AI-Assisted Farming For Crop Recommendation & Farm Yield Prediction Application

Team:
hack.py
Video: https://youtu.be/XmeU1cPr8wU
Github: https://github.com/kanavmittal/Agri-Cultivate
Website: https://agri-cultivate.vercel.app/

Team Members:
Sarthak Tyagi
Karmabir Chakraborty
Kanav Mittal
Samank Gupta

1.1) Overview

Precision agriculture is in trend nowadays. It helps the farmers to get informed decisions about the farming strategy. Older farming techniques are outdated and are not very useful as they lead to crop wastage, poor cultivation, etc. Farmers also do not know the yield and revenue they will be able to generate which makes them vulnerable to third-party people who offer them false prices.

To overcome these problems we will be coming up with a system that will help the farmers to recommend which crops to cultivate, sow, and other activities related to farming taking in various inputs such as nitrogen value of soil, Ratio of Phosphorus content in the soil, Ratio of Potassium content in the soil, etc. The system will also predict the profit as well as the yield for the cultivated land. Our portal will have two major sections containing recommendation and profit calculation web pages. The user can then go to any desired section and enter the input values to get the desired output.
For the recommendation section, the crop details along with the nutrients and other specifications about it will be rendered on the UI.
For the profit section, the user can enter the details needed for the input and the user can find out whether or not they will be profited.

1.2) Purpose

A Farmer can access our website easily as it's deployed on Vercel . One of the most attractive things about our Website is the interactive UI which makes it easy to access and use by anyone.
We provide additional benefits like crop details upon recommendation of a crop, chatbot to let the user know about the different components and top crops for different States.
The purpose of this project is to make it easy for farmers to access resources to help them cultivate crops suited to their soil and conditions. Profit is calculated for the given inputs like total cultivation etc.

2) Literature Survey

2.1 Existing problem / Existing approaches or methods to solve this problem

One of the major problems which a new farmer faces is what crop to grow on what type of soil and how much profit it might yield at the end. This lack of knowledge may lead to loss of farmer's time and as well as money. For an existing farmer, agricultural productivity has become stagnant due to the availability of fertilizers and pesticides. However, not correctly taking all the parameters and lack of awareness of field parameters leads to a decrease in productivity.
Farming techniques presently used in India are outdated due to which farmers aren't able to get maximum profit and in some cases the farmers even face loss. They are not even able to predict the estimated yield at the end of the season which makes them agree to the middlemen or the vendors who offer them false prices. A considerable amount of research has been done on precision farming yet in India, very few farmers have practiced precision farming technologies to reorganize towards low-input, high-efficiency and sustainable agriculture. This is because most of the work done and the technologies worked are tough to implement. This is where our technology comes

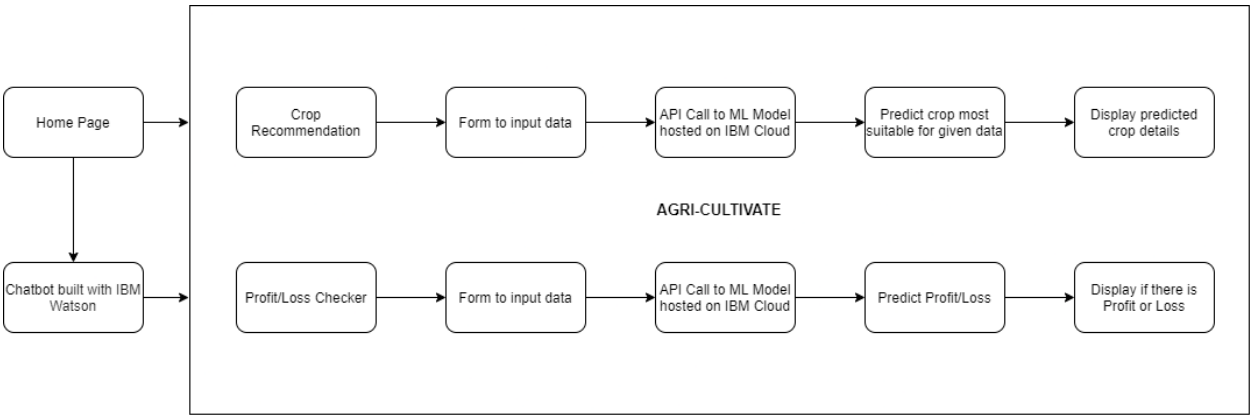
into play. Our website is free and easy to use and any farmer or user can easily check what crop to grow on a field and how much profit it will yield.

