Soil Analysis and Crop Prediction

A Project Report

Submitted for the partial fulfillment for the award of the degree of

Bachelor in Technology

Submitted by:

Neha Adlakha	2016204
Pragati Chaturvedi	2016224
Prateeksha	2016231
Priyanshi Chaudhary	2016238

Under the supervision of:

Name of the Mentor

Prof. Saurabh Mukherjee

Name of Coordinator(s)
(Name, Designation, Organization)
Dr. Neelam Sharma

Department of Mathematics & Computing
Banasthali Vidyapith

Banasthali - 304022

Session: 2022-23

Certificate

Certified that **Priyanshi Chaudhary** has carried out the project work titled **"Soil Analysis and Crop Prediction"** from 19-12-2022 to 15-04-2023 for the award of the **Bachelor in Technology** from **Banasthali Vidyapith** under my supervision. The thesis embodies result of original work and studies carried out by Student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else.

Prof. Saurabh Mukherjee

Designation: Professor

Place: Banasthali Vidyapith

Date: 15-04-2023

Department of Mathematics & Computing
Banasthali Vidyapith

Banasthali - 304022

Session: 2022-23

Certificate

Certified that **Prateeksha** has carried out the project work titled **"Soil Analysis and Crop Prediction"** from 19-12-2022 to 15-04-2023 for the award of the **Bachelor in Technology** from **Banasthali Vidyapith** under my supervision. The thesis embodies result of original work and studies carried out by Student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else.

Prof. Saurabh Mukherjee

Designation: Professor

Place: Banasthali Vidyapith

Date: 15-04-2023

Department of Mathematics & Computing
Banasthali Vidyapith

Banasthali - 304022

Session: 2022-23

Certificate

Certified that **Neha Adlakha** has carried out the project work titled **"Soil Analysis and Crop Prediction"** from 19-12-2022 to 15-04-2023 for the award of the **Bachelor in Technology** from **Banasthali Vidyapith** under my supervision. The thesis embodies result of original work and studies carried out by Student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else.

Prof. Saurabh Mukherjee

Designation: Professor

Place: Banasthali Vidyapith

Date: 15-04-2023

Department of Mathematics & Computing
Banasthali Vidyapith

Banasthali - 304022

Session: 2022-23

Certificate

Certified that **Pragati Chaturvedi** has carried out the project work titled **"Soil Analysis and Crop Prediction"** from 19-12-2022 to 15-04-2023 for the award of the **Bachelor in Technology** from **Banasthali Vidyapith** under my supervision. The thesis embodies result of original work and studies carried out by Student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else.

Prof. Saurabh Mukherjee

Designation: Professor

Place: Banasthali Vidyapith

Date: 15-04-2023

Abstract

Earlier, crop cultivation was undertaken on the basis of farmer's hands-on expertise. However, climate change has begun to affect crop yields badly. Consequently, farmers are unable to choose the right crop/s based on soil and environmental factors, and the process of manually predicting the choice of the right crop/s of land has, more often than not, resulted in failure. Accurate crop prediction results in increased crop production. This is where machine learning playing a crucial role in the area of crop prediction. Crop prediction depends on the soil, geographic and climatic attributes. Selecting appropriate attributes for the right crop/s is an intrinsic part of the prediction undertaken by feature selection techniques.

In this work, a comparative study of various classification techniques is carried out for crop prediction and soil analysis that suggest the suitable crop/s for land and the conditions required to grow a particular crop. The experimental results show the Random Forest classifier outperforms the others.

Acknowledgement

We would like to express our profound gratitude to Dr. C.K. Jha, HOD of Department of Mathematics & Computing, and Mr./Mrs. _____ (Dean) of Department of Mathematics & Computing of Banasthali Vidyapith for their contributions to the completion of our project titled Soil Analysis and Crop Prediction.

We would like to express our special thanks to our mentor Prof. Saurabh Mukherjee for his time and efforts he provided throughout the semester. Your useful advice and suggestions were really helpful to us during the project's completion. In this aspect, We are eternally grateful to you.

We would like to acknowledge that this project was completed entirely by us and not by someone else.

