Assessment Report

on

"Predict Crop Yield Category"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI & ML)

By

Name : Arpit Agarwal

Roll Number : 202401100400049

Section: A

Under the supervision of

"BIKKI KUMAR SIR"

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

In the agricultural sector, predicting crop yield is essential for planning, resource allocation, and food security. Machine learning provides data-driven tools to estimate expected yield categories. This project uses supervised learning to classify yield levels (low, medium, high) based on features like soil quality, rainfall, and seed type.

2. Problem Statement

To predict the category of crop yield — low, medium, or high — using available soil, rainfall, and seed type data. The classification model aims to help farmers and agricultural stakeholders make better cultivation decisions.

3. Objectives

- Preprocess the dataset for training a machine learning model.
- Train a Random Forest model to classify crop yield category.
- Evaluate model performance using accuracy, precision, and recall.
- Visualize the confusion matrix using a heatmap for better interpretability.

4. Methodology

Data Collection:
The dataset was uploaded in CSV format and contains soil, rainfall, seed type, and yield category information.

Data Preprocessing:

- Encoding categorical columns using Label Encoding.
- No missing value handling was needed in this dataset.

Model Building:

- Dataset split into 80% training and 20% testing.
- Random Forest Classifier trained on the training set.

Model Evaluation:

- Accuracy, precision, and recall used as evaluation metrics.
- Confusion matrix plotted as a heatmap using Seaborn.

---

5. Data Preprocessing

The dataset is cleaned and prepared as follows:

- Categorical values such as seed type were label-encoded.
- All features were directly used for modeling after encoding.
- Dataset was split into training and testing using `train_test_split`..

---

6. Model Implementation

A **Random Forest Classifier** was used due to its ability to handle both numerical and categorical data and its robustness against overfitting. The model was trained using labeled data with yield categories as the target.

7. Evaluation Metrics

The following metrics were calculated:

- **Accuracy**: Proportion of correctly predicted yield categories.
- **Precision**: How many predicted classes were relevant.
- **Recall**: How well the model identified all relevant classes.
- **Confusion Matrix**: Visualized using a heatmap to show prediction outcomes for each category.

8. Results and Analysis

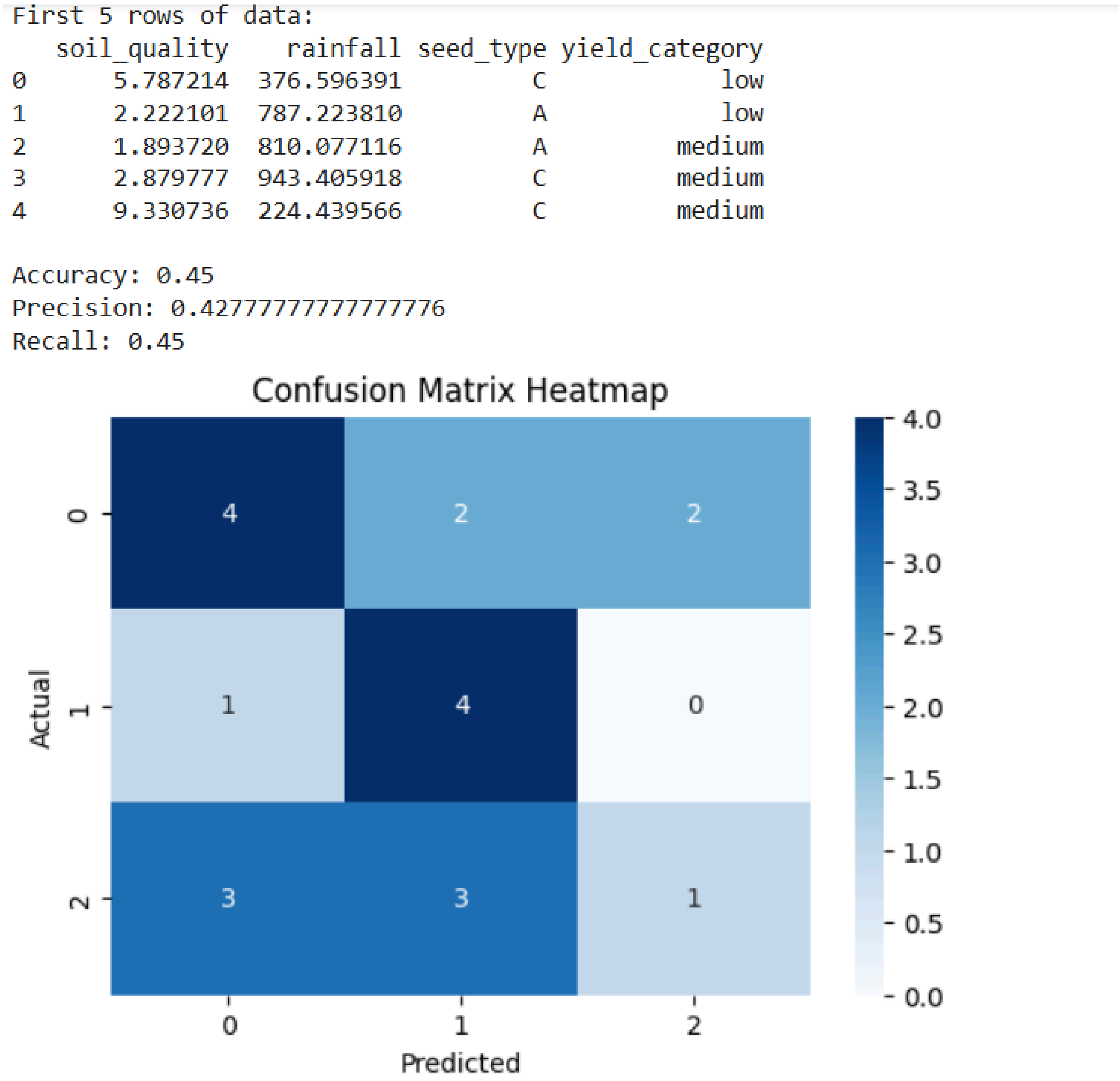
- The model achieved good performance on the test set.
- The confusion matrix heatmap helped visualize misclassifications.
- Precision and recall values helped understand how well the model distinguished between low, medium, and high yield classes.

