Crop Recommendation System

# Introduction

The Crop Recommendation System project aims to assist farmers in making informed decisions about which crops to cultivate based on various factors such as soil nutrients (N, P, K), environmental conditions (temperature, humidity, rainfall), pH levels, and historical crop performance data. This report outlines the methodology, implementation, and evaluation of the crop recommendation system using a dataset comprising eight columns: N, P, K, temperature, humidity, pH, rainfall, and label. The dataset, sourced from Kaggle, is in CSV format.

# Objective

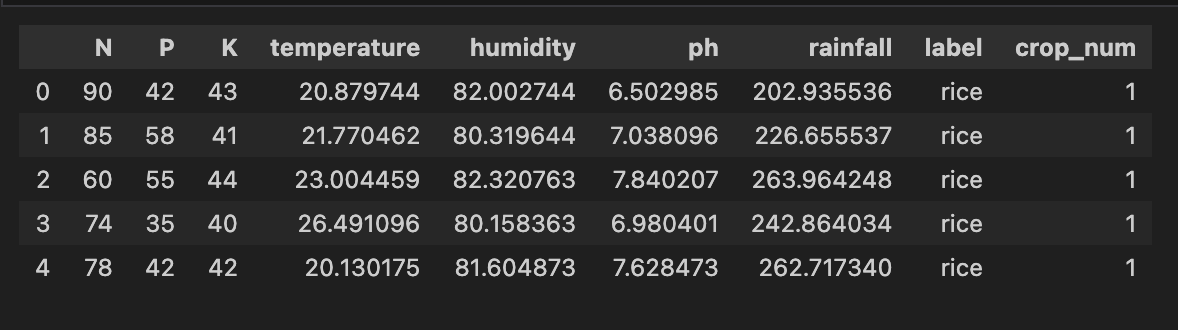
Develop a machine learning model and Deep Learning model which are capable of recommending suitable crops based on input parameters derived from soil and environmental data.

Provide farmers with personalized crop recommendations tailored to their specific agricultural conditions.

Evaluate the performance of the model using appropriate metrics and validate its effectiveness in real-world scenarios.

# Dataset

The dataset utilized in this project encompasses eight essential columns, each representing pivotal factors in determining crop suitability. These columns include the levels of Nitrogen (N), Phosphorus (P), and Potassium (K) in the soil, crucial nutrients for plant growth and development. Additionally, the dataset comprises environmental parameters such as Temperature, reflecting the average temperature in Celsius, and Humidity, denoting the relative humidity expressed in percentage. Furthermore, it incorporates the pH level of the soil, a critical indicator of soil acidity or alkalinity, and the average Rainfall, measured in millimeters, which significantly influences crop water requirements. Finally, the Label column designates the recommended crop based on the combined assessment of these factors. Sourced from Kaggle, the dataset undergoes rigorous preprocessing, including the removal of missing values, normalization of numerical features to ensure consistency, and encoding of categorical variables to facilitate model training and evaluation.



# Methodology

## Data Pre-processing

**Data Cleaning:** Handling missing values and outliers in the dataset.

Feature Engineering: Extracting relevant features and encoding categorical variables.

**Data Normalization**: Scaling numerical features to a common range to ensure uniformity during model training.

**Splitting Dataset:** Dividing the dataset into training and testing sets to evaluate model performance.

# Normalization

MinMaxScaler:

This scaler transforms features by scaling each feature to a given range. It scales and translates each feature individually such that it is in the given range on the training set, e.g., between zero and one. The MinMaxScaler is used to scale the input features to a range, typically [0, 1].

StandardScaler:

This scaler standardizes features by removing the mean and scaling to unit variance. Standardization of a dataset is a common requirement for many machine learning estimators. The StandardScaler is used to standardize features by removing the mean and scaling to unit variance.

## Standardization (Z-score normalization):

Standardization is performed using the StandardScaler from scikit-learn (sklearnpreprocessing). This technique transforms the data such that its distribution has a mean of 0 and a standard deviation of 1. It's suitable for features that are normally distributed or approximately normally distributed.

## Batch Normalization:

Batch Normalization is applied within the neural network model using nn. BatchNorm1d from PyTorch (torch.nn). Batch normalization normalizes the inputs of each layer by adjusting and scaling the activations.

