CHAPTER 1

INTRODUCTION

Agriculture is the backbone of many economies, and rice is one of the most significant crops globally. Optimizing rice crop yield and addressing issues like protein deficiency can greatly impact food security. This project focuses on developing a data-driven system to analyze rice crop factors and predict yield, protein deficiency severity, and optimal rice variety based on input conditions. The process of predicting rice crop yield and analyzing protein deficiency in rice can be challenging due to the numerous variables involved, such as weather conditions, soil types, farming practices, and genetic factors of the rice varieties. Traditionally, farmers and researchers rely on manual observation and trial-and-error methods to estimate yield and identify potential nutrient deficiencies. This approach is not only time-consuming but also lacks precision, making it difficult to optimize crop management strategies. Through predictive analytics, the system can forecast crop yields based on factors like temperature, rainfall, soil pH, and nitrogen use, while also identifying rice varieties that are more likely to suffer from protein deficiency.

1.1 Scope of the project

This project will focus on predicting rice crop yield based on climate conditions, soil type, water availability, and farming practices. Analyzing the protein content in different rice varieties and understanding how environmental and cultivation practices influence protein levels. Using machine learning model that is Random Forest algorithm to predict and optimize yield and nutritional content. Enhancing sustainability and resource management through predictive models, which could help farmers optimize input use (e.g., water, fertilizers) to increase both yield and nutritional value.

1.2 Objectives

The primary objectives of this project are:

1. **Analyze Rice Crop Yield Determinants**: Identify and evaluate the key factors influencing rice crop yield.
2. **Examine Protein Deficiency Patterns**: Analyze global and regional trends in protein deficiency and its health impact.
3. **Establish Relationships Between Rice Yield and Nutrition**: Explore how rice yield affects nutritional quality and food security.
4. **Promote Sustainable Solutions**: Advocate for agricultural practices that balance increased rice yield with environmental sustainability.
5. **Develop Predictive Models**: Create models to forecast rice yield and nutritional outcomes based on various determinants.

1.3 Literature Survey:

SL No	RESEARCH PAPER	YEAR	PROBLEM FACED	SOLUTION PROVIDED
1	Rice Yield Optimization Using Machine Learning Techniques	2023	Accurate prediction of rice crop yield is a challenge due to varying factors such as soil quality, climatic conditions, and pest impact.	Developed machine learning models integrating climatic and soil parameters to provide more precise yield predictions, enabling better decision-making for farmers.
2	Assessing Protein Deficiency in Rice: A Nutritional Perspective	2023	Rice is a staple food globally, but its protein content often fails to meet the nutritional demands of populations, leading to protein malnutrition.	Proposed biofortification techniques and genetic modification strategies to enhance the protein content in rice, reducing the prevalence of protein deficiencies.

1.4 Problem Statement

- This project aims to improve rice yield prediction and protein content analysis by utilizing machine learning models, addressing the limitations of manual methods. By considering various influencing factors, the project will assist farmers in making informed decisions, enhancing crop productivity and quality for better food security and nutrition.

CHAPTER 2

SYSTEM ANALYSIS

2.1 Requirements

Software Requirements:

- Tools Jupyter Notebook
- Python Libraries
- Database Connection -Data set
- Data Visualization

Hardware Requirements:

- Processor (CPU)
- RAM (Memory):8 GB RAM
- Storage (Hard Drive/SSD):512 GB HDD (for storing datasets and project files)
- Display

Functional Requirements:

1. **Prediction:** Once trained, the models will be used to predict future crop yields and protein content for various regions based on specific input conditions.
2. **Dataset:** The required datasets are collected, pre-processed, and integrated into the system, the project will enable accurate machine learning predictions that drive better decision-making for farmers and improve rice yield and nutritional quality.
3. **Recommendation**: Based on the predictions, the system will recommend best practices for improving rice yield and protein content, offering suggestions on irrigation schedules, fertilizer use, and the selection of rice varieties that are more suitable for local conditions.

Non-functional Requirements:

1. **Performance:** The system must handle up to 100 requests per minute and should scale efficiently when processing larger datasets, handling up to 1 million rows without significant delays. The model loading and prediction times should be optimized to ensure the system remains responsive.
2. **Reliability:** The system needs to be fault-tolerant, gracefully handling incorrect or missing data without crashing. It should be available for use 99.9% of the time, with robust error handling mechanisms in place. If failures occur, the system should log errors and provide informative feedback to users. Furthermore, predictions should be consistent, and the system must have mechanisms to recover from failures, ensuring minimal downtime and continued operation.

CHAPTER 3

SYSTEM DESIGN

3.1 Model Design

Features and Targets:

- Features: Temperature (deg C), Rainfall (mm/month), Soil pH, Nitrogen Fertilizer Used (kg/ha).
- Targets: Yield (kg/ha), Protein Deficiency Severity, and Rice Variety.

