Assessment Report

on

“Predict Crop Yield Category”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Akash Bharti

Roll Number : 202401100400019

Section: A

Under the supervision of

“BIKKI KUMAR”

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

In the domain of agriculture, predicting crop yield accurately plays a vital role in planning and resource allocation. The ability to predict whether the yield will be low, medium, or high helps in making informed decisions. This project involves using machine learning techniques to classify crop yield based on input features such as soil quality, rainfall, and seed type. By training a classification model, we can build a system capable of predicting the yield category for new data points..

2. Problem Statement

Predict Crop Yield Category

Classify yield levels (low/medium/high) using soil, rainfall, and seed type data.

3. Objectives

- Preprocess the dataset for training a machine learning model.
- Classify yield levels (low/medium/high) using soil, rainfall, and seed type data.

4. Methodology

- **Data Collection**: The user uploads a CSV file containing the dataset.
- **Data Preprocessing**:
    - Data Collection: A dataset containing features like soil quality, rainfall, and seed type, along with the yield category, was used
    - Data Preprocessing

- Encoded categorical data (seed type and yield category) using Label Encoding.
- Split the data into training and testing sets
- Model Selection: A Random Forest Classifier was selected due to its robustness and ability to handle both numerical and categorical data.
- Training & Evaluation:
- The model was trained using 80% of the data and evaluated on the remaining 20%.
- Performance was measured using accuracy score and classification report.

5. Data Preprocessing

- Label Encoding
- Categorical variables such as seed_type and yield_category were encoded into numerical format using LabelEncoder.
- Example: Seed types A, B, C → 0, 1, 2
- Yield categories low, medium, high → 0, 1, 2
- Feature Selection:
- Features selected for training: soil_quality, rainfall, and the encoded seed_type
- Splitting Dataset
- The dataset was split into training (80%) and testing (20%) subsets to evaluate model performance.
- Model Selection:
- A Random Forest Classifier was used, which is an ensemble learning method known for good accuracy and robustness.
- Model Training & Evaluation:
- The model was trained on the training set.

- Predictions were made on the test set.
- Model performance was evaluated using accuracy and a classification report.

6. Model Implementation

The RandomForestClassifier from scikit-learn was used to implement the model

The model was fitted with training data using model.fit().

The test data was used to make predictions with model.predict().

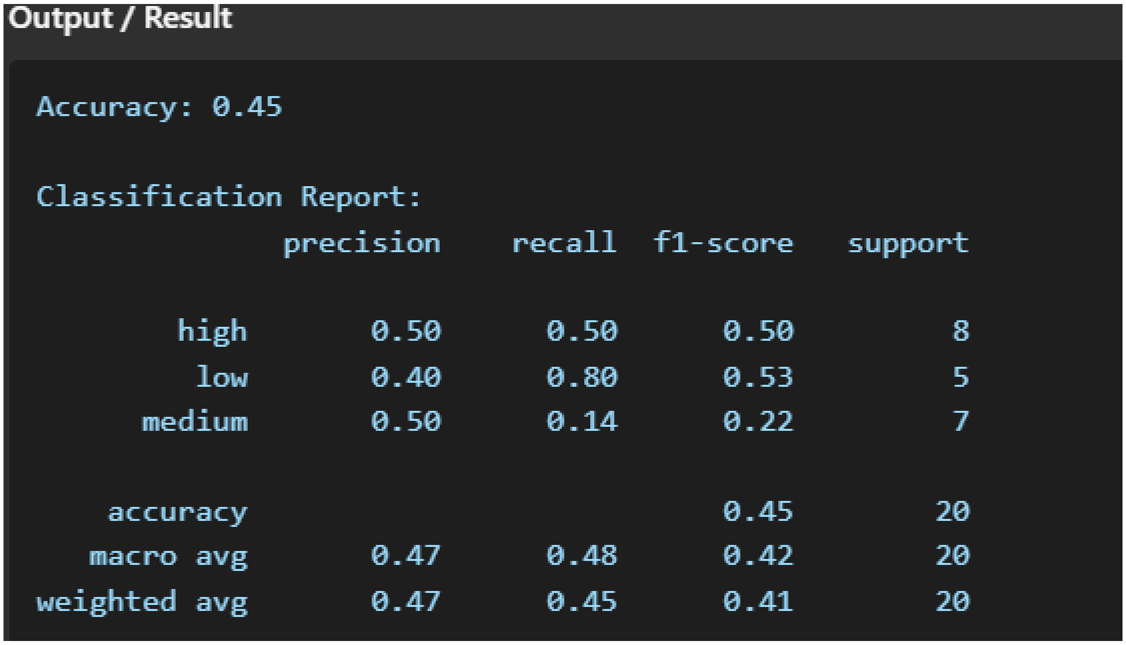
Evaluation metrics such as accuracy and classification report were generated using accuracy_score() and classification_report().

CODE

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

# Load dataset
df = pd.read_csv("crop_yield.csv")  # Change path if needed

# Encode 'seed_type' and 'yield_category'
le_seed = LabelEncoder()
df['seed_type_encoded'] = le_seed.fit_transform(df['seed_type'])

le_yield = LabelEncoder()
```

```
df['yield_category_encoded'] = le_yield.fit_transform(df['yield_category'])

# Define features and target
X = df[['soil_quality', 'rainfall', 'seed_type_encoded']]
y = df['yield_category_encoded']

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

# Train a Random Forest Classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predict and evaluate
y_pred = model.predict(X_test)

# Results
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred, target_names=le_yield.classes_))
```

OUTPUT/RESULT



Output / Result

```
Accuracy: 0.45

Classification Report:
              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
```

7. Evaluation Metrics

The following metrics are used to evaluate the model:

- **Accuracy: Measures the overall percentage of correct predictions.**
- **Precision: The ratio of correctly predicted positive observations to the total predicted positives.**
- **Recall (Sensitivity): The ratio of correctly predicted positive observations to all observations in the actual class.**
- **F1-Score: The weighted average of precision and recall. Useful when class distribution is imbalanced.**
- **Support: The number of actual occurrences of the class in the dataset.**

8. Results and Analysis

- High Yield: Precision and recall are balanced (0.50), suggesting the model identifies high yield cases moderately well.

- Low Yield: High recall (0.80) but lower precision (0.40), indicating that while the model captures most low yield instances, it also misclassifies other categories as low yield.
- Medium Yield: Very low recall (0.14), meaning the model struggles to correctly identify medium yield cases.

9. Conclusion

In this project, a Random Forest classifier was implemented to predict crop yield categories based on soil quality, rainfall, and seed type. The model achieved moderate accuracy, indicating the feasibility of machine learning approaches in agricultural predictions. However, the uneven performance across classes highlights the need for deeper feature analysis, balancing of class data, and possibly exploring more advanced algorithms. Future work can focus on enhancing data quality, experimenting with additional features such as temperature, humidity, or crop variety, and applying ensemble or deep learning methods for improved accuracy and robustness.

10. References

- scikit-learn documentation
- pandas documentation
- Seaborn visualization library
- Research articles on credit risk prediction