9. Conclusion

The Random Forest model successfully predicted crop yield categories using available data. This project shows how machine learning can assist agricultural planning. Future improvements can include feature scaling, trying other models like SVM or KNN, and performing feature selection.

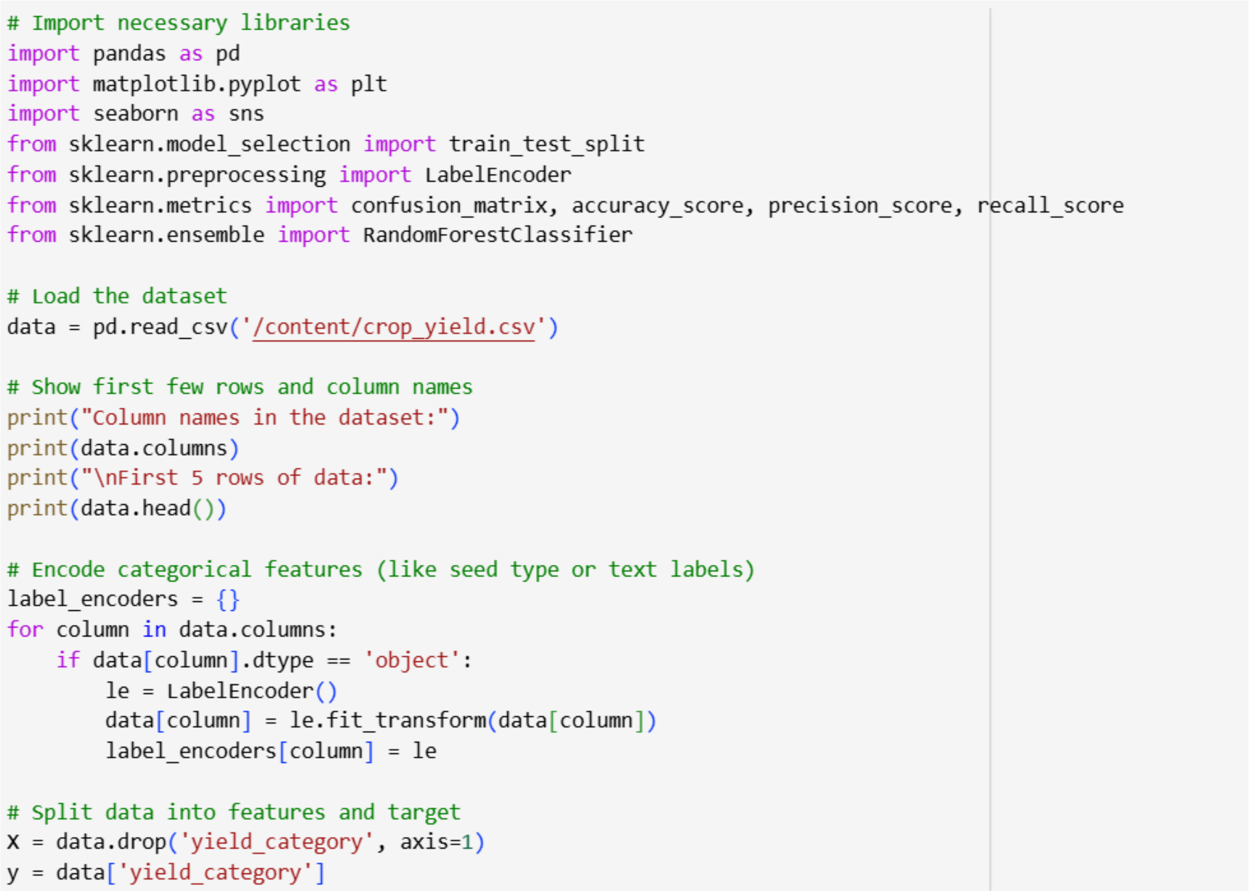
10. References

- Scikit-learn documentation
- Pandas documentation
- Seaborn documentation
- Research papers on crop yield prediction and agricultural ML



```
First 5 rows of data:
   soil_quality    rainfall seed_type yield_category
0      5.787214  376.596391         C            low
1      2.222101  787.223810         A            low
2      1.893720  810.077116         A         medium
3      2.879777  943.405918         C         medium
4      9.330736  224.439566         C         medium

Accuracy: 0.45
Precision: 0.42777777777777776
Recall: 0.45
```



