!pip install scikit-learn pandas joblib

import pandas as pd

df = pd.read\_csv("/content/crop\_recommendation.csv")

df.head()

import numpy as np

import pickle

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import accuracy\_score, classification\_report

import matplotlib.pyplot as plt

import seaborn as sns

print("Dataset Info:")

print(df.info())

df = df.drop\_duplicates()

def remove\_outliers(data):

    numeric\_data = data.select\_dtypes(include=['number'])  # Select only numeric columns

    Q1 = numeric\_data.quantile(0.25)  # 25th percentile

    Q3 = numeric\_data.quantile(0.75)  # 75th percentile

    IQR = Q3 - Q1  # Interquartile Range

    lower\_bound = Q1 - 1.5 \* IQR

    upper\_bound = Q3 + 1.5 \* IQR

    filtered\_data = data[~((numeric\_data < lower\_bound) | (numeric\_data > upper\_bound)).any(axis=1)]

    return filtered\_data

df\_cleaned = remove\_outliers(df)

print(f"Original Rows: {df.shape[0]}, Cleaned Rows: {df\_cleaned.shape[0]}, Deleted: {df.shape[0] - df\_cleaned.shape[0]}")

X = df\_cleaned.drop(columns=['label']) # Features

y = df\_cleaned['label'] # Target variable

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

model = SVC(kernel='linear', random\_state=42)

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Model Accuracy: {accuracy}")

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

import matplotlib.pyplot as plt

import seaborn as sns

# Boxplot for detecting outliers

plt.figure(figsize=(10,5))

sns.boxplot(data=df[['n', 'p', 'k', 'temperature', 'humidity', 'ph', 'rainfall']])

plt.show()

from sklearn.metrics import accuracy\_score, classification\_report

# Predict on test data

y\_pred = model.predict(X\_test)

# Evaluate performance

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Model Accuracy: {accuracy}")

custom\_input = np.array([[90, 42, 43, 20.5, 82.0, 6.5, 200]])

custom\_input\_scaled = scaler.transform(custom\_input)

predicted\_crop = model.predict(custom\_input\_scaled)

print("Recommended Crop:", predicted\_crop[0])

test\_samples = np.array([

    [100, 30, 50, 25, 100, 6.8, 250],  # Sample 1

    [70, 50, 100, 28, 85, 6.0, 300],   # Sample 2

])

# Scale input features

test\_samples\_scaled = scaler.transform(test\_samples)

# Predict crops

predictions = model.predict(test\_samples\_scaled)

print("Predicted Crops:", predictions)

import pickle

# Save the trained model

with open("/content/crop\_recommender.pkl", "wb") as file:

    pickle.dump(model, file)

# Save the scaler (needed for future predictions)

with open("/content/scaler.pkl", "wb") as file:

    pickle.dump(scaler, file)

print("Model and scaler saved successfully!")