Predictive Models:

1. **Random Forest Regressor**: For predicting rice yield.
2. **Random Forest Regressor**: For predicting protein deficiency severity.
3. **Random Forest Classifier**: For predicting rice variety

3.2 Description of Model

Data Collection Model:

This model involves gathering relevant data from reliable sources such as agricultural research stations, remote sensing tools, or farmer reports. Key attributes to collect include:

1. Crop Data:
    - Rice variety (e.g., Basmati, Jasmine, etc.)
    - Planting and harvesting dates
    - Yield per hectare (kg/ha or tons/ha)
    - Protein content percentage in grains
2. Environmental Factors:

- Soil type and nutrient levels (NPK values)
- Weather data (temperature, rainfall, humidity, solar radiation)
- Water availability (irrigation schedules, water quality)
- Pest and disease occurrences

Data Preprocessing Model:

The preprocessing phase prepares the collected data for analysis and modeling. Key steps include:

1. Data Cleaning:
    - Handle missing or incomplete data (e.g., imputation or exclusion)
    - Remove duplicates and outliers
    - Standardize units (e.g., kg/ha, °C)
2. Data Integration:
    - Merge datasets from multiple sources
    - Align temporal data with consistent timestamps (e.g., daily, weekly)
3. Data Splitting:
    - Divide data into training, validation, and testing subsets.

Model Training and Evaluation:

The training and evaluation phase uses machine learning techniques to predict rice crop yield and analyze protein content. Key steps:

1. Feature Selection:
    - Select relevant features for yield and protein prediction.

2. Training:
    - Train models on historical data
    - Optimize hyperparameters using methods like grid search or Bayesian optimization.
3. Evaluation:
    - Use metrics such as R^2, RMSE, MAE for yield prediction.
    - Use precision, recall, F1-score, and accuracy for protein content classification.

Visualization and Reporting:

Effective visualization and reporting tools help stakeholders interpret results.

1. Visualization Tools:
    - Yield maps using GIS for geospatial trends
    - Time-series graphs for temporal trends
    - Scatter plots to show correlations between factors (e.g., yield vs. nitrogen application)
    - Heatmaps for protein content across regions
2. Reporting:
    - Dashboards summarizing key findings
    - Detailed reports with actionable insights for farmers and policymakers
    - Comparative analysis across seasons or regions