Neha Adlakha (2016204)
Pragati Chaturvedi (2016224)
Prateeksha (2016231)
Priyanshi Chaudhary (2016238)

Table of Contents

Introduction

In India, farming is done by traditional method, farmer’s plant crops traditionally without knowing the content of soil and quality of that soil. Contribution of the Agriculture sector in the Indian economy is much higher than the world's average (6.1%). But Traditional farms in India still have some of lowest per capita productivity and farmer incomes. As a result, farmers will not gain sufficient profit from their farming. The existing method of soil testing is a manual method which starts by taking soil samples and then sends them to laboratories for testing. This manual process is time consuming and not so feasible. So, there is a need for an automated process for soil testing and crop prediction. Testing of soil is important because soil testing helps to determine fertility of soil and thus crop prediction can be done. So, we proposed a system which will have a handheld device which gives pH value and we will estimate Nitrogen (N), Phosphorus (P) and Potassium (K) from the pH of that soil. We are using classification algorithms to predict suitable crops based on the values we get from our device and we will also provide suitable fertilizers required for that land. There's always a significant risk factor to the farmers when deciding to grow a particular crop during a particular season, on a particular piece of land. Irrespective of the capital put in terms of soil nutrients, water and seed quality, the crop may fail bringing disastrous losses to the farmer and his family, eventually leading to more serious problems like debt and suicide. This field is a subject of a lot of research still, allowing umpteen scopes to pick the combinations various strategies and factors apply them to bring out a model that suits the requirements.

Purpose

Precision agriculture is a modern farming technique that uses the data of soil characteristics, soil types, crop yield data, weather conditions and suggests the farmers with the most optimal crop to grow in their farms for maximum yield and profit. This technique can reduce crop failures and will help the farmers to take informed decisions about their farming strategy. A proper soil test will help ensure the application of enough fertilizer to meet the requirements of the crop while taking advantage of the nutrients already present in the soil and also, what other nutrients will be needed for a particular crop. Today in agriculture, this can allow crops to be grown at much higher precision, enabling farmers to treat plants and animals almost individually, which in turn significantly increases the effectiveness of farmers' decisions.

Using this can develop means to even predict harvest yields and evaluate crop quality for individual plant species to detect crop disease and weed infestations which were previously impossible! Purpose of this software is to recommend best possible crops to be cultivated by farmers based on several parameters and help them make an informed decision before cultivation by tracking the exact amount of soil nutrients, a farmer can easily adjust fertilization in accordance with soil and crop requirements. Additionally, soil analysis facilitates crop nutrient management by revealing the current soil pH level. In order to mitigate the agrarian crisis in the current status quo, there is a need for better recommendation systems to alleviate the crisis by helping the farmers to make an informed decision before starting the cultivation of crops. It helps the farmers to get informed decisions about the farming strategy.

Scope

Technological change in agriculture is reflected directly on the prediction of crops. It enhances the productivity of land and improves the capability of agricultural labor.

The name we are giving to our software product is **KRISHI-SUVIDHA**.

It is mainly proposed for the betterment of crop production across the country and help in food production. To develop initial machine learning models for crop yield predictions to analyze the performance of the model developed and also to explore the spatial & temporal granularity, and constraints therein in the database.

Factors that we are taking into consideration that influence crop growth:

1. It includes mainly the prediction of soil nature that comprises of moisture
2. Determining pH of soil
3. Other factors such as irrigation salinity
4. Rainfall and growing methods.

What can our system do?

1. Our system can analyze the soil parameters and nutrients given in the soil like NPK which can facilitate work out the fertility level of that soil.

2. Our system will not solely give soil analysis however additionally predict the crops.
3. Farmers will check the soil multiple ranges of times throughout the cultivation method and take necessary precautions to induce smart yield.
4. At the top reports are generated thus farmers will keep a record of their fertility

A soil test is important for several reasons:

1. to optimize crop production
2. to protect the environment from contamination by runoff and leaching of excess fertilizers
3. to aid within the identification of plant culture issues
4. to improve the nutritional balance of the growing media and to save lots of cash and provide an indication of potential nutrient deficiencies, pH imbalance or excess soluble salts.

- The limitations of the model are that we are taking only 22 crops into account.
- This software will help farmers to know about the soil's nature and suitable crops that can be grown in it.
- It will also include the information about if a farmer wants to grow a particular crop in the soil, then what will be the requirements of the soil to produce that crop.

Definitions, Acronyms, and Abbreviations.