2.2 Proposed solution / What is the method or solution suggested by you?

The main functionality of our website is to predict what crops should be grown and how much profit it will yield(approximately) while taking into account multiple parameters. As far as which crop to grow is concerned, we have taken the amount of nitrogen, phosphorus, potassium, the average temperature of the soil, humidity, average pH value and annual rainfall into consideration. This model is able to predict among 22 major crop species which are grown in India. For the prediction of the profit from the crop yield, we have taken the parameters crop, state, cost of cultivation per hectare, cost of production, yield(quintal/hectare) and support price. This website is made free and easy to use. If any user faces any difficulty regarding the usage, they can also use the chatbot to know about our project and the user can even ask about the top 5 crops in a state.The Machine Learning models are created by setting up a deployment space on IBM Watson Studio where we created an asset for the jupyter notebook on which we shipped our code to create an API Endpoint so that the desired output can be presented on our website.

3 <u>THEORETICAL ANALYSIS</u>

3.1 Block diagram :



3.2 Hardware / Software designing :

Agri Cultivate was built using the ReactJS framework which allows easy integration with other data sources like weather, soil type, humidity and many more. This ensures farmers get all the information they need in order to make the best decision for their farms.

Website Framework design :
The Front-end design is done in Figma which is a powerful tool for UX/UI Designs.

User Interface :

ReactJS is a JavaScript library that's used on the front end of a website. Its main focus is to have the DOM reflect the underlying data. By using a virtual DOM, that can make your apps more efficient with the way they handle updates.

While working with the ReactJS framework, we have ensured that all of the components are reusable and responsive, as well as being simple to update and change.

We have used Tailwind CSS which is a lightweight, responsive, & highly customizable CSS framework that gives us the building blocks to create a stunning website.

The majority of the code has been designed in such a way that if a user encounters difficulty when understanding a certain component, the website will notify the user as to why the website is responding in this manner and what the customer may do to resolve the issue.

4 Experimental Investigations

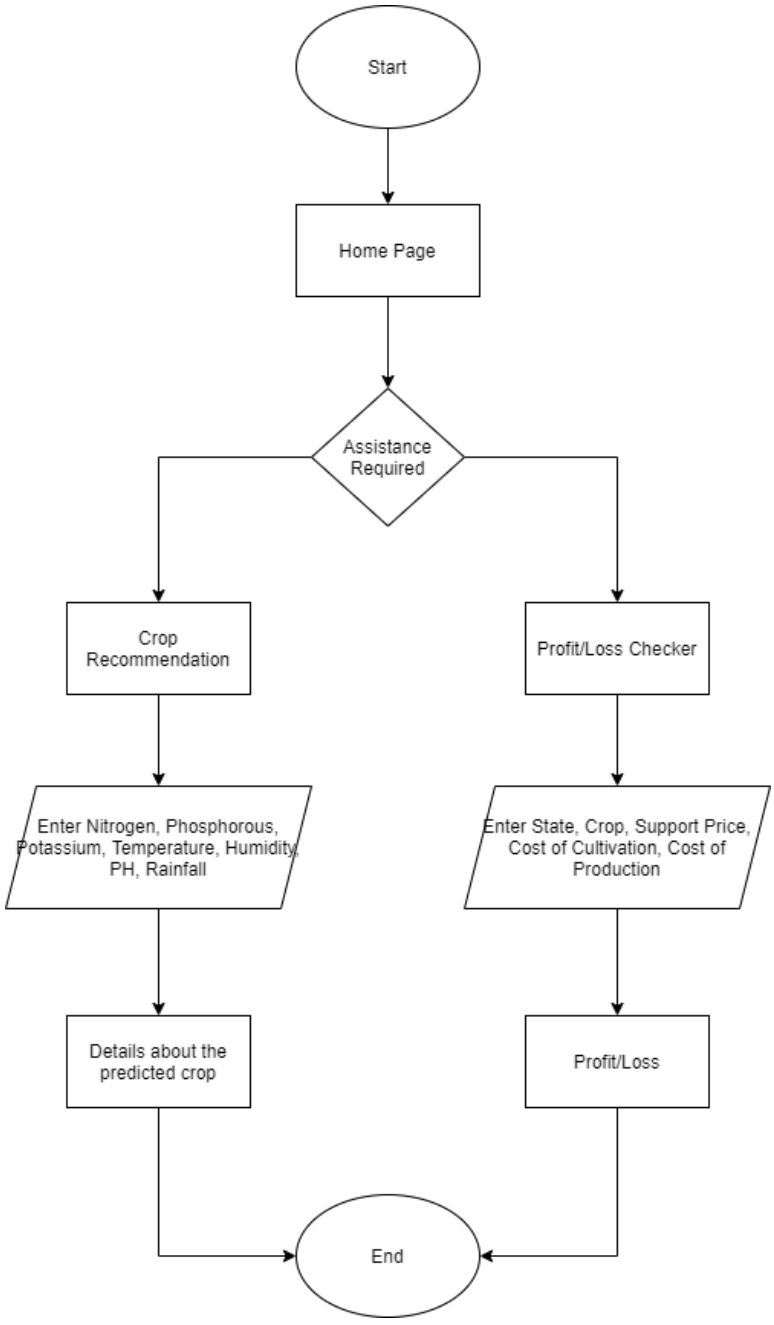
With the exponential growth of population and demand for food, agricultural production has also increased to meet the demand. However, the current cultivation methods are unsustainable in terms of resource use, environmental pollution and

