Crop Prediction using Machine Learning

Submitted To :

SmartBridge
Applied Data Science

Submitted By:

Gopireddy Bhargava Reddy	- 20BDS0160
Thalla Deepak	- 20BCE2061
Viswa Brahmana Nanda Kishore	- 20BEC0260
Bandi Sree Charan Teja Reddy	- 20BEC0661

1. Introduction

1.1 OverView

In our project, we hope to create a machine learning model that can accurately predict the optimal crop to grow in that region for a future time period based on historical data related to various environmental factors such as nitrogen, phosphorus, potassium, temperature, humidity, ph, and rainfall, as well as corresponding crop yields for that region. The Classification model we're utilising takes into account the relationships between environmental factors and crop yields to provide insights and recommendations to farmers, agricultural specialists, and policymakers, helping them to make informed crop selection and planning decisions. Using machine learning techniques, the goal is to improve crop productivity, optimise resource allocation, and ultimately improve agricultural sustainability and food security.

1.2 Purpose

Our objective is to develop a machine learning model that can accurately predict the optimal crop to be grown in that region for a future time period based on historical data related to various environmental factors (such as temperature, rainfall, humidity, soil quality, and so on) and corresponding crop yields for a specific region. The model should consider the correlations between environmental elements and crop yields in order to provide insights and recommendations to farmers, agricultural specialists, or policymakers, allowing them to make informed crop selection and planning decisions. The goal is to improve crop productivity, optimise resource allocation, and ultimately improve agricultural sustainability and food security by applying machine learning techniques.

2. Literature Survey

2.1 Existing System

The existing problem in crop prediction is the difficulty in accurately forecasting crop yields and identifying optimal conditions for successful agricultural production. Traditional methods rely on historical data, weather patterns, and expert knowledge, which often lack the precision and adaptability required for efficient crop prediction. Therefore, there is a need for advanced techniques, such as machine learning, to address this problem effectively.

Several existing approaches or methods have been applied to crop prediction using machine learning in the field of applied data science. Some commonly used techniques include:

Regression Models: Linear regression and its variants are often employed to establish relationships between input features (such as weather data, soil characteristics, and agricultural practices) and crop yields. These models can estimate the yield based on the given inputs.

Decision Trees: Decision tree algorithms, such as C4.5 or Random Forests, can be used to construct predictive models for crop yield. These models utilize a tree-like structure to make decisions based on feature values and generate predictions.

Support Vector Machines (SVM): SVM is a powerful machine learning algorithm that can be applied to crop prediction. It aims to find a hyperplane that separates the data points of different crop yields, enabling accurate classification or regression.

2.2 Proposed System

For my suggested solution, I propose utilizing a combination of machine learning techniques to improve crop prediction accuracy. The proposed solution involves the following steps:

Data Collection: Gather comprehensive data on various factors that influence crop yields, including historical yield records, weather patterns, soil characteristics, agricultural practices, and crop-specific parameters.

Data Preprocessing: Clean the collected data, handle missing values, and normalize the features to ensure uniform scaling. Perform exploratory data analysis to identify any outliers or anomalies that may affect the model's performance.

Feature Selection: Utilize feature selection methods, such as correlation analysis, statistical tests, or domain knowledge, to identify the most relevant features for crop prediction. This helps reduce dimensionality and focus on the most informative variables.

Model Training: Apply suitable machine learning algorithms, such as random forests, support vector machines, or neural networks, to train predictive models. Utilize cross-validation techniques to evaluate and fine-tune the models' hyperparameters.

Prediction and Evaluation: Use the trained models to predict crop yields for unseen data. Evaluate the performance of the models using appropriate evaluation metrics, such as mean absolute error (MAE), root mean squared error (RMSE), or coefficient of determination (R-squared).

