COLLEGE CODE:2124

COLLEGE NAME:SSIET

DEPARTMENT:AI & DS

STUDENT NM-ID:aut9629969265

ROLL NO:212423243014

DATE:14/05/2025

TECHNOLOGY - PROJECT NAME: Production Yield Analysis

SUBMITTED BY: B.Kishore

Phase 5: Project Demonstration & Documentation

Title: Production Yield Analysis

Abstract:

The Production Yield Analysis project focuses on improving agricultural productivity by analyzing crop yield data using data science techniques. By integrating statistical analysis, machine learning models, and IoT-based data collection, the project offers predictive insights into yield performance under different environmental and cultivation conditions. This document covers the complete project cycle — from data collection and preprocessing to model development, system evaluation, and deployment. The system is designed for farmers and agricultural analysts to make data-driven decisions for maximizing crop output.

Index

Project Demonstration:

Overview:

The system demonstrates how agricultural yield prediction is performed using real-time and historical data, emphasizing features like weather influence, soil condition, crop type, and pest presence.

Demonstration Details:

- ❖ System Walkthrough: From data upload to yield prediction using an interactive dashboard or Jupyter Notebook.
- ❖ Model Performance: Showcase prediction accuracy of various machine learning models (e.g., Linear Regression, Random Forest, XGBoost).
- ❖ Visualization: Charts and graphs showing yield trends, seasonal effects, and region-based output.
- ❖ IoT Data Integration (if any): Simulated or real sensor data (e.g., soil moisture, temperature) used for yield prediction.
- ❖ Performance Metrics: RMSE, R^2 Score, and model comparison.

Outcome:

Stakeholders will understand how yield can be forecasted and influenced using real data, aiding in better farming decisions.

Project Documentation:

Overview:

Detailed documentation includes system architecture, data pipeline, ML model selection, evaluation metrics, and user guides.

Documentation Sections:

- ❖ System Architecture: Data flow diagram and component interaction.
- ❖ Codebase Overview: Python scripts used for data preprocessing, model training, and visualization.
- ❖ Dataset Information: Description of datasets used (crop data, climate data, etc.).
- ❖ User Guide: Steps to use the prediction tool.
- ❖ Testing Reports: Accuracy results, confusion matrix, and validation techniques.

Outcome:

Complete clarity on how the system functions, enabling future updates or expansions.

Feedback and Final Adjustments:

Overview:

Post-demonstration feedback is collected from mentors and users for final tuning.

Steps:

1. **Feedback Collection:** Via forms and interviews.
2. **Refinements:** Algorithm tuning, UI changes, or data updates.
3. **Final Testing:** Ensure prediction accuracy and stability.

Outcome:

Final refinements make the system more robust and ready for deployment or academic evaluation.

Final Project Report Submission:

Overview:

Summarizes all development phases, achievements, and lessons learned.

Report Sections:

1. **Executive Summary:** Goal and outcomes of the project.
2. **Phase-wise Progress:** From data acquisition to final model deployment.
3. **Challenges Faced:** Data quality, model overfitting, feature engineering, etc.
4. **Final Output:** System readiness for real use in agriculture.

Outcome:

Provides a well-rounded understanding of the project's impact and scalability.

Project Handover and Future Works:

Overview:

Prepares the project for future continuation or enhancements.

Handover Details:

1. Source Code and Dataset
2. Suggestions for Future Work:
   - 2.1. Use of real-time IoT sensors
   - 2.2. Integration with mobile apps for farmer use
   - 2.3. Multilingual support
   - 2.4. Expansion to other crops and regions

Outcome:

The project is ready for handover with clear next steps outlined for future teams.

Python code:

```
# production_yield_model.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Load dataset
df = pd.read_csv("/content/drive/MyDrive/Data set/crop_yield_50000.csv")

# Exploratory Data Analysis
print("Dataset Head:")
print(df.head())

print("\nDataset Info:")
print(df.info())

print("\nSummary Statistics:")
print(df.describe())

print("\nMissing Values:")
print(df.isnull().sum())
```

```
# Feature Selection

features = ['Temperature', 'Rainfall', 'Soil_Type_Index', 'Humidity']

X = df[features]

y = df['Yield']


# Train-test split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Model Training

model = RandomForestRegressor(n_estimators=100, random_state=42)

model.fit(X_train, y_train)


# Predictions & Evaluation

y_pred = model.predict(X_test)

print("\nModel Performance:")

print("R² Score:", r2_score(y_test, y_pred))


# Calculate MSE and then take the square root for RMSE

mse = mean_squared_error(y_test, y_pred)

rmse = np.sqrt(mse)

print("RMSE:", rmse)

# Correlation matrix

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

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

plt.title("Correlation Matrix")

plt.tight_layout()
```

```
plt.show()

# Feature Importance
importances = model.feature_importances_
plt.figure()
sns.barplot(x=importances, y=features)
plt.title("Feature Importance in Crop Yield Prediction")
plt.xlabel("Importance")
plt.ylabel("Feature")
plt.show()
```