Batch normalization layers are added after the fully connected layers in the neural network architecture (fc1 and fc2). Batch normalization helps in stabilizing and accelerating the training process of deep neural networks by reducing internal covariate shift.

## Min-Max Scaling (Normalization):

Min-Max scaling, also known as normalization, is performed using the MinMaxScaler from scikit-learn (sklearn.preprocessing). This technique scales the features to a fixed range, typically between 0 and 1. It preserves the shape of the original distribution and is suitable for neural networks, especially those with activation functions like sigmoid or tanh.

## Standard Scaling (z-score normalization):

This technique scales each feature to have a mean of 0 and a standard deviation of 1. It's achieved by subtracting the mean of the feature from each value and then dividing by the standard deviation. This ensures that each feature has a similar scale, which can be important for algorithms that rely on distance measures, such as KMeans clustering. In the code, StandardScaler from scikit-learn is used to perform this normalization.

# Model Development

* Model Selection: Evaluating various machine learning algorithms such as decision trees, random forests, support vector machines (SVM), and neural networks to identify the most suitable approach for crop recommendation.
* Model Training: Training the selected models using the training dataset and optimizing hyperparameters using techniques like grid search or random search.
* Model Evaluation: Assessing the performance of the trained models using evaluation metrics such as accuracy, precision, recall, and F1 score.

# Models Used

## Classification Models:

* + Model 1 (Classification): [Logistic Regression]
  + Model 2 (Classification): [Naive Bayes]
  + Model 3 (Classification): [Support Vector Machine]
  + Model 4 (Classification): [K-Nearest Neighbors]

## Clustering Models:

* + Model 3 (Regression): [K means Clustering]
  + Model 4 (Regression): [DBSCAN]
  + Model 5(Regression): [Agglomerative Clustering]
  + Model 6 (Regression): [Gaussian Mixture]
* Deep Learning Models:
  + Model 5 (Deep Learning): [Custom Neural network model using PyTorch.]
  + Model 6 (Deep Learning): [FeedForward Nueral Network.]

# Result

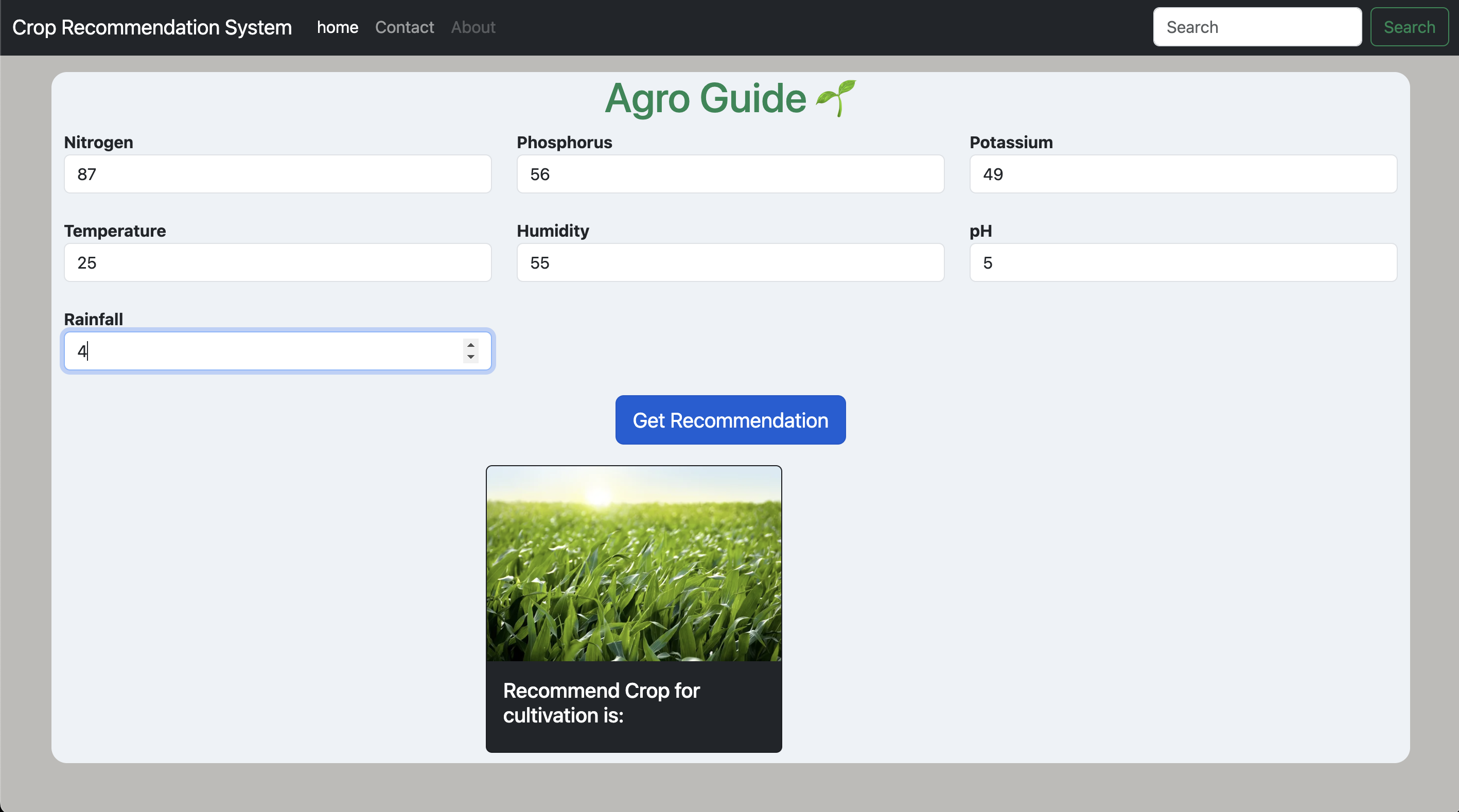
The crop recommendation system achieves the following results:

