Assessment Report

on

"Crop Recommendation System"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI) Section C

By

GROUP NO. :- 04

Raja Yadav (202401100300193)
Rishabh Kumar Singh (202401100300199)
Rishabh Singh (202401100300201)
Rituraj Kumar (202401100300201)
Saksham Yadav (202401100300201)

Under the supervision of

"Mr. Mayank Lakhotia"

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

India is an agrarian country where crop selection based on soil and weather conditions plays a critical role in optimizing agricultural yield. This project proposes a machine learning-based Crop Recommendation System that analyzes parameters like soil nutrients (NPK), temperature, humidity, pH, and rainfall to suggest the most suitable crop. This approach aims to support farmers in making data-driven crop decisions, enhancing productivity and sustainability.

2. Problem Statement

To develop a classification model that recommends the best crop to cultivate based on environmental factors including soil nutrients and weather conditions.

3. Objectives

- To preprocess and analyze agricultural and climatic data.
- To build a classification model using Random Forest to predict the best-suited crop.
- To evaluate model performance using standard classification_metrics.
- To visualize prediction performance using confusion matrix and heatmaps.

4. Methodology

Data Collection:
The dataset was sourced from a publicly available Crop Recommendation Dataset, which includes features such as nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall.

Data Preprocessing:
- Removed missing values (none present).
- No categorical encoding needed as all features were numeric.
- Split data into training (80%) and testing (20%) sets.

Model Building:
- Applied Random Forest Classifier due to its efficiency and robustness.
- Trained on features to classify among 22 possible crops.

Model Evaluation:
- Used accuracy, precision, recall, F1-score as performance metrics.
- Used confusion matrix heatmap for detailed analysis.

5. Data Preprocessing

- All data was numeric and clean.
- Features included: N, P, K, temperature, humidity, pH, rainfall.
- Dataset was split into 80% training and 20% testing.
- No scaling was needed for tree-based models like Random Forest.

6. Model Implementation

Random Forest Classifier was chosen due to its ability to handle non-linear data and reduce overfitting. The model was trained using scikit-learn and achieved high accuracy in predicting the correct crop for given conditions.

7. Evaluation Metrics

- Accuracy: Evaluates the proportion of correct predictions.
- Precision: Ratio of true positives to predicted positives.
- Recall: Measures how many actual positive cases were correctly predicted.
- F1 Score: Harmonic mean of precision and recall.
- Confusion Matrix: Visualized using Seaborn heatmap to show model performance across all crop classes.

8. Results and Analysis

- The model achieved excellent accuracy (~98%).
- The confusion matrix showed minimal misclassifications.
- Model performance was consistent across various crop types.
- Precision and recall metrics confirmed robustness of predictions.

9. Conclusion

The Crop Recommendation System effectively leverages environmental and soil data to guide crop selection using machine learning. The Random Forest classifier demonstrated high performance, offering a promising tool for smart agriculture. Future work may include real-time API integration and broader geographic data to make predictions location-aware.

10. References

- scikit-learn documentation
- pandas documentation
- Seaborn visualization library
- Kaggle Crop Recommendation Dataset
- Research articles on smart farming and agri-tech AI systems

Code:

```python
# Step 1: Install Required Libraries
!pip install -q pandas numpy scikit-learn matplotlib seaborn
```

```python
# Step 2: Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

```python
# Step 3: Load Dataset (using your file path)
df = pd.read_csv("/content/Crop_recommendation dataset.csv")
df.head()
```

```python
# Step 4: Explore Dataset
print("Shape of dataset:", df.shape)
print("Columns:", df.columns.tolist())
print("\nMissing values:\n", df.isnull().sum())
```

```python
# Step 5: Visualize Feature Correlation (only numeric columns)
plt.figure(figsize=(10, 6))
numeric_df = df.select_dtypes(include=['float64', 'int64'])  # Only numeric columns
sns.heatmap(numeric_df.corr(), annot=True, cmap="YlGnBu")
plt.title("Feature Correlation Heatmap")
plt.show()
```

```python
# Step 6: Prepare Data
X = df.drop("label", axis=1)  # Features
y = df["label"]               # Target (crop label)

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Step 7: Train the Model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```python
# Step 8: Evaluate the Model
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Confusion Matrix
plt.figure(figsize=(12, 6))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap="Blues")
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

```python
# Step 9: Predict on New Input
# Example: [N, P, K, temperature, humidity, pH, rainfall]
sample = np.array([[90, 42, 43, 20.5, 82.0, 6.5, 202.0]])
predicted_crop = model.predict(sample)
print("Recommended Crop:", predicted_crop[0])
```