agriculture products distribution.

We discovered that there is a lot of information that can be given to a farming community that may assist them in growing or sowing better crops, or if not better crops, then crops that are grown or sown in a more efficient or effective manner. Implemented Logistic Regression Algorithm to correctly classify if the farmer will be profited by his/her cultivation or would undergo deficit and used Decision Trees Algorithm to recommend crops similar to conditions in that particular geographical area. Machine Learning in the field of Agriculture is used for Crop Management, Determining crop quality and Disease Detection. The techniques will further improve and maximize crop production and profit for our farmers.
One of the companies named Rentokil is using AI to kill all the bugs and vermin. Other companies are making use of the Android app which is developed by Accenture to find bugs. The app takes the pictures of the bug and runs the app called PestID. When a bug is found app will provide an immediate solution that helps the technician to take further actions. It will also recommend the chemical to be used to kill the bugs.

We tested a large number of APIs as well as other kinds of farmer-related assistance that is accessible in current technological form and that can be included in our solution. We made use of postman, which is a programme that allows us to send get and post requests to an API without having to write any code. By doing so, we were able to test a large number of APIs but were only able to discover one that returned all of the data to fulfil the requirements, which was provided by https://www.getambee.com.
Analysis of the investigation made while working on the solution.

5 FLOWCHART

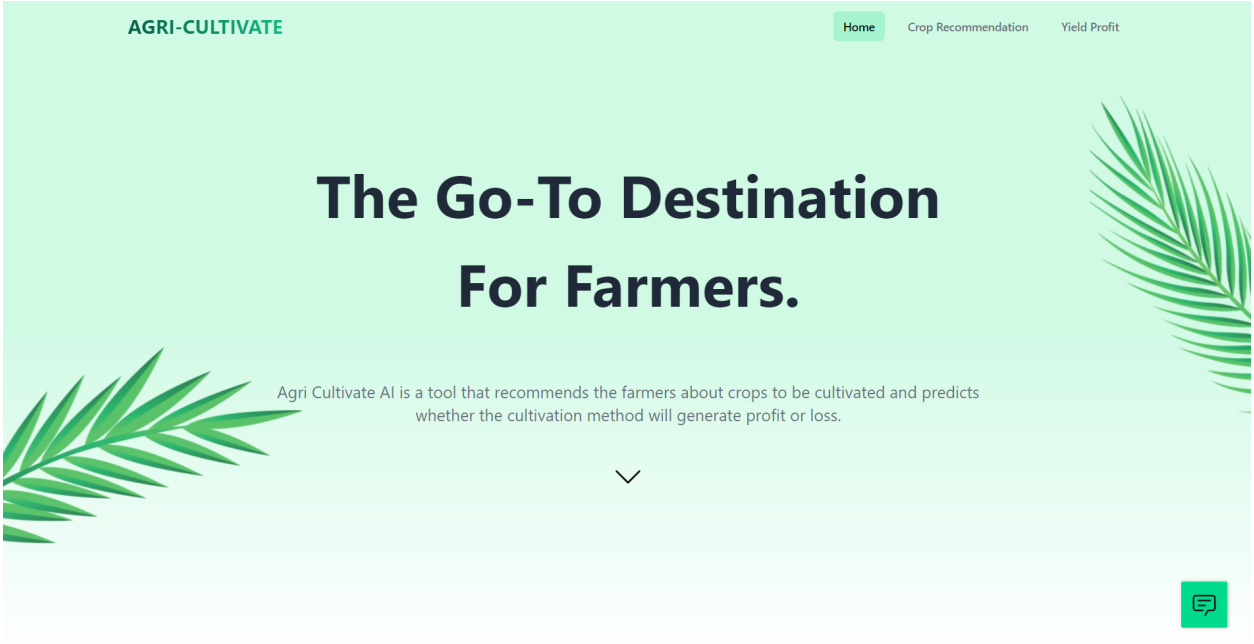


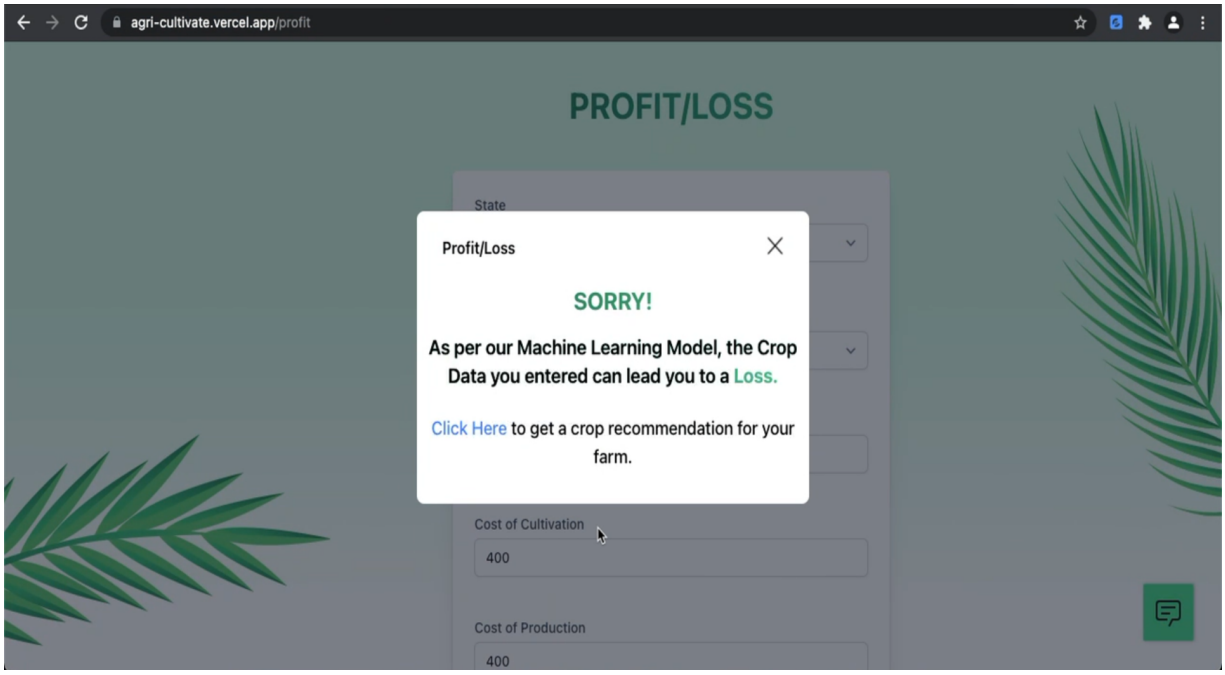
6 RESULT

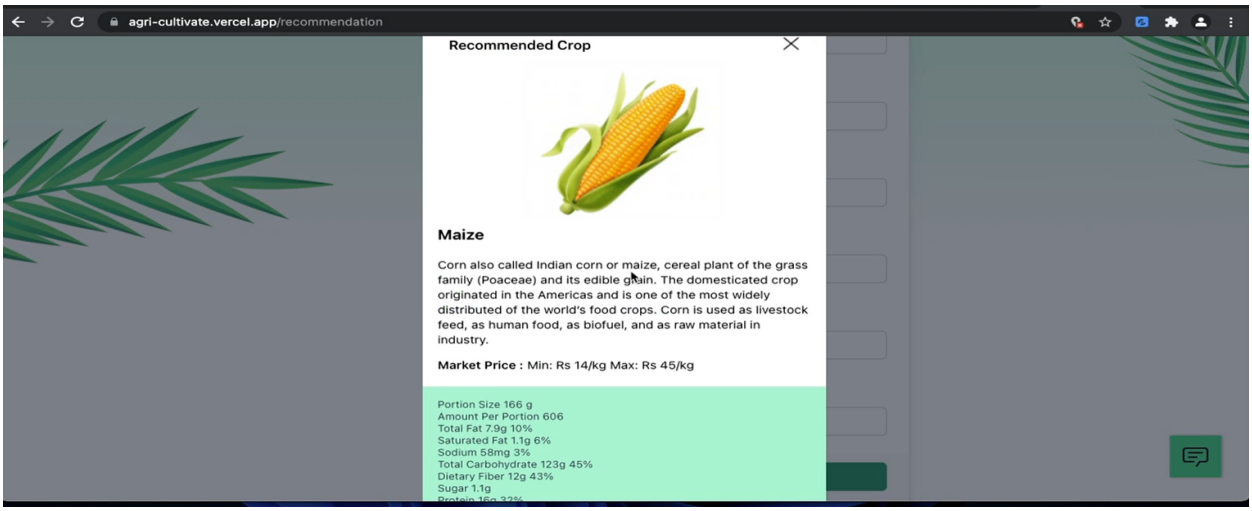
Our application was successfully able to -

- Predict a crop on the basis of the user's farm data.