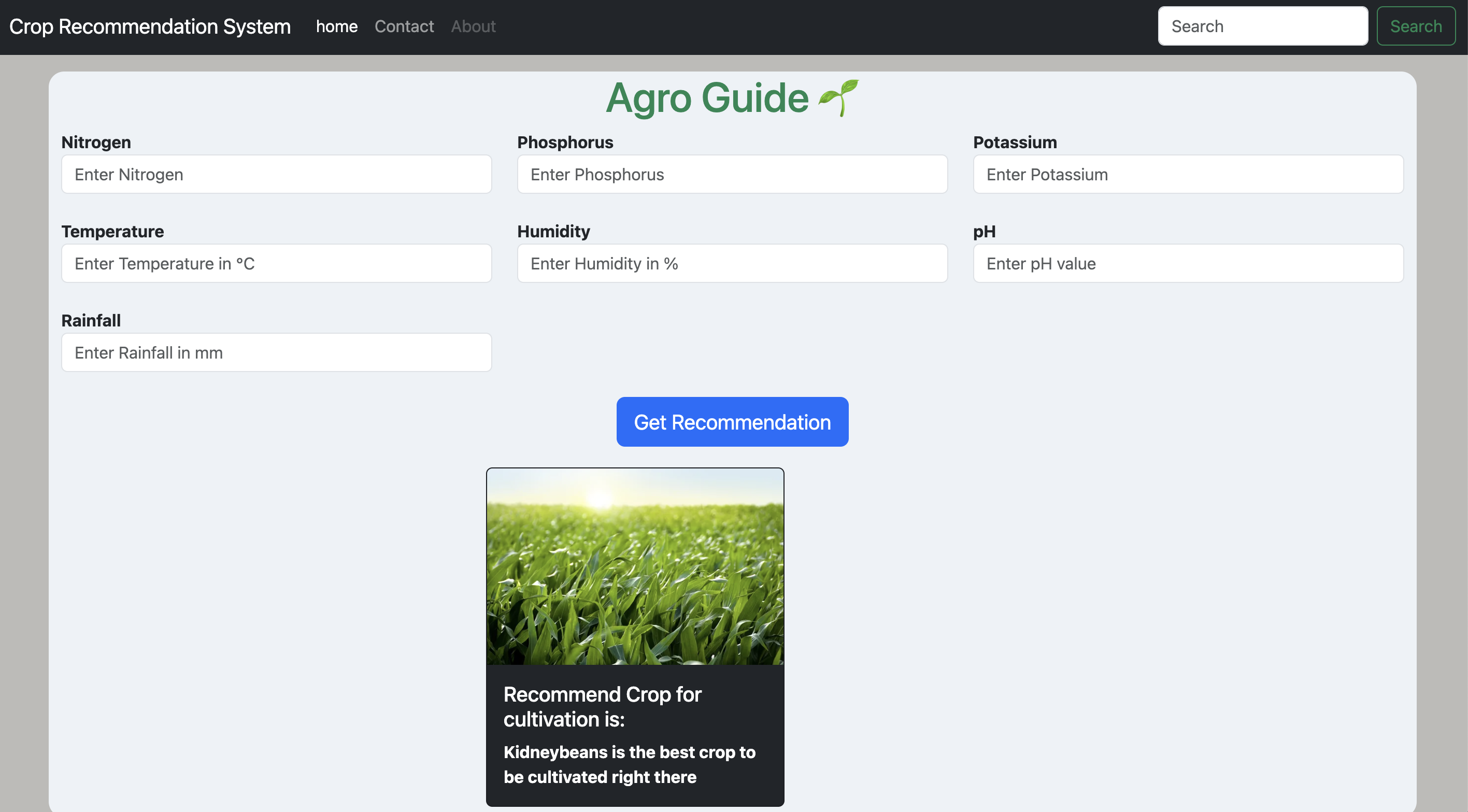
The performance metrics indicate that the model effectively predicts suitable crops based on input parameters, demonstrating its potential utility in agricultural decision-making.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy(%) | Precision(%) | Recall(%) | F1-Score(%) |
| Logistic Regression | 96.36 | 95.00 | 95.34 | 95.00 |
| Naive Bayes | 99.54 | 99.00 | 89.54 | 94.22 |
| Support vector Machine | 96.81 | 89.00 | 94.00 | 91.00 |
| K-Nearest Neighbors | 95.90 | 89.00 | 94.00` | 81.00 |