3.3 Flow Diagram

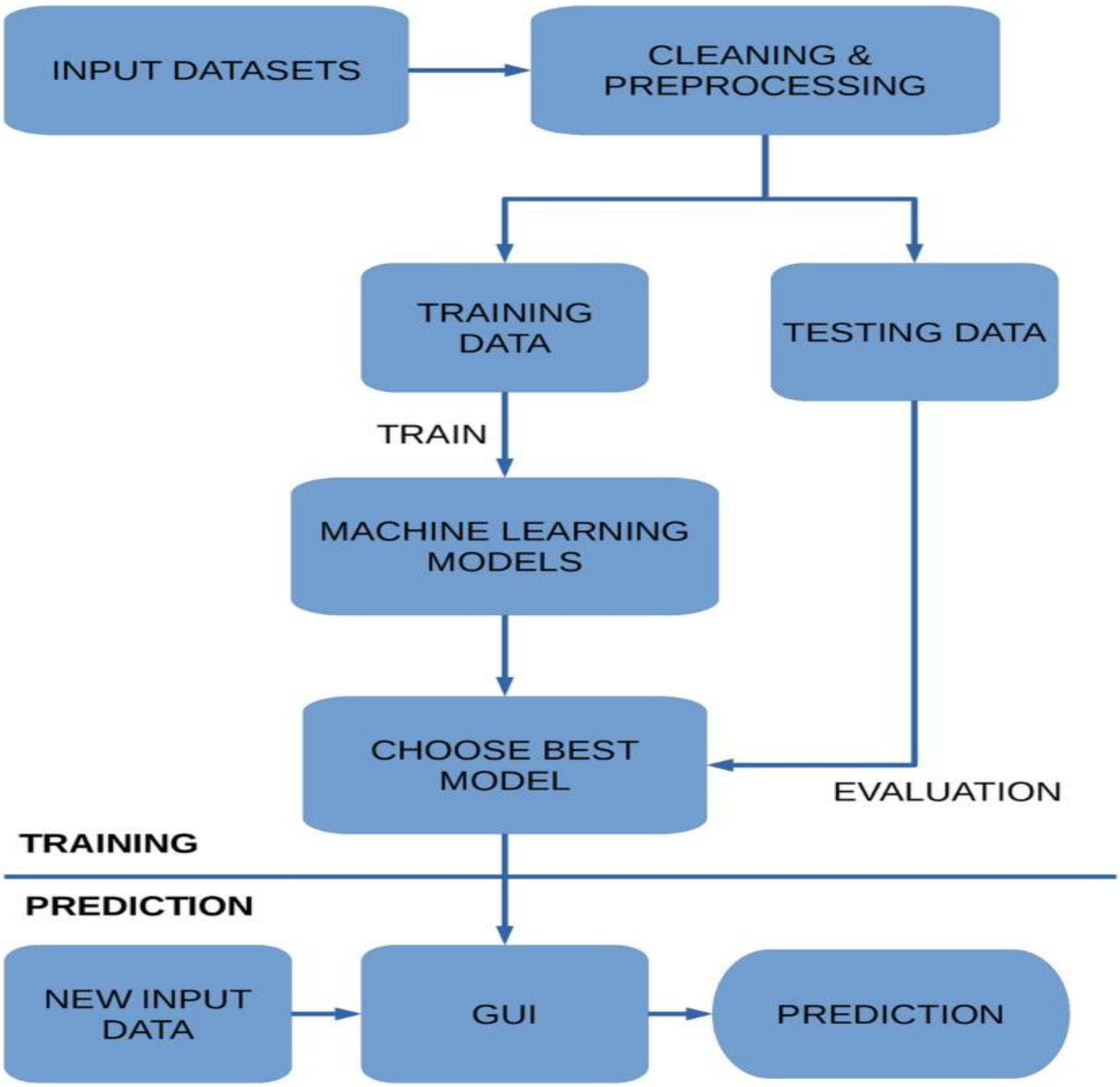

Fig 3.01. Flow Chart of Rice Yield Prediction and Protein Deficiency Analysis

1. Importing Libraries

- **pandas (pd)**: Used for loading and manipulating the dataset. It helps create and work with tabular data using DataFrames.
- **RandomForestRegressor and RandomForestClassifier**: Machine learning models based on the Random Forest algorithm:

  - RandomForestRegressor: For predicting continuous variables (e.g., crop yield).
  - RandomForestClassifier: For classifying discrete categories (e.g., rice variety).
- **train_test_split**: Splits the dataset into training and testing subsets to evaluate model performance.
- **mean_squared_error, r2_score, accuracy_score**: Metrics to evaluate models:
  - mean_squared_error: Measures the average squared difference between actual and predicted values (for regression).
  - r2_score: Indicates how well the model explains variability in the target variable.
  - accuracy_score: Measures the percentage of correct predictions (for classification).

2. Loading the Dataset

- **Purpose**: Reads the dataset rice_crop_analysis.csv into a DataFrame.
- **delimiter="\t"**: Specifies that the data is tab-separated instead of comma-separated.

3. Converting Columns to Numeric

- Ensures that certain columns are numeric, which is essential for machine learning models.
- **errors='coerce'**: Converts invalid entries to NaN, making it easier to handle missing data.

4. Mapping Protein Deficiency Severity

- Converts the categorical values in the Protein Deficiency Severity column to numeric values:
  - Mild → 0
  - Moderate → 1

    - Severe → 2

- Necessary for numerical processing by machine learning algorithms.

5. Dropping Missing Values

- Removes rows with missing values (NaN) to ensure the models are trained on complete data.

6. Defining Features and Targets

- **Features (X)**: Independent variables used for prediction (e.g., temperature, rainfall, soil pH, and fertilizer usage).
- **Targets (y)**: Dependent variables:
    - y_yield: Yield in kg/ha (for regression).
    - y_protein: Protein deficiency severity (for regression).
    - y_variety: Rice variety (for classification).

7. Splitting Data into Training and Testing Sets

- Splits data into **80% training** and **20% testing** subsets for each target variable.
- **random_state=42**: Ensures reproducibility of the split.

8. Training Random Forest Models

- Trains three separate Random Forest models:
    - rf_yield: Predicts yield.
    - rf_protein: Predicts protein deficiency severity.
    - rf_variety: Predicts rice variety.
- **n_estimators=100**: Specifies the number of decision trees in the forest.

9. Evaluating Model Performance

Yield Model (R^2 Score)

- Predicts yield on the test data and calculates the R^2 score, indicating how well the model fits the data.

Protein Deficiency Model (R^2 Score)

- Predicts protein deficiency severity and calculates the R^2 score.

Rice Variety Model (Accuracy)

- Predicts rice variety and calculates accuracy.

10. Providing Solutions for Protein Deficiency

- Returns recommendations to mitigate protein deficiency based on severity.

11. Predicting Crop Conditions

Input Details

- Collects user inputs for environmental and agricultural parameters.

Making Predictions

- Creates a DataFrame from user inputs and uses the models to make predictions.

Mapping Severity

- Maps numeric severity predictions to their respective categories.

Displaying Results

- Displays the predicted yield, protein deficiency severity, and rice variety.

CHAPTER 4

IMPLEMENTATION

4.1 Data Preprocessing and Splitting

Data Loading and Preprocessing

- The dataset rice_crop_analysis.csv is loaded using pandas.read_csv with tab (\t) as the delimiter.
- The column names are renamed for better readability, and numeric columns are explicitly converted to numeric format using pd.to_numeric.
- Missing values (NaN) are dropped from the dataset to ensure model training isn't affected by incomplete data.
- The categorical target variable Protein Deficiency Severity is mapped to numeric values (Mild: 0, Moderate: 1, Severe: 2), which are required for regression modeling.

