Project Title: Crop Yield Prediction using Classification Techniques

Problem Statement: To classify crop yield levels (low, medium, high) based on soil quality, rainfall, and seed type data.

Name: Akash Bharti
Roll No: 202401100400019
Course: Introduction to AI
Institution: KIET Group of Institution

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.

Methodology

1. **Data Collection**: A dataset containing features like soil quality, rainfall, and seed type, along with the yield category, was used.
2. **Data Preprocessing**:
    - Encoded categorical data (seed type and yield category) using Label Encoding.
    - Split the data into training and testing sets.
3. **Model Selection**: A Random Forest Classifier was selected due to its robustness and ability to handle both numerical and categorical data.
4. **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.

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
```

References / Credits

- Scikit-learn Documentation: https://scikit-learn.org/
- Pandas Documentation: https://pandas.pydata.org/
- Dataset provided as part of coursework / project.
- Developed using Python 3 and Jupyter Notebook / IDE of choice.