Crop Recommendation System

Group name: Harvestifier
Project team members:
Subhajit Mukherjee (Email Id: smukherjee081202@gmail.com)
Dhrupad Dutta (Email Id: dhrupaddutta20@gmail.com)

May 1, 2024

1 Abstract

The Crop Recommendation System is an intelligent application powered by machine learning algorithms designed to assist farmers in making informed decisions about crop selection based on environmental factors. By analyzing key parameters such as soil nutrient levels, temperature, humidity, pH, and rainfall, the system predicts the most suitable crop for cultivation on a given piece of land. This project aims to alleviate the challenges faced by farmers in selecting appropriate crops, ultimately increasing agricultural productivity and profitability. Through an intuitive user interface, farmers can input their land's characteristics, and the system provides real-time recommendations for optimal crop choices. By harnessing the power of data-driven insights, the Crop Recommendation System empowers farmers to make efficient and sustainable agricultural decisions, leading to improved yields and economic outcomes.

2 Introduction

2.1 What is this project:

The Crop Recommendation System is a pioneering application designed to assist farmers in making informed decisions about crop selection by harnessing the power of machine learning algorithms. It analyzes various environmental factors such as soil nutrient levels, temperature, humidity, pH, and rainfall to provide tailored recommendations for optimal crop choices.

2.2 Why this project:

With the increasing complexity of agricultural practices and the growing need for sustainable farming solutions, there is a pressing demand for advanced technologies to streamline crop selection processes. This project addresses the challenges faced by farmers in selecting suitable crops by offering a data-driven approach that optimizes productivity and resource utilization while minimizing environmental impact.

2.3 How this is solved:

The Crop Recommendation System employs sophisticated machine learning techniques to analyze large datasets of environmental parameters. By training predictive models on this data, the system can accurately predict the most suitable crops for cultivation on a given piece of land. Through an intuitive user interface, farmers can input their land's characteristics, and the system provides real-time recommendations based on the analysis of these factors. This approach revolutionizes traditional crop selection methods, empowering farmers with the tools and insights needed to achieve sustainable and profitable farming outcomes.

3 Literature review

3.1 FARMING ASSISTANCE FOR SOIL FERTILITY IMPROVEMENT AND CROP PREDICTION USING XGBOOST – MANGESH DESHMUKH, AMITKUMAR JAISWAR, OMKAR JOSHI, AND RAJASHREE SHEDGE – DEPARTMENT OF COMPUTER ENGINEERING, RAMRAO ADIK INSTITUTE OF TECHNOLOGY, NERUL, NAVI MUMBAI, INDIA

This proposed work is a recommendation system in which Machine Learning techniques are used to recommend best three crops based on soil and weather parameters. The top three crops are recommended because farmers may not have access to a particular crop if only one crop is recommended. Previous studies in this field have been done by using different Machine Learning algorithms such as Random Forest, KNN, Naïve Bayes, etc. In this proposed system XGBoost Machine Learning algorithm is used which gives better results than other algorithms. In addition, the system provides information about how to improve the soil for growing the desired crop and gives the weather forecast for next five days. Afinancial losses while also increasing crop productivity. This system has the accuracy 90%.

3.2 CROP RECOMMENDATION SYSTEM USING MACHINE LEARNING ALGORITHM – LAKSHMAN KUMAR SERU, SAI MAANAS GANDHAM, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF COMPUTING, SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

This Crop Recommendation System for agriculture is based on various input parameters. This proposes a hybrid model for recommending crops to south Indian states by considering various attributes such as soil type, Rainfall, Groundwater level, Temperature, Fertilizers, Pesticides and season. The recommender model is built as a hybrid model using the classifier machine learning algorithm. Based on the appropriate parameters, the system will recommend the crop. Technology based crop recommendation system for agriculture helps the farmers to increase the crop yield by recommending a suitable crop for their land with the help of geographic and the climatic parameters. This system has the accuracy 91% using Random Forest Classifier.

3.3 INTELLIGENT CROP RECOMMENDATION SYSTEM USING ML - AYUSH KUMAR, OMEN RAJENDRA POONIWALA, SWAPNEEL CHAKRABORTY, VISVESVARAYA TECHNOLOGICAL UNIVERSITY

In this project, they are building an intelligent system, which intends to assist the Indian farmers in making an informed decision about which crop to grow depending on the sowing season, his farm's geographical location and soil characteristics. Further the system will also provide the farmer, the yield prediction if he plants the recommended crop. This is an intelligent system that would consider environmental parameters (temperature, rainfall, geographical location in terms of state) and soil characteristics (pH value, soil type and nutrients concentration) before recommending the most suitable crop to the user.

3.4 AGRO CONSULTANT - INTELLIGENT CROP RECOMMENDATION SYSTEM USING MACHINE LEARNING ALGORITHMS - ZEEL DOSHI, SUBHASH NADKARNI, RASHI AGRAWAL, PROF. NEEPA SHAH

This paper, proposed and implemented an intelligent crop recommendation system, which can be easily used by farmers all over India. This system would assist the farmers in making an informed decision about which crop to grow depending on a variety of environmental and geographical factors. We have also implemented a secondary system, called Rainfall Predictor, which predicts the rainfall of the next 12 months.

3.5 CROP RECOMMENDATION SYSTEM FOR PRECISION AGRICULTURE - S.PUDUMALAR, E.RAMANUJAM, R.HARINE RAJASHREE, C.KAVYA, T.KIRUTHIKA, J.NISHA

This paper, proposes a recommendation system through an ensemble model with majority voting technique using Random tree, CHAID, K-Nearest Neighbor and Naive Bayes as learners to recommend a crop for the site specific parameters with high accuracy and efficiency.

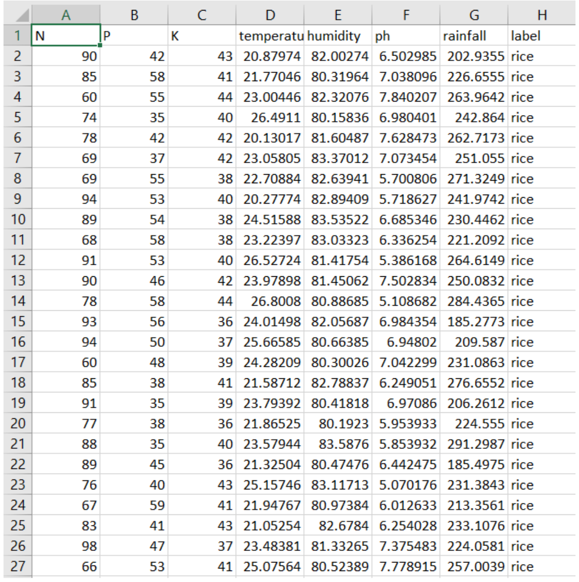
4 Proposed methodology

4.1 Data Collection:

The first step in implementing the Crop Recommendation System is to collect relevant data on soil nutrient levels (Nitrogen, Phosphorus, Potassium), temperature, humidity, pH, rainfall. This data serves as the foundation for training the machine learning models.

In our dataset, there are 8 columns in which 7 are the input parameters and the last one is the crop type which is classified based on these input factors.

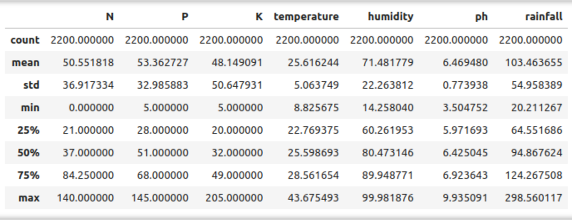
The screenshot of the raw CSV file of our dataset is given in Figure 1.