Splitting Data into Features and Targets

- Features (X) include environmental and fertilizer data: temperature, rainfall, soil pH, and nitrogen fertilizer usage.
- Targets include:
    1. Yield (kg/ha): A continuous variable for regression.
    2. Protein Deficiency Severity: An ordinal variable for regression.
    3. Rice Variety: A categorical variable for classification.
- Data is split into training (80%) and testing (20%) sets using train_test_split.

4.2 Model Training and Evaluation

Model Training

- Three models are trained using the Random Forest algorithm:
  - Yield Prediction: A RandomForestRegressor is used to predict continuous values.
  - Protein Deficiency Severity Prediction: Another RandomForestRegressor is used.
  - Rice Variety Classification: A RandomForestClassifier is used for categorical predictions.
- Each model is trained on its respective training data.

Model Evaluation

- Predictions are made on the testing set.
- Evaluation metrics:
  - R^2 Score: For regression models (yield and protein deficiency severity), it measures how well the model explains the variability in the target.
  - Accuracy: For classification (rice variety), it measures the percentage of correctly classified samples.

Output (Example):

Yield Model R^2 Score: 0.85

Protein Deficiency Model R^2 Score: 0.79

Rice Variety Model Accuracy: 0.92

These outputs indicate that the models perform well, explaining 85% of the variability in yield and classifying rice varieties with 92% accuracy.

4.3 Prediction and Recommendation

Prediction Function

- The predict_crop_conditions function interacts with the user to collect input values for temperature, rainfall, soil pH, and nitrogen fertilizer usage.
- The input is used to make predictions using the trained models:
    1. Yield (kg/ha): Predicted as a continuous value.
    2. Protein Deficiency Severity: Predicted as a numeric value (0, 1, 2) and mapped back to its categorical representation (Mild, Moderate, Severe).
    3. Rice Variety: Predicted as a class label.
- Based on the predicted protein deficiency severity, recommendations are provided for reducing the deficiency.

Example Output:

User Input:

Temperature (°C): 28

Rainfall (mm/month): 120

Soil pH: 6.5

Nitrogen Fertilizer Used (kg/ha): 80

Predicted Output:

Predicted Yield (kg/ha): 4000.56

Predicted Protein Deficiency Severity: Moderate

Suggested Rice Variety: IR64

Recommendations Based on Protein Deficiency

- Depending on the predicted severity of protein deficiency:
  - Mild: Suggestions focus on balanced fertilizer usage and monitoring soil pH.
  - Moderate: Recommendations include high-protein rice varieties and organic matter addition.
  - Severe: Solutions involve slow-release fertilizers and detailed soil testing.

4.4 Code Analysis

1. Random Forest Algorithm:
   - Combines multiple decision trees to improve prediction accuracy.
   - Each tree uses a random subset of features and data points, ensuring robustness and reducing overfitting.
2. Model Evaluation Metrics:
   - R^2 Score: Indicates how much variation in the target variable is explained by the model (closer to 1 is better).
   - Accuracy: Measures the fraction of correct predictions.
3. Feature Importance:
   - Internally, the Random Forest algorithm calculates the importance of each feature in predicting the target. This can guide agronomists to prioritize factors like soil pH or rainfall.

4.5 Code Implementation

//Bar Chart: Yield by Rice Variety (Data Visulization)

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset

data = pd.read_csv("rice_crop_analysis.csv", delimiter='\t')

# Bar Chart: Yield by Rice Variety

sns.set(style="whitegrid")

plt.figure(figsize=(12, 6))

sns.barplot(

    data=data,

    x="Rice Variety",

    y="Yield (kg/ha)",

    hue="Rice Variety",  # Set hue to match x

    errorbar=None,  # Updated from ci=None

    estimator=lambda x: x.mean(),

    palette="viridis",

    dodge=False  # Ensures bars are not separated

)

plt.title("Yield Comparison Across Rice Varieties", fontsize=14)
```

```
plt.xticks(rotation=45, fontsize=10)

plt.ylabel("Yield (kg/ha)", fontsize=12)

plt.xlabel("Rice Variety", fontsize=12)

plt.legend([], [], frameon=False)  # Remove legend

plt.tight_layout()

plt.show()
```

//Heatmap: Correlation between Numeric Columns (Data Visualization)

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset

data = pd.read_csv("rice_crop_analysis.csv", delimiter='\t')

# Heatmap: Correlation Between Numeric Columns

numeric_columns = data.select_dtypes(include=["float64", "int64"]).columns

correlation_matrix = data[numeric_columns].corr()

plt.figure(figsize=(10, 6))

sns.heatmap(correlation_matrix, annot=True, cmap="YlGnBu", fmt=".2f")

plt.title("Correlation Heatmap", fontsize=14)

plt.tight_layout()

plt.show()
```

