SmartInternz HackChallenge 2021

AI assisted Farming for Crop Recommendation & Farm Yield Prediction Application

Team: FOUR HORSEMAN

- ## Introduction
  - **Overview**

    An Artificial Intelligence assisted digital farming application is used to recommend Crop to sow in farm based on soil nutrients values and weather data, it also predicts the yield based on recommended crop and location of farm.

    It will be beneficial for farmers to estimate their cost and efforts to put and expect the maximum output using this system.

  - **Purpose**

    The purpose of the system to recommend the most suitable Crop to sow in the Farm based on its land nutrients and weather of the region and also forecast the expected yield to be produced on basis of past yield production of that region. Thus, the farmers can utilize the time, efforts and money to get maximum output.

- Literature Survey
    - **Existing Problem**

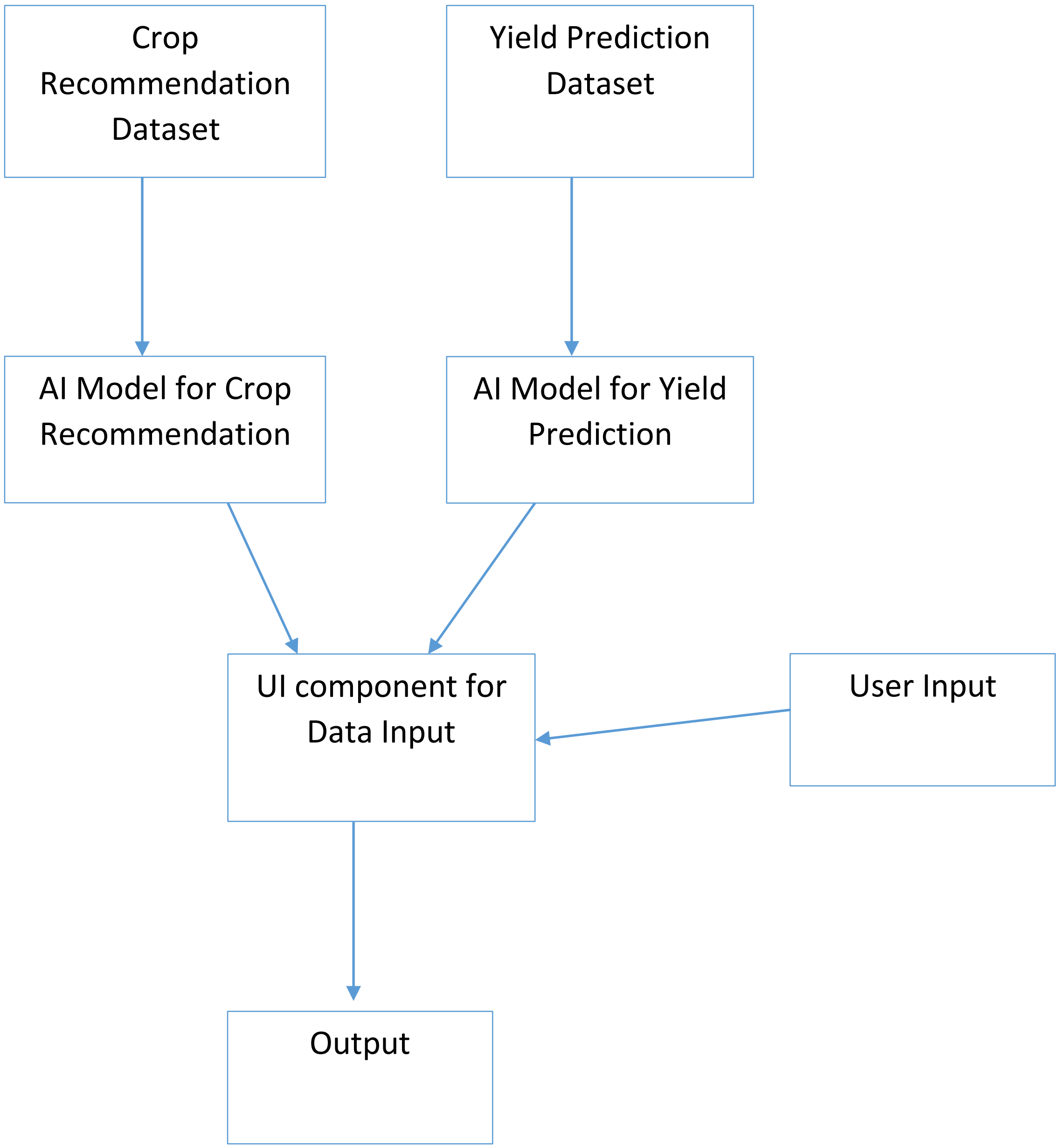
      In farming climatic factors such as rainfall, temperature and humidity plays an important role in the agriculture lifecycle. Increasing deforestation and pollution result in climatic changes, so it's difficult for farmers to take decision to prepare the soil, sow seeds and harvest.