	A	B	C	D	E	F	G	H
1	N	P	K	temperatu	humidity	ph	rainfall	label
2	90	42	43	20.87974	82.00274	6.502985	202.9355	rice
3	85	58	41	21.77046	80.31964	7.038096	226.6555	rice
4	60	55	44	23.00446	82.32076	7.840207	263.9642	rice
5	74	35	40	26.4911	80.15836	6.980401	242.864	rice
6	78	42	42	20.13017	81.60487	7.628473	262.7173	rice
7	69	37	42	23.05805	83.37012	7.073454	251.055	rice
8	69	55	38	22.70884	82.63941	5.700806	271.3249	rice
9	94	53	40	20.27774	82.89409	5.718627	241.9742	rice
10	89	54	38	24.51588	83.53522	6.685346	230.4462	rice
11	68	58	38	23.22397	83.03323	6.336254	221.2092	rice
12	91	53	40	26.52724	81.41754	5.386168	264.6149	rice
13	90	46	42	23.97898	81.45062	7.502834	250.0832	rice
14	78	58	44	26.8008	80.88685	5.108682	284.4365	rice
15	93	56	36	24.01498	82.05687	6.984354	185.2773	rice
16	94	50	37	25.66585	80.66385	6.94802	209.587	rice
17	60	48	39	24.28209	80.30026	7.042299	231.0863	rice
18	85	38	41	21.58712	82.78837	6.249051	276.6552	rice
19	91	35	39	23.79392	80.41818	6.97086	206.2612	rice
20	77	38	36	21.86525	80.1923	5.953933	224.555	rice
21	88	35	40	23.57944	83.5876	5.853932	291.2987	rice
22	89	45	36	21.32504	80.47476	6.442475	185.4975	rice
23	76	40	43	25.15746	83.11713	5.070176	231.3843	rice
24	67	59	41	21.94767	80.97384	6.012633	213.3561	rice
25	83	41	43	21.05254	82.6784	6.254028	233.1076	rice
26	98	47	37	23.48381	81.33265	7.375483	224.0581	rice
27	66	53	41	25.07564	80.52389	7.778915	257.0039	rice

Figure 1: Some rows of our dataset

In figure 2, the dataset is statistically shown.

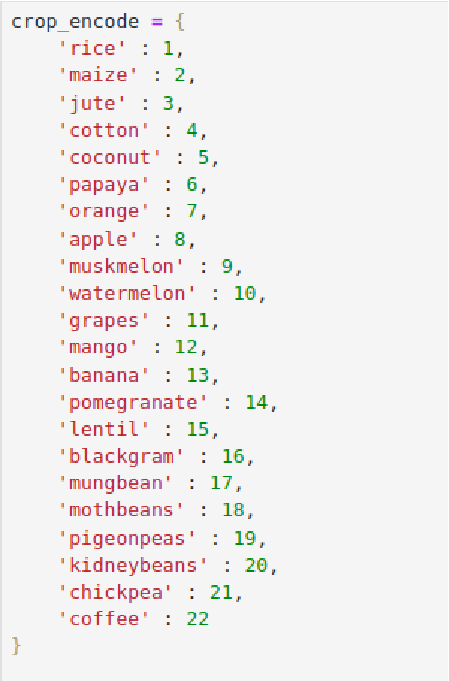


	N	P	K	temperature	humidity	ph	rainfall
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.211267
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.551686
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.867624
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.267508
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.560117

Figure 2: Statistical view of the dataset

4.2 Data Preprocessing:

Once the data is collected, it undergoes preprocessing to clean and prepare it for analysis. This includes handling missing values, encoding categorical variables, and scaling numerical features to ensure uniformity and consistency across the dataset. In our dataset there was no null values and no duplicate values so we didn't need to modify these. In next stage, we have encoded the categorical column of our dataset (In our dataset, 'label' field) into numerical values in which 1st type name is encoded by '1', 2nd by '2' and so on. Figure 3 describes the encoding of the categorical values.



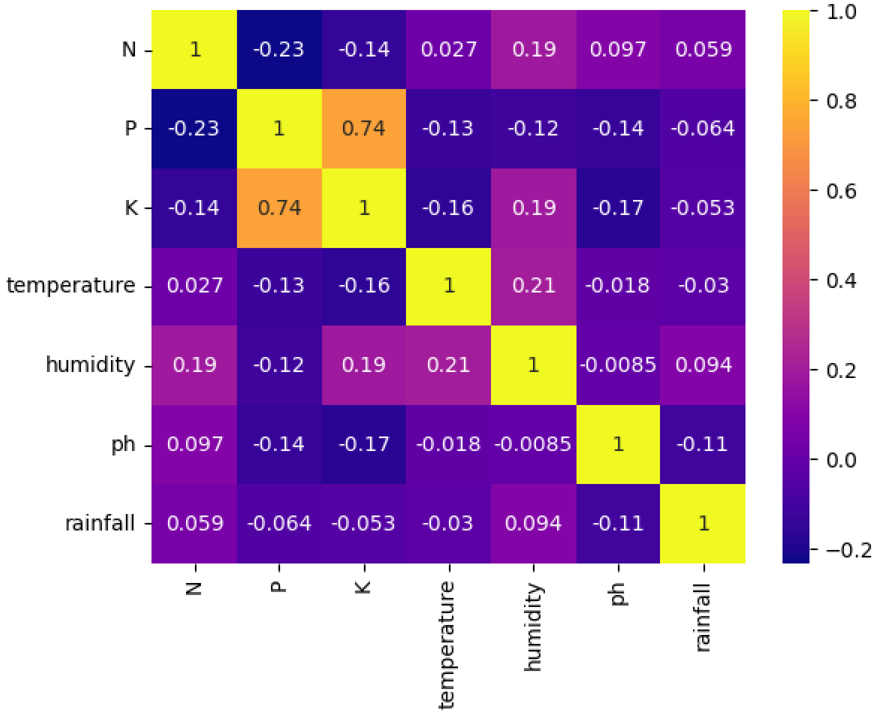
```
crop_encode = {
    'rice' : 1,
    'maize' : 2,
    'jute' : 3,
    'cotton' : 4,
    'coconut' : 5,
    'papaya' : 6,
    'orange' : 7,
    'apple' : 8,
    'muskmelon' : 9,
    'watermelon' : 10,
    'grapes' : 11,
    'mango' : 12,
    'banana' : 13,
    'pomegranate' : 14,
    'lentil' : 15,
    'blackgram' : 16,
    'mungbean' : 17,
    'mothbeans' : 18,
    'pigeonpeas' : 19,
    'kidneybeans' : 20,
    'chickpea' : 21,
    'coffee' : 22
}
```

Figure 3: Encoding of crop names

4.3 Exploratory Data Analysis (EDA):

EDA is performed to gain insights into the relationships between different environmental factors and crop yields. Visualization techniques such as histograms, scatter plots, and correlation matrices are used to identify patterns and trends in the data.

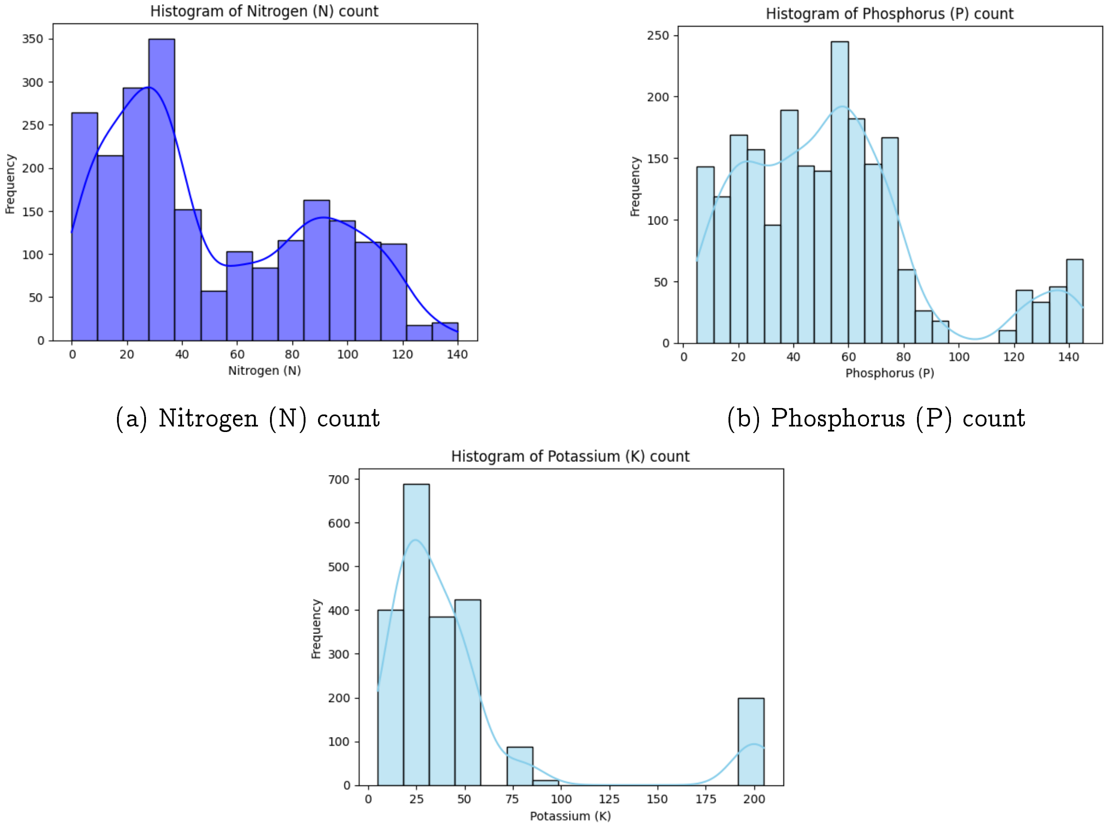
At first, the co-relations between different environmental input factors are determined and plotted as co-relation matrix using Seaborn heatmap function. Figure 4 shows the co-relation matrix.