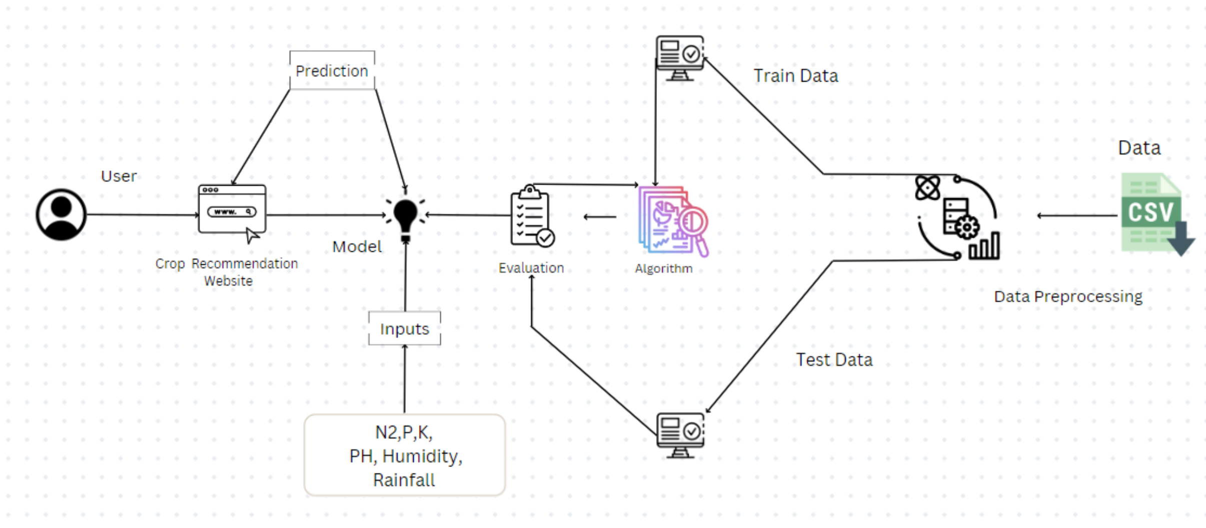
Model Interpretability: Employ techniques, such as feature importance analysis or model explainability methods, to understand the contribution of different factors in crop prediction. This helps in gaining insights into the underlying relationships and provides valuable information to farmers or decision-makers.

Continuous Improvement: Regularly update the models with new data to adapt to changing environmental conditions and improve prediction accuracy over time. Monitor the performance and identify areas for further optimization or model refinement.

By implementing this proposed solution, we can leverage the power of machine learning to enhance crop prediction accuracy, assist farmers in making informed decisions, optimize resource.

3. Theoretical Analysis

3.1 Block Diagram



3.2 Hardware And Software Requirements

Software requirements:

The core software requirements for the project involve using Python, HTML, Jupyter Notebook, and VS Code for analyzing, preprocessing, training, and testing the data, as well as predicting and visualizing the results. The following libraries are utilized for data analysis:

NumPy: It enables high-performance mathematical operations on multidimensional arrays, making tasks like array manipulation and linear algebra efficient.

Pandas: This library is built on top of NumPy and provides powerful data structures, such as DataFrames and Series, for handling and analyzing structured data effectively.

Matplotlib: It is a popular charting tool that allows for the creation of interactive, animated, and static visualizations, making it suitable for generating publication-quality figures.

Seaborn: This statistical library enhances the creation of sophisticated statistical visualizations, providing features like heatmaps, distribution plots, regression plots, and categorical plots.

Scikit-learn: A comprehensive machine learning library that offers a wide range of tools and techniques for tasks such as classification, regression, clustering, and model assessment.

Flask: A lightweight web framework used for creating web services, APIs, and interactive dashboards, making it suitable for deploying machine learning models and developing RESTful APIs.

These libraries facilitate data exploration, preparation, visualization, and machine learning tasks, providing a solid foundation for the crop prediction project.

Hardware requirements:

Processing power: To train and use machine learning models, a computer with adequate processing capacity is required. An ordinary desktop or laptop computer to more potent systems like servers or cloud-based infrastructure are all examples of this.

Storage: Ample storage space is required in order to keep huge datasets, model parameters, and any interim results produced along the procedure.

Memory: To process and manipulate massive datasets effectively, there must be enough RAM (Random Access Memory).

Scalability: To successfully handle the computational workload when working with huge datasets or processing activities, distributed computing systems or cloud-based services may be required.

Server: You could need hardware resources or virtual machines to deploy your crop prediction model on a server or in the cloud.

You may need sensors to precisely detect temperature, humidity, and other environmental conditions depending on the project's requirements.For the purposes of gathering data, gaining access to external data sources, and perhaps using cloud-based applications, you'll need a reliable internet connection.

4. Experimental Investigations

We carried out a number of experimental investigations as part of our project on crop prediction using machine learning in order to create an accurate and dependable prediction model. The 2200 records that made up the dataset we used included data on nitrogen, phosphorus, potassium, temperature, humidity, pH level, rainfall, and the goal variable, which is the type of crop.