- Predict if the user would gain profit or suffer a loss on the basis of the data they entered.







7 ADVANTAGES & DISADVANTAGES

This application really helps to solve a real-world issue that on daily basis farmers across the country face. The major advantage of this application is that it is easy, safe and reliable to use. The phrase “Simplicity at its best” is something that people can really use for this web application. The user interface is clean yet impressive. People using this will be able to use it quite easily without much difficulty and even if they face some issue, we have installed a chatbot for their help.

But like with any other application, it has some disadvantages. The chatbot still has some constraints. It is not able to hold all the conversations and the language of conversation is only English. Secondly, the dataset chosen for the prediction does not consider all the situations. For example in India, farmers have many different species of crops but we have taken only 22.

8 Applications

1. Any farmer or user can check which crop to grow on which type of soil, weather and geographical topography.
2. The farmer can even check the approximate profit which they can earn for each state.
3. Getambee is used to display details like air quality, soil, weather and water

vapour content in the air.

4. Users can even use chatbots for their own ease.

9 Conclusion

Our project's aim is to help the farmers throughout the country sow and grow crops that are best suited to their soil, weather and geographical region. This in turn will help the farmers to generate more profit. The Website's design is such that anyone can easily understand his/her way around it by easy navigation and access to various features provided by us.

10 Future Scope

To scale our project to a level where it is recognized and trusted by a large community of farmers is to integrate multilingual features which will help farmers throughout the country to be able to use it without any language constraints. We aim to get more data which will, in turn, lead to a better generalization of results. There are certain problems that are faced by farmers if their crops tend to get infected, Though it was not a part of our problem statement we would like to implement Image classification/detection methods to help them identify the infected crops and the best ways to deal with it.

11 Bibliography

Some of the references we made use of while building our project are:

(1) https://www.researchgate.net/publication/309465979_Applications_of_Machine_Learning_Techniques_in_Agricultural_Crop_Production_A_Review_Paper

(2)https://technostacks.com/blog/machine-learning-in-agriculture

(3)https://tailwindcomponents.com/

(4)http://dribbble.com/

(5) Tutorial 43-Random Forest Classifier and Regressor/

12 Code Snippets

```
Name: profit_label, dtype: int64

In [49]: len(X_test)

Out[49]: 10

In [50]: y_test

Out[50]: 32    1
         12    0
         25    0
         46    1
         33    1
         22    1
         3     0
         30    1
         26    0
         1     0
         Name: profit_label, dtype: int64

In [53]: #os1=RandomOverSampler(sampling_strategy=0.50)
         #X_test_os,y_test_os=os1.fit_resample(X_test,y_test)

In [54]: from sklearn.linear_model import LogisticRegression
         from sklearn.model_selection import KFold,GridSearchCV
         from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

In [55]: lr=LogisticRegression()

In [56]: lr.fit(X_train,y_train)

Out[56]: LogisticRegression()

In [57]: lr.predict(X_train)

Out[57]: array([1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0,
                0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0], dtype=int64)

In [58]: weights=np.linspace(0,0.8,100)

In [59]: grid={'penalty':['l2'],
               'class_weight':[{0:1-x,1.0:x} for x in weights],
                "max_iter":[5000]
              }

In [60]: lr_clf=GridSearchCV(estimator=lr,param_grid=grid,cv=10)

In [61]: lr_clf.get_params().keys()

Out[61]: dict_keys(['cv', 'error_score', 'estimator__C', 'estimator__class_weight', 'estimator__dual', 'estimator__fit_intercept', 'esti
         mator__intercept_scaling', 'estimator__l1_ratio', 'estimator__max_iter', 'estimator__multi_class', 'estimator__n_jobs', 'estima
         tor__penalty', 'estimator__random_state', 'estimator__solver', 'estimator__tol', 'estimator__verbose', 'estimator__warm_start',
```

jupyter crop profit Last Checkpoint: 08/08/2021 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

75%	24731.060000	35423.480000	2228.970000	38.610000	5515.000000
max	66335.060000	91442.630000	5777.480000	1015.450000	7196.000000