	N	P	K	temperature	humidity	ph	rainfall
N	1	-0.23	-0.14	0.027	0.19	0.097	0.059
P	-0.23	1	0.74	-0.13	-0.12	-0.14	-0.064
K	-0.14	0.74	1	-0.16	0.19	-0.17	-0.053
temperature	0.027	-0.13	-0.16	1	0.21	-0.018	-0.03
humidity	0.19	-0.12	0.19	0.21	1	-0.0085	0.094
ph	0.097	-0.14	-0.17	-0.018	-0.0085	1	-0.11
rainfall	0.059	-0.064	-0.053	-0.03	0.094	-0.11	1

Figure 4: Co-relation matrix of the environmental input factors

Plots of input counts can provide insights into the distribution of different classes or categories within the dataset. Examining input counts can reveal potential data pre-processing steps that may be necessary before model training. So here the Histogram plots of the environmental input parameters are done for visualization. There are 7 input parameters such as ratio of Nitrogen (N), Phosphorus (P), Potassium (K), Temperature, Humidity, pH value of soil and Rainfall of that geographical location. Figure 5 and 6 shows all the Histogram plots for input factors.

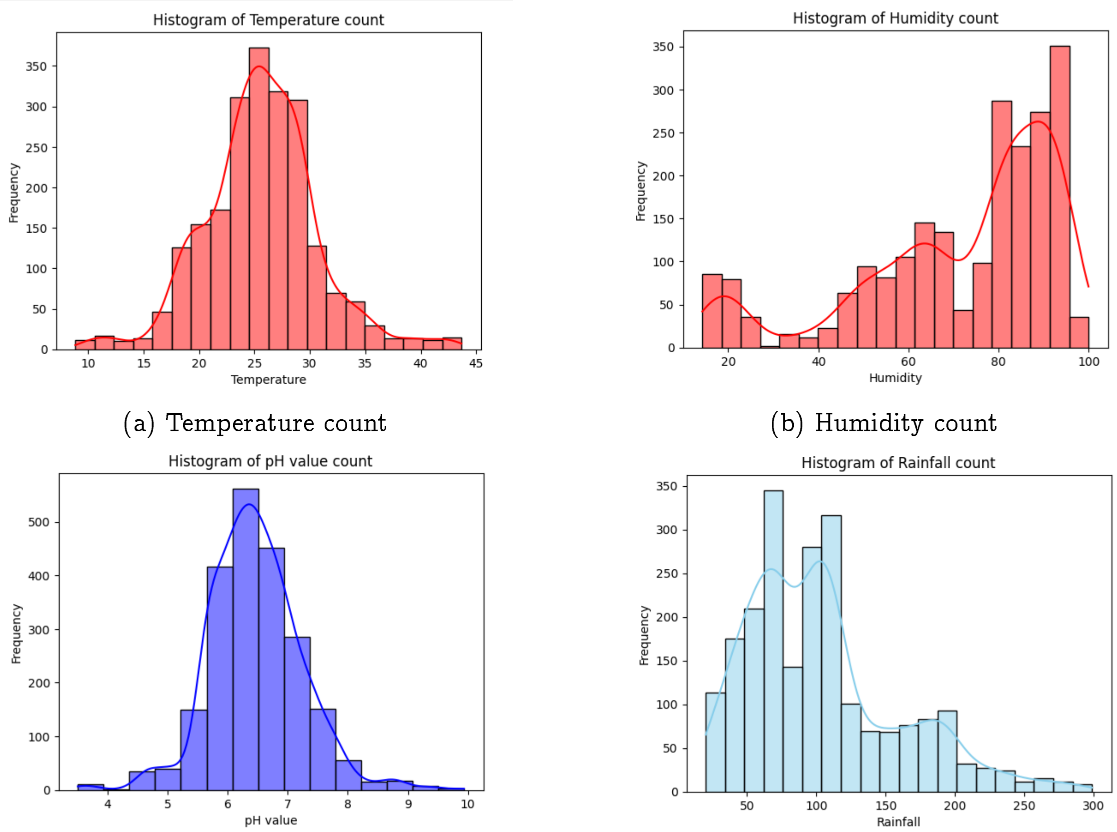


(a) Nitrogen (N) count

(b) Phosphorus (P) count

(c) Potassium (K) count

Figure 5: Histogram plots for Nitrogen (N), Phosphorus (P) and Potassium (K) counts



(a) Temperature count

(b) Humidity count

(c) pH count

(d) Rainfall count

Figure 6: Histogram plots for Temperature, Humidity, pH and Rainfall count

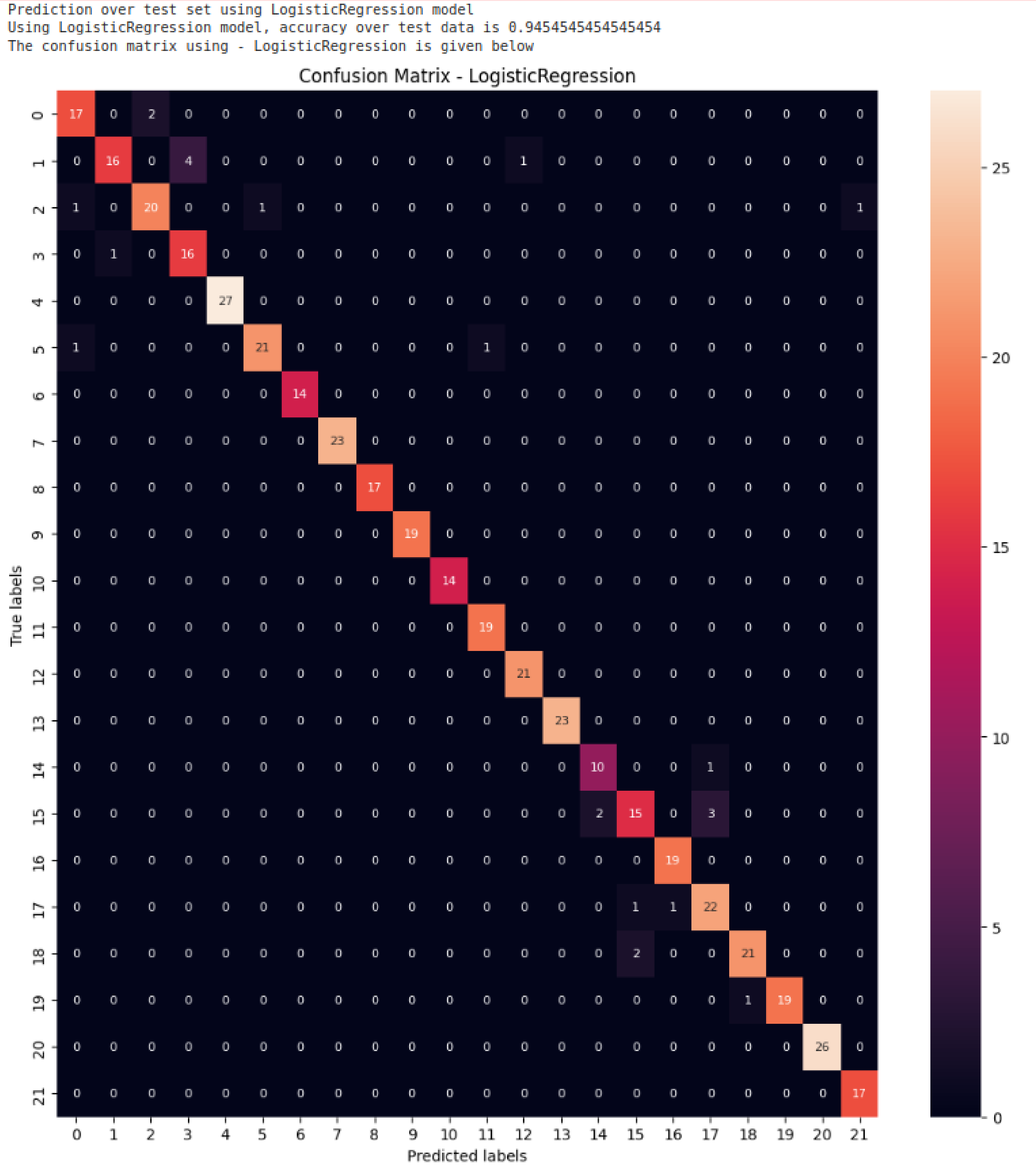
4.4 Model Selection:

After pre-processing the data, various machine learning models are evaluated to determine which one performs best for the task of crop recommendation. Models such as Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, Support Vector Machine, and K-Nearest Neighbors are considered. We have applied these models for training upon our dataset. We found the accuracy scores of all these models upon our dataset which are given in Table 1.