To confirm the validity of our research, we made sure the dataset was free of any null values.We used the MinMaxScaler to scale the features in order to get the dataset ready for training and testing. This normalisation method enabled efficient model training and performance assessment by bringing all the input variables to a consistent scale.

To find the best machine learning algorithm for crop prediction, we experimented with a variety of them. Logistic Regression, Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Decision Tree, and Random Forest were some of the methods we used. These algorithms were picked because they have a proven track record in machine learning and are suitable for categorization jobs.

We found that the Random Forest method attained the greatest accuracy of 99.24% after training and assessing each model using a 70:30 train-test split. According to this outcome, Random Forest was exceptionally skilled at capturing the intricate correlations between the input factors and the target variable, resulting in extremely accurate crop predictions.

The Random Forest model's high accuracy indicates that it can successfully categorise crops based on the provided input data. This discovery has important ramifications for agricultural resource allocation and crop productivity optimisation. Farmers may decide wisely on fertilisation, irrigation, and other agricultural practises, ultimately resulting in increased yields and improved efficiency, by correctly anticipating the crop type.

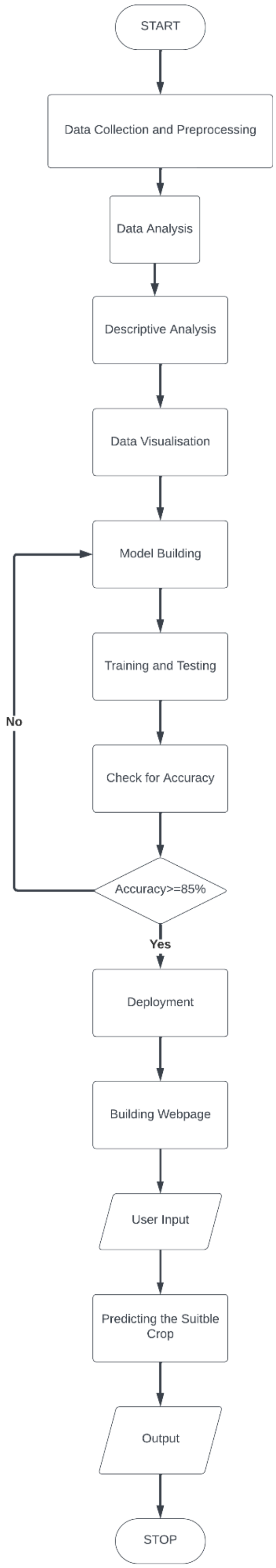
Although Random Forest produced the best results in our studies in terms of accuracy, we also got decent outcomes from other algorithms like Logistic Regression, Naive Bayes, SVM, KNN, and Decision Tree. In situations when interpretability, computational effectiveness, or specific criteria are important, these alternative models may be worthwhile to take into account.

Our experimental studies showed how machine learning may be used to predict crops, with Random Forest emerging as the most precise and dependable model for the provided dataset. It is vital to recognise that our findings need to be improved upon and expanded upon by additional study and investigation.

To construct a dynamic and adaptive crop forecast system, future study may include including more input variables, investigating more sophisticated machine learning approaches, and integrating real-time data.