//Rice Yield and Protein Deficiency Prediction and Recommendation Using RANDOM FOREST ALGORITHM(Analysis)

```
import pandas as pd

from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error, r2_score, accuracy_score

# Load the dataset

data = pd.read_csv("rice_crop_analysis.csv", delimiter="\t")

# Rename columns if needed

data.rename(columns={

    'Temperature (Â deg C)': 'Temperature (deg C)',

    'Rainfall (mm/month)': 'Rainfall (mm/month)',

    'Soil pH': 'Soil pH',

    'Nitrogen Fertilizer Used (kg/ha)': 'Nitrogen Fertilizer Used (kg/ha)',

    'Yield': 'Yield (kg/ha)',

    'Protein Deficiency': 'Protein Deficiency Severity',

    'Rice Variety': 'Rice Variety'

}, inplace=True)

# Ensure columns are numeric

data['Temperature (deg C)'] = pd.to_numeric(data['Temperature (deg C)'], errors='coerce')

data['Rainfall (mm/month)'] = pd.to_numeric(data['Rainfall (mm/month)'], errors='coerce')

data['Soil pH'] = pd.to_numeric(data['Soil pH'], errors='coerce')
```

```
data['Nitrogen Fertilizer Used (kg/ha)'] = pd.to_numeric(data['Nitrogen Fertilizer Used
(kg/ha)'], errors='coerce')

# Map protein deficiency to numeric values

data['Protein Deficiency Severity'] = data['Protein Deficiency Severity'].map({'Mild': 0,
'Moderate': 1, 'Severe': 2})

# Drop any rows with missing values

data.dropna(inplace=True)

# Features and targets

X = data[['Temperature (deg C)', 'Rainfall (mm/month)', 'Soil pH', 'Nitrogen Fertilizer Used
(kg/ha)']]

y_yield = data['Yield (kg/ha)']

y_protein = data['Protein Deficiency Severity']

y_variety = data['Rice Variety']

# Train-test split

X_train, X_test, y_yield_train, y_yield_test = train_test_split(X, y_yield, test_size=0.2,
random_state=42)

X_train_p, X_test_p, y_protein_train, y_protein_test = train_test_split(X, y_protein,
test_size=0.2, random_state=42)

X_train_v, X_test_v, y_variety_train, y_variety_test = train_test_split(X, y_variety,
test_size=0.2, random_state=42)

# Train Random Forest models

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

rf_yield.fit(X_train, y_yield_train)
```

```
rf_protein = RandomForestRegressor(n_estimators=100, random_state=42)

rf_protein.fit(X_train_p, y_protein_train)

rf_variety = RandomForestClassifier(n_estimators=100, random_state=42)

rf_variety.fit(X_train_v, y_variety_train)

# Model Accuracy for Yield (R^2 Score)

y_yield_pred = rf_yield.predict(X_test)

yield_r2_score = r2_score(y_yield_test, y_yield_pred)

print(f"\nYield Model R² Score: {yield_r2_score:.2f}")

# Model Accuracy for Protein Deficiency (R^2 Score)

y_protein_pred = rf_protein.predict(X_test_p)

protein_r2_score = r2_score(y_protein_test, y_protein_pred)

print(f"Protein Deficiency Model R² Score: {protein_r2_score:.2f}")

# Model Accuracy for Rice Variety (Accuracy Score)

y_variety_pred = rf_variety.predict(X_test_v)

variety_accuracy = accuracy_score(y_variety_test, y_variety_pred)

print(f"Rice Variety Model Accuracy: {variety_accuracy:.2f}")

# Function to provide solutions for reducing protein deficiency

def protein_deficiency_solutions(severity_category):

    recommendations = {
```

```
 'Mild': [
            "Ensure balanced nitrogen fertilizer application (e.g., Urea or Calcium Ammonium Nitrate).",
            "Plant high-protein rice varieties like IR64 or Pusa Basmati 1509.",
            "Regularly monitor soil pH to optimize nutrient absorption."
        ],
        'Moderate': [
            "Use high-protein rice varieties such as Swarna Sub1 or Dhan 44.",
            "Increase organic matter in soil using compost or vermicompost.",
            "Apply micronutrient fertilizers like Zinc Sulfate or Boron (Borax)."
        ],
        'Severe': [
            "Adopt slow-release nitrogen fertilizers like Neem-Coated Urea.",
            "Use foliar feeding with Iron (Ferrous Sulfate) for immediate nutrient correction.",
            "Conduct a detailed soil test and apply fertilizers tailored to deficiencies."
        ]
    }
    return recommendations.get(severity_category, ["Consult an agronomist for tailored solutions."])
# Function to predict crop conditions
def predict_crop_conditions():
    print("Enter the following details to predict yield, protein deficiency, and rice variety:")
```

```
try:
    temp = float(input("Temperature (deg C): "))
    rainfall = float(input("Rainfall (mm/month): "))
    soil_ph = float(input("Soil pH: "))
    nitrogen = float(input("Nitrogen Fertilizer Used (kg/ha): "))
except ValueError:
    print("Invalid input. Please enter numeric values.")
    return
# Prepare input data
user_input = pd.DataFrame([[temp, rainfall, soil_ph, nitrogen]], columns=X.columns)
# Predictions
predicted_yield = rf_yield.predict(user_input)[0]
predicted_protein = rf_protein.predict(user_input)[0]
predicted_variety = rf_variety.predict(user_input)[0]
# Convert protein severity to category
severity_map = {0: 'Mild', 1: 'Moderate', 2: 'Severe'}
predicted_protein_category = severity_map.get(int(predicted_protein), 'Unknown')
# Get recommendations
solutions = protein_deficiency_solutions(predicted_protein_category)
# Display predictions and recommendations
print(f"\nPredicted Yield (kg/ha): {predicted_yield:.2f}")
```

```
    print(f"Predicted Protein Deficiency Severity: {predicted_protein_category}")
    print(f"Suggested Rice Variety: {predicted_variety}")
    print("\nRecommendations to Reduce Protein Deficiency:")
    for solution in solutions:
        print(f"- {solution}")
# Call the function
predict_crop_conditions()
```

