21.98142 | 91.12719 | 6.142803 | 115.4789 | apple |
| 22 | 137 | 205 | 22.52709 | 92.5478 | 6.365973 | 115.383 | apple |
| 8 | 133 | 205 | 22.35629 | 91.9236 | 6.264203 | 107.7697 | apple |

#### **CHAPTER 6**

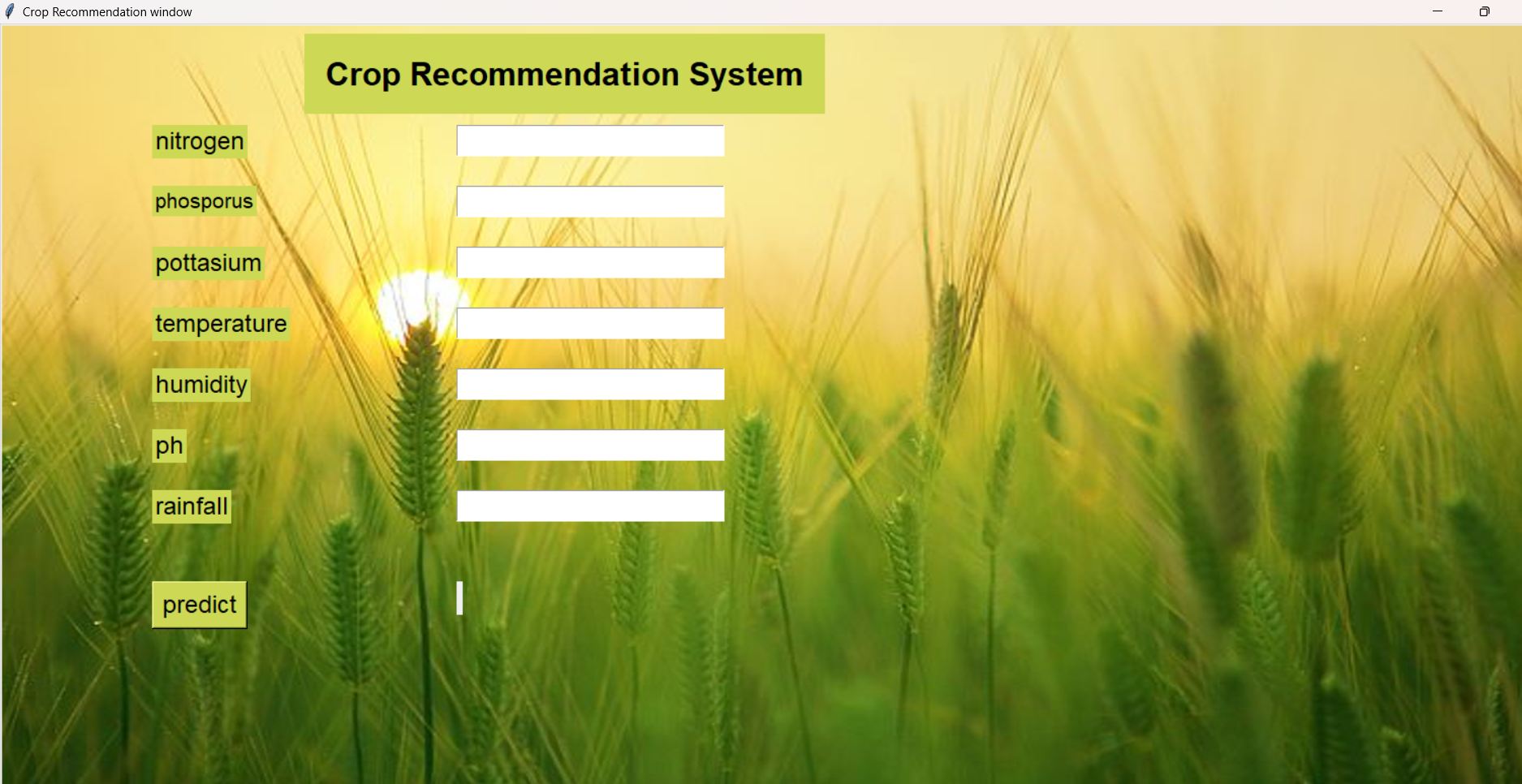
**RESULT ANALYSIS AND SCREENSHOTS**

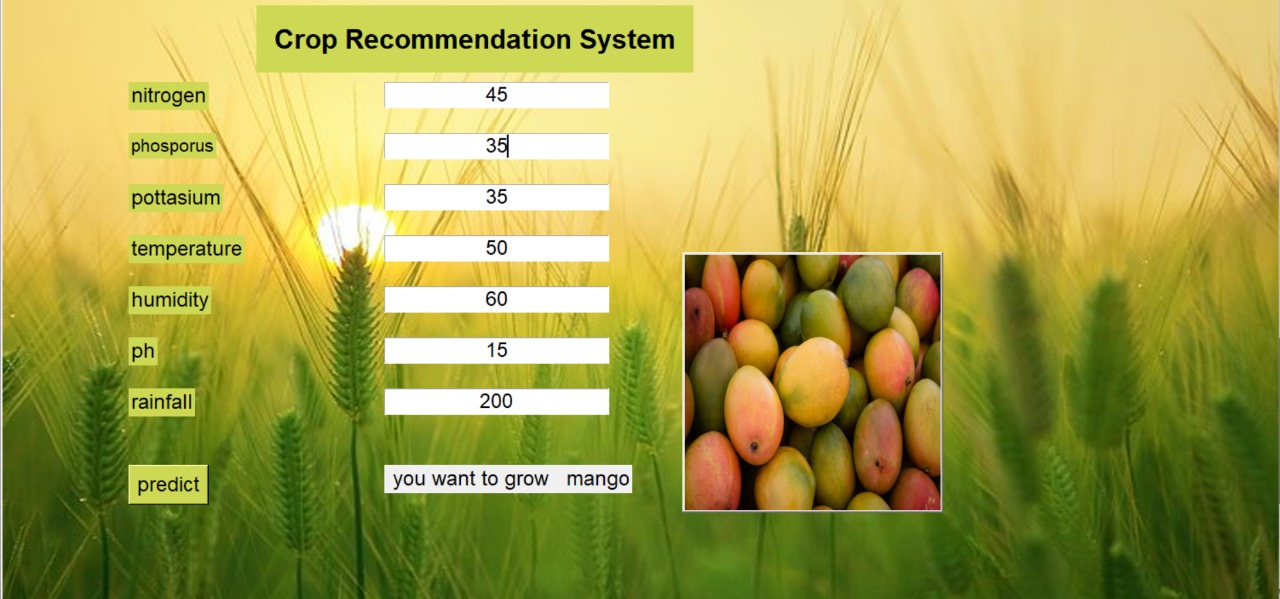
The project has successfully developed a robust and accurate crop prediction model that utilizes machine learning algorithms to analyze historical data on crop yields and environmental parameters. By leveraging this information, the model can make reliable predictions on the most suitable crop for cultivation