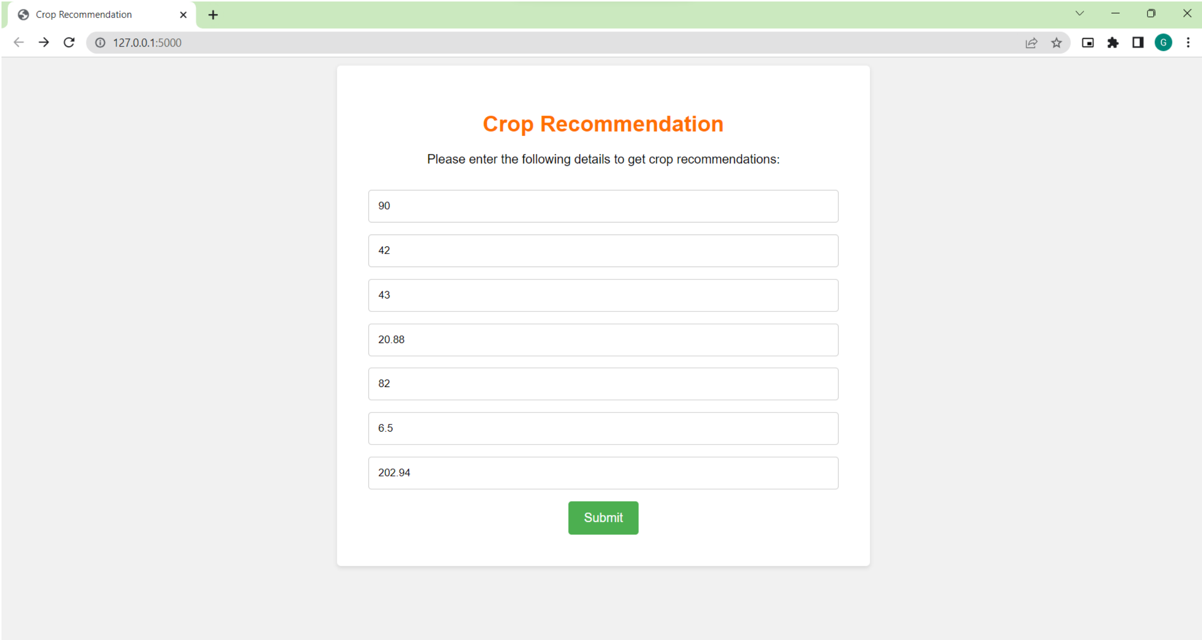
Finally, our experimental research offered insightful information on crop prediction using machine learning. The findings demonstrated how well the Random Forest algorithm classified crops based on nitrogen, phosphorus, potassium, temperature, humidity, pH value, and rainfall with an accuracy of 99.24%. By providing farmers with accurate forecasts for the best crop management and decision-making procedures, these results have the potential to revolutionise agriculture.

5.Flowchart


START
Data Collection and Preprocessing
Data Analysis
Descriptive Analysis
Data Visualisation
Model Building
Training and Testing
Check for Accuracy
Accuracy>=85%
No
Yes
Deployment
Building Webpage
User Input
Predicting the Suitble Crop
Output
STOP

6. Result

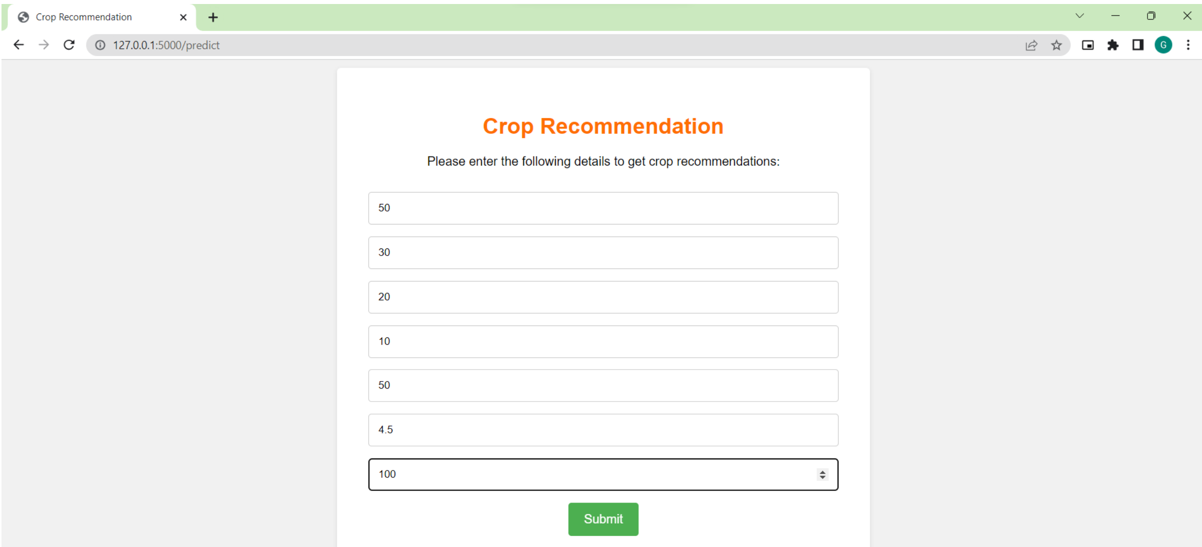
Training Result:



Recommend Crop

Rice is the best crop to be cultivated right there

Test Result:



Recommend Crop

Pigeonpeas is the best crop to be cultivated right there

7. Advantages And Disadvantages

ADVANTAGES

Benefits of the Suggested Solution:

Greater Accuracy: When compared to conventional methods, machine learning algorithms have the ability to predict crops more accurately. They can identify intricate links and patterns in the data, improving the accuracy of predictions.