- N: Sodium
- P: Phosphorus
- K: Potassium
- SRS: Software Requirements Specification

- TCP/IP: Transmission Control Protocol / Internet Protocol, the suite of communication protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being TCP and IP.
- RAM: Random Access Memory
- HTML: Hyper Text Markup Language is a markup language used to design static web pages.
- OS: Operating Systems
- GUI: Graphical User Interface
- ML: Machine Learning
- HDD: Hard Disk Drive
- IE: Internet Explorer
- TP: True Positive
- FP: False Positive
- TN: True Negative
- FN: False Negative
- P: Precision
- R: Recall is the capability of a classifier to discover all positive cases from the confusion matrix. It is calculated as the ratio of true positives to the sum of true positives and false negatives for each class.
- Precision-Positive Prediction Accuracy
- F1 score is a weighted harmonic mean of precision and recall, with 0.0 being the worst and 1.0 being the best. Since precision and recall are used in the computation, F1 scores are often lower than accuracy measurements.

Overview

The further SRS contains the overall description of the product. It describes the general factors that affect the product and its requirements, the major functions the product will perform, the logical characteristics of each interface between the software product and the hardware components of the system, the use of other required software products and interfaces with other application systems. The various

interfaces to communications, general characteristics of the intended users of the product including educational level, experience, and technical expertise, constraints, all the software requirements, the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs.

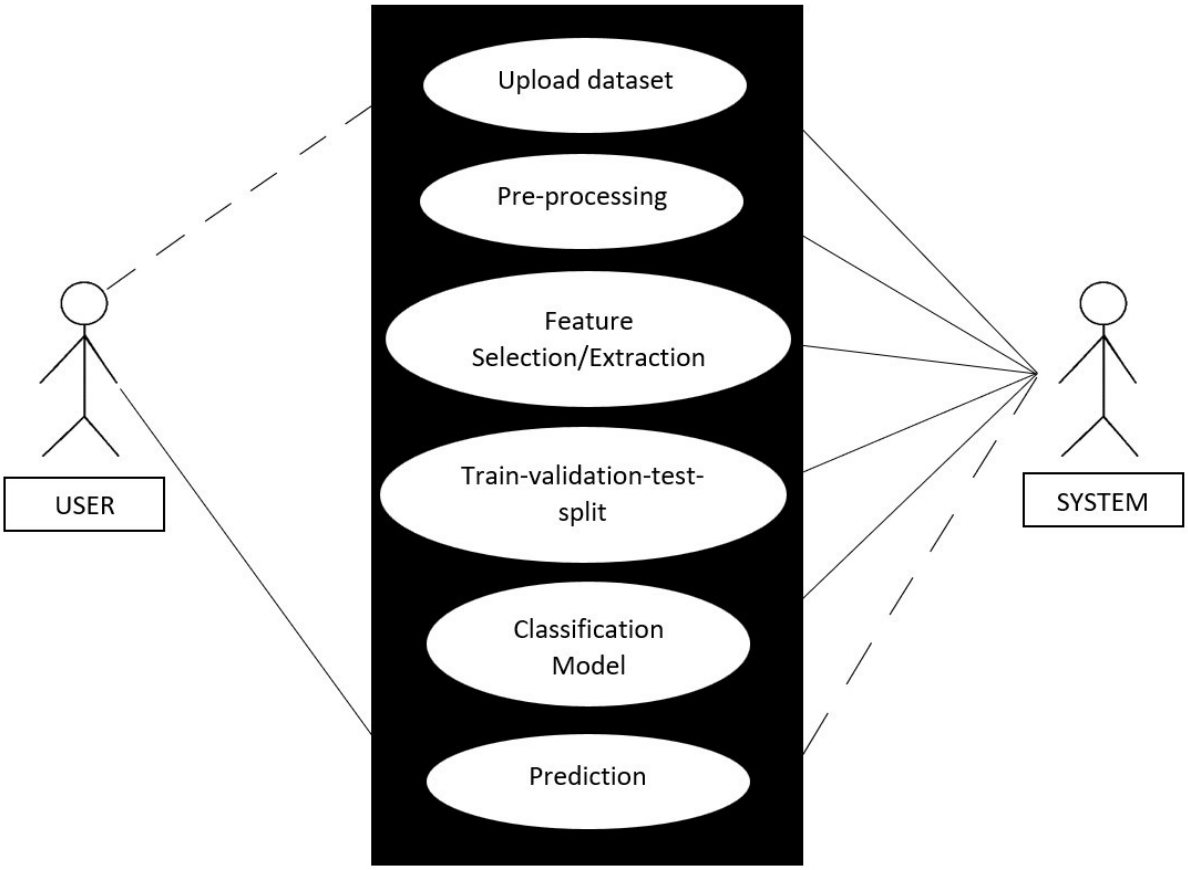
General Description

Agriculture plays a significant role in increasing the economic development of our nation. Crop production has greatly affected due to changes in weather patterns. Emerging technologies can be used to improve productivity of the crops by converting traditional farming to precision farming. The new technologies that are

