AI-FARM ASSISTANT

SUBMITTED BY: JINU JOJI

REG NO: 23BCE8495

Smart Farming with AI: Helping Farmers Make Better Decisions

This tool is here to help farm workers make better decisions about their crops by using artificial intelligence and the evaluation of information. By studying a wide number of factors such as soil moisture, temperature, humidity, and many other key conditions, the model accurately predicts the degree to which a crop is healthy or specifically recommends the best crops to grow. The system takes actual farm data, tidies it up, processes it, along with training a machine learning model (Random Forest Classifier) to recognize many patterns and create precise predictions. With this, farmers can:

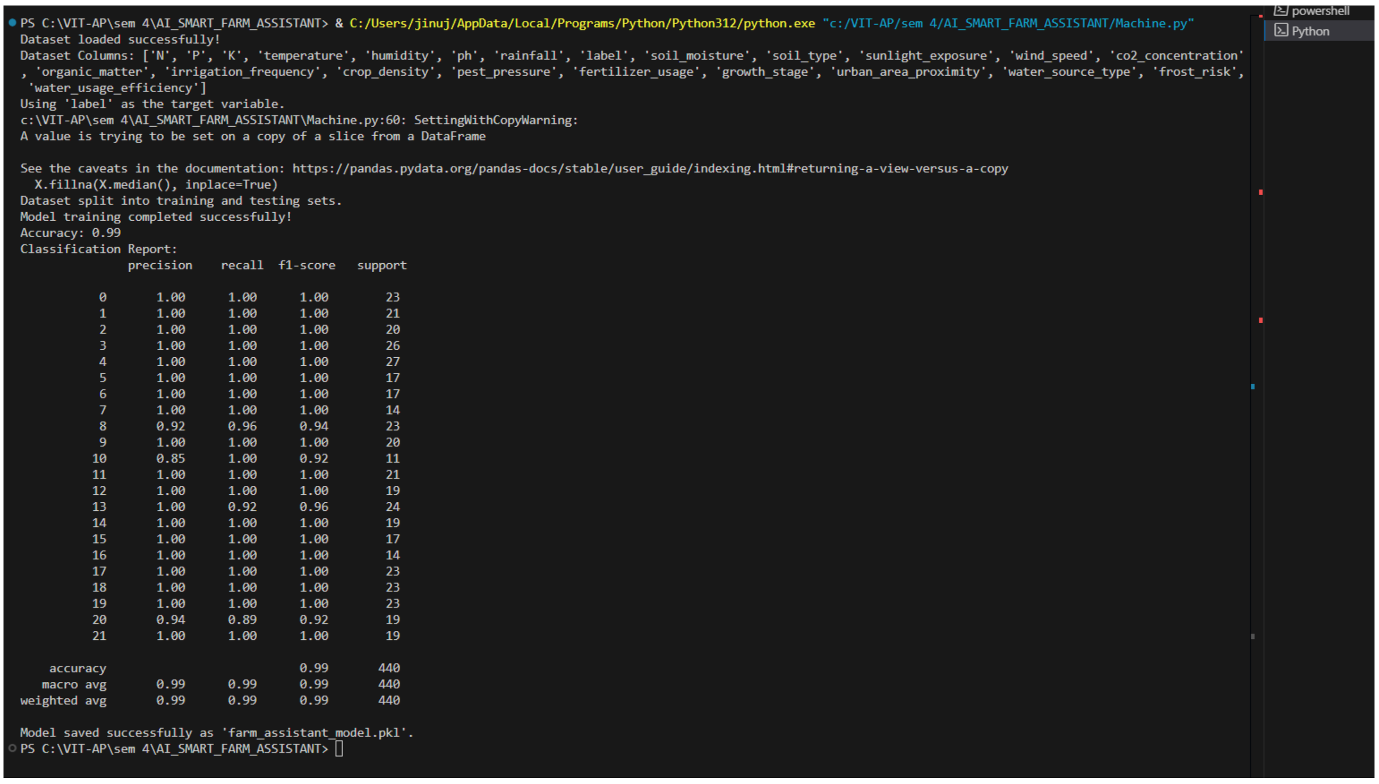
Recognize when crops are unhealthy quickly and then act decisively to prevent wide-ranging harm. To cut down on waste, use water, fertilizers, along with other resources in an efficient way.

Make many informed decisions. Base these decisions on real-time farm conditions.

This tool helps farmers increase their yields to a great extent, substantially reduce risks, and make farming more sustainable and profitable to a large degree in the long run by using AI in agriculture.