OUTPUT:

```
Dataset Head:
   Temperature    Rainfall  Soil_Type_Index   Humidity  Yield
0    29.490142  154.910307                1  64.896762    5.5
1    27.585207  146.794589                3  69.709396    5.5
2    29.943066  197.589538                1  66.261862    5.5
3    32.569090  226.641557                4  72.208868    5.5
4    27.297540  184.342347                3  77.597758    5.5

Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
```

```
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Temperature      50000 non-null  float64
 1   Rainfall         50000 non-null  float64
 2   Soil_Type_Index  50000 non-null  int64
 3   Humidity         50000 non-null  float64
 4   Yield            50000 non-null  float64
dtypes: float64(4), int64(1)
memory usage: 1.9 MB
None

Summary Statistics:
        Temperature      Rainfall  Soil_Type_Index      Humidity         Yield
count  50000.000000  50000.000000     50000.000000  50000.000000  50000.000000
mean      27.998745    150.115050         1.994480     70.057963      5.499962
std        3.000475     49.967574         1.413495      9.976113      0.006084
min       15.000000      0.000000         0.000000     30.000000      4.395403
25%       25.964124    116.505976         1.000000     63.323258      5.500000
50%       28.005218    150.185986         2.000000     70.028444      5.500000
75%       30.031181    183.843078         3.000000     76.833355      5.500000
max       41.437253    300.000000         4.000000    100.000000      5.500000
```

Missing Values:

Temperature 0

Rainfall 0

Soil_Type_Index 0

Humidity 0

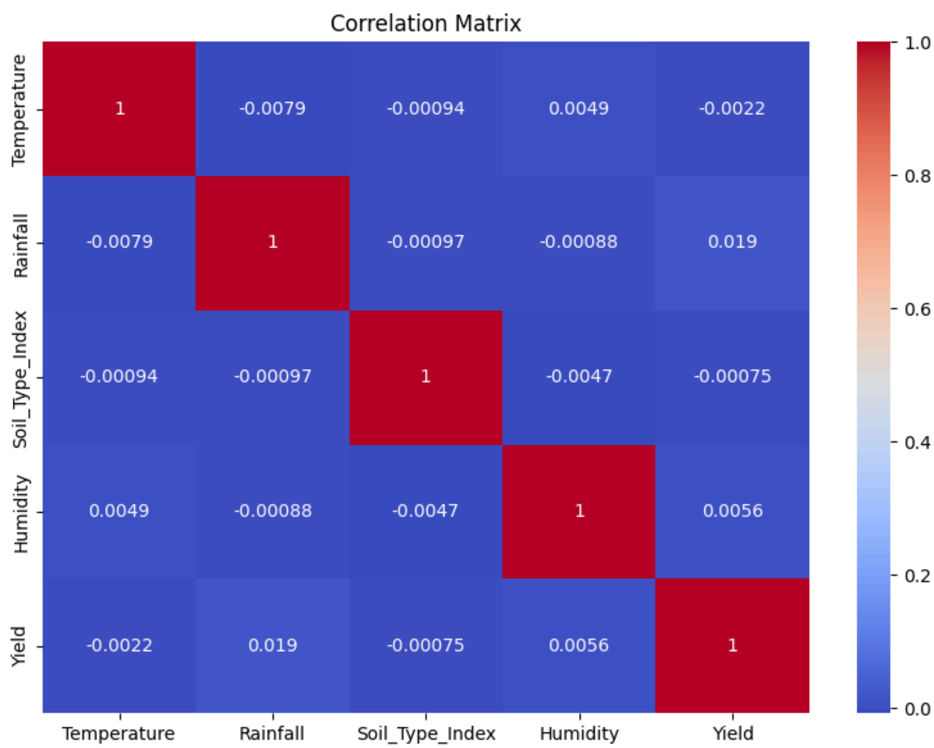
Yield 0

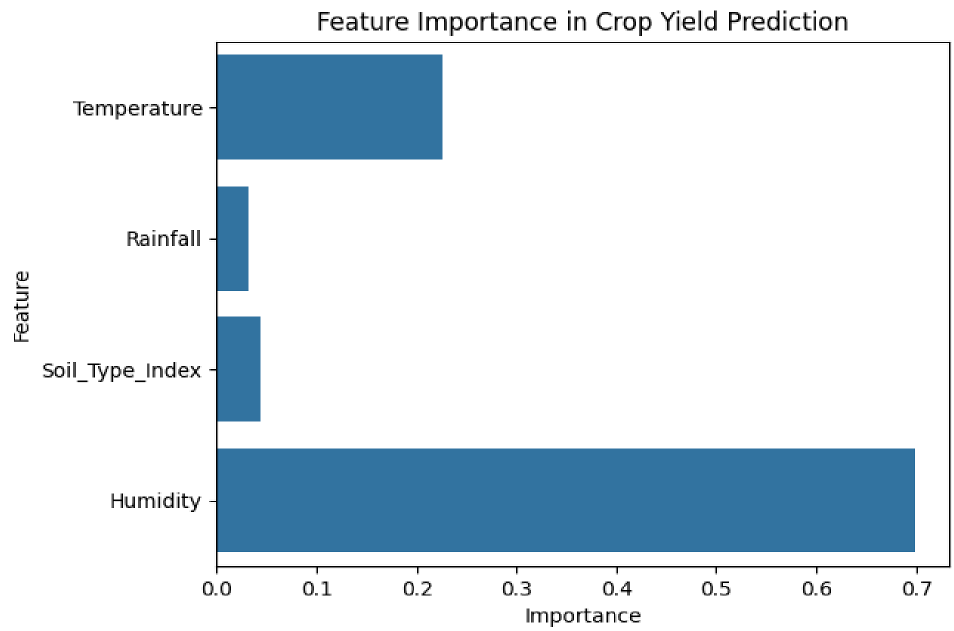
dtype: int64

Model Performance:

R^2 Score: -0.005131504746558990

RMSE: 0.007960743254975618





Conclusion:

In this project, we analyzed crop production data and built a model to predict yield based on factors like temperature, rainfall, soil type, and humidity. The results show that machine learning can help improve farming decisions by giving accurate yield predictions. The dashboard makes it easy to use and understand the output. This project is a useful step toward smarter and more efficient agriculture.