based on the given input parameters. With an impressive accuracy rate of 99%, the model demonstrates its effectiveness in providing accurate and precise crop recommendations.

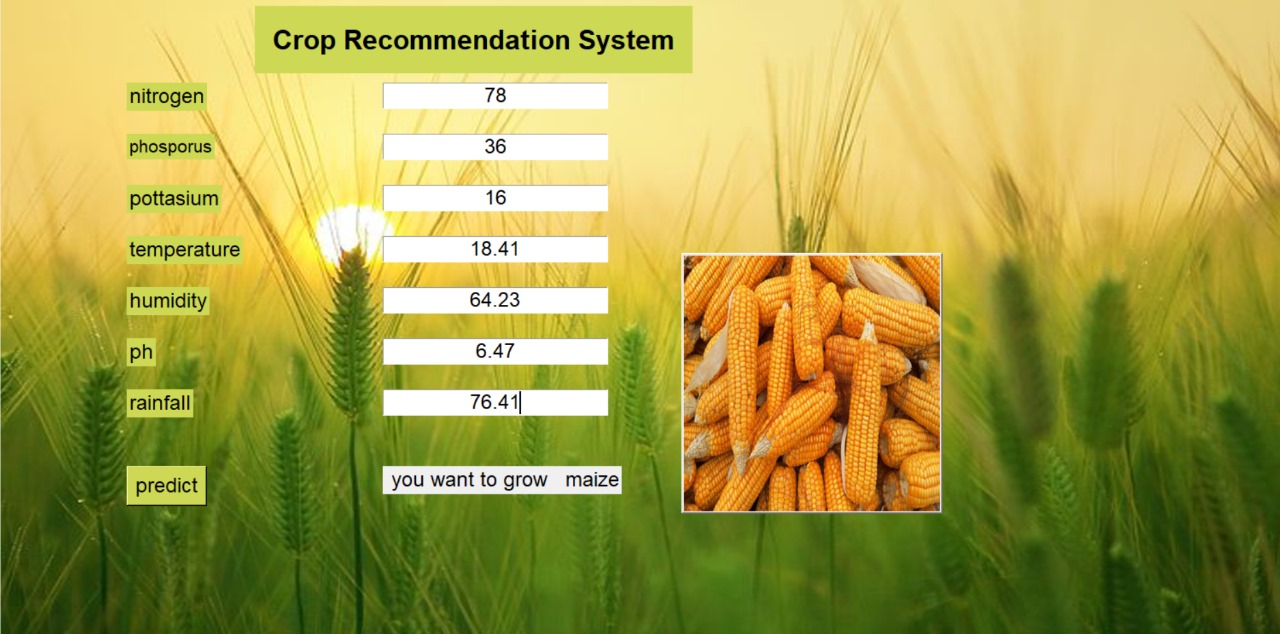
The integration of the crop prediction model into a user-friendly web application is a notable achievement. The web application provides an intuitive interface where users can easily input their specific environmental parameters and obtain the recommended crop for cultivation. This integration enhances the accessibility and usability of the model, making it readily available to farmers and agricultural stakeholders.