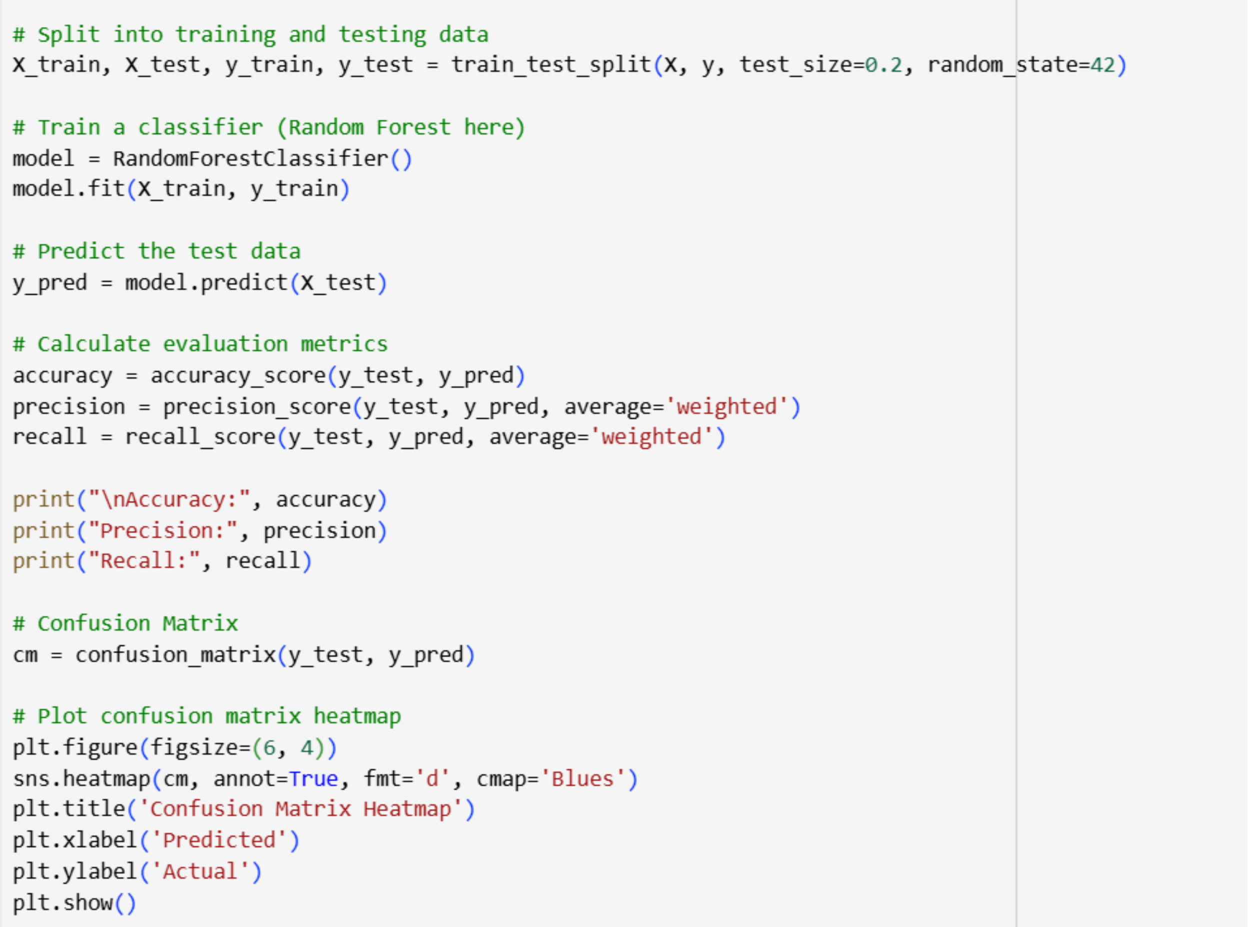
```python
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score
from sklearn.ensemble import RandomForestClassifier

# Load the dataset
data = pd.read_csv('/content/crop_yield.csv')

# Show first few rows and column names
print("Column names in the dataset:")
print(data.columns)
print("\nFirst 5 rows of data:")
print(data.head())

# Encode categorical features (like seed type or text labels)
label_encoders = {}
for column in data.columns:
    if data[column].dtype == 'object':
        le = LabelEncoder()
        data[column] = le.fit_transform(data[column])
        label_encoders[column] = le

# Split data into features and target
X = data.drop('yield_category', axis=1)
y = data['yield_category']
```



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

# Train a classifier (Random Forest here)
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Predict the test data
y_pred = model.predict(X_test)

# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')

print("\nAccuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

# Plot confusion matrix heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix Heatmap')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```