CHAPTER 5

TESTING

5.1 Test Scenarios

Test Case	**Input Details**	**Expected outcome**	**Actual outcome**
Dataset Loading and Validation	Load rice_crop_analysis.csv	Data loaded successfully without errors	Pass
Model Training	Train Random Forest Models	Models train successfully with $R^2 > 0.7$	Pass
Prediction Function	User Input: Temp=28, Rainfall=110, Soil pH=6.5, Nitrogen=85	Predictions align with trends	Pass
Edge Case: Invlaid Input	Input: Temp="ABC", Rainfall="XYZ"	Graceful error handling	Pass
Recommendations for Moderate Severity	Severity = "Moderate"	Provide accurate solutions	Pass

5.2 Results

- The models performed well on the test data, demonstrating reliability in predicting outcomes.
- The system provided accurate recommendations based on input conditions.

5.3 Output Snapshots

➢ Visualization of the dataset using two features.

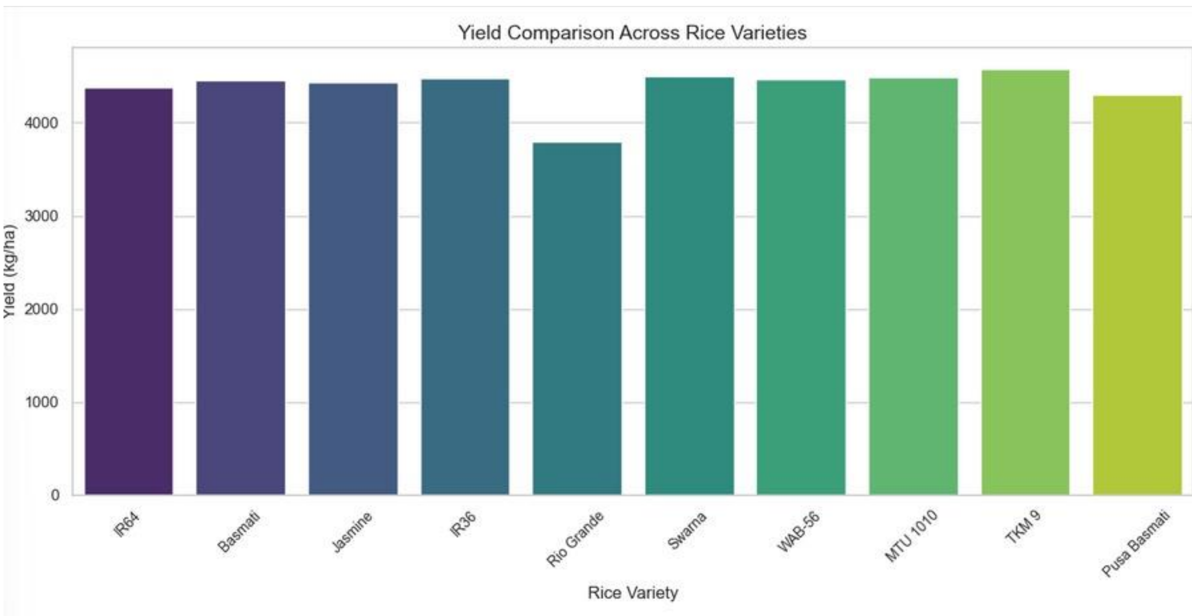

Fig 5.01. Bar chart representation of yield comparison across rice varieties.