Sl no	Model Name	Accuracy score
1	LogisticRegression	0.94
2	DecisionTreeClassifier	0.98
3	RandomForestClassifier	0.99
4	GradientBoostingClassifier	0.98
5	SupportVectorMachine	0.96
6	KNeighborsClassifier	0.97

Table 1: Accuracy Scores of Various Models

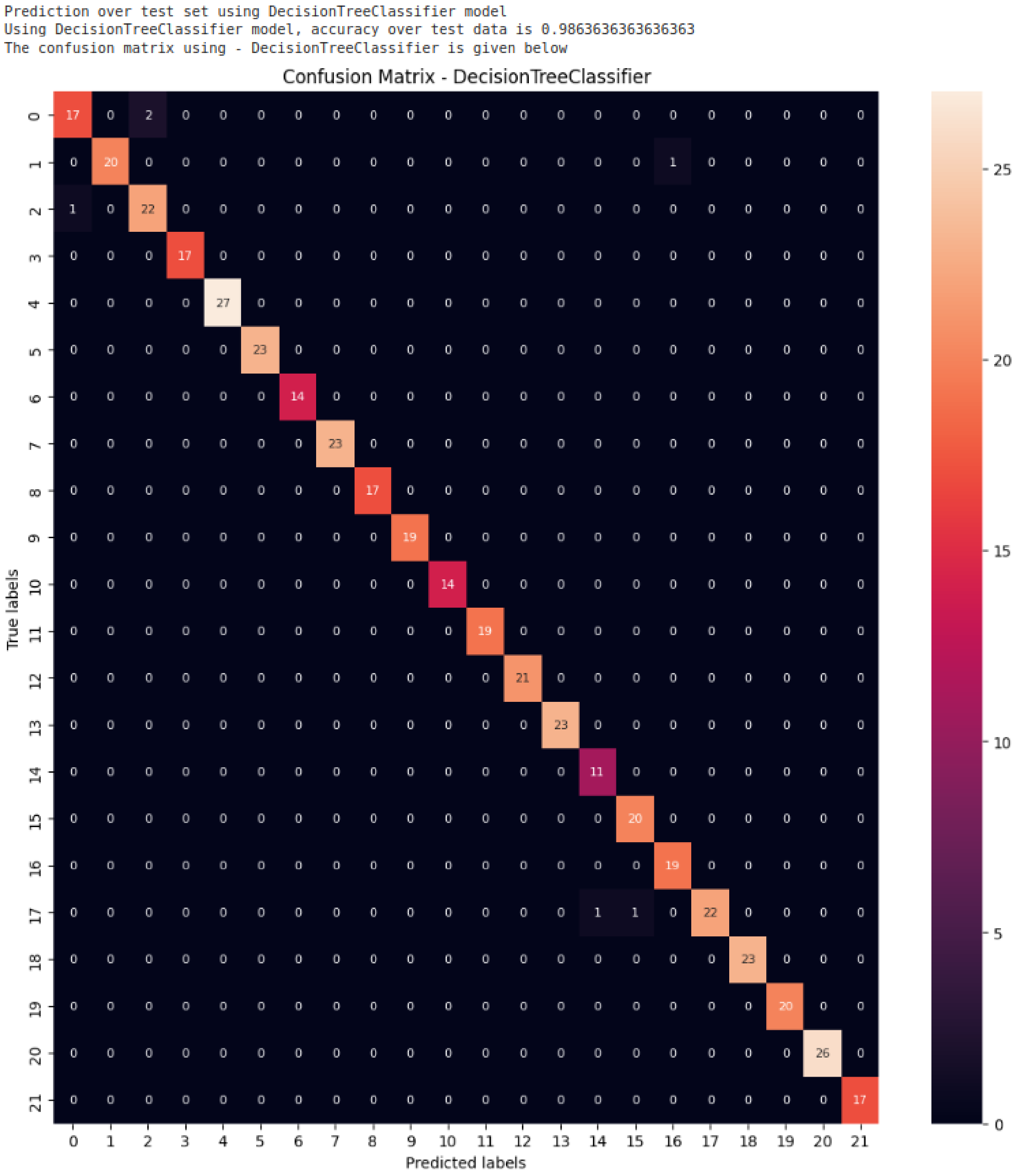
Confusion matrices provide a comprehensive summary of the performance of a classification model by tabulating the actual and predicted class labels. It allows for a detailed analysis of the types of errors made by the model. By breaking down the predictions into true positives, true negatives, false positives, and false negatives, one can identify which classes are often confused with each other and understand the specific nature of the model's mistakes. So for model selection, we have used Confusion matrices to facilitate the comparison of multiple models by providing a standardized framework for evaluating their performance. Figure 7 - 12 shows all the Confusion Matrices.



```
Prediction over test set using LogisticRegression model
Using LogisticRegression model, accuracy over test data is 0.9454545454545454
The confusion matrix using - LogisticRegression is given below
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	17	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	16	0	4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
2	1	0	20	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	0	1	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	21	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	1	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	15	0	3	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	22	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	21	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	19	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17

Figure 7: Confusion Matrix for LogisticRegression model with accuracy 94%

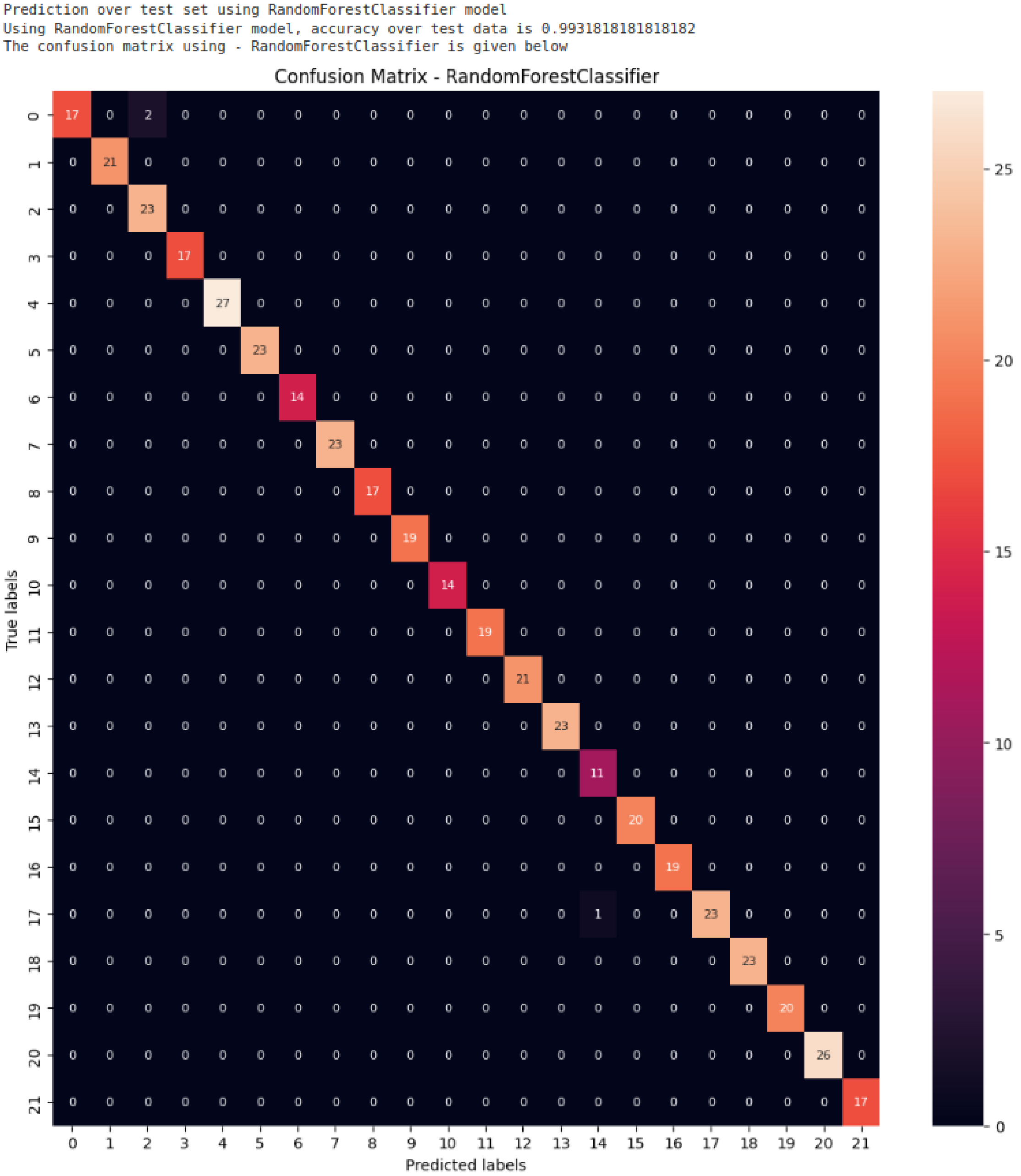


