A

Assesment Report on

"Crop Recommendation System"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25 in

CSE(AIML)

By

Shubham Chaudhary (202401100400184)

Sudhakar Kumar (202401100400190)

Sudhanshu Gill (202401100400191)

Utkarsh Kumar Singh (202401100400202)

Vikrant Baliyan (202401100400211)

Under the supervision of

Abhishek Shukla

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Agriculture plays a crucial role in the economy of many countries, and choosing the right crop for cultivation is vital for increasing productivity and ensuring sustainable farming.
Traditionally, farmers make crop choices based on experience or generic recommendations, which may not consider real-time environmental and soil factors.
This project aims to build a **machine learning-based Crop Recommendation System** that suggests the most suitable crop based on key soil and climatic conditions such as Nitrogen, Phosphorus, Potassium content, temperature, humidity, pH, and rainfall.

Methodology

1 Data Collection

The dataset used in this project is **Crop_recommendation.csv**, which contains agricultural data points including:

- Soil nutrients: **Nitrogen (N)**, **Phosphorus (P)**, **Potassium (K)**
- Environmental data: **Temperature**, **Humidity**, **pH**, **Rainfall**
- Target label: **Crop Name**

2 Data Preprocessing

- Loaded the dataset using pandas.
- Encoded the target crop labels using LabelEncoder.
- Checked for null values and data consistency.

3 Model Selection

We used a **Random Forest Classifier**, a powerful ensemble method that combines multiple decision trees to improve accuracy and avoid overfitting.

4 Model Training

- Split the dataset using an 80:20 train-test ratio.
- Trained the model using training data.
- Evaluated performance using accuracy score, classification report, and confusion matrix.

5 Crop Recommendation Function

We developed a simple function recommend_crop() that takes seven input parameters and returns the recommended crop based on the trained model's prediction.

Code

```
# ☑ STEP 1: Install required libraries (only if missing in Colab)
!pip install -q scikit-learn

# ☑ STEP 2: Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# ☑ STEP 3: Upload CSV File
from google.colab import files
uploaded = files.upload()  # Select "Crop_recommendation.csv" here

# ☑ STEP 4: Load dataset
filename = next(iter(uploaded))
df = pd.read_csv(filename)
print("☑ Dataset Preview:")
print(df.head())

# ☑ STEP 5: Encode the target labels
le = LabelEncoder()
df['label_encoded'] = le.fit_transform(df['label'])

# ☑ STEP 6: Prepare features and target
X = df.drop(['label', 'label_encoded'], axis=1)
y = df['label_encoded']

# ☑ STEP 7: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# ☑ STEP 8: Train Random Forest Classifier
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)

# ☑ STEP 9: Model Evaluation
y_pred = clf.predict(X_test)
acc = accuracy_score(y_test, y_pred)
```

```
print(f"\n🎯 Accuracy: {acc:.2f}")
print("\n📊 Classification Report:")
print(classification_report(y_test, y_pred))

# ☑ Confusion Matrix
plt.figure(figsize=(10, 5))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='YlGnBu')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# ☑ STEP 10: Crop Recommendation Function
def recommend_crop(N, P, K, temperature, humidity, ph, rainfall):
    input_data = np.array([[N, P, K, temperature, humidity, ph, rainfall]])
    prediction = clf.predict(input_data)
    crop = le.inverse_transform(prediction)
    return crop[0]

# ☑ STEP 11: Take user input for prediction
print("\n🌿 Enter values to get crop recommendation:")
N = int(input("Nitrogen (N): "))
P = int(input("Phosphorous (P): "))
K = int(input("Potassium (K): "))
temperature = float(input("Temperature (°C): "))
humidity = float(input("Humidity (%): "))
ph = float(input("pH value: "))
rainfall = float(input("Rainfall (mm): "))

# ☑ STEP 12: Get prediction
recommended_crop = recommend_crop(N, P, K, temperature, humidity, ph, rainfall)
print(f"\n☑ Recommended Crop: {recommended_crop}")
```

Output

Choose Files Crop_reco...endation.csv

- **Crop_recommendation.csv**(text/csv) - 150034 bytes, last modified: 5/27/2025 - 100% done

Saving Crop_recommendation.csv to Crop_recommendation.csv

✅ Dataset Preview:

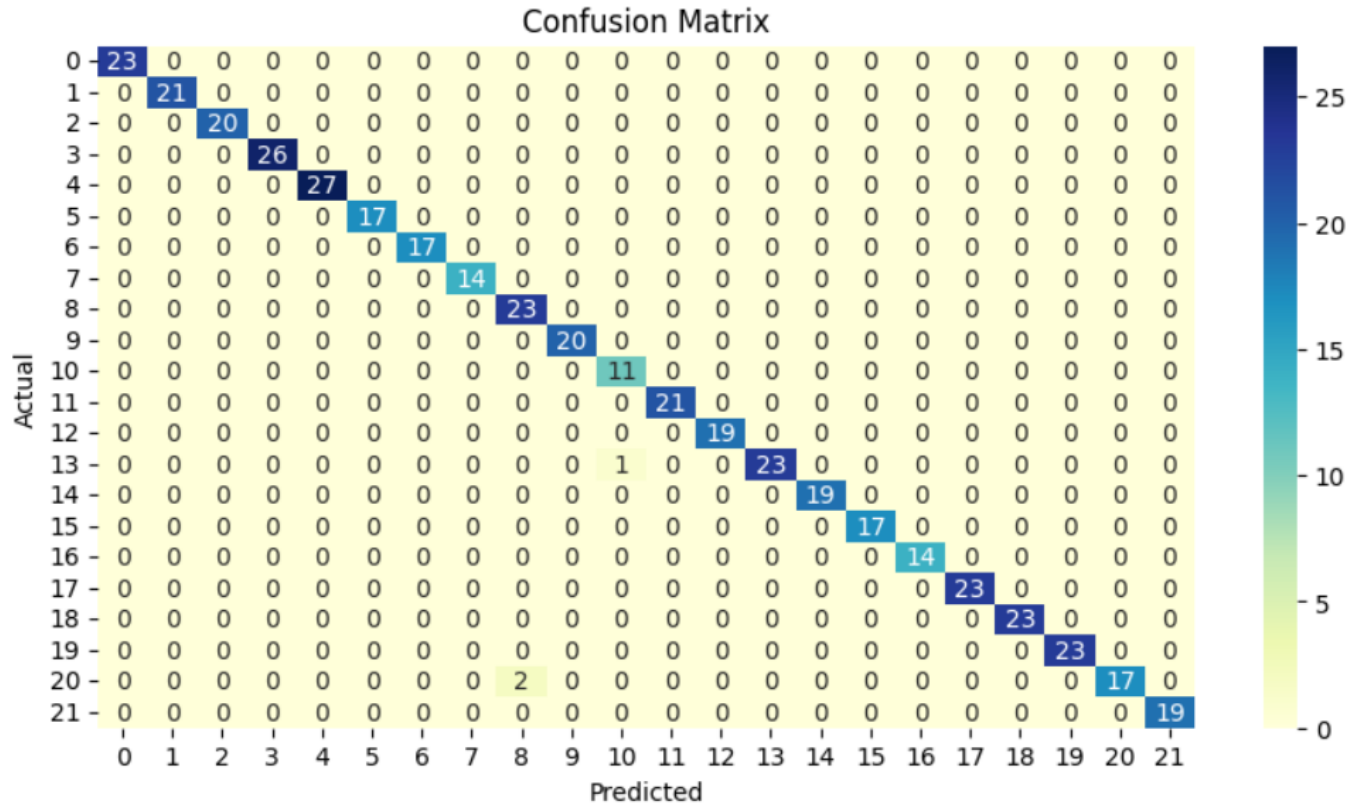
	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

🎯 Accuracy: 0.99

📊 Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	1.00	1.00	1.00	21
2	1.00	1.00	1.00	20
3	1.00	1.00	1.00	26
4	1.00	1.00	1.00	27
5	1.00	1.00	1.00	17
6	1.00	1.00	1.00	17
7	1.00	1.00	1.00	14
8	0.92	1.00	0.96	23
9	1.00	1.00	1.00	20
10	0.92	1.00	0.96	11
11	1.00	1.00	1.00	21
12	1.00	1.00	1.00	19
13	1.00	0.96	0.98	24
14	1.00	1.00	1.00	19
15	1.00	1.00	1.00	17
16	1.00	1.00	1.00	14
17	1.00	1.00	1.00	23
18	1.00	1.00	1.00	23
19	1.00	1.00	1.00	23
20	1.00	0.89	0.94	19
21	1.00	1.00	1.00	19

accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440



🌿 Enter values to get crop recommendation:
Nitrogen (N): 200
Phosphorous (P): 80
Potassium (K): 200
Temperature (°C): 30
Humidity (%): 75
pH value: 7

Rainfall (mm): 2000

✅ Recommended Crop: banana

References

- Dataset: Crop Recommendation Dataset (Kaggle)
- Scikit-learn Documentation: https://scikit-learn.org/
- Pandas Documentation: https://pandas.pydata.org/
- Streamlit Documentation (for future deployment): https://docs.streamlit.io/

Credits

This Crop Recommendation System project was inspired and made possible by various open-source tools, communities, and resources. Special thanks to:

- **Kaggle** for providing the Crop Recommendation Dataset.
- **Scikit-learn** for its powerful machine learning tools and documentation.
- **Google Colab** for providing a free and accessible cloud-based environment for running Python code.
- **Pandas**, **NumPy**, **Seaborn**, and **Matplotlib** for data manipulation and visualization.
- Online contributors and educators who share tutorials and projects that help learners build practical skills in data science and machine learning.