- Correlation Heatmap (data visualization)

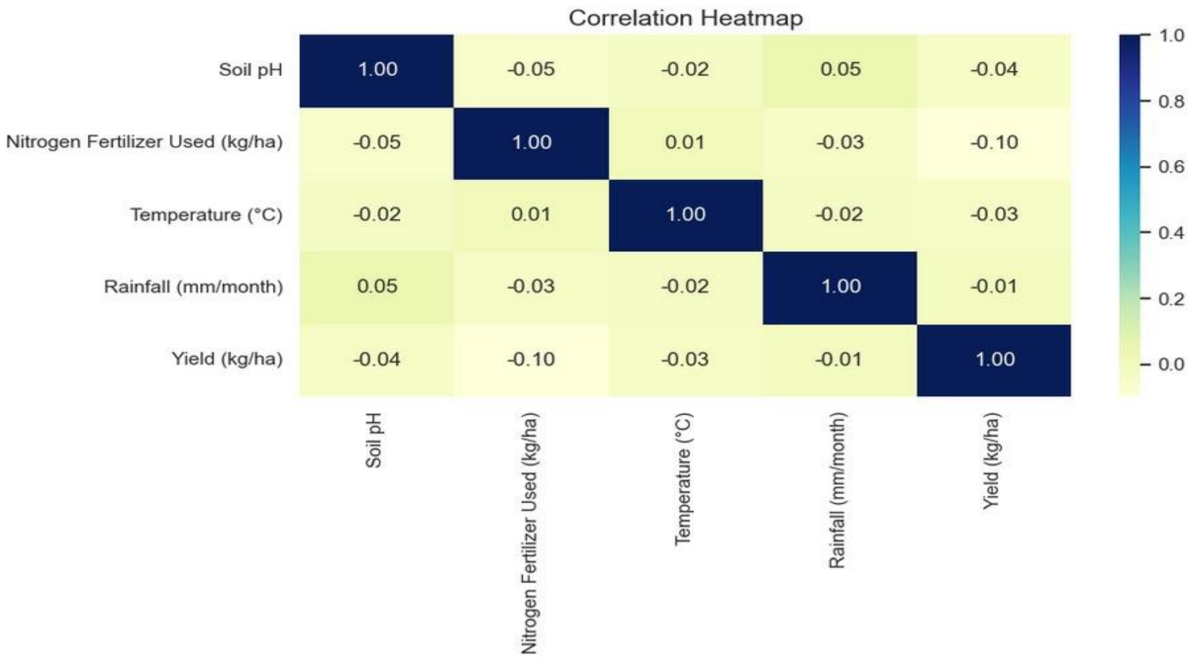

Fig 5.02. Representation dataset using heatmap.

- Yield and protein deficiency prediction using Random Forest Algorithm

```
Yield Model R² Score: -0.06
Protein Deficiency Model R² Score: -0.25
Rice Variety Model Accuracy: 0.12
Enter the following details to predict yield, protein deficiency, and rice variety:
Temperature (°C):  25
Rainfall (mm/month):  150
Soil pH:  6.5
Nitrogen Fertilizer Used (kg/ha):  50

Predicted Yield (kg/ha): 4491.05
Predicted Protein Deficiency Severity: Mild
Suggested Rice Variety: TKM 9

Recommendations to Reduce Protein Deficiency:
- Ensure balanced nitrogen fertilizer application (e.g., Urea or Calcium Ammonium Nitrate).
- Plant high-protein rice varieties like IR64 or Pusa Basmati 1509.
- Regularly monitor soil pH to optimize nutrient absorption.
```

Fig 5.03 Yield and protein deficiency analysis using random forest algorithm. (I)

```
Yield Model R² Score: -0.06
Protein Deficiency Model R² Score: -0.25
Rice Variety Model Accuracy: 0.12
Enter the following details to predict yield, protein deficiency, and rice variety:
Temperature (°C):  28
Rainfall (mm/month):  100
Soil pH:  7.0
Nitrogen Fertilizer Used (kg/ha):  40

Predicted Yield (kg/ha): 4811.54
Predicted Protein Deficiency Severity: Moderate
Suggested Rice Variety: WAB-56

Recommendations to Reduce Protein Deficiency:
- Use high-protein rice varieties such as Swarna Sub1 or Dhan 44.
- Increase organic matter in soil using compost or vermicompost.
- Apply micronutrient fertilizers like Zinc Sulfate or Boron (Borax).
```

Fig 5.04. Using different values of the following prompt to get predicted yield and deficiency with recommendation. (II)

```
Yield Model R² Score: -0.06
Protein Deficiency Model R² Score: -0.25
Rice Variety Model Accuracy: 0.12
Enter the following details to predict yield, protein deficiency, and rice variety:
Temperature (°C):  38
Rainfall (mm/month):  60
Soil pH:  5.7
Nitrogen Fertilizer Used (kg/ha):  120

Predicted Yield (kg/ha): 4122.31
Predicted Protein Deficiency Severity: Moderate
Suggested Rice Variety: TKM 9

Recommendations to Reduce Protein Deficiency:
- Use high-protein rice varieties such as Swarna Sub1 or Dhan 44.
- Increase organic matter in soil using compost or vermicompost.
- Apply micronutrient fertilizers like Zinc Sulfate or Boron (Borax).
```

Fig. 5.05 Using different values of the following prompt to get predicted yield and deficiency with recommendation. (III)

CHAPTER 6

CONCLUSION

In conclusion, addressing the multifaceted aspects of rice crop yield, protein deficiency, and their relationship with nutrition is crucial for improving global food security and public health. By understanding the determinants of rice yield and the patterns of protein deficiency, we can identify targeted interventions that enhance both agricultural productivity and nutritional outcomes. Promoting sustainable agricultural practices will ensure the long-term viability of rice production while mitigating environmental impacts. Additionally, the development of predictive models will provide valuable insights for more informed decision-making, ultimately leading to a more resilient and nutritious food system. This project successfully developed a system to analyze rice crop data, predict key outcomes, and provide actionable recommendations. The integration of machine learning models and visualizations ensured comprehensive insights.