      Every Crop need specific nutrition from soil. The deficiency can lead to poor quality of crop. The climatic situation is also important factor which can impact the production. As well as it is necessary for soil to shuffle the crop to maintain balance in its nutrients. Harvesting the same crop again and again may lead the nutrition deficiency of soil.

    - **Proposed Solution**

      This system collects the land nutrients values using an IoT device, in addition to humidity and rainfall and recommend the most suitable crop based on analysis of the past data where the same scenario given the maximum output. It also gives yield production if the recommended crop is cultivated which is also on based of the analysis of the past data.

- ## Theoretical Analysis
    - ### **Block Diagram**



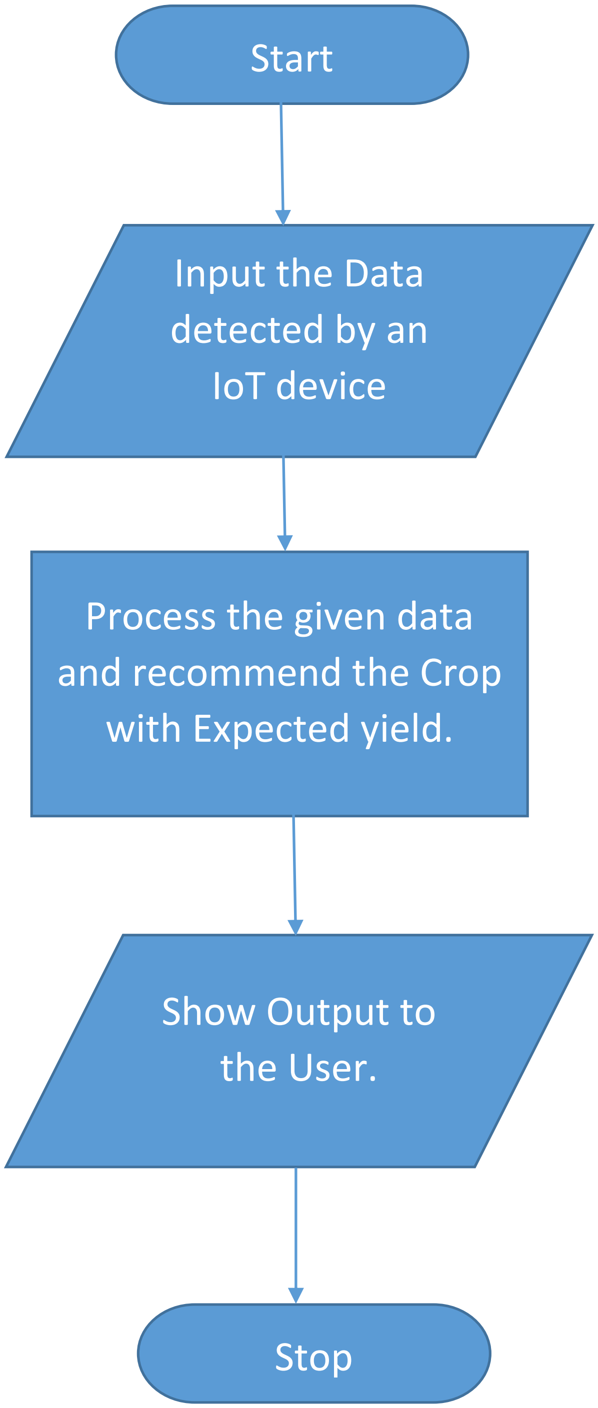
- **Hardware and Software Requirements**
    - Software Requirements
        - IBM Cloud
        - Flutter Framework
        - System with Browser
    - Hardware Requirements
        - IoT device