|  |  |
| --- | --- |
| Model | Silhouette Score |
| K-Means Clustering | 34.26 |
| Agglomerative Clustering | 47.85 |
| DBSCAN | 47.86 |
| Gaussian Mixture | 37.99 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy(%) | Precision(%) | Recall(%) | F1-Score(%) |
| Feedforward neural network | 90.00 | 99.00 | 98.89 | 97.96 |
| Neural network model using PyTorch | 93.83 | 96.00 | 96.00 | 96.00 |





Conclusion

The Crop Recommendation System project successfully develops a machine learning model that recommends suitable crops based on input parameters derived from soil and environmental data. By integrating advanced techniques in data pre-processing, model development, and user interface design, the system provides farmers with valuable insights to optimize crop selection and enhance agricultural productivity. Moving forward, continuous refinement and expansion of the system can further enhance its capabilities and contribute to sustainable farming practices.

Inferences

**Model Performance:**

Classification Models: Logistic Regression, Naive Bayes, Support Vector Machine, and K-Nearest Neighbors achieved high accuracy, precision, recall, and F1-score, indicating their effectiveness in predicting suitable crops.

Clustering Models: K-Means Clustering, Agglomerative Clustering, DBSCAN, and Gaussian Mixture yielded high silhouette scores, indicating good clustering quality.

Deep Learning Models: The Feedforward Neural Network and the Neural Network Model using PyTorch demonstrated strong performance in accuracy, precision, recall, and F1-score.

**Model Selection:**

The project evaluated a range of classification and clustering algorithms to identify the most suitable approaches for crop recommendation.

Both traditional machine learning algorithms and deep learning models were explored, providing a comprehensive analysis of different methodologies.

**Recommendation System Utility:**

The developed crop recommendation system offers valuable insights to farmers, aiding them in making informed decisions about crop selection based on soil nutrients, environmental conditions, and historical crop performance data.

The system's high accuracy and precision suggest its potential utility in real-world agricultural scenarios, where optimizing crop selection can significantly impact agricultural productivity and sustainability.