```
Prediction over test set using DecisionTreeClassifier model
Using DecisionTreeClassifier model, accuracy over test data is 0.9863636363636363
The confusion matrix using - DecisionTreeClassifier is given below
```

Confusion Matrix - DecisionTreeClassifier

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	17	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
2	1	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	22	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17

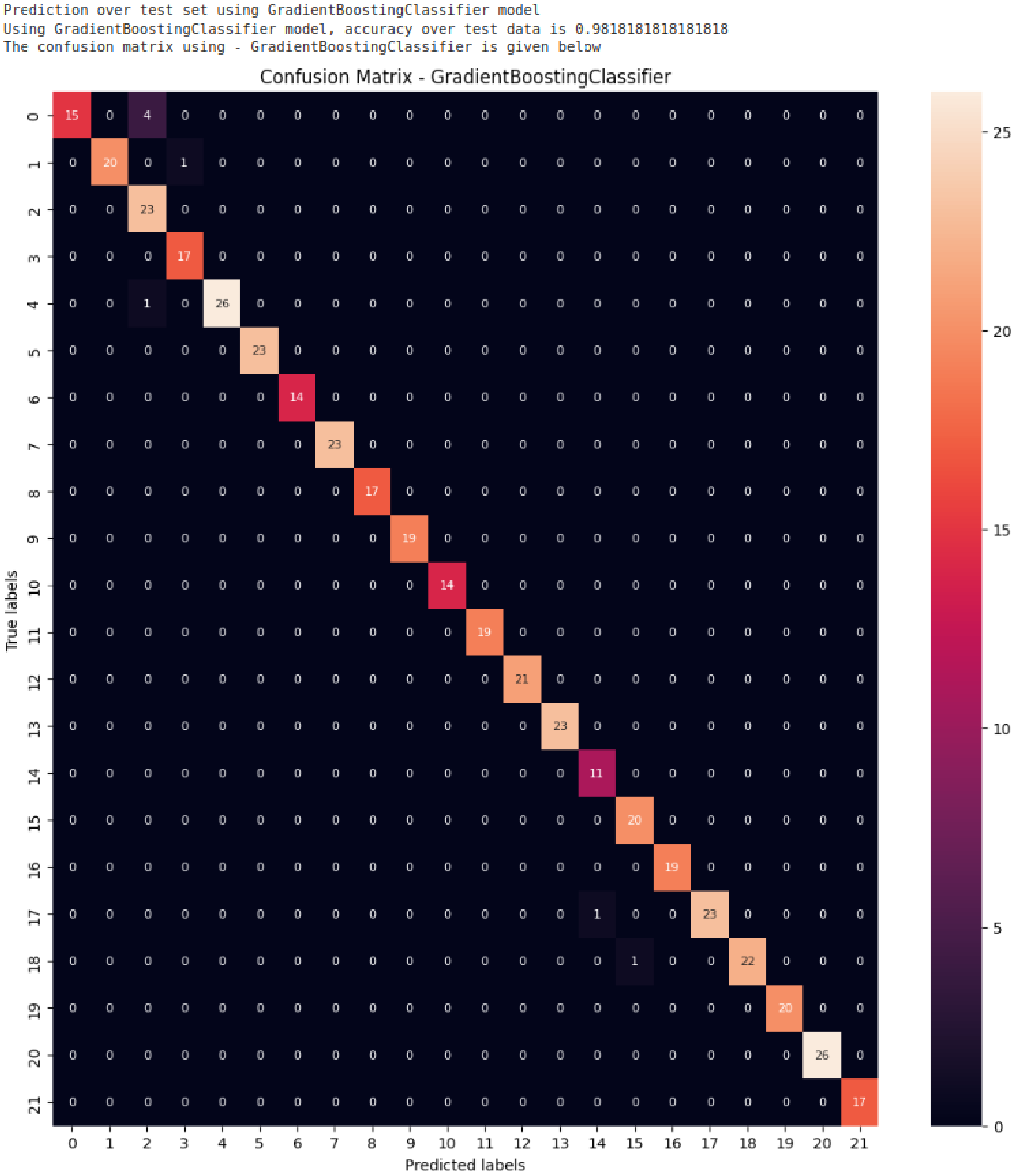
Figure 8: Confusion Matrix for DecisionTreeClassifier model with accuracy 98%



```
Prediction over test set using RandomForestClassifier model
Using RandomForestClassifier model, accuracy over test data is 0.9931818181818182
The confusion matrix using - RandomForestClassifier is given below
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	17	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	23	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17

Figure 9: Confusion Matrix for RandomForestClassifier model with accuracy 99%

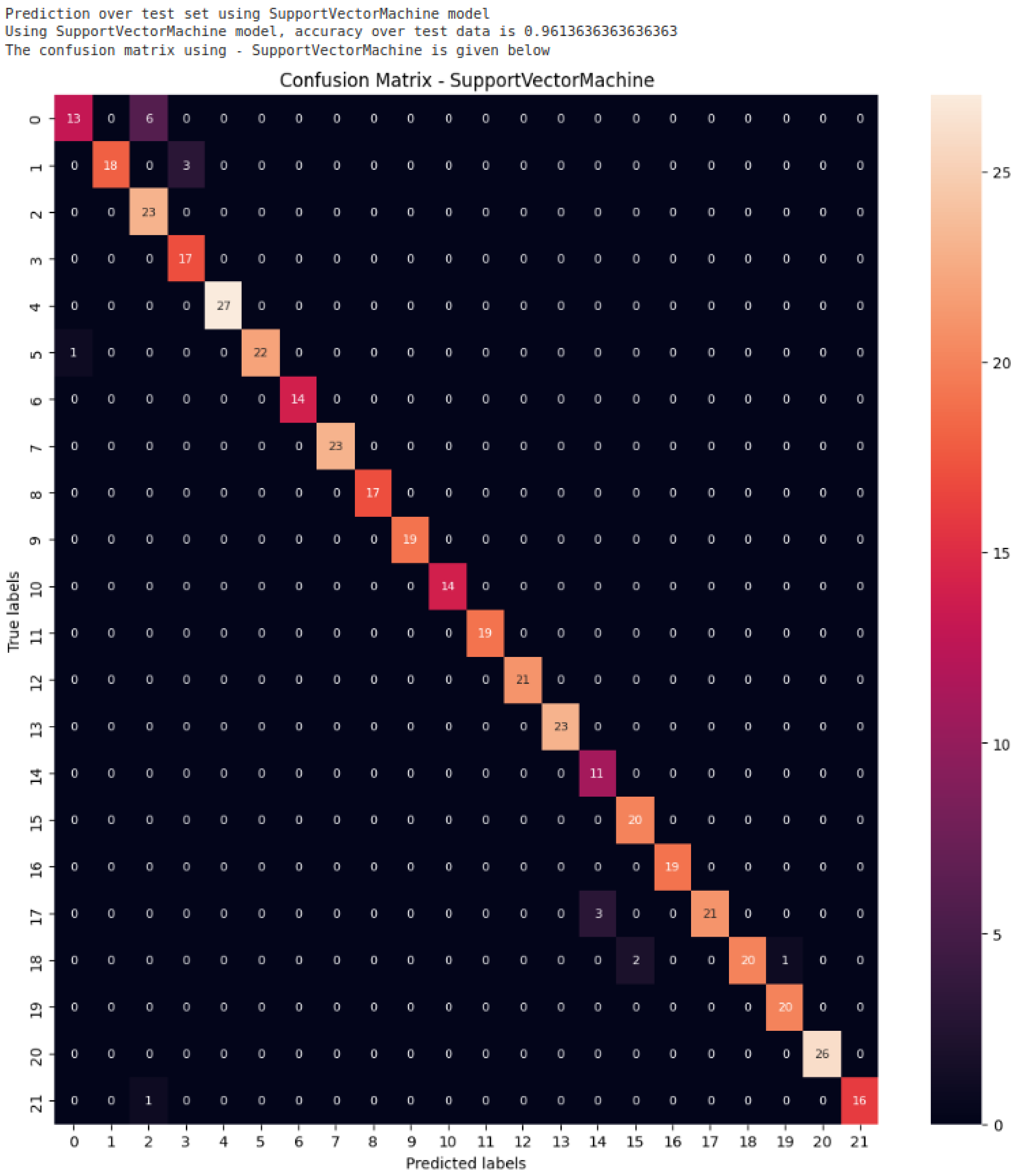


```
Prediction over test set using GradientBoostingClassifier model
Using GradientBoostingClassifier model, accuracy over test data is 0.9818181818181818
The confusion matrix using - GradientBoostingClassifier is given below
```