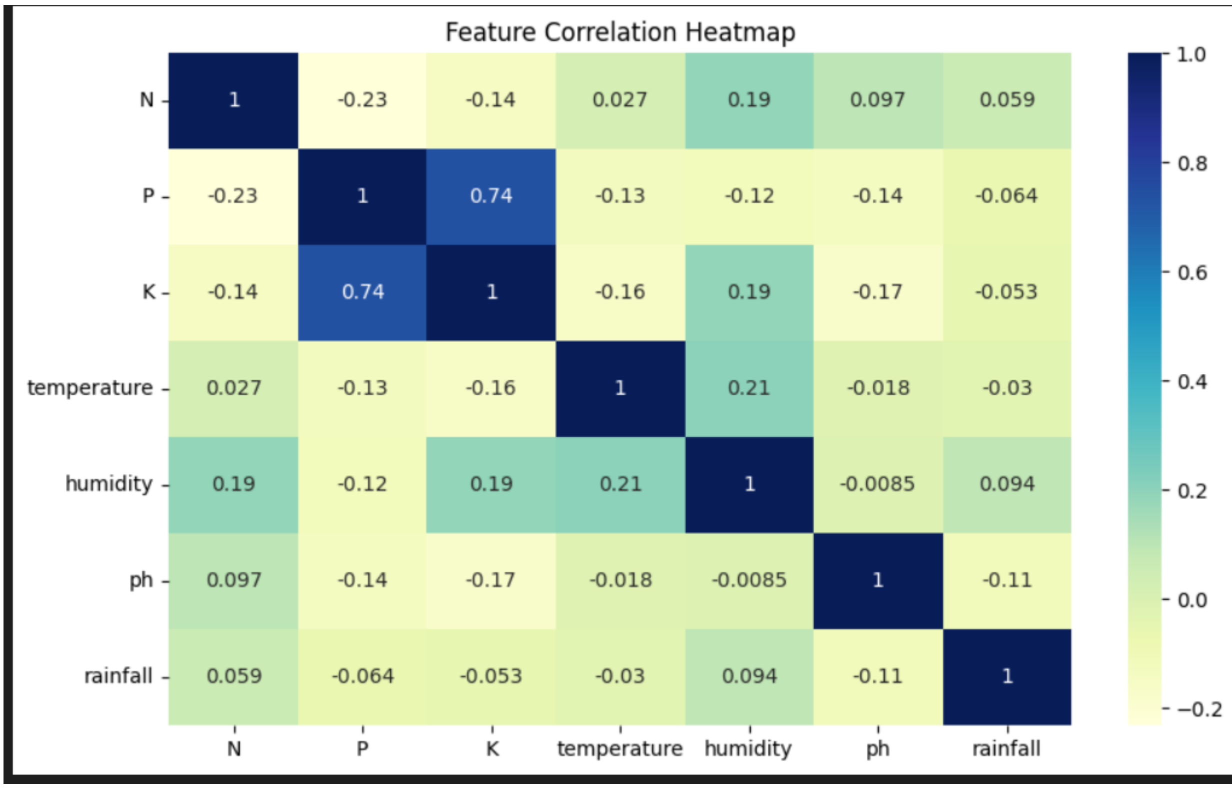
```
Recommended Crop: rice
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```

```python
# Step 10: Save the Model (Optional)
import joblib
joblib.dump(model, "crop_recommendation_model.pkl")
```

Output:

```
Shape of dataset: (2200, 8)
Columns: ['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label']

Missing values:
 N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```



```
Accuracy: 0.9931818181818182

Classification Report:
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        23
      banana       1.00      1.00      1.00        21
   blackgram       1.00      1.00      1.00        20
    chickpea       1.00      1.00      1.00        26
     coconut       1.00      1.00      1.00        27
      coffee       1.00      1.00      1.00        17
      cotton       1.00      1.00      1.00        17
      grapes       1.00      1.00      1.00        14
        jute       0.92      1.00      0.96        23
 kidneybeans       1.00      1.00      1.00        20
      lentil       0.92      1.00      0.96        11
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        19
   mothbeans       1.00      0.96      0.98        24
    mungbean       1.00      1.00      1.00        19
   muskmelon       1.00      1.00      1.00        17
      orange       1.00      1.00      1.00        14
      papaya       1.00      1.00      1.00        23
  pigeonpeas       1.00      1.00      1.00        23
 pomegranate       1.00      1.00      1.00        23
...
    accuracy                           0.99       440
   macro avg       0.99      0.99      0.99       440
weighted avg       0.99      0.99      0.99       440
```

```
Recommended Crop: rice
```