used include data analysis. The major issue yet to be resolved is cultivating a precise crop at precise time. This can be done with the help of machine learning algorithms which is found to be an effective method for predicting the suitable crop. The soil parameters such as soil moisture, temperature, humidity and pH are collected from the sensors and given to Graphical User Interface (GUI). GUI gets the inputs and suggests the suitable crops. The system developed using ML greatly helps the farmers to make a valuable decision.

This system suggests crop based on soil classification with an assembling classifiers system has been developed after examining several previous systems on crop prediction.

Product Perspective



We have several modules:

- Uploading the data
- Pre-processing
- Feature selection/ extraction
- Train-validation-test split
- Classification model
- Prediction

Product Function

Major functions that the software will perform:

- Analyze the soil based on NPK, pH values of the soil.
- Predict the best crop that could be grown according to the particular conditions of soil.

"DDS" will allow:

1. Then the user will have two options: "SOIL ANALYSIS" or "CROP PREDICTION"
2. The user will be asked a set of questions based on what he wants to do.
3. Based on which results will be shown as which crop is suitable in the conditions or the result for analysis of soil.

Hardware Interfaces

Server Side:

- Processor – 1-2 GHz (ex. P-IV) or onwards,
- RAM – 512 MB,
- HDD – 20 GB or more (free space excluding data size)

Client Side

- Processor – 450 MHz (ex. P-II) or onwards
- RAM – 128 MB
- HDD – 10 GB or more (free space excluding data size) Developer Side
- Processor – 2.50 GHz Core i5
- RAM – 4 GB
- HDD – 20GB or more (free space excluding data size)

Software Interfaces

Server Side:

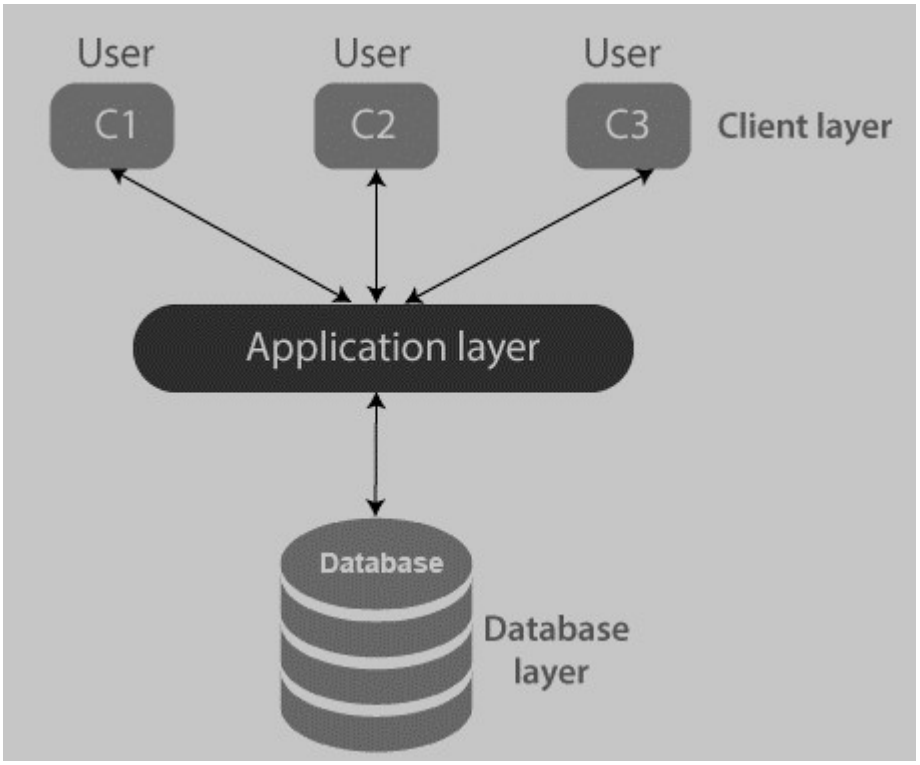
- OS-Windows Server 2003 or onwards
- Web Server-IIS 5.0 or onward
- Database Server- SQL Server 2005 Client Side:
- OS – Any Operating System
- Browser – Any browser compatible with IE 5.0 or onwards Developer

Side:

- OS – 64-bits of Windows 10
- TOOLS – Python version 3, Jupyter Notebook version jupyter
- Browser – Any browser compatible with IE 5.0 or onwards

Communications Interfaces

Protocols - TCP / IP, HTTP



User Characteristics

- As our software product is mainly for farmers, so there won't be many literary constraints. The software will be interactive enough for the user.
- It will be easy to understand and will not require much training, only beginner computer operating skills are required.