Confusion Matrix - GradientBoostingClassifier

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	15	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	20	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	23	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	22	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17

Figure 10: Confusion Matrix for GradientBoostingClassifier model with accuracy 98%



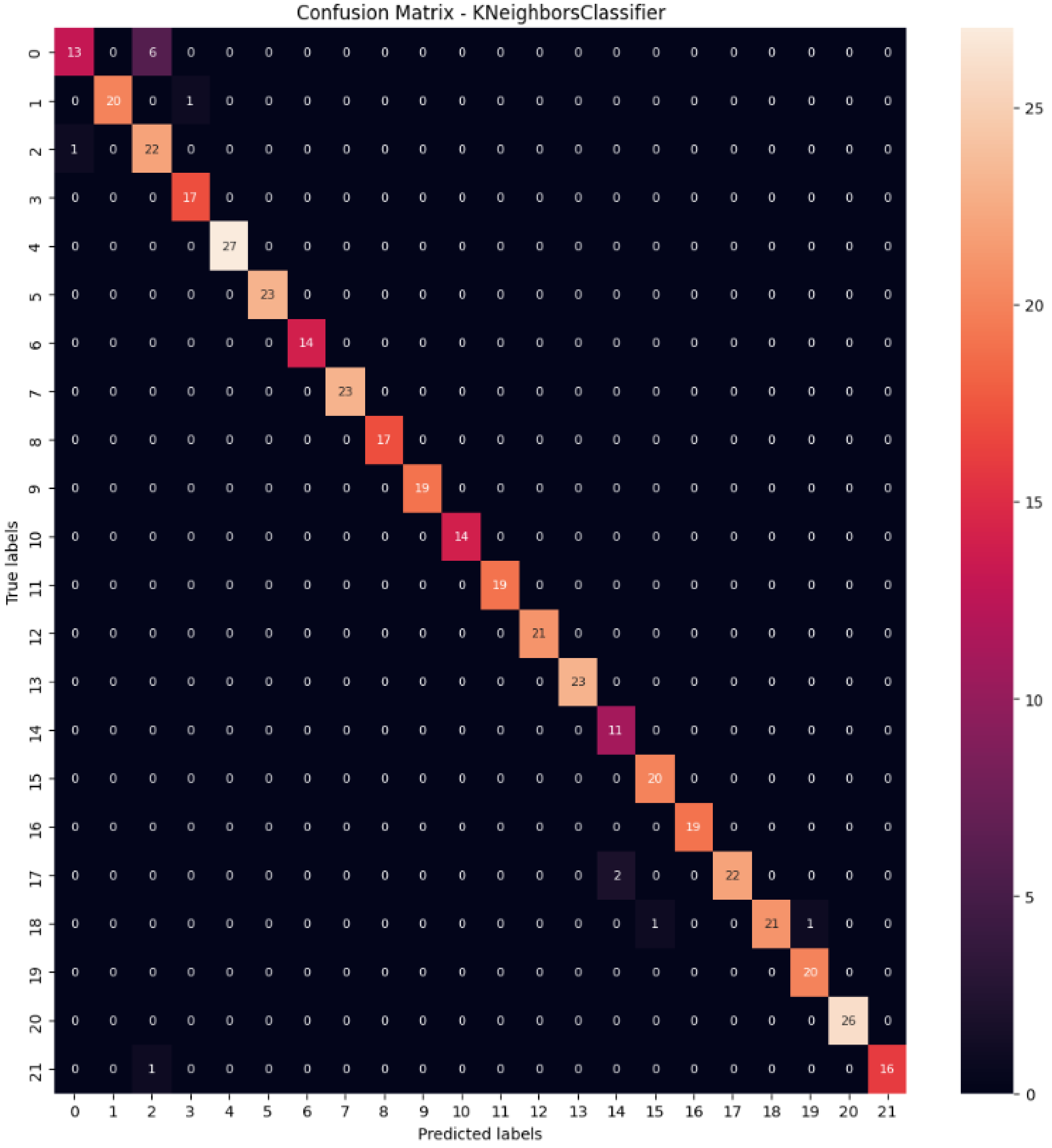
```
Prediction over test set using SupportVectorMachine model
Using SupportVectorMachine model, accuracy over test data is 0.9613636363636363
The confusion matrix using - SupportVectorMachine is given below
```

Confusion Matrix - SupportVectorMachine

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	13	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	18	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	21	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	20	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0
21	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16

Figure 11: Confusion Matrix for SupportVectorMachine model with accuracy 96%

```
Prediction over test set using KNeighborsClassifier model
Using KNeighborsClassifier model, accuracy over test data is 0.9704545454545455
The confusion matrix using - KNeighborsClassifier is given below
```



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	13	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	20	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	22	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	21	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0
21	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16

Figure 12: Confusion Matrix for KNeighborsClassifier model with accuracy 97%

4.5 Model Training:

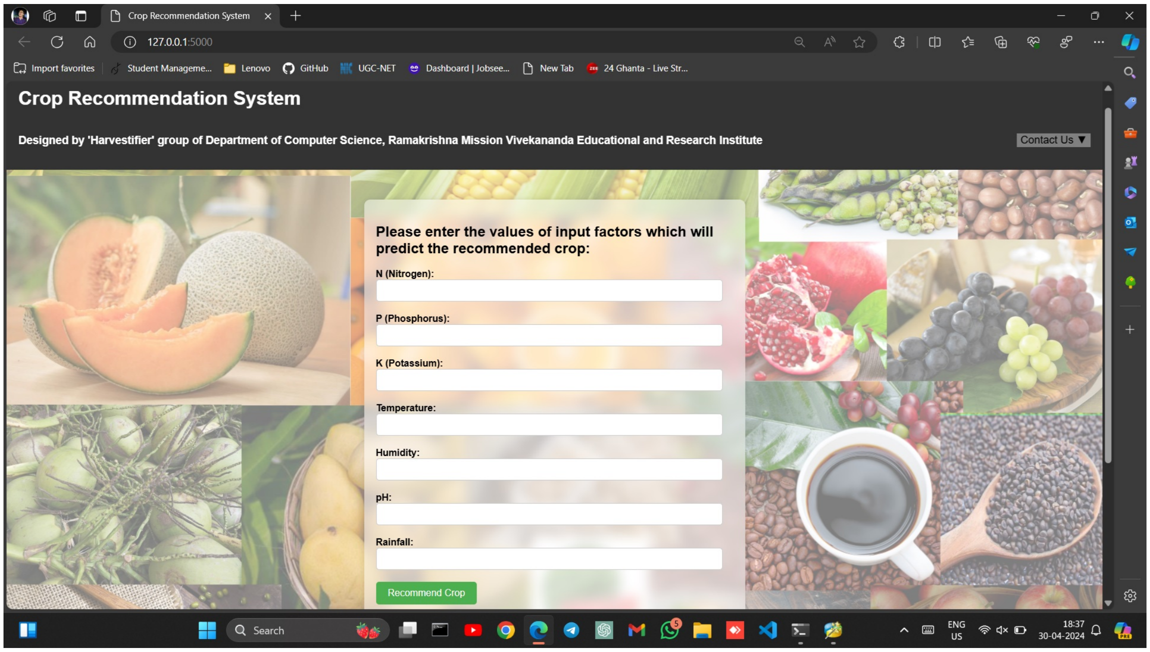
The selected machine learning model (Random Forest Classifier in this case) is trained on the pre-processed dataset. The model learns to map the input environmental factors to the corresponding crop recommendations based on the historical data. We considered Random Forest Classifier model as best_model for us. After splitting our dataset into 80-20 split for training-testing, we trained it by importing the model from Scikit-learn python package and applying fit() function on RandomForestClassifier.