CODE:

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification_report, accuracy_score

import joblib

from sklearn.preprocessing import LabelEncoder


try:
    data = pd.read_csv('Crop_recommendationV2.csv')
    print("Dataset loaded successfully!")
except FileNotFoundError:
    print("Error: Dataset file 'Crop_recommendationV2.csv' not found!")
    exit()
except pd.errors.EmptyDataError:
    print("Error: Dataset is empty!")
    exit()
except pd.errors.ParserError:
    print("Error: Dataset file is corrupt or improperly formatted!")
    exit()

print("Dataset Columns:", list(data.columns))

possible_targets = ['crop_health', 'label', 'growth_stage']
target_column = None

for col in possible_targets:
    if col in data.columns:
        target_column = col
        break

if target_column is None:
    print("Error: No valid target variable found in dataset!")
    exit()

print(f"Using '{target_column}' as the target variable.")

if data[target_column].dtype == 'object':
    le = LabelEncoder()
    y = le.fit_transform(data[target_column])
else:
    y = data[target_column]

numerical_columns = data.select_dtypes(include=['number']).columns.tolist()
exclude_columns = [target_column]  # Exclude target variable from features
feature_columns = [col for col in numerical_columns if col not in exclude_columns]

if not feature_columns:
    print("Error: No numerical features available for training!")
    exit()

X = data[feature_columns]

X.fillna(X.median(), inplace=True)

try:
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    print("Dataset split into training and testing sets.")
except ValueError as e:
    print(f"Error during train-test split: {e}")
    exit()

try:
    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    print("Model training completed successfully!")
except Exception as e:
    print(f"Error during model training: {e}")
    exit()

try:
    y_pred = model.predict(X_test)
except Exception as e:
    print(f"Error during prediction: {e}")
    exit()


try:
    accuracy = accuracy_score(y_test, y_pred)
    print(f"Accuracy: {accuracy:.2f}")
    print("Classification Report:\n", classification_report(y_test, y_pred))
except Exception as e:
    print(f"Error during evaluation: {e}")
try:
    joblib.dump(model, 'farm_assistant_model.pkl')
    print("Model saved successfully as 'farm_assistant_model.pkl'.")
except Exception as e:
    print(f"Error while saving the model: {e}")
```



```
PS C:\VIT-AP\sem 4\AI_SMART_FARM_ASSISTANT> & C:/Users/jinuj/AppData/Local/Programs/Python/Python312/python.exe "c:/VIT-AP/sem 4/AI_SMART_FARM_ASSISTANT/Machine.py"
Dataset loaded successfully!
Dataset Columns: ['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label', 'soil_moisture', 'soil_type', 'sunlight_exposure', 'wind_speed', 'co2_concentration'
, 'organic_matter', 'irrigation_frequency', 'crop_density', 'pest_pressure', 'fertilizer_usage', 'growth_stage', 'urban_area_proximity', 'water_source_type', 'frost_risk',
 'water_usage_efficiency']
Using 'label' as the target variable.
c:\VIT-AP\sem 4\AI_SMART_FARM_ASSISTANT\Machine.py:60: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X.fillna(X.median(), inplace=True)
Dataset split into training and testing sets.
Model training completed successfully!
Accuracy: 0.99
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        23
           1       1.00      1.00      1.00        21
           2       1.00      1.00      1.00        20
           3       1.00      1.00      1.00        26
           4       1.00      1.00      1.00        27
           5       1.00      1.00      1.00        17
           6       1.00      1.00      1.00        17
           7       1.00      1.00      1.00        14
           8       0.92      0.96      0.94        23
           9       1.00      1.00      1.00        20
          10       0.85      1.00      0.92        11
          11       1.00      1.00      1.00        21
          12       1.00      1.00      1.00        19
          13       1.00      0.92      0.96        24
          14       1.00      1.00      1.00        19
          15       1.00      1.00      1.00        17
          16       1.00      1.00      1.00        14
          17       1.00      1.00      1.00        23
          18       1.00      1.00      1.00        23
          19       1.00      1.00      1.00        23
          20       0.94      0.89      0.92        19
          21       1.00      1.00      1.00        19

    accuracy                           0.99       440
   macro avg       0.99      0.99      0.99       440
weighted avg       0.99      0.99      0.99       440

Model saved successfully as 'farm_assistant_model.pkl'.
PS C:\VIT-AP\sem 4\AI_SMART_FARM_ASSISTANT>
```