**Crop Prediction Using Machine Learning**

**Table of Contents**

1. **Introduction**

1.1 Overview

1.2 Purpose

1. **Literature Survey**

2.1 Existing Problem

2.2 Proposed Solution

1. **Theoretical Analysis**

3.1 Block Diagram

3.2 Hardware/Software Designing

1. **Experimental Investigations**
   * Data Collection & Preparation
   * Exploratory Data Analysis
   * Model Building
2. **Flowchart**
   * Control Flow of the Solution
3. **Results**
   * Model Performance Metrics
   * Visualizations
4. **Advantages & Disadvantages**
   * Pros of the Proposed Solution
   * Cons of the Proposed Solution
5. **Applications**
   * Areas Where the Solution Can Be Applied
6. **Conclusion**
   * Summary of the Project and Findings
7. **Future Scope**
   * Potential Enhancements
8. **BIBILOGRAPHY**
   * Links to Relevant Literature and Resources

15.**APPENDIX**

## 1. Introduction

### 1.1 Overview

Farming is a fundamental pillar of human civilization, responsible for sustaining the global population and ensuring food security. However, farmers face a significant challenge in choosing the most suitable crops for their farms. This decision is influenced by various factors, including soil composition, nutrient content (Nitrogen, Phosphorus, Potassium), pH levels, and regional climate conditions (humidity, rainfall, temperature).

### 1.2 Purpose

The primary objective of this project is to create a machine learning model that can assist farmers in making informed decisions regarding crop selection. By analyzing soil content and climate-oriented factors, this model aims to recommend the most suitable crop for a specific farm. This data-driven approach seeks to optimize farm yield and profitability while simplifying the decision-making process for farmers.

## 2. Literature Survey

### 2.1 Existing Problem

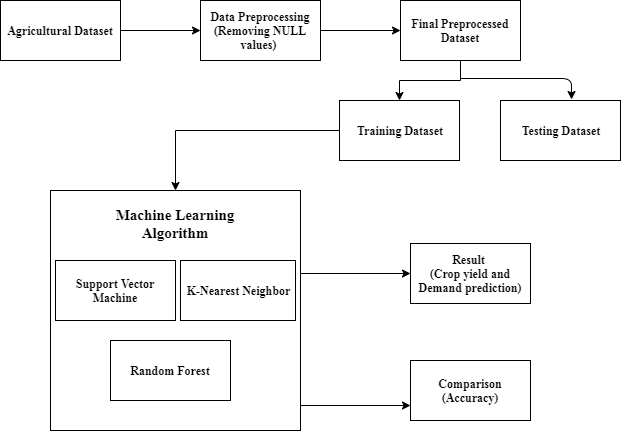
The agricultural industry faces a persistent challenge in determining the most appropriate crops for cultivation on a given farm. This problem stems from the variability in soil characteristics and regional climate conditions, making it essential to tailor crop selection to these factors.

### 2.2 Proposed Solution

This project proposes a machine learning-based solution to predict the ideal crop for a farm. By leveraging data on soil content and climate-related variables, the model can offer personalized crop recommendations. This solution has the potential to alleviate the complexity of crop selection for farmers.

## 3. Theoretical Analysis

### 3.1 Block Diagram



### 3.2 Hardware/Software Designing

#### Hardware:

* CPU : Intel/AMD processors
* Ram : 4GB
* Memory : 256 GB

#### Software:

* Python: The primary programming language used for developing and implementing the machine learning model.
* Flask: A web framework for creating a user interface and integrating the model.
* Scikit-Learn: A machine learning library for training and evaluating models.
* Pandas: Used for data manipulation and preprocessing.
* Seaborn and Matplotlib: Employed for data visualization.

## 4. Experimental Investigations

### Data Collection & Preparation

* Dataset: The project utilizes a dataset from Kaggle.
* Data Cleaning: Data preprocessing steps were undertaken to handle missing values, outliers, and ensure data quality.

### Exploratory Data Analysis

* Descriptive Statistical Analysis: Statistical summaries, including means, medians, and standard deviations, were generated for key dataset features.
* Visual Analysis: Visualizations, such as histograms and box plots, were used to gain insights into the distribution of soil and climate data.

