## 📄 ****Crop Yield Prediction - Report Structure****

### 1. ****Title Page****

**Title of the Assignment** (Crop Yield Prediction using Machine Learning)

AYAAN VASHU

CSEAI (B)

### 2. ****Introduction****

In this report, we aim to predict the crop yield category (low, medium, high) based on three input variables: soil quality, rainfall, and seed type. These predictions can help in improving crop management and forecasting crop yield.

To achieve this, we will use two machine learning techniques:

1. \*\*Supervised Learning (Random Forest Classifier)\*\*: For predicting the yield category (low, medium, high) based on labeled data.2. \*\*Unsupervised Learning (KMeans Clustering)\*\*: For identifying natural clusters or patterns in the data that may not be immediately obvious.

The dataset contains information on soil quality, rainfall, and seed type, with the yield category as the target variable.

### 3. ****Methodology****

· **Supervised Learning (Random Forest Classifier)**:

A decision tree-based model used for **classification** tasks (Low, Medium, High yield categories).

We trained the model with historical data, where the input features (soil quality, rainfall, seed type) predict the target variable (yield category).

**Why Random Forest?**

Random Forest is a robust algorithm that handles both classification and regression problems. It works well with **non-linear** data and reduces overfitting by averaging multiple decision trees.

· **Unsupervised Learning (KMeans Clustering)**:

KMeans clustering is used to find natural groupings or **clusters** in the data. We used it to identify potential patterns or groups based on soil quality, rainfall, and seed type, without predefined labels (like Low/Medium/High yield).

### 3.CODE

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.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix

from sklearn.cluster import KMeans

from sklearn.decomposition import PCA

from google.colab import files

uploaded = files.upload()

import io

df = pd.read\_csv(io.BytesIO(list(uploaded.values())[0]))

df.head()

# Encode categorical variables

le\_seed = LabelEncoder()

df['seed\_type\_encoded'] = le\_seed.fit\_transform(df['seed\_type'])

le\_yield = LabelEncoder()

df['yield\_encoded'] = le\_yield.fit\_transform(df['yield\_category'])

# Feature scaling

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(df[['soil\_quality', 'rainfall', 'seed\_type\_encoded']])

y = df['yield\_encoded']

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

# Train the Random Forest Classifier

clf = RandomForestClassifier(n\_estimators=100, random\_state=42)

clf.fit(X\_train, y\_train)

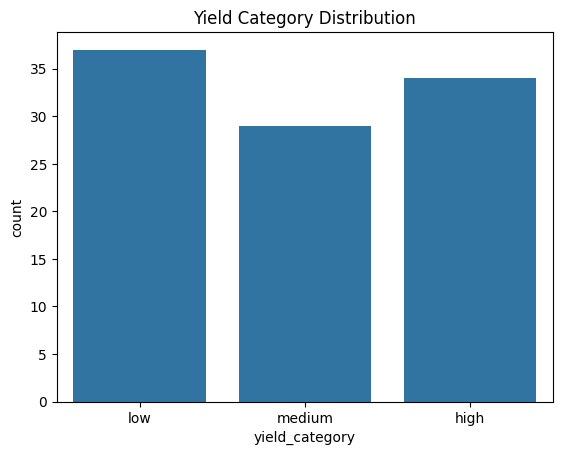
# Predict and evaluate

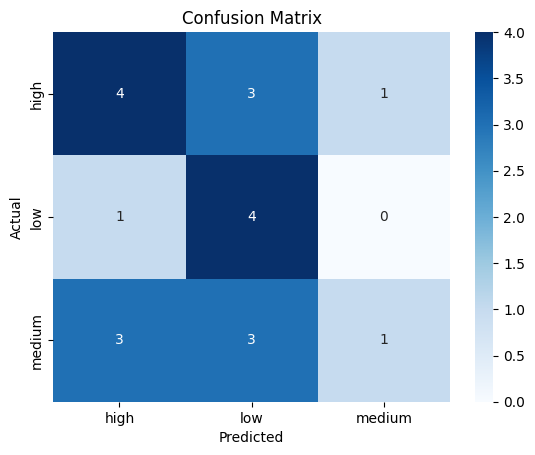
y\_pred = clf.predict(X\_test)