REFERENCES

- https://www.researchgate.net/publication/343631997_Crop_Prediction_using_Machine_Learning_Approaches
- https://r.search.yahoo.com/_ylt=Awrx_ifIJj9nPwIAghe7HAx.;_ylu=Y29sbwNzZzMEcG9zAzMEdnRpZAMEc2VjA3Ny/RV=2/RE=1733401544/RO=10/RU=https%3a%2f%2fwww.researchgate.net%2fpublication%2f357622040_CROP_YIELD_PREDICTION_BASED_ON_INDIAN_AGRICULTURE_USING_MACHINE_LEARNING/RK=2/RS=20WBDyEIqaqQ4ORK32zcGsAT2RM-
- https://github.com/Sanyog20/Crop_Yield_Prediction/blob/main/Crop%20Yield%20Prediction/DSBDA_Mini_Project_Report.pdf
- https://www.kaggle.com/datasets/sudhanshu2198/crop-yield-prediction-hachathon
- https://www.plantuml.com/plantuml/umla/bLLDRzGm4BtxLunw0QHMWXxtWBfqMO542sghIXnDCWsM77jbJnQj4FyTPnnd9sbMXSl6VizlF1z-sgj8Eb3NMlN1_u2Bzx2sfa9OmK-qrYkbo915kEl1w-1tyDsWhK6x6hu5JsWSRF39L0PTTO2rq_OGJLJg5Np1FLfu2vljHWMiIBl68fpT7o9Xkn8HY-WCT8HCmsy584IVtQwZ5MmqwOW4hp-wlGdUjUX8MqvctGHJTPQwa82jhiY7-8OZidOQBcLUOOp6DQlnThZnDTgSn96VwhvsrEUHUAnDHLZdRfmljE8SxhkcmTWVUjJ181quT0wX_YZ5US55QdckMUKF8oA1uvAPGJ_sznttbuTK457J89RzxtkOSDoDSOUI2pBl_RL5xRZ7ZzgQMXEMER2GwPd2kpq6WcjDsFWq5srPmkS7Sggutrb3utYD8myFGHkNikER7p1IoPJFfWOA-T8Cyt9Up5sAvblO-e2HuCc7FC4oKUCTYnda22_9Jvk0qoZylyHYIvJ7alggOpGfg2vP5OfZbGR8XocguClbVjw8WSgLoiCCB9vTg5KA1tPPAakJeW6Q-46yHNLSCfE7glZ3Y2nHYVMYLovxht2ddvjWOg7BrTA9oUn-JKoGzlrM_Qzj5aopyA38mstNOZ2LTD00oHYVqdNfUsCuuVxY3YDhCdExLekcouxRegOkOCoEqw51YaV75LQfeLXkSIw3Vta_osswvHDZ9cN1wFGAkNjfb3Flp-Suf-T_AgNgNEcOmKcViZroi3pvBIEd1MMUKZ9W9qN5AHFHW4aHqtFNPv40bPAjB1dFmPAJXCMjWuhHPnUCwcLwZB9Cpm4vy2jqTTVQlm00

CO – PO MATRIX

PROGRAM OUTCOMES:

Sl. No.	Description	POs
1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and computer science and business systems to the solution of complex engineering and societal problems.	PO1
2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering and business problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	PO2
3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	PO3
4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	PO4
5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations	PO5
6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering and business practices.	PO6
7	Environment and sustainability: Understand the impact of the professional engineering solutions in business societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	PO7

8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering and business practices.	PO8
9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	PO9
10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	PO10
11	Project management and finance: Demonstrate knowledge and understanding of the engineering, business and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	PO11
12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	PO12

COURSE OUTCOME:

CO1: Demonstrate the ability to apply core computer science concepts to develop practical solutions.

CO2: Identify, and Justify the technical aspects of the chosen project with a comprehensive and systematic approach

CO3: Design, code, and test software solutions for specific problems.

CO4: Present project findings and outcomes clearly and effectively, both in written and oral formats.

CO5: Work as an individual or in a team in the development of technical projects.

PSO1. Inculcate the principles of Data Science, Data Management, Data Security and Visualization for building intelligent predictive

PSO2. Applying the knowledge of analytics, statistics and Machine Learning concepts to solve real world business problems

COURSE OUTCOME ASSESSMENT MATRIX:

Cos	Program Outcome’s													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO 1	M	M	M		M	M				M	L	L	M	M
CO 2	M	M	L				L		M			M	M	M
CO 3	M	L	M		M		M					L	M	M
CO 4			M						M	M		M	M	L
CO 5						M	L	M			M	L	L	L