Constraints

- The limitations of the model are that we are taking 22 crops into account.
- The limitations of machine learning models depend on the model, problem being solved, and data set used to train the model. Machine learning models can be limited by their accuracy, by the types of problems they can solve, and by the quality of the data used to train them.

Technologies Used

- Machine Learning
- Predictive modeling algorithms include support vector machines, random forests & decision trees.
- Also, we are using Streamlit library to develop our frontend side. **S**treamli**t** is an open-source app framework for Machine Learning and Data Science teams.
- Also, we have used Pickle to connect our frontend framework to that of backend.

Specific Requirements

The software requirements at a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies

those requirements are Operating System can be any for the client and the server, for developer it is Windows 10, Tools used by the developer are Python version 3, Jupyter Notebook version Jupyter v6. 5.3.

Determining levels of nitrogen, phosphorus, potassium. Analyzing soil pH. determining the humidity, rainfall, and temperature.

Soil testing involves collecting soil samples, preparation for analysis, chemical or physical analysis, interpretation of analysis results, and finally making fertilizer and lime recommendations for the crops.

Functional

Collecting the Raw Data

The practice of culminating and Data collection is a way to keep track of past occurrences so that one can utilize the repetitive patterns. The 'Crop Recommendation' dataset is collected from the Kaggle website. The dataset takes into account 5 different crops

1. Nitrogen content ratio (N)
2. Phosphorus content ratio (P)
3. Potassium content ratio (K) in the soil
4. Temperature expressed in degree Celsius
5. Percentage of Relative Humidity
6. pH value
7. Rainfall measured in millimeters.

Data Preprocessing

The process of modifying raw data into a form that analysts and data scientists can use in machine learning algorithms to find insights or forecast outcomes is called Data preprocessing. In this project, the data processing method is to find missing values. Getting every data point for every record in a dataset is tough. Empty cells, values

like null or a specific character, such as a question mark, might all indicate that data is missing. The dataset used in the project did not have any missing values.

Train and Test Split

It is a process of splitting the dataset into a training dataset and testing dataset using train_test_split() method of scikit learn module. 2200 data in the dataset has been divided as 80% of a dataset into training dataset-1760 and 20% of a dataset into testing dataset-440 data.

Fitting the model

Modifying the model's parameters to increase accuracy is referred to as fitting. To construct a machine learning model, an algorithm is performed on data for which the target variable is known. The model's accuracy is determined by comparing the model's outputs to the target variable's actual, observed values. Model fitting is the ability of a machine learning model to generalize data comparable to that with which it was trained. When given unknown inputs, a good model fit refers to a model that properly approximates the output.

Checking the score over a training dataset

Scoring, often known as prediction, is the act of creating values from new input data using a trained machine learning model. Using accuracy_score() method calculating the score of each model over a training dataset shows how well the model has learned.

Predicting the model

When forecasting the likelihood of a specific result, "prediction" refers to the outcome of an algorithm after it has been trained on a previous dataset and applied to new data. Predicting the model using predict() method using test feature dataset. It has given the output as an array of predicted values.

Confusion Matrix and Classification Report

Confusion Matrix and Classification Report are the methods imported from the metrics module in the scikit learn library that are calculated using the actual labels of test datasets and predicted values.

Confusion Matrix gives the matrix of frequency of true negatives, false negatives, true positives and false positives.

Classification Report is a metric used for evaluating the performance of a classification algorithm's predictions. It gives three things: Precision, Recall and f1-score of the model.

Precision refers to a classifier's ability to identify the number of positive predictions which are relatively correct. It is calculated as the ratio of true positives to the sum of true and false positives for each class.

Precision= TP / TP+FP

Recall= TP/TP+FN

F1 Score= 2*PR/(P+R)

Accuracy

The number of correct predictions divided by the total number of predictions is known as model accuracy.