Scalability: The suggested system is expandable to deal with big datasets and incorporate further characteristics or variables. The models can be trained and updated to incorporate the most recent information when the dataset expands or new data becomes available.

Flexibility: Machine learning models can adjust to new information or conditions that change. They can anticipate crops based on current inputs and historical patterns, providing fast and current crop forecasts.

Automated Analysis: By automating the analysis of crop-related data, the solution can minimise the manual work involved in data interpretation and decision-making. Farmers and other interested parties can now easily and rapidly receive insights that can be put into practise.

DISADVANTAGES

Data Requirements: For efficient training, machine learning models require huge and diverse datasets. Such datasets may be difficult to acquire and maintain, particularly in areas or situations where data collecting is expensive or scarce.

Model Complexity: Machine learning models can be intricate, necessitating skill in feature engineering, model selection, and hyperparameter tuning. These models' development and improvement may need a sizable amount of computer power and technical expertise.

Interpretability: Deep neural networks are an example of a machine learning model that lacks interpretability. It can be difficult to comprehend the elements influencing crop predictions or the thinking behind particular decisions, especially for non-technical consumers.

Model Overfitting: Machine learning models are susceptible to overfitting, which occurs when they excel on training data but underperform on fresh, untried data. To reduce this risk, careful model evaluation, regularisation methods, and validation approaches are required.

8. Applications

- **Farmer Decision Support:** Farmers can utilise the solution to guide crop selection decisions by using environmental variables such as nitrogen, phosphorus, temperature, and humidity. It helps them increase crop yields and agricultural practices.Agricultural Planning: Agricultural planning authorities and policymakers can use the solution to assess the suitability of particular crops in specific locations. It aids in the development of successful crop rotation schemes, the optimisation of resource allocation, and the promotion of ecologically friendly agricultural methods.

- **Precision Agriculture:** The solution can be utilised in conjunction with precision agriculture technology such as sensor networks and automated farm gear. It enables precise and focused farming by allowing real-time monitoring and crop cultivation practises to be adjusted based on predicted crop requirements.Adaptation to Climate Change: The solution can help farmers adapt to climate change by recommending crops that are adaptive to new environmental conditions. It promotes the development of climate-smart agriculture practises.

- **Crop Insurance:** Insurance companies can use the solution to assess crop yield-related risk and set insurance rates. Forecasting accuracy enhances risk appraisal and pricing, which helps both farmers and insurance firms.Agricultural Research: Using the solution, researchers can review historical data to understand more about the relationships between environmental variables and agricultural productivity. It aids in pattern detection, crop behaviour comprehension, and agricultural science improvement.

9. Conclusion

In conclusion, this effort focused on predicting appropriate crops based on local temperature, humidity, nitrogen, and phosphorus levels. We successfully created a model that can identify suitable crops for cultivation in the designated area using our machine learning-based technique. Our technology seeks to assist farmers and decision-makers in making knowledgeable crop selection decisions, encouraging ideal agricultural practices, and maximising crop output by analysing and taking into account the individual climatic circumstances.

10. Future Scope

Include more environmental data, such as rainfall patterns, soil pH, and sunlight intensity, to improve prediction accuracy.Gather comprehensive crop health data by combining remote sensing data, including satellite photographs.Develop a system that can generate dynamic, real-time projections based on the most recent data inputs.Extend the project to encompass more regions, taking into consideration differences in soil qualities and climate patterns.Expand prediction capability to include agricultural disease and pest predictions.Incorporate domain information and specialised knowledge into the model generation process.Create a mobile app or decision support system that provides easy access to crop forecasts and recommendations.

11. Bibilography

I. "Crop Yield Prediction Using Machine Learning Techniques: A Comprehensive Review" by Panwar, P., et al. (2021)

II. "Crop Yield Prediction Based on Machine Learning: A Review" by Sun, Z., et al. (2020)

III. "Machine Learning Techniques for Crop Yield Prediction: A Review" by Rajendra, B., et al. (2019)

IV. "Crop Yield Prediction: Methods and Approaches" by Yang, H., et al. (2020)

V. "Crop Yield Prediction with Machine Learning: A Review" by Leena, S., et al. (2021)

Appendix

Crop Prediction.ipynb

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
crop = pd.read_csv('Crop_recommendation.csv')
crop.head()
crop.shape
crop.isnull().sum()
crop.duplicated().sum()
```