4.6 Model Evaluation:

Once trained, the model is evaluated using a separate test dataset to assess its performance. Metrics such as accuracy score and confusion matrix are used to measure the model's effectiveness in predicting crop recommendations. Our training dataset contains 1760 records and testing dataset contains 440 records.

4.7 Integration with UI:

The trained model is integrated into a web application using HTML/CSS for the user interface. The application allows users to input their land's characteristics through text input boxes styled with CSS, providing a visually appealing and intuitive experience. Upon submission of the input data, the application sends a request to the backend, where the machine learning model generates crop recommendations based on the provided parameters. Flask framework is used for backend. The recommended crops are then displayed to the user on the web page, allowing farmers to make informed decisions about crop selection for their agricultural land directly through the custom-designed user interface. Our Web UI looks like the following figure.



Crop Recommendation System

Designed by 'Harvestifier' group of Department of Computer Science, Ramakrishna Mission Vivekananda Educational and Research Institute

Please enter the values of input factors which will predict the recommended crop:

N (Nitrogen):

P (Phosphorus):

K (Potassium):

Temperature:

Humidity:

pH:

Rainfall:

Figure 13: Screenshot of Webpage User Interface

By following this methodology, the Crop Recommendation System provides a robust and user-friendly solution for optimizing crop selection processes and improving agricultural productivity.

5 Experimental result

The experimental results demonstrate the effectiveness of the proposed method in accurately predicting crop recommendations based on environmental factors. The system achieves competitive performance compared to state-of-the-art methods while maintaining reasonable time complexity for real-time application.

5.1 Used Datasets:

The Crop Recommendation System was evaluated using a standard dataset obtained from the site: https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset?resource=download

This dataset comprises records of soil nutrient levels (such as Nitrogen, Phosphorus, Potassium), temperature, humidity, pH, rainfall, and corresponding crop names which is suitable for yielding based on these environmental factors.

5.2 Experimental Settings:

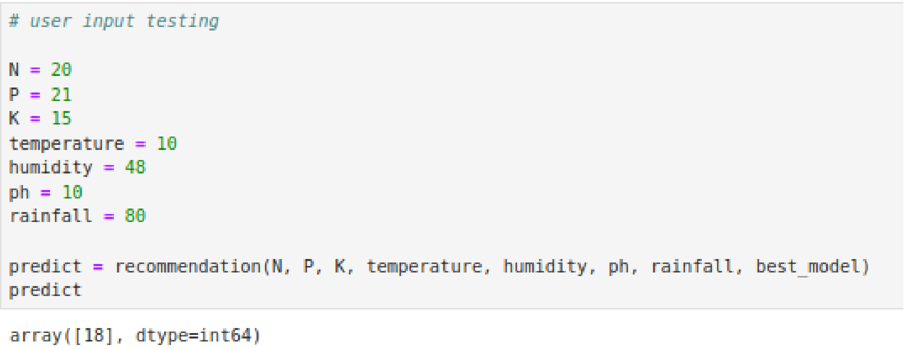
The machine learning models were trained and evaluated using a 80-20 train-test split. The dataset was preprocessed to handle missing values, encode categorical variables, and scale numerical features. Various machine learning algorithms, including Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, Support Vector Machine, and K-Nearest Neighbors, were implemented and compared by the Confusion Matrices which were shown before (Figure 7).

5.3 Experimental Results:

After deploying the whole system, it requires a lot of testing and experiments for understanding how effective is the model's performance and reliability of its performance. We have used our test dataset values for giving into the input variables during our testing and experimental stage.

5.3.1 In Python Notebook:

The screenshot is provided where we have tested the model for prediction using input values such as, N = 20, P = 21, K = 15, Temperature = 10, Humidity = 48, pH = 10, Rainfall = 80.

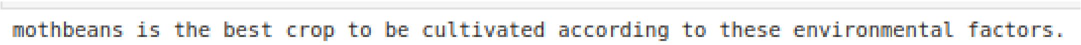


```
# user input testing

N = 20
P = 21
K = 15
temperature = 10
humidity = 48
ph = 10
rainfall = 80

predict = recommendation(N, P, K, temperature, humidity, ph, rainfall, best_model)
predict
```

```
array([18], dtype=int64)
```

Figure 14: Predicting crop type in numerical encoded format

Then again the crop names are decoded into the original names from the numerical encoding.

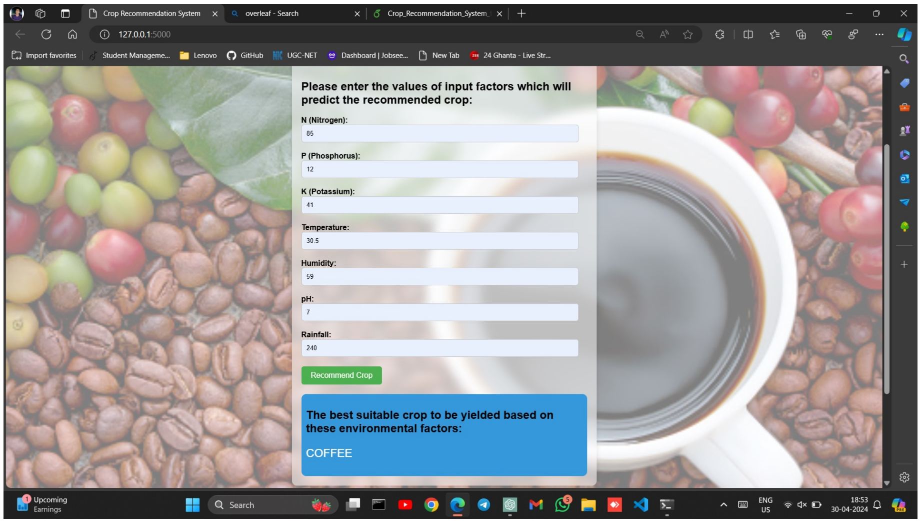


```
mothbeans is the best crop to be cultivated according to these environmental factors.
```

Figure 15: Predicting crop name

5.3.2 In Web User Interface:

Some screenshots are provided where we have tested with some input values for checking the result is accurate or not.



Please enter the values of input factors which will predict the recommended crop:

N (Nitrogen): 85

P (Phosphorus): 12

K (Potassium): 41

Temperature: 30.5

Humidity: 59

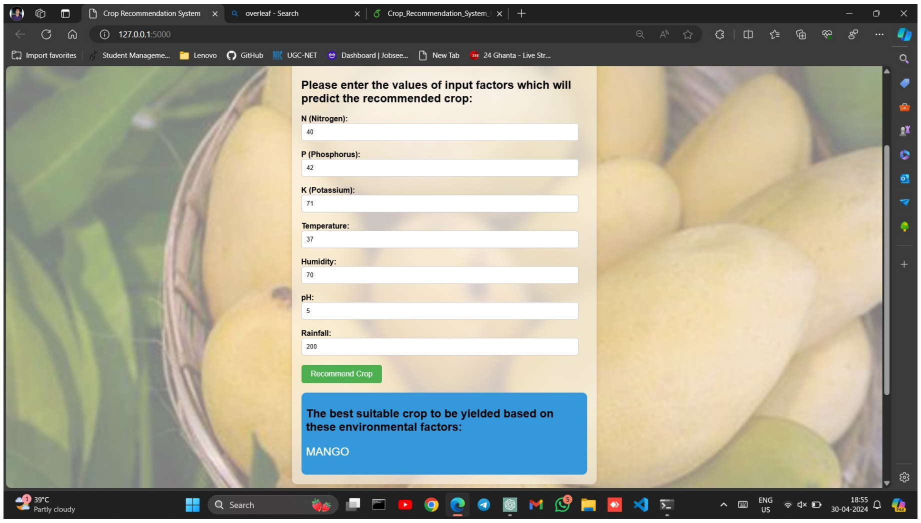
pH: 7

Rainfall: 240

The best suitable crop to be yielded based on these environmental factors:

COFFEE

Figure 16: Value Testing in User Interface



Please enter the values of input factors which will predict the recommended crop:

N (Nitrogen): 40

P (Phosphorus): 42

K (Potassium): 71

Temperature: 37

Humidity: 70

pH: 5

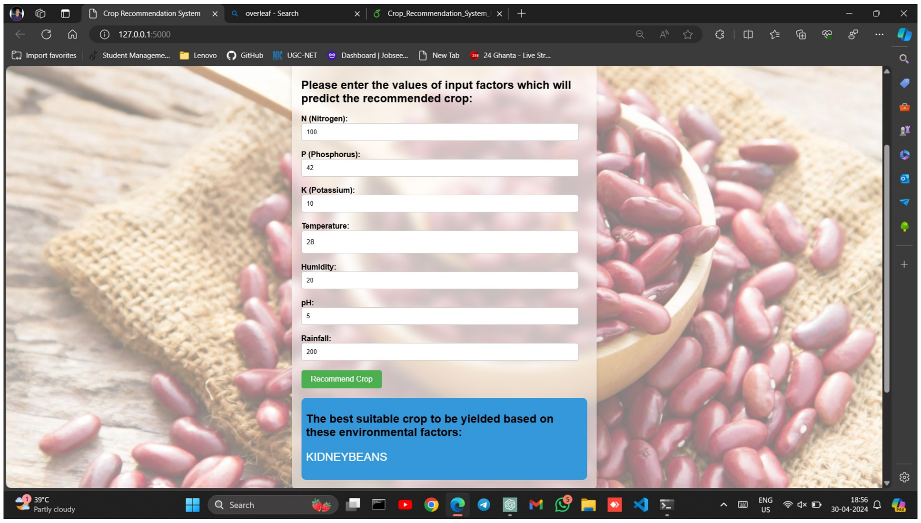
Rainfall: 200

Recommend Crop

The best suitable crop to be yielded based on these environmental factors:

MANGO

Figure 17: Value Testing in User Interface



Please enter the values of input factors which will predict the recommended crop:

N (Nitrogen): 100

P (Phosphorus): 42

K (Potassium): 10

Temperature: 28

Humidity: 20

pH: 5

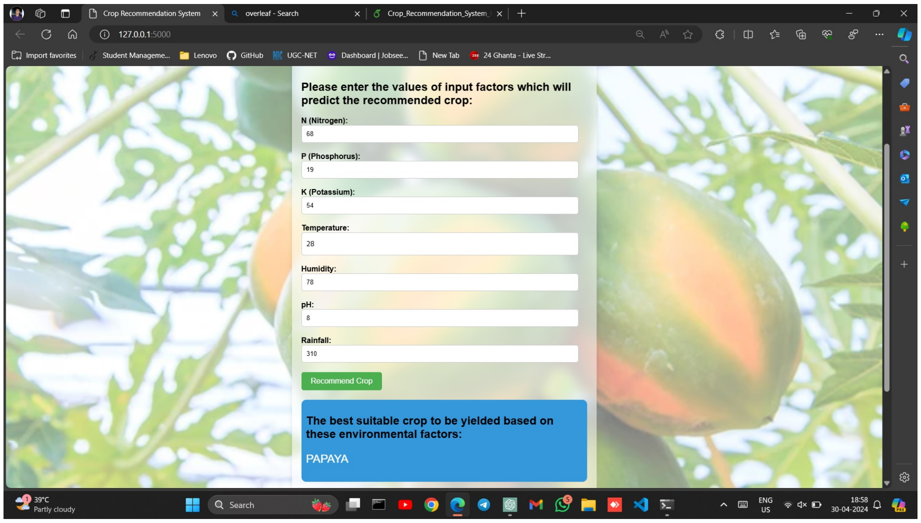
Rainfall: 200

Recommend Crop

The best suitable crop to be yielded based on these environmental factors:

KIDNEYBEANS

Figure 18: Value Testing in User Interface



Please enter the values of input factors which will predict the recommended crop:

N (Nitrogen): 68

P (Phosphorus): 19

K (Potassium): 54

Temperature: 28

Humidity: 78

pH: 8

Rainfall: 310

Recommend Crop

The best suitable crop to be yielded based on these environmental factors:

PAPAYA

Figure 19: Value Testing in User Interface

5.4 Time Complexity:

The time complexity of the Crop Recommendation System project can vary depending on several factors, including:

5.4.1 Data Preprocessing:

Time complexity of data preprocessing steps such as handling missing values, encoding categorical variables, and splitting the dataset into training and testing sets depends on the size of the dataset and the complexity of preprocessing operations. Generally, these operations have a linear time complexity, but certain operations like one-hot encoding of categorical variables may have higher time complexity.

5.4.2 Model Training:

The time complexity of training the RandomForestClassifier model depends on the size of the training dataset (n) and the number of features (m). For each decision tree in the random forest, the time complexity of training is approximately $O(n \cdot m \cdot \log(n))$.

5.4.3 Web Interface:

The time complexity of serving web requests and rendering the HTML/CSS interface primarily depends on the server's performance and the complexity of the web application logic. Simple web applications with basic functionality typically have low time complexity for handling user requests.

Overall, the time complexity of the Crop Recommendation System project is influenced by the size of the dataset, the complexity of preprocessing operations, the number of decision trees in the RandomForestClassifier model, and the complexity of the web application logic. As such, it's challenging to provide a precise time complexity analysis without specific details about the dataset size, model parameters, and server performance. However, in practice, the system is designed to handle real-time user requests efficiently, making it suitable for practical applications in agriculture.

6 Summary

In conclusion, the Crop Recommendation System project represents a significant step towards leveraging machine learning technology to address agricultural challenges. By integrating the Machine Learning model with a user-friendly HTML/CSS web interface, we have created a versatile tool capable of providing tailored crop recommendations based on key environmental parameters. This system has the potential to revolutionize traditional farming practices by offering data-driven insights to farmers and agriculture professionals.

Through extensive data exploration, preprocessing, and model selection processes, we have identified the Random Forest Classifier as the most suitable model for predicting crop recommendations with high accuracy. Its ability to handle complex relationships within the dataset and provide robust predictions makes it an ideal choice for our application.

The user interface design emphasizes simplicity, intuitiveness, and accessibility, allowing users to easily input their environmental parameters and receive instant recommendations. Input validation mechanisms ensure that users provide valid input values, enhancing the reliability of the system's output.

Furthermore, the system's output display seamlessly integrates the predicted crop recommendation into the same webpage, providing users with immediate feedback and eliminating the need for navigating to separate pages.

Overall, the Crop Recommendation System holds immense potential to empower farmers with actionable insights, optimize crop selection processes, increase agricultural productivity, and ultimately contribute to sustainable food production. As we continue to refine and improve the system, it will play a pivotal role in shaping the future of precision agriculture and fostering innovation in the farming industry.

References

[1] Swapneel Chakraborty Ayush Kumar, Omen Rajendra Pooniwala. Intelligent crop recommendation system using ml. Technical report, Visvesvaraya Technological University, 2019.

[2] BDA. Big data analytics. http://cs.rkmvu.ac.in/academics-msc-in-big-data-analytics-data-science/, 2016.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[4] Bernhard E Boser, Isabelle M Guyon, and Vladimir Naumovich Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

[5] Leo Breiman. Random forests. *Machine Learning*, 45(1):5 – 32, 2001.

[6] Google. Crops images. http://www.google.com/images.

[7] Sai Manas Gandham Lakshman Kumar Seru. Crop recommendation system using machine learning algorithm. Technical report, Department of Computer Science and Engineering, School of Computing, Sathyabama Institute of Science and Technology, 2022.

[8] Omkar Joshi Rajashree Shedge Mangesh Deshmukh, Amit Kumar Jaiswar. Farming assistance for soil fertility improvement and crop prediction using xgboost. Technical report, Department of Computer Engineering, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai, 2022.

[9] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning.* The MIT Press, 2nd edition, 2018.

[10] R. Harine Rajashree C. Kavya T. Kiruthika J. Nisha S. Pudumalar, E. Ramanujam. Crop recommendation system for precision agriculture. Technical report, 2017.

[11] Rashi Agrawal Prof. Neepa Shah Zeel Doshi, Subhash Nadkarni. Agro consultant - intelligent crop recommendation system using machine learning algorithms. Technical report, 2018.