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49 entries, 0 to 48
Data columns (total 7 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Crop                                  49 non-null     object
 1   State                                 49 non-null     object
 2   Cost of Cultivation (`/Hectare) A2+FL  49 non-null     float64
 3   Cost of Cultivation (`/Hectare) C2    49 non-null     float64
 4   Cost of Production (`/Quintal) C2     49 non-null     float64
 5   Yield (Quintal/ Hectare)              49 non-null     float64
 6   Support price                         49 non-null     int64
dtypes: float64(4), int64(1), object(2)
memory usage: 2.8+ KB
```

In [10]:
```
df['Crop'].unique()
```

Out[10]:
```
array(['ARHAR', 'COTTON', 'GRAM', 'GROUNDNUT', 'MAIZE', 'MOONG', 'PADDY',
       'RAPESEED AND MUSTARD', 'SUGARCANE', 'WHEAT'], dtype=object)
```

In [11]:
```
df.rename(columns={"Cost of Cultivation (`/Hectare) A2+FL":"A2+FL",
                   "Cost of Cultivation (`/Hectare) C2":"CC2",
                   "Cost of Production (`/Quintal) C2":"CP2",
                   "Yield (Quintal/ Hectare)":"Yield"
                   }
                    ,inplace=True)
```

In [12]:
```
df.head()
```

Out[12]:

	Crop	State	A2+FL	CC2	CP2	Yield (Quintal/ Hectare)	Support price
0	ARHAR	Uttar Pradesh	9784.05	23076.74	1941.55	9.83	6000
1	ARHAR	Karnataka	10593.15	16528.68	2172.46	7.47	6000
2	ARHAR	Gujarat	13468.82	19551.90	1898.30	9.59	6000
3	ARHAR	Andhra Pradesh	17051.66	24171.65	3670.54	6.42	6000
4	ARHAR	Maharashtra	17130.55	25270.26	2776.80	8.72	6000

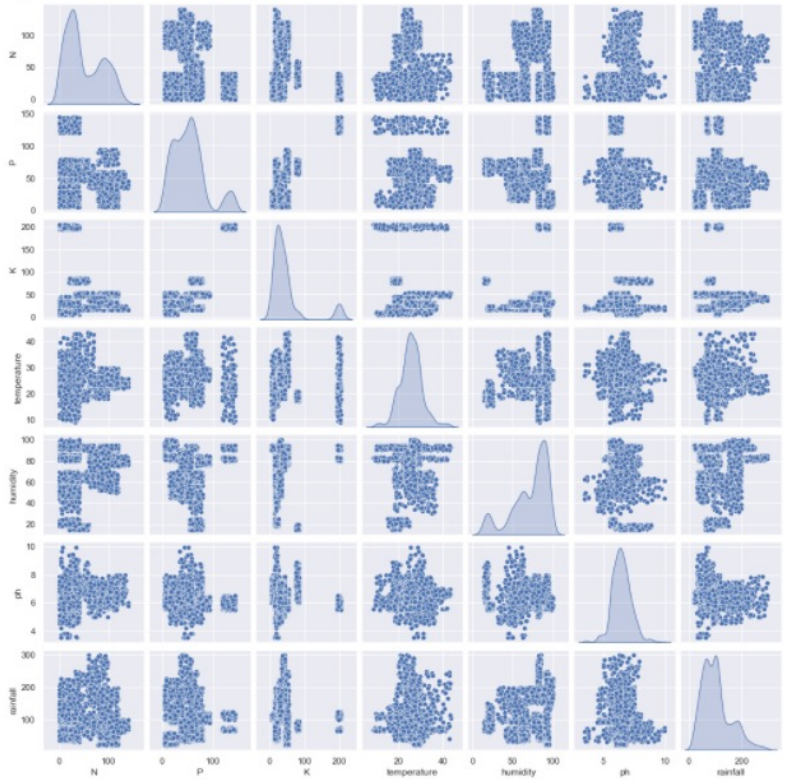
In [13]:
```
def CalProfit():
    df['Profit']=(df.iloc[:,5]*df.iloc[:,6])-(df.iloc[:,2]-df.iloc[:,3]+(df.iloc[:,5]*df.iloc[:,4]))
```

In [14]:
```
CalProfit()
```

In [15]:
```
df['Profit']
```

Out[15]:
```
0      53177.2535
1      34527.2538
2      45418.3830
3      22075.1232
4      36254.7340
```

In [6]:
```
sns.set()
sns.pairplot(df, height=2, kind='scatter', diag_kind='kde');
```

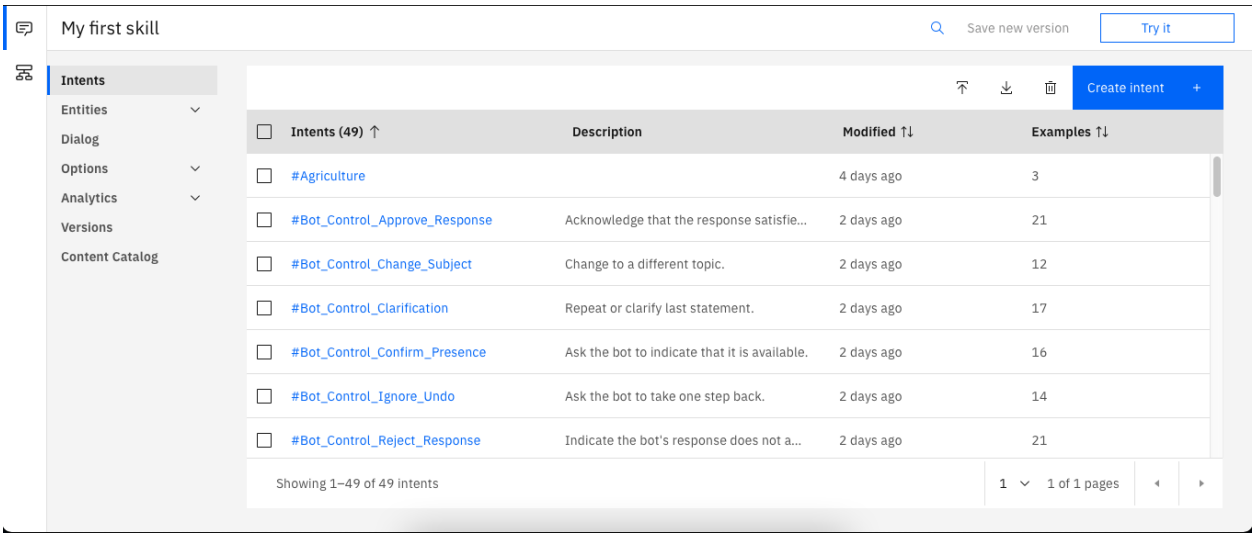