        (Note: In this project Simulated Data of IoT device is used instead of live data from IoT device)

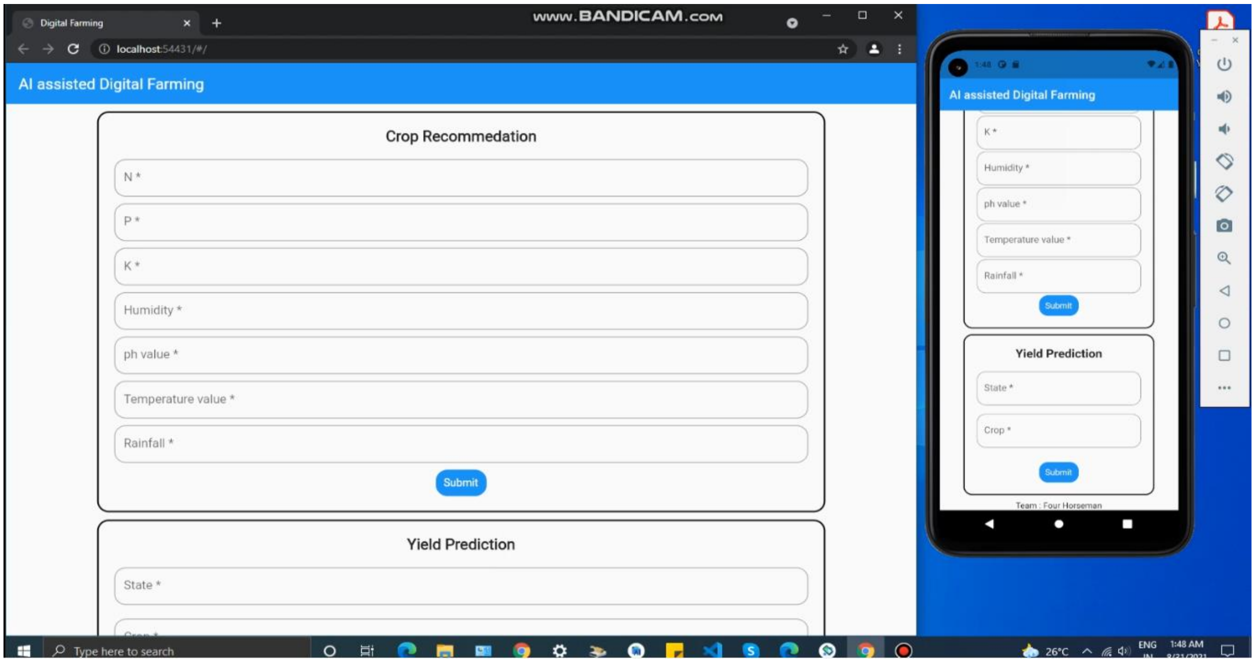
Experimental Investigation

India is very vast in terms of Agricultural Concept. Every region is very different in terms of weather, soil fertility, climatic situation and the method used to cultivate the respective crop and its varies region to region and person to person. Thus it becomes difficult to predict any factor related to farming. It may lead to get unexpected outcome.

