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)

By

Name : Divya Pal

Roll Number : 202401100300103

Section: B

Under the supervision of

"MR. SHIVANSH PRASAD"

KIET Group of Institutions, Ghaziabad

April, 2025

1. Introduction

Agriculture is one of the most critical sectors in any economy. Accurate crop yield prediction can significantly help farmers and policymakers make informed decisions about resource allocation, crop selection, and food security planning. This project aims to classify the crop yield into categories — **Low**, **Medium**, or **High** — using features like **soil quality**, **rainfall**, and **seed type** through a machine learning model.

---

2. Problem Statement

The problem is to predict the **crop yield category** (low, medium, or high) using key agricultural inputs like **soil quality**, **rainfall**, and **seed type**. The objective is to build a machine learning model that can accurately classify yield levels to support better farming decisions.

---

3. Methodology

- **Dataset**

The dataset consists of:

- **Soil_quality**: A numeric value indicating fertility
- **Rainfall**: Amount of rainfall received (in mm)
- **Seed_type**: Type of seed used (categorical: A, B, C)
- **Yield_category**: Target variable (low, medium, high)

- **Preprocessing Steps**

1. **Label Encoding**: Convert text labels into numbers.
2. **Feature Scaling**: Normalize numerical inputs.
3. **Train-Test Split**: Use 80% of data for training and 20% for testing.
4. **Model Selection**: Use Random Forest Classifier for prediction.

---

4. Code Used:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from google.colab import files

# Step 1: Upload and load data
print("Please upload your file")
uploaded = files.upload()
df = pd.read_csv("crop_yield.csv")

# Step 2: Encode categorical data
le_seed = LabelEncoder()
df['seed_type'] = le_seed.fit_transform(df['seed_type'])

le_yield = LabelEncoder()
df['yield_category'] = le_yield.fit_transform(df['yield_category'])

# Step 3: Split into features and target
X = df.drop('yield_category', axis=1)
```

```
y = df['yield_category']


# Step 4: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Step 5: Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)


# Step 6: Train model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)


# Step 7: Evaluate model
y_pred = model.predict(X_test_scaled)
print(classification_report(y_test, y_pred, target_names=le_yield.classes_))
```

---

5. Output:

```
please upload your file
Choose Files crop_yield.csv
• crop_yield.csv(text/csv) - 4376 bytes, last modified: 4/22/2025 - 100% done
Saving crop_yield.csv to crop_yield (1).csv
              precision    recall  f1-score   support

        high       0.50      0.50      0.50         8
         low       0.40      0.80      0.53         5
      medium       0.50      0.14      0.22         7

    accuracy                           0.45        20
   macro avg       0.47      0.48      0.42        20
weighted avg       0.47      0.45      0.41        20
```

---

6. Results and Analysis

- **Accuracy**: 45%
- The model performs reasonably well for the low category.
- **Poor performance** for medium and high suggests overlapping features or insufficient data.
- **Recommendations**:
    - Use more features (like temperature, humidity, crop type).
    - Try models like XGBoost or tune Random Forest.
    - Balance the dataset if one category has fewer examples.

---

7. Conclusion

Machine learning offers powerful tools for predicting agricultural outcomes. While this basic model demonstrates potential, enhancing the dataset and trying advanced models could yield better results and provide real-world decision-making support for farmers and agricultural planners.

---

8. References

- Scikit-learn Documentation
- Pandas Documentation: https
- Random Forest Classifier Theory
- Label Encoding & Standard Scaling

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

from google.colab import files
print("please upload your file")
uplaoded =files.upload()
df = pd.read_csv("crop_yield.csv")

# 2. Convert 'seed_type' and 'yield_category' from text to numbers
le_seed = LabelEncoder()
df['seed_type'] = le_seed.fit_transform(df['seed_type'])  # A, B, C -> 0, 1, 2

le_yield = LabelEncoder()
df['yield_category'] = le_yield.fit_transform(df['yield_category'])  # low, medium, high -> 0, 1, 2

# 3. Split into input (X) and output (y)
X = df.drop('yield_category', axis=1)
y = df['yield_category']

# 4. Split into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 5. Scale features to have similar range
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 6. Train a Random Forest classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 6. Train a Random Forest classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)

# 7. Make predictions on the test set
y_pred = model.predict(X_test_scaled)

# 8. Show how well it did
print(classification_report(y_test, y_pred, target_names=le_yield.classes_))
```

```
please upload your file
Choose Files crop_yield.csv
```

- **crop_yield.csv**(text/csv) - 4376 bytes, last modified: 4/22/2025 - 100% done

```
Saving crop_yield.csv to crop_yield (1).csv
              precision    recall  f1-score   support

        high       0.50      0.50      0.50         8
         low       0.40      0.80      0.53         5
      medium       0.50      0.14      0.22         7

    accuracy                           0.45        20
   macro avg       0.47      0.48      0.42        20
weighted avg       0.47      0.45      0.41        20
```