```
from sklearn.metrics import classification_report, confusion_matrix, accura

print ('Confusion Matrix :')
print(confusion_matrix(y_test, predicted))
print ('Accuracy Score :',accuracy_score(y_test, predicted))
print ('Report : ')
print (classification_report(y_test, predicted))
```

```
Confusion Matrix :
[[23  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  6  0]
 [ 0 32  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 41  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 41  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 41  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 37  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 33  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 29  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 28  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 28  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 34  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 25  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0 23  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 26  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 38  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 37  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 35  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 33  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 35]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 33]]
Accuracy Score : 0.9903581267217634
Report :
              precision    recall  f1-score   support

           1       1.00      0.79      0.88        29
           2       1.00      1.00      1.00        32
           3       1.00      1.00      1.00        41
           4       1.00      1.00      1.00        41
           5       1.00      1.00      1.00        41
           6       1.00      0.97      0.99        38
           7       1.00      1.00      1.00        33
           8       1.00      1.00      1.00        29
           9       0.97      1.00      0.98        28
          10       1.00      1.00      1.00        40
          11       1.00      1.00      1.00        28
          12       1.00      1.00      1.00        34
          13       1.00      1.00      1.00        25
          14       1.00      1.00      1.00        23
          15       1.00      1.00      1.00        26
          16       1.00      1.00      1.00        38
          17       1.00      1.00      1.00        27
          18       1.00      1.00      1.00        37
          19       1.00      1.00      1.00        35
          20       1.00      1.00      1.00        33
          21       0.85      1.00      0.92        35
          22       1.00      1.00      1.00        33

    accuracy                           0.99       726
   macro avg       0.99      0.99      0.99       726
weighted avg       0.99      0.99      0.99       726
```

```
return (
    <div className="flex">
        <dialog id="modal" className="bg-transparent z-0 relative w-screen h-screen">
            <div className="p-6 flex justify-center items-center fixed left-0 top-0 w-full h-full bg-gray-900 bg-opacity-50 z-50 transition-opacity duration-300 opaci
                <div className="bg-white rounded-lg md:w-2/3 lg:w-1/3 relative">
                    <div>
                        <div className="px-7 pt-6 pb-2 grid grid-cols-2">
                            <h1 className="font-semibold text-base">Recommended Crop</h1>
                            <img src="https://upload.wikimedia.org/wikipedia/commons/thumb/4/49/Multiplication_Sign.svg/1024px-Multiplication_Sign.svg.png" alt="Close
                        </div>
                        <div className="overflow-y-auto">
                            <Card cardId={returnData!==''?parseInt(returnData)-1:0}/>
                        </div>
                    </div>
                </div>
            </div>
        </dialog>