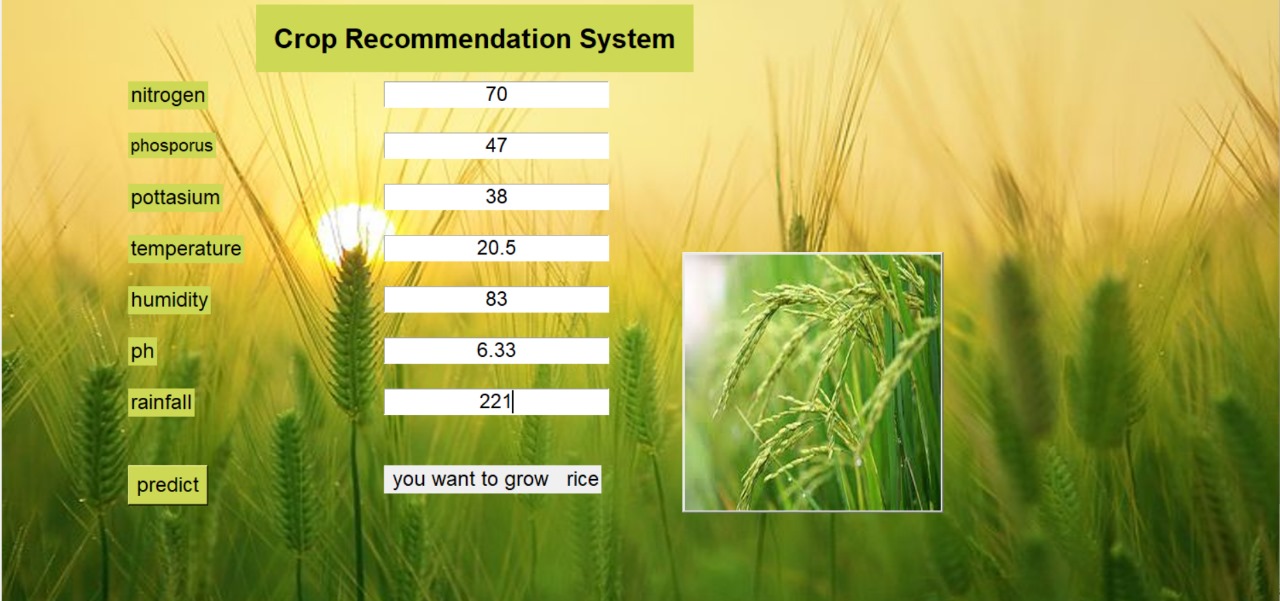
To validate the accuracy and effectiveness of the model, extensive testing and validation have been performed. The model was trained on a comprehensive dataset covering a wide range of crops. During the training phase, the model learned the complex relationships between the input parameters and the corresponding crop yields, achieving an impressive accuracy rate of 99%. This high accuracy demonstrates the model's ability to accurately predict suitable crops based on environmental factors, providing valuable insights for optimizing agricultural practices and maximizing profitability.The web application connected to the model offers additional features that enhance its usefulness. Along with crop recommendations and fertilizer requirements, the application displays current market prices for the recommended crops and provides an approximate yield estimation. These details empower farmers to make informed decisions about crop selection based on market conditions and expected yields.

In conclusion, the successful development and integration of the crop prediction model into a user-friendly web application represent a significant achievement. The model's exceptional accuracy rate of 99%, coupled with the additional information provided by the application, positions it as a highly valuable tool for farmers and agricultural stakeholders. The project's findings contribute to improved crop selection strategies, enhanced agricultural productivity, and increased profitability.