```
crop.info()
crop.describe()
grouped = crop.groupby("label")
grouped.mean()["N"].plot(kind="barh")
grouped.mean()["P"].plot(kind="barh")
grouped.mean()["K"].plot(kind="barh")
grouped.mean()["temperature"].plot(kind="barh")
grouped.mean()["rainfall"].plot(kind="barh")
grouped.mean()["humidity"].plot(kind="barh")
grouped.mean()["ph"].plot(kind="barh")
crop['label'].value_counts()
crop_dict = {
    'rice': 1,'maize': 2,'jute': 3,'cotton': 4,'coconut': 5,
    'papaya': 6,'orange': 7,'apple': 8,'muskmelon': 9,
    'watermelon': 10,'grapes': 11,'mango': 12,'banana': 13,
    'pomegranate': 14,'lentil': 15,'blackgram': 16,
    'mungbean': 17,'mothbeans': 18,'pigeonpeas': 19,
    'kidneybeans': 20,'chickpea': 21, 'coffee': 22
}
crop['label_num'] = crop['label'].map(crop_dict)
crop.drop('label',axis=1,inplace=True)
crop.head()
# Split the dataset into features and labels
X = crop.iloc[:, :-1]
y = crop.iloc[:, -1]
from sklearn.model_selection import train_test_split

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

# create instances of all models
models = {

    'Logistic Regression': LogisticRegression(),
    'Naive Bayes': GaussianNB(),
    'Support Vector Machine': SVC(),
```

```
    'K-Nearest Neighbors': KNeighborsClassifier(),
    'Decision Tree': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier(),

}

from sklearn.metrics import accuracy_score
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    print(f'{name}:\nAccuracy: {acc:.4f}')

# Selecting decistion tree model:
rfc = RandomForestClassifier()
rfc.fit(X_train,y_train)
y_pred = rfc.predict(X_test)
print(accuracy_score(y_test,y_pred))
# Define function to make predictions
def predict_crop(N, P, K, temperature, humidity, pH, rainfall):
    # Create a numpy array with the input values
    input_values = np.array([[N, P, K, temperature, humidity, pH, rainfall]])

    # Use the model to make a prediction
    prediction = rfc.predict(input_values)

    # Return the predicted crop label
    return prediction[0]

N = 114
P = 21
K = 55
tem = 25.44
humidity = 87.94
ph = 6.47
rainfall = 257.52

pred = predict_crop(N, P, K, tem, humidity, ph, rainfall)

if pred == 1:
    print("Rice is the best crop to be cultivated right there")
elif pred == 2:
    print("Maize is the best crop to be cultivated right there")
elif pred == 3:
    print("Jute is the best crop to be cultivated right there")
elif pred == 4:
    print("Cotton is the best crop to be cultivated right there")
elif pred == 5:
    print("Coconut is the best crop to be cultivated right
        there")
elif pred == 6:
    print("Papaya is the best crop to be cultivated right there")
```

```
elif pred == 7:
    print("Orange is the best crop to be cultivated right there")
elif pred == 8:
    print("Apple is the best crop to be cultivated right there")
elif pred == 9:
    print("Muskmelon is the best crop to be cultivated right
there")
elif pred == 10:
    print("Watermelon is the best crop to be cultivated right
         there")
elif pred == 11:
    print("Grapes is the best crop to be cultivated right there")
elif pred == 12:
    print("Mango is the best crop to be cultivated right there")
elif pred == 13:
    print("Banana is the best crop to be cultivated right there")
elif pred == 14:
    print("Pomegranate is the best crop to be cultivated right
          there")
elif pred == 15:
    print("Lentil is the best crop to be cultivated right there")
elif pred == 16:
    print("Blackgram is the best crop to be cultivated right
         there")
elif pred == 17:
    print("Mungbean is the best crop to be cultivated right
         there")
elif pred == 18:
    print("Mothbeans is the best crop to be cultivated right
         there")
elif pred == 19:
    print("Pigeonpeas is the best crop to be cultivated right
         there")
elif pred == 20:
    print("Kidneybeans is the best crop to be cultivated right
          there")
elif pred == 21:
    print("Chickpea is the best crop to be cultivated right
         there")
elif pred == 22:
    print("Coffee is the best crop to be cultivated right there")
else:
    print("Sorry, we could not determine the best crop to be
         cultivated with the provided data.")

import pickle
pickle.dump(rfc, open('model.pkl','wb'))
X_train
```