        <div className="min-w-full w-full h-screen bg-gradient-to-b from-green-100 via-green-100">
            <Navbar name="Crop Recommendation"></Navbar>
            <div className='relative h-4/5 w-full flex justify-center'>
                <div className="absolute right-0 top-12 opacity-0 lg:opacity-80"><img className="" src={leaves} alt="leaves1" /></div>
                <div className="absolute left-0 top-60 opacity-0 lg:opacity-80"><img className="transform rotate-90" src={leaves2} alt="leaves2" /></div>
                <div className="w-4/5 lg:w-1/3 my-auto">
                    <div className="mx-auto">
            <div className="md:grid md:gap-6">
                <h1 className="text-3xl sm:text-4xl bg-clip-text text-transparent font-bold bg-gradient-to-r from-green-800 to-green-500 text-center lg:ml-12 mt-8">CR
                <div className="my-5 mb-16 lg:ml-12 shadow-xl">
                    <form id="recommendationForm" onSubmit={handleSubmit}>
                        <div className="shadow overflow-hidden sm:rounded-md">
                            <div className="px-4 py-5 bg-white sm:p-6">
                                <div className="grid gap-y-10 md:gap-10">
                                    <div className="col-span-8 md:col-span-12">
                                        <label htmlFor="Nitrogen" className="block text-sm font-medium text-gray-700">
                                            Nitrogen
                                        </label>
                                        <input
                                            type="number"
                                            step="any"
```

```
async function getToken() {
    try {
        const { data } = await axios.post('https://eba8b471.eu-gb.apigw.appdomain.cloud/token/?grant_type=urn:ibm:params:oauth:grant-type:apikey&apikey=YAUKpiPPjvKaqE
            setToken(data.access_token) ;
    } catch (error) {
        console.log(error.response.data)
    }
}

async function profitApi() {
    var configuration = {
        headers: {
            Authorization: "Bearer " + token,
            'X-IBM-Client-Id': 'd6e01114-4e36-43cf-806d-c6c22b1f1773',
        },
    };
    try {
        const { data } = await axios.post('https://eba8b471.eu-gb.apigw.appdomain.cloud/farm?version=2021-08-06', postData, configuration);
            setReturnData(data['predictions'][0]['values'][0][0]) ;
    } catch (error) {
        console.log(error.response.data)
    }
}

React.useEffect(()=>{
    if(returnData!==''){
        document.getElementById('profitForm').reset()
        document.getElementById('state').getElementsByTagName('option')[0].selected = "selected";
        document.getElementById('crop').getElementsByTagName('option')[0].selected = "selected";
        setVal({ "state": null, "crop": null, "sp": null, "cc2": null, "cp2": null })
        openModal('profitmodal')
        setLoading(false)
    }
},[returnData])

React.useEffect(()=>{
    if(postData!==''){
        getToken()
```




```jsx
import React from "react";
import Navbar from "./Navbar.component";
import leaves from "../assets/images/plant.png";
import leaves2 from "../assets/images/plant2.png";
import "./leaves.css";

class MainHeader extends React.Component {
    render() {
        return (
            <div className="min-w-full w-full h-screen bg-gradient-to-b from-green-100 via-green-100">
                <Navbar name="Home"></Navbar>
                <div className='relative h-4/5 w-full flex justify-center'>
                    <div className="absolute right-0 top-12 opacity-0 sm:opacity-80 overflow-hidden"><img className="leaf1" src={leaves} alt="leaves1" /></div>
                    <div className="absolute left-0 top-60 opacity-0 sm:opacity-80"><img className="transform rotate-90 leaf2" src={leaves2} alt="leaves2" /></div>
                    <div className="w-3/5 flex flex-col my-auto">
                        <span className="text-gray-900 sm:text-gray-800 font-sans text-4xl sm:text-7xl font-bold text-center sm:leading-normal sm:p-6 pb-4">The Go-To Dest
                        <span className="text-gray-600 sm:text-gray-500 text-center text-xl py-10">Agri Cultivate AI is a tool that recommends the farmers about crops to
                        <div className="flex justify-center animate-bounce ">
                        <a href="#arrow"><svg xmlns="http://www.w3.org/2000/svg" class="h-12 w-12" fill="none" viewBox="0 0 24 24" stroke="currentColor">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="1" d="M19 9l-7 7-7-7" />
                        </svg></a>
                        </div>
                    </div>
                </div>
            </div>
        )
    }
}
export default MainHeader;
```