### Model Building

The project involved training and evaluating multiple machine learning algorithms to predict crop recommendations.

### 5. Flowchart

START

USER INTERFACE

DATA INPUT PROCESSING

MACHINE LEARNING MODEL

DATA OUTPUT PROCESSING

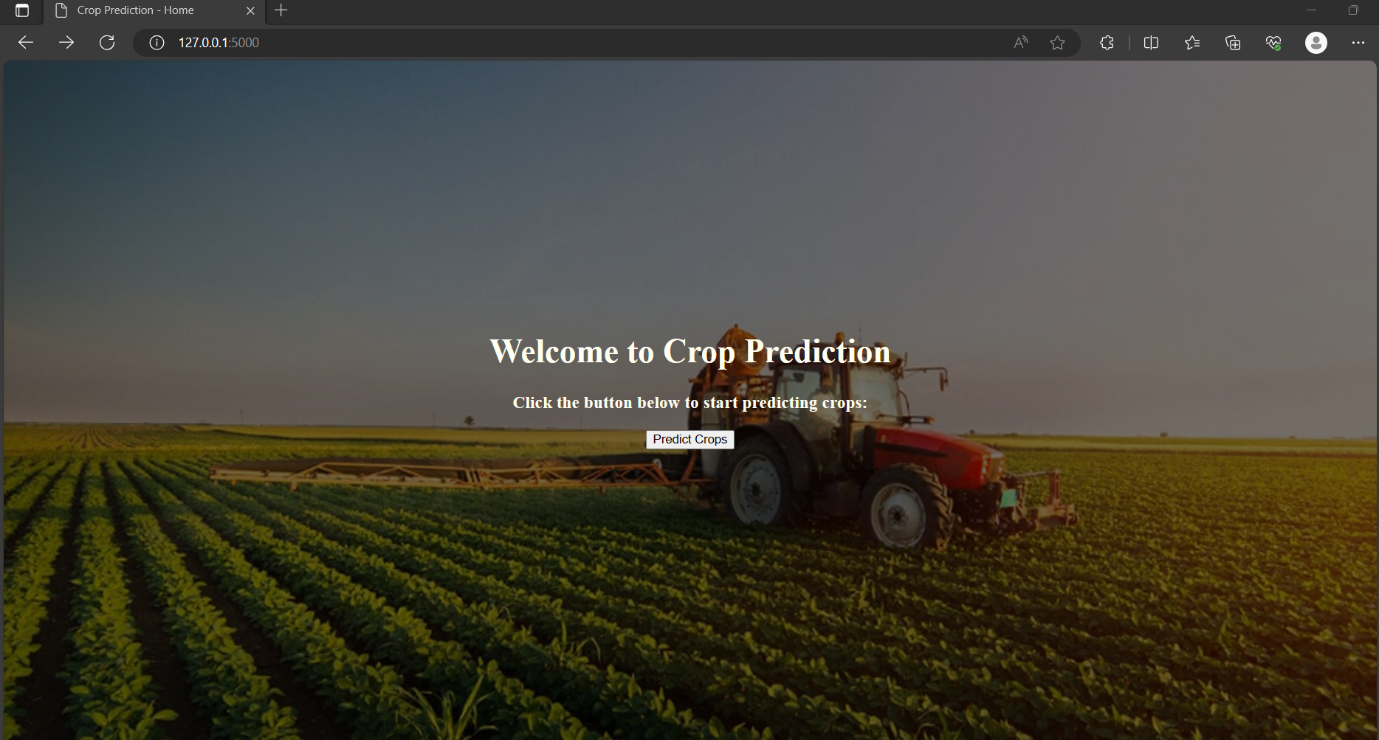
END

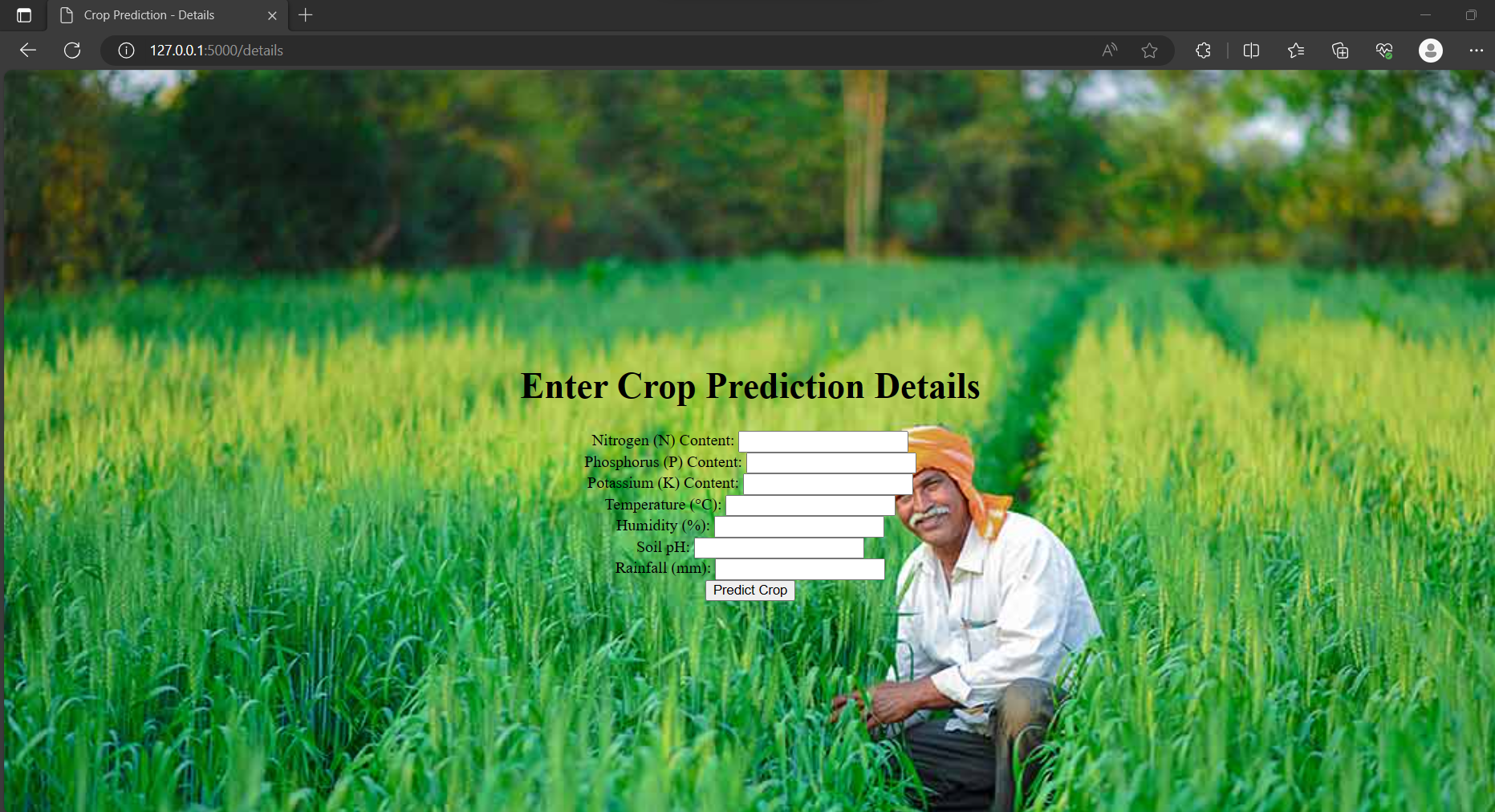
### 6. Results

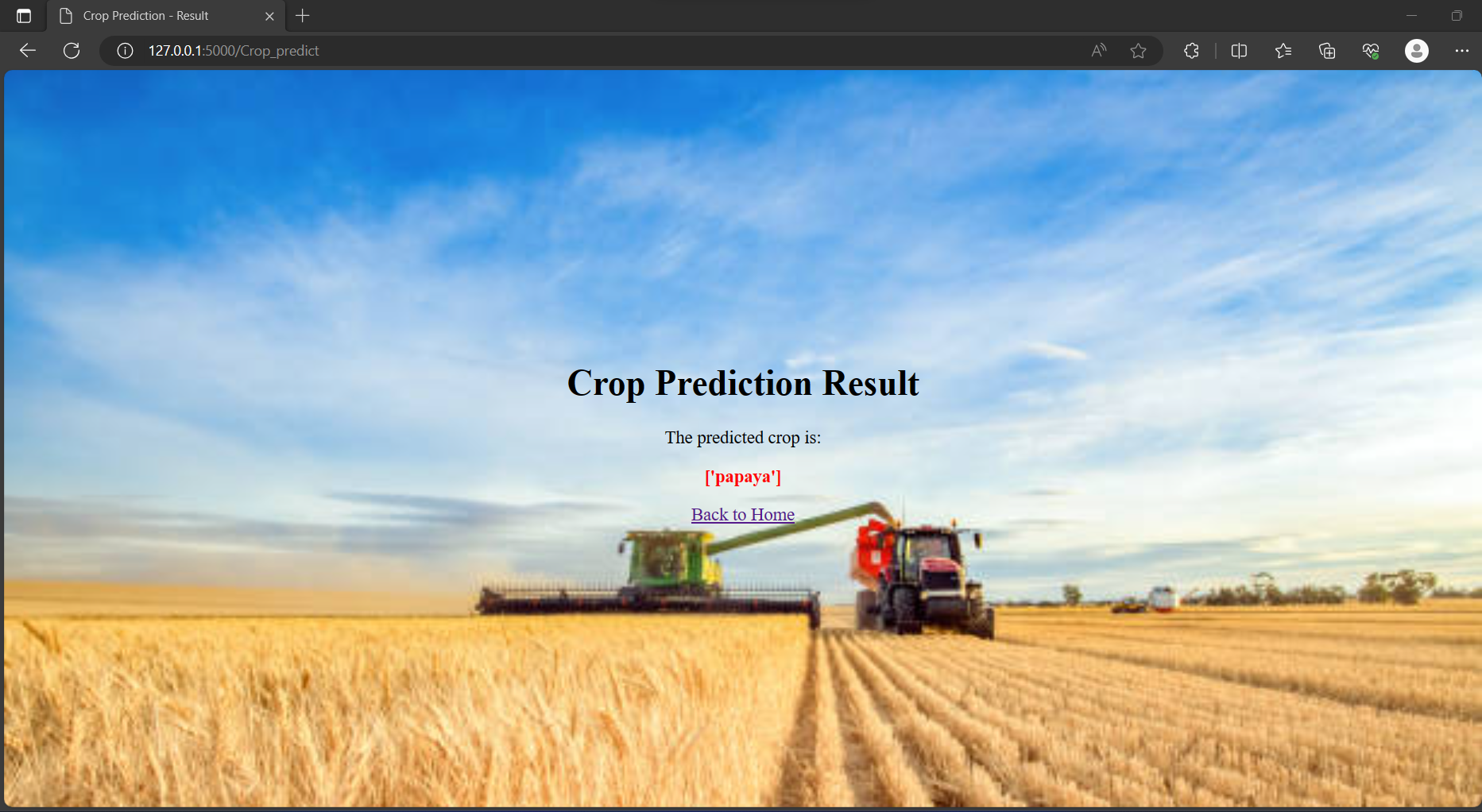
#### Model Performance Metrics

* KNN Model Accuracy: Achieved an accuracy of 98% on both training and testing data.
* SVM Model Accuracy: Achieved an accuracy of 97% on both training and testing data.
* Decision Tree Model Accuracy: Achieved an accuracy of 73.4% on testing data.
* Random Forest Model Accuracy: Achieved an accuracy of 99.5% on testing data.

#### Visualizations







### 7. Advantages & Disadvantages

#### Advantages:

* Data-driven crop recommendations enable farmers to make informed decisions.
* Improved food security through optimized crop selection.
* Promotes sustainable farming practices.

#### Disadvantages:

* Dependency on the availability and accuracy of data.
* Model limitations in complex and rapidly changing environmental conditions.

