**Agricultural Yield Analysis & Prediction**

🎯 Objective:

To analyze agricultural yield data, evaluate the influence of key factors like soil quality, water availability, and fertilizer use, and predict crop yield using machine learning techniques.

---

🧰 Tools & Libraries:

pandas: Data manipulation

numpy: Numerical operations

matplotlib, seaborn: Visualization

sklearn: Machine learning models, data splitting, metrics

---

📊 Dataset Summary:

✅ Sample Dataset Features:

Farm\_ID: Unique identifier for each farm

Crop\_Type: Type of crop (Wheat or Corn)

Soil\_Quality: Rating from 1–10

Water\_Availability: Water access level

Fertilizer\_Usage: Amount of fertilizer used

Yield: Crop yield output

✅ Complete Dataset (from CSV):

Additional columns such as Rainfall are included

Realistic agricultural features affecting yield

---

📌 Step-by-Step Process:

---

🔍 Step 1: Data Creation / Loading

Created a sample dataset manually

Later loaded a more detailed dataset (complete\_agricultural\_yield\_data.csv) using pandas

---

📈 Step 2: Exploratory Data Analysis (EDA)

🔹 Summary Statistics:

Mean, min, max, std deviation computed

Provided insight into feature ranges

🔹 Correlation Matrix:

Heatmap showed strong positive correlations between Soil\_Quality, Fertilizer\_Usage, and Yield

In full dataset: Rainfall also showed positive correlation with Yield

🔹 Boxplot: Yield by Crop Type

Compared Yield for Wheat vs Corn

Identified Corn as generally having higher yield

🔹 Scatterplots:

Soil Quality vs Yield: Showed upward trend, especially for Corn

Rainfall vs Yield (in extended dataset): Positive correlation, rainfall influenced yield more in Corn

---

🔧 Step 3: Predictive Modeling

🔹 Model Used:

Linear Regression

🔹 Input Features (X):

Soil\_Quality, Water\_Availability, Fertilizer\_Usage

🔹 Target Variable (y):

Yield

🔹 Model Training & Evaluation:

Dataset split: 80% training, 20% testing

Model trained and tested

Mean Squared Error (MSE) reported for prediction accuracy

Scatter plot of actual vs predicted yield values visualized model performance

---

💾 Step 4: Dataset Export

Final dataset saved as 'agricultural\_yield\_data.csv' for reuse or reporting

---

✅ Key Insights:

Soil Quality and Fertilizer Usage are strong indicators of crop yield

Corn tends to outperform Wheat in yield under similar conditions

Rainfall also plays a significant role (as shown in the extended dataset)

Linear Regression provided a good baseline model for predicting yield

---

📌 Use Cases & Recommendations:

For Farmers: Optimize soil and water management for better yield

For Government/Agencies: Use prediction models to forecast food supply

Future Work:

Test more advanced models like Random Forest, XGBoost

Include more environmental features (e.g., temperature, humidity)

Use time-series data for seasonal trend analysis

---

# Import necessary libraries

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

# Step 1: Create Sample Dataset

print("Step 1: Creating Sample Dataset...")

data = {

'Farm\_ID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

'Crop\_Type': ['Wheat', 'Corn', 'Wheat', 'Corn', 'Wheat', 'Corn', 'Wheat', 'Corn', 'Wheat', 'Corn'],

'Soil\_Quality': [5, 7, 6, 8, 5, 7, 6, 8, 5, 7],

'Water\_Availability': [3, 4, 3, 4, 3, 4, 3, 4, 3, 4],

'Fertilizer\_Usage': [2, 3, 2, 3, 2, 3, 2, 3, 2, 3],

'Yield': [10, 15, 12, 16, 11, 14, 13, 17, 12, 15]

}

# Create DataFrame

df = pd.DataFrame(data)

print("\nSample Dataset:")

print(df)

# Step 2: Data Analysis

print("\nStep 2: Performing Data Analysis...")

# Summary statistics

print("\nSummary Statistics:")

print(df.describe())

# Correlation matrix

print("\nCalculating Correlation Matrix...")

corr\_matrix = df.corr()

plt.figure(figsize=(8, 6))

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Matrix')

plt.show()

# Yield distribution by Crop Type

print("\nPlotting Yield Distribution by Crop Type...")

plt.figure(figsize=(8, 6))

sns.boxplot(x='Crop\_Type', y='Yield', data=df)

plt.title('Yield Distribution by Crop Type')

plt.show()

# Yield vs Soil Quality

print("\nPlotting Yield vs Soil Quality...")

plt.figure(figsize=(8, 6))

sns.scatterplot(x='Soil\_Quality', y='Yield', hue='Crop\_Type', data=df)

plt.title('Yield vs Soil Quality')

plt.show()

# Step 3: Predictive Modeling

print("\nStep 3: Performing Predictive Modeling...")

# Prepare data for modeling

X = df[['Soil\_Quality', 'Water\_Availability', 'Fertilizer\_Usage']]

y = df['Yield']

# Split the data

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

# Train the model

print("\nTraining Linear Regression Model...")

model = LinearRegression()

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

# Predict

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

# Evaluate the model

mse = mean\_squared\_error(y\_test, y\_pred)

print(f'\nMean Squared Error: {mse}')

# Plot actual vs predicted

print("\nPlotting Actual vs Predicted Yield...")

plt.figure(figsize=(8, 6))

plt.scatter(y\_test, y\_pred)

plt.xlabel('Actual Yield')

plt.ylabel('Predicted Yield')

plt.title('Actual vs Predicted Yield')

plt.show()

# Step 4: Save the Dataset

print("\nStep 4: Saving Dataset...")

df.to\_csv('agricultural\_yield\_data.csv', index=False)

print("Dataset saved as 'agricultural\_yield\_data.csv'.")

print("\nProgram completed successfully!")

Data set

You can use the dataset to perform the following analyses:

Correlation Analysis: Check how different factors (e.g., Soil\_Quality, Rainfall) correlate with Yield.

Crop Comparison: Compare yields for different crop types (e.g., Wheat vs Corn).

Predictive Modeling: Train a machine learning model to predict Yield based on other features.

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Load the dataset

df = pd.read\_csv('complete\_agricultural\_yield\_data.csv')

# Display the dataset

print("Dataset Loaded:")

print(df)

# Correlation matrix

plt.figure(figsize=(10, 8))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title('Correlation Matrix')

plt.show()

# Yield distribution by Crop Type

plt.figure(figsize=(8, 6))

sns.boxplot(x='Crop\_Type', y='Yield', data=df)

plt.title('Yield Distribution by Crop Type')

plt.show()

# Yield vs Rainfall

plt.figure(figsize=(8, 6))

sns.scatterplot(x='Rainfall', y='Yield', hue='Crop\_Type', data=df)

plt.title('Yield vs Rainfall')

plt.show()