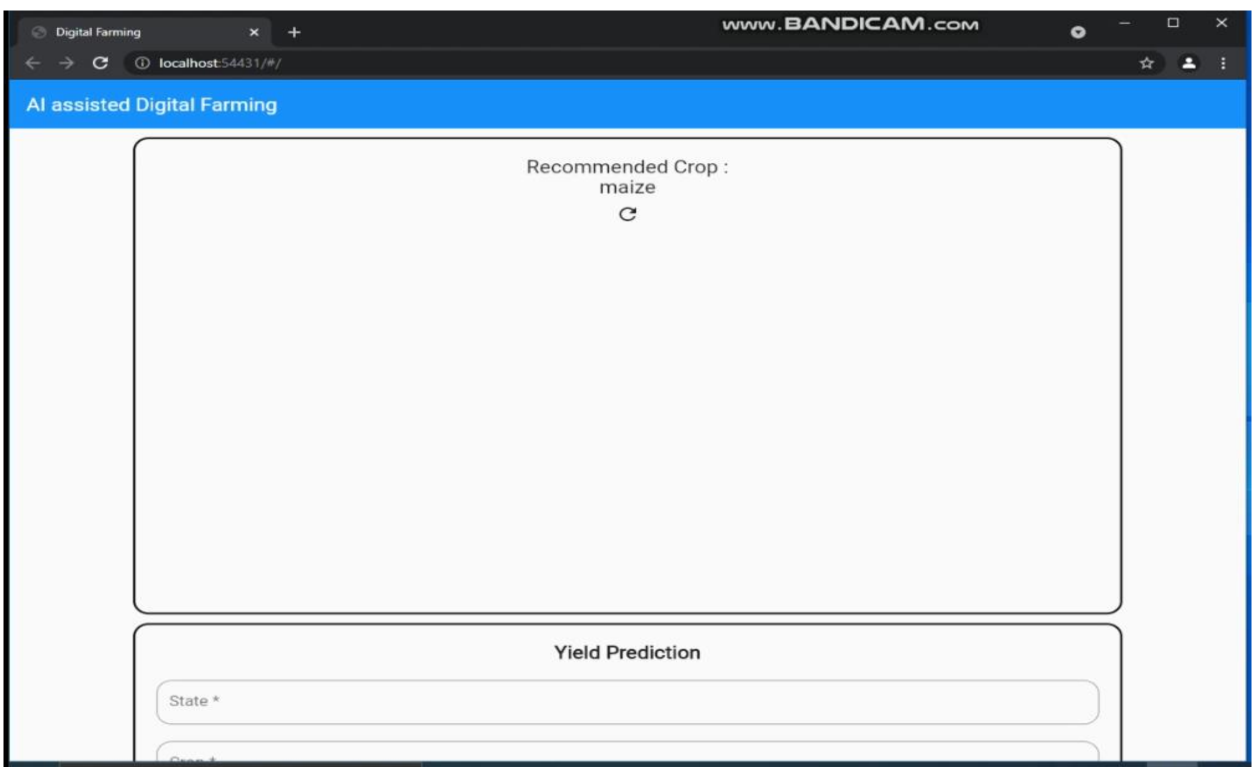
- Flowchart


Start
Input the Data detected by an IoT device
Process the given data and recommend the Crop with Expected yield.
Show Output to the User.
Stop