### 8. Applications

The solution's versatility allows it to be applied in various agricultural regions to assist farmers in optimizing crop selection.

### 9. Conclusion

In conclusion, this project has successfully developed a machine learning model that predicts crop recommendations based on soil and climate factors. The model's high accuracy and user-friendliness have the potential to revolutionize agricultural decision-making.

### 10. Future Scope

Future enhancements may include real-time weather data integration and mobile application development for farmer accessibility.

### 11. BIBILOGRAPHY

ML Concepts:

O Supervised learning: https://www.javatpoint.com/supervised-machine-learning o K Nearest Neighbours: https://www.javatpoint.com/k-nearestneighboralgorithm- for-machine-learning

o SVM: https://www.javatpoint.com/machine-learning-support- vectormachinealgorithm

o Decision tree: https://www.javatpoint.com/machine-learning- decisiontreeclassificationalgorithm

o Random forest: https://www.javatpoint.com/machine-learning- randomforestalgorithm

o Evaluation metrics: https://www.analyticsvidhya.com/blog/2019/08/11important- modelevaluationerror-metrics/

• Flask Basics: <https://www.youtube.com/watch?v=lj4I_CvBnt0>

**APPENDIX**

* 1. **Source Code**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Crop Prediction - Home</title>

    <link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='styles.css') }}">

</head>

<body class="index">

    <h1>Welcome to Crop Prediction</h1>

    <h3>Click the button below to start predicting crops:</h3>

    <a href="/details"><button >Predict Crops</button></a>

</body>

</html>

**Details.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Crop Prediction - Details</title>

    <link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='styles.css') }}">

</head>

<body class="details">

    <h1>Enter Crop Prediction Details</h1>

    <form action="/Crop\_predict" method="POST">

        <label for="N">Nitrogen (N) Content:</label>

        <input type="number" id="N" name="N" required><br>

        <label for="P">Phosphorus (P) Content:</label>

        <input type="number" id="P" name="P" required><br>

        <label for="K">Potassium (K) Content:</label>

        <input type="number" id="K" name="K" required><br>

        <label for="temperature">Temperature (°C):</label>

        <input type="number" id="temperature" name="temperature" required><br>

        <label for="humidity">Humidity (%):</label>

        <input type="number" id="humidity" name="humidity" required><br>

        <label for="ph">Soil pH:</label>

        <input type="number" id="ph" name="ph" step="0.01" required><br>

        <label for="rainfall">Rainfall (mm):</label>

        <input type="number" id="rainfall" name="rainfall" required><br>

        <input type="submit" value="Predict Crop">

    </form>

</body>

</html>

**Crop\_predict.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Crop Prediction - Result</title>

    <link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='styles.css') }}">

</head>

<body class="predict">

    <h1>Crop Prediction Result</h1>

    <p>The predicted crop is:</p>

    <p class="result-text"><strong>{{ prediction\_text }}</strong></p>

    <p><a href="/">Back to Home</a></p>

</body>

</html>

**App.py**

import pickle

from flask import Flask, render\_template, request

import pandas as pd

import numpy as np

model = pickle.load(open('model.pkl', 'rb'))

app = Flask(\_\_name\_\_, static\_folder='static')

@app.route('/')

def home():

    return render\_template('index.html')

@app.route('/details') # rendering the html template

def index() :

    return render\_template('details.html')

@app.route('/Crop\_predict', methods=['GET','POST'])

def predict() :

    # loading model which we saved

    N = float(request.form['N'])

    P = float(request.form['P'])

    K = float(request.form['K'])

    temperature= float(request.form['temperature'])

    humidity = float(request.form['humidity'])

    ph= float(request.form['ph'])

    rainfall = float(request.form['rainfall'])

    prediction =model.predict(pd.DataFrame([[N,P,K, temperature, humidity, ph, rainfall]], columns= ['N', 'P', 'K', 'temperature', 'humidity', 'ph',

       'rainfall']))

    return render\_template('Crop\_predict.html', prediction\_text ="{}".format(prediction))

if \_\_name\_\_ == '\_\_main\_\_':

    app.run()