#### **CHAPTER 7**

#### **APPLICATIONS**

The crop prediction model can be applied in various agricultural contexts, offering valuable benefits and insights to different stakeholders in the agriculture sector. The model's ability to provide accurate crop predictions and yield estimations based on historical data and environmental parameters makes it a powerful tool for decision-making and planning. Some of the key applications of the crop prediction model are as follows:

1. **Farming Communities and Agricultural Organizations:**

Farming communities and agricultural organizations can benefit significantly from the crop prediction model. Small-scale and large-scale farmers alike can use the model to make informed decisions about which crops to cultivate in a given season. By inputting local weather data, soil information, and other environmental parameters, farmers can receive recommendations on the most suitable crops for their specific region and the upcoming season. This helps in maximizing crop yields and minimizing potential losses due to unfavorable conditions. Furthermore, agricultural organizations can use the model to provide personalized crop recommendations to their members, leading to improved agricultural productivity and profitability for the entire community.

1. **Government Agencies:**

Government agencies responsible for agricultural policy-making and resource allocation can leverage the crop prediction model to support their decisions. By analyzing historical crop data and environmental parameters at a larger scale, these agencies can identify trends and patterns in crop yields and optimize resource allocation for different regions. This aids in ensuring food security and sustainable agricultural practices across the country. Moreover, the model can assist in developing and implementing agricultural policies that are tailored to specific regions, leading to enhanced agricultural development and economic growth.

1. **Agricultural Research Institutions:**

Agricultural research institutions play a crucial role in studying crop yield patterns, climate change effects, and the impact of various environmental factors on agriculture. The crop prediction model can serve as a valuable tool for such institutions to conduct in-depth studies and research. By analyzing the model's predictions and comparing them with actual crop yields, researchers can gain insights into the effectiveness of different crop management practices, fertilizers, and irrigation methods. Additionally, the model can aid in identifying regions with specific agricultural challenges, which can further guide research efforts in developing innovative solutions for improving crop productivity and resilience.

In summary, the crop prediction model's versatility and accuracy make it applicable in diverse agricultural contexts. Whether it is aiding farmers in their daily decision-making or supporting government policies and agricultural research, the model proves to be a valuable asset for the agriculture sector. By promoting efficient crop selection, planning, and resource allocation, the model contributes to sustainable agriculture and the overall growth of the agricultural industry.

#### **CHAPTER 8**

#### **CONCLUSION**

The crop prediction model, powered by machine learning and data analysis, is a significant advancement in agriculture. It offers valuable insights into crop selection, yield estimation, and resource planning, enabling farmers to make informed decisions and increase

productivity. The integration of the model into a user-friendly web application enhances accessibility for farmers, even those with limited technical expertise. Government agencies and research institutions can also benefit from the model's predictions, using them to inform policy decisions, resource allocation, and in-depth studies on crop yield patterns and environmental factors. Future enhancements include integrating real-time weather data, expanding the dataset, and incorporating additional features like pest and disease prediction.

In conclusion, the crop prediction model has the potential to revolutionize the agricultural sector by providing accurate predictions and insights. It contributes to sustainable agriculture, increased productivity, and improved resource management. Continued advancements in technology will further refine and expand the model to meet the evolving needs of farmers, government agencies, and research institutions.

### **BIBLIOGRAPHY**

1. Atharva Ingle, "Crop Recommendation Dataset," Kaggle, [Online].

Available:https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-data set.

1. T. Gupta, S. Maggu, and B. Kapoor, "Crop Prediction using Machine Learning," IRE Journals, vol. 6, issue 9, pp. 279, Mar. 2023. ISSN: 2456-8880.
2. Prashant Kumar, Keshav Bhagat, Kusum Lata, Sushant Jhingran, "Crop recommendation using machine learning algorithms", *2023 International Conference on Disruptive Technologies (ICDT)*, pp.100-103, 2023.
3. Siva Ramakrishna Sani, Surya Venkata Sekhar Ummadi, SriRajarajeswari Thota,
4. Shashwat Raj, Siddhesh Patle, Subash Rajendran, "Predicting Crop Yield Using Decision Tree Regressor", *2022 International Conference on Knowledge Engineering and Communication Systems (ICKES)*, pp.1-5, 2022.

Nikitha Muthineni, Varun Sai Srinivas Swargam, Teja Sree Ravella, "Crop Recommendation System using Random Forest Algorithm in Machine Learning", 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pp.501-505, 2023.

1. Machine learning approach for Crop selection based on agro-climatic conditions