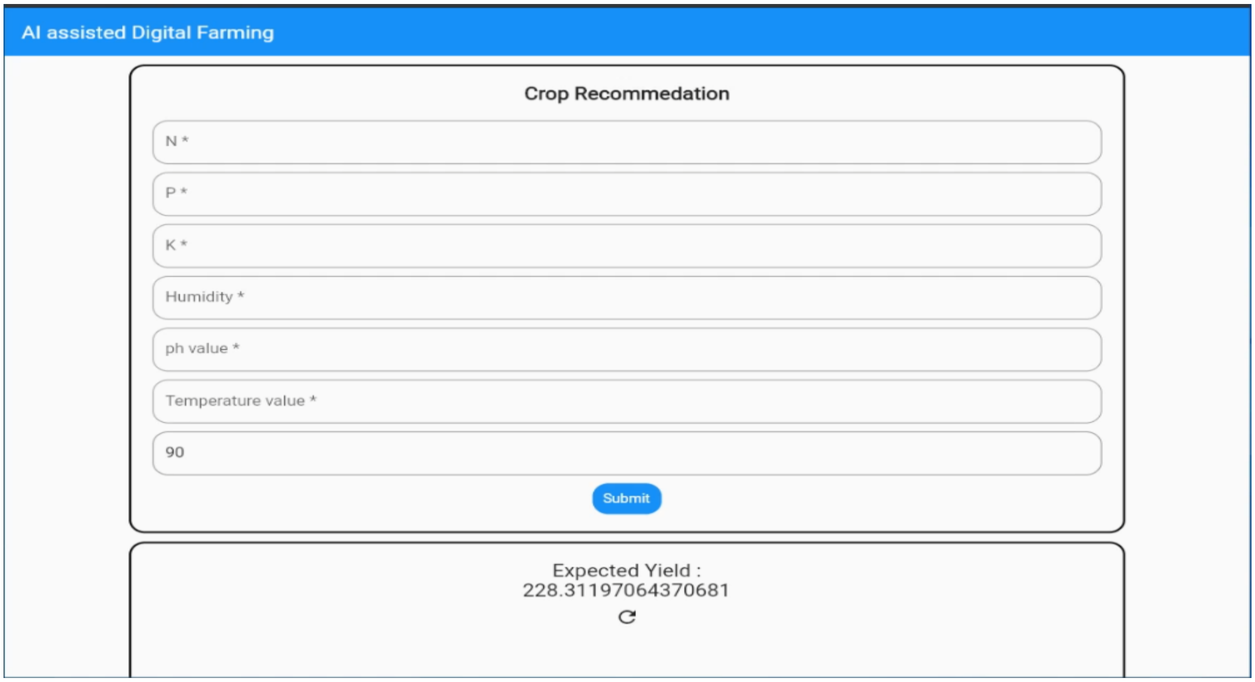
- Result



www.BANDICAM.com
Digital Farming
localhost:54431/#/
AI assisted Digital Farming
Crop Recommedation
N *
P *
K *
Humidity *
ph value *
Temperature value *
Rainfall *
Submit
Yield Prediction
State *
State *
Crop *
Submit
Team : Four Horseman
Type here to search
26°C
ENG
1:48 AM



www.BANDICAM.com
Digital Farming
localhost:54431/#/
AI assisted Digital Farming
Recommended Crop :
maize
Yield Prediction
State *


AI assisted Digital Farming
Crop Recommedation
N *
P *
K *
Humidity *
ph value *
Temperature value *
90
Submit
Expected Yield :
228.31197064370681


1:49
AI assisted Digital Farming
45
50
56
45
7
30
99
Submit
Yield Prediction
State *
Crop *


1:49
AI assisted Digital Farming
Recommended Crop :
pigeonpeas
Yield Prediction
State *

- ## Advantages and Disadvantages
  - **Advantages**
    - The result is based on large number of data collected in past thus the accuracy is high.
    - The system helps farmers to take decision to sow the crop.
    - Artificial intelligence helps farmers get more from the land while using resources more sustainably.
    - Artificial intelligence can improve plant breeding and crop management practices.
    - It promotes digital farming.
    - This solution is build using flutter framework which support Android, iOS, MacOs, Web and Windows Platforms.(Currently supports Android and Web)
  - **Disadvantages**
    - The result does not cover all the region of India.
    - The result does not cover each and every agriculture product harvested in all the regions.

- ## Application
    - The system can be used to get suitable crop to sow in the field.
    - The system also predict the production value.