The accuracy of model is calculated using accuracy_score() method of scikit learn metric module Accuracy=TP+TN/TP+TN+FP+FN

Nonfunctional Requirements (Software System Attributes)

Availability

The availability of this web-site is up to the internet connection of the client. Since this is a client-server related web-site shall be attainable all the time. User should have an account to enter the system; if the user does not have an account, then the user can only see the information which will be displayed on the homepage of the web-site.

Security

The authorization mechanism of the system will block the unwanted attempts to the server and also let the system decide on which privileges may the user have. The system has different types of users so there are different levels of authorization.

Reliability

A backup file is maintained so that in case of system crash, the data will not be affected.

Portability

The system is developed using python version 3 and on jupyter notebook version jupyter v6. 5.3 which provides a framework for developing.

Maintainability

This website will follow the modular structure so it will be easy to maintain.

System Architectural Design

Unit Tests

Unit tests are most done by developers on their own machines or on a common server that is very volatile. It is not necessary that the unit test machines be the same platform and operating system as the target deployment environment, but the movement from the unit test environment to other testing environments should not require material code changes by developers. A plan for one machine per developer plus one small server should be included in the overall system architecture.

System Tests

The system test environment allows multiple modules to be connected together and executed as in a typical use-case scenario. The choice as to whether this is done on a separate machine from unit testing is up to the implementation and test team. If the target deployment environment is different from the unit test environment, the system test environment should contain a machine that matches the target environment. Although the system test machine need not match the size of the deployment box, it should have the same platform and operating system. A good rule of thumb is to prepare to add one more box for system tests of a smaller size, but the same operating system as the target environment. Again, this will be a relatively volatile environment, so it should not be viewed as a place to do industrial-strength testing by a large team.

Integration or Regression Tests

To perform integration and regression tests, it is advisable to have a separate environment that is similar to the target environment. Generally, one server will be enough at this point. However, the contents of this server should be strictly controlled. Either the test coordinator or his or her designate should make all software changes to this environment. Stability and auditability are essential to ensuring the accuracy of test results. Plan for at least one more server at this stage in testing.

Stress Tests

Stress tests should be done in an environment identical to the target hosting environment. In the first development cycle, this can be done in the production site, before the cutover to production. For subsequent development cycles, a separate environment will have to be maintained for stress testing.

Plan to replicate the deployment environment as part of the test bed for at least the second development cycle of the site. It has also been observed that the most common problem after performance in a high stress environment is database deadlock due to improper programming. Deadlocks are typically difficult to detect and fix and may not show up until the site is highly stressed in production. So, it is important that these conditions be stringently tested during the stress test phase.

Acceptance Tests and Staging

Acceptance tests are generally performed in the same environment as the stress tests, so additional hardware is not needed to support this phase of testing. Again, during the initial development cycle, the production environment can be used to perform both acceptance testing and staging, but a new environment should be created for subsequent development cycles.

Coding

Importing libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_selection import f_classif, SelectKBest
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, plot_confusion_matrix
```

Uploading dataset

In [2]:
```
df=pd.read_csv('Crop_recommendation.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

In [4]:
```
df.shape
```

Out[4]:
```
(2200, 8)
```

In [5]:
```
df.label.unique()
```

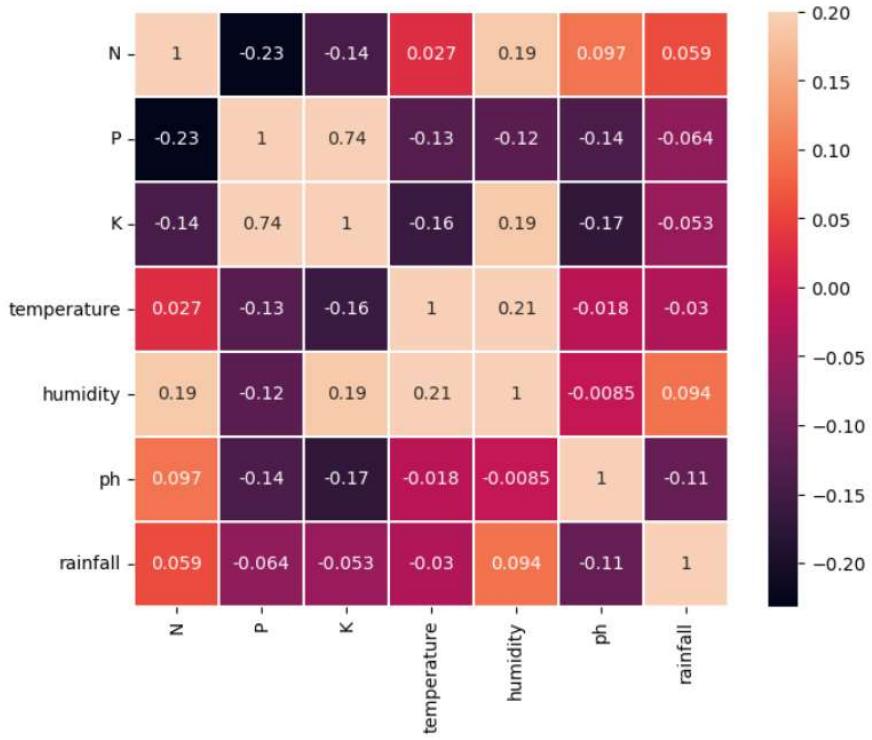
Out[5]:
```
array(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',
       'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',
       'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',
       'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],
      dtype=object)
```