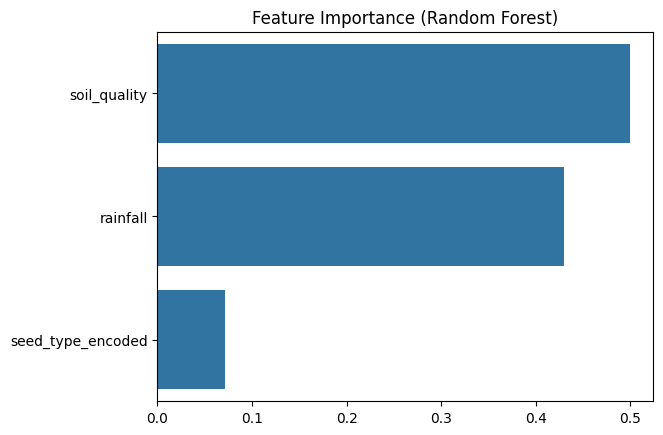
print(accuracy\_score(y\_test, y\_pred))

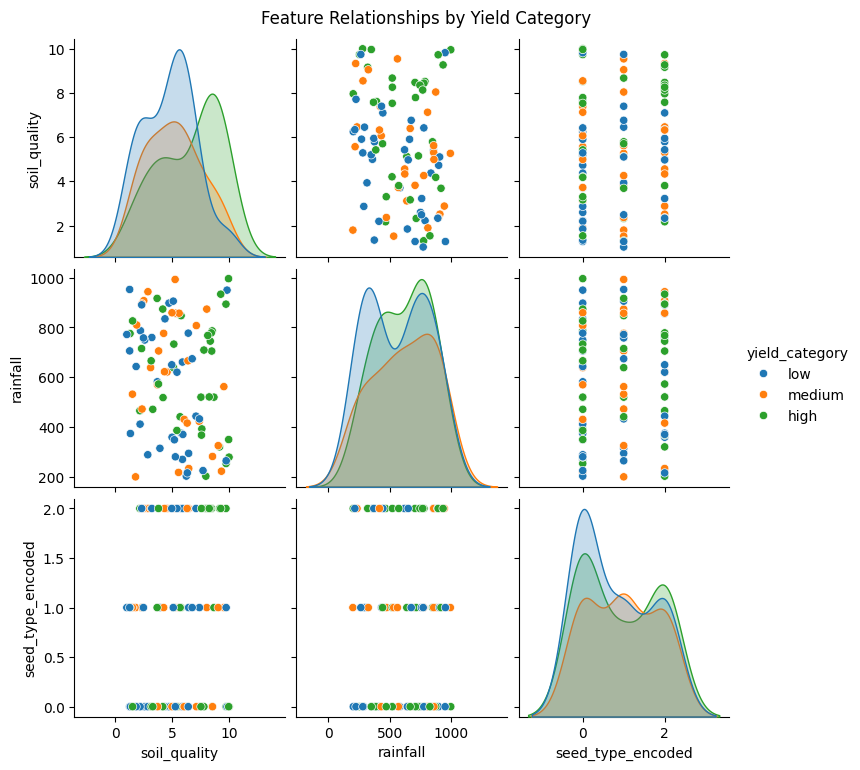
print(classification\_report(y\_test, y\_pred))

### 4.IMAGES and OUTPUTS









### 5. ****REFERENCE****

<https://github.com/VashuKiet/Crop-Yield-Prediction>

https://colab.research.google.com/drive/1zlIeW1sRt5tEhJeUpete5KAtq66ZgF8l?usp=sharing