- Conclusion
    - The proposed solution consist the analysis done by IBM cloud based service IBM Auto AI and recommend the Crop to sow and estimate the Yield. It also warns user about expected disease related with recommended crop. It displays output using the Python Django web Framework. Any user with understanding of Internet will be able to use this system.

- Future Scope
        - The Live weather forecasting system can be integrated with this system.
        - Soil nutrient booster or fertilizer recommendation system will be beneficial if system detect any soil nutrient deficiency.
        - Suitable fertilizer and pesticide recommendation system can be implemented along with this system.
        - In addition to this system, cultivation cost can also be predicted.

- Source code

This are the main 3 method by which I access the endpoint and fetch results rest part is UI:

```
Future getAccessToken() async {
  setState(() {
   isLoading = true;
  });
  final response =
     await http.post(Uri.parse("https://iam.cloud.ibm.com/identity/token"),
        body: {
          "grant_type": "urn:ibm:params:oauth:grant-type:apikey",
          "apikey": "jWZ_xbe0joKsCADmU5ipeE6jQdffE2iUVtK-7n9xuQDV"
        },
        headers: <String, String>{
          "Content-Type": "application/x-www-form-urlencoded",
          "Accept": "application/json",
          "Access-Control-Allow-Origin": "*",
        },
        encoding: Encoding.getByName("utf-8"));
  Map res = jsonDecode(response.body);
  print('getting token successfull');
  iamToken = res['access_token'];
  setState(() {
   isLoading = false;
  });
 }

 Future getRecommendedCrop() async {
  setState(() {
   isLoading = true;
  });
  final body = jsonEncode({
   "input_data": [
    {
     "fields": [
      "N",
      "P",
      "K",
      "temperature",
```

```
          "humidity",
          "ph",
          "rainfall"
        ],
        "values": [
          [n, p, k, temp, humidity, ph, rainfall]
        ]
      }
    ]
  });
  try {
    final cropres = await http.post(
      Uri.parse(
          "https://us-south.ml.cloud.ibm.com/ml/v4/deployments/c22115ce-b6de-4d38-a2b6-
413ae20cfe7b/predictions?version=2021-08-30"),
      body: body,
      headers: <String, String>{
        "Content-Type": "application/json",
        "Accept": "application/json",
        "Access-Control-Allow-Origin": "*",
        "Authorization": "Bearer $iamToken",
      },
      encoding: Encoding.getByName("utf-8"),
    );

    Map finalres = jsonDecode(cropres.body);
    crop = finalres['predictions'][0]['values'][0][0].toString();
  } catch (e) {
    final snackBar = SnackBar(content: Text(e.toString()));
    ScaffoldMessenger.of(context).showSnackBar(snackBar);
  }

  setState(() {
    cdone = true;
    isLoading = false;
  });
 }

 Future getYieldPrediction() async {
  setState(() {
    isLoading = true;
  });
  final body = jsonEncode({
    "input_data": [
```

```
      {
        "fields": ["Crop", "State"],
        "values": [
          [ycrop, state]
        ]
      }
    ]
  });
  try {
    final yieldres = await http.post(
      Uri.parse(
          "https://us-south.ml.cloud.ibm.com/ml/v4/deployments/8e82e510-8071-46b6-9918-
250906e53165/predictions?version=2021-08-30"),
      body: body,
      headers: <String, String>{
        "Content-Type": "application/json",
        "Accept": "application/json",
        "Access-Control-Allow-Origin": "*",
        "Authorization": "Bearer $iamToken",
      },
      encoding: Encoding.getByName("utf-8"),
    );

    Map result = jsonDecode(yieldres.body);

    predictedYield = result['predictions'][0]['values'][0][0].toString();
  } catch (e) {
    final snackBar = SnackBar(content: Text(e.toString()));
    ScaffoldMessenger.of(context).showSnackBar(snackBar);
  }
  setState(() {
    ydone = true;
    isLoading = false;
  });
}
```