We have imported the dataset from Kaggle: Your Home for Data Science website.

Here we are taking 22 crops into consideration for our model ie. RICE, MAIZE, CHICKPEA, KIDNEYBEANS, PIGEONPEAS, MOTHBEANS, MUGBEANS, BLACKGRAM, LENTIL, POMEGRANTE, BANANA, MANGO, GRAPES, WATERMELON, MUSKMELON, APPLE, ORANGE, PAPAYA, COCONUT, COTTON, JUTE, COFFEE

Correlation using heatmap

In [15]:
```
plt.figure(figsize=(8,6))
sns.heatmap(df.corr(),  cmap='rocket', vmax=.2, center=0,
            square=True, linewidths=.2,annot=True)
plt.show()
```



Data pre-processing

In [17]:
```
lb=LabelEncoder()
```

In [18]:
```
df['label1'] = lb.fit_transform(df['label'])
```

In [19]:
```
df.label.unique()
```

Out[19]:
```
array(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',
       'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',
       'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',
       'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],
      dtype=object)
```

In [20]:
```
df.label1.unique()
```

Out[20]:
```
array([20, 11,  3,  9, 18, 13, 14,  2, 10, 19,  1, 12,  7, 21, 15,  0, 16,
       17,  4,  6,  8,  5])
```

['rice'=20, 'maize'=11, 'chickpea'=3, 'kidneybeans'=9, 'pigeonpeas'=18,

'mothbeans'=13, 'mungbean'=14, 'blackgram'=2, 'lentil'=10, 'pomegranate'=19,

'banana'=1, 'mango'=12, 'grapes'=7, 'watermelon'=21, 'muskmelon'=15, 'apple'=0,

'orange'=16, 'papaya'=17, 'coconut'=4, 'cotton'=6, 'jute'=8, 'coffee'=5]

In [21]:
```
df=pd.DataFrame(df)
```

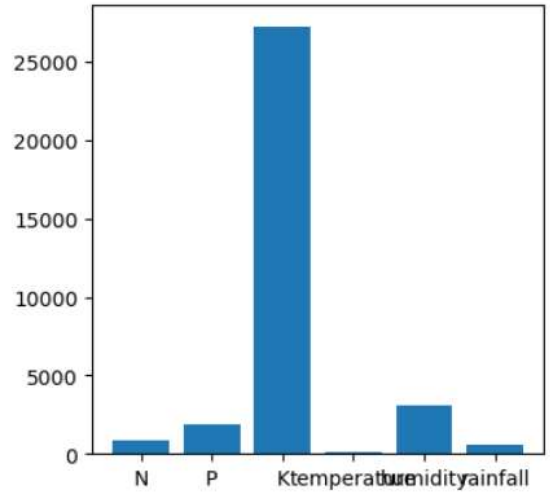
In [22]:
```
df=df.drop(['label'],axis=1)
```

In [23]:
```
y=df['label1']
x=df.drop(['label1'],axis=1)
```

Feature selection

In [56]:
```
fs = SelectKBest(score_func=f_classif, k='all')
fs.fit(x, y)
feature_names=x.columns
for i in range(len(fs.scores_)):
    print('Importance of ' +feature_names[i]+' is %f' % (fs.scores_[i]))
# plot the scores
plt.rcParams["figure.figsize"] = (4,4)
plt.bar([i for i in x.columns], fs.scores_)
plt.show()
```

```
Importance of N is 897.568186
Importance of P is 1885.657859
Importance of K is 27238.362067
Importance of temperature is 102.186981
Importance of humidity is 3103.708891
Importance of rainfall is 605.527966
```



In [27]:
```
x=x.drop(['ph'],axis=1)
```

Spliting the data

In [30]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=2)
```

In [31]:
```
acc = []
model = []
```

Model training

In [32]:
```
dtc = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)
svc = SVC()
rfc = RandomForestClassifier()
```

In [33]:
```
model.append('Decision Tree Classifier')
model_dtc = dtc.fit(x_train,y_train)
```

In [34]:
```
model.append('SVC')
model_svc = svc.fit(x_train,y_train)
```

In [35]:
```
model.append('Random Forest Classifier')
model_rfc = rfc.fit(x_train,y_train)
```

Testing

Prediction ¶

In [36]:
```
pred1 = model_dtc.predict(x_test)
pred2 = model_svc.predict(x_test)
pred3 = model_rfc.predict(x_test)
```

Accuracy

In [37]:
```
tc1 = model_dtc.score(x_train,y_train)
tc2 = model_svc.score(x_train,y_train)
tc3 = model_rfc.score(x_train,y_train)
print("Training Accuracy\nDecision tree",tc1*100,"\nNaive Bayes",tc2*100,"\nRandom Forest",tc3*100)
```

```
Training Accuracy
Decision tree 88.18181818181819
Naive Bayes 98.01136363636364
Random Forest 100.0
```

In [38]:
```
acc1 = accuracy_score(y_test,pred1)
acc.append(acc1)
acc1*100
```

Out[38]:
```
90.45454545454545
```

In [39]:
```
acc2 = accuracy_score(y_test,pred2)
acc.append(acc2)
acc2*100
```

Out[39]:
```
97.72727272727273
```

In [40]:
```
acc3 = accuracy_score(y_test,pred3)
acc.append(acc3)
acc3*100
```

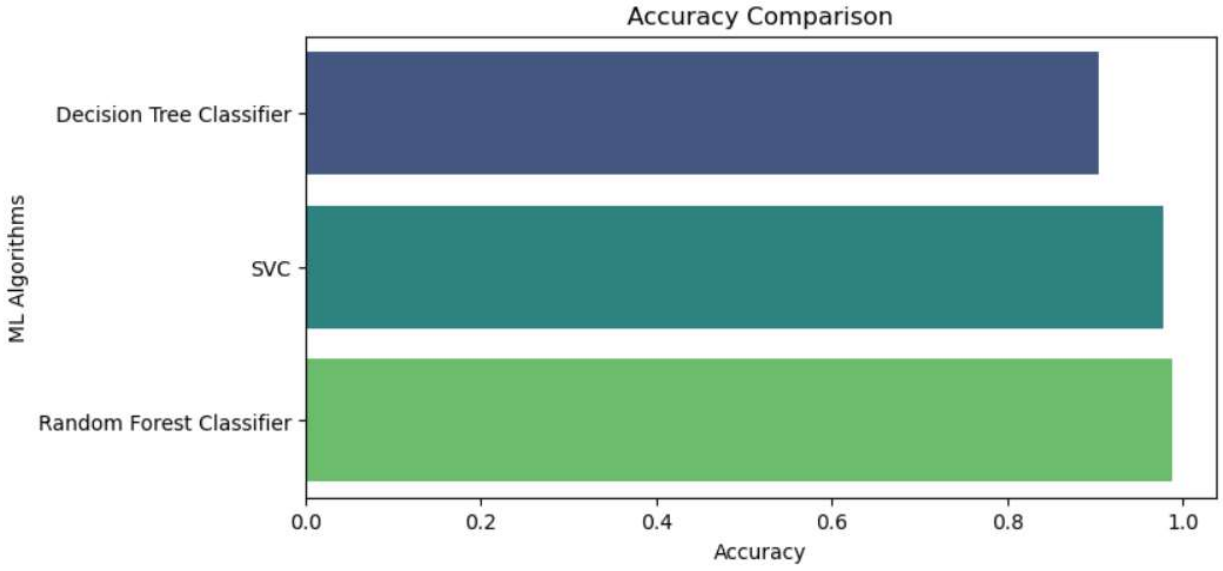
Out[40]:
```
99.0909090909091
```

In [41]:
```
print("Testing Accuracy\nDecision tree",acc1*100,"\nNaive Bayes",acc2*100,"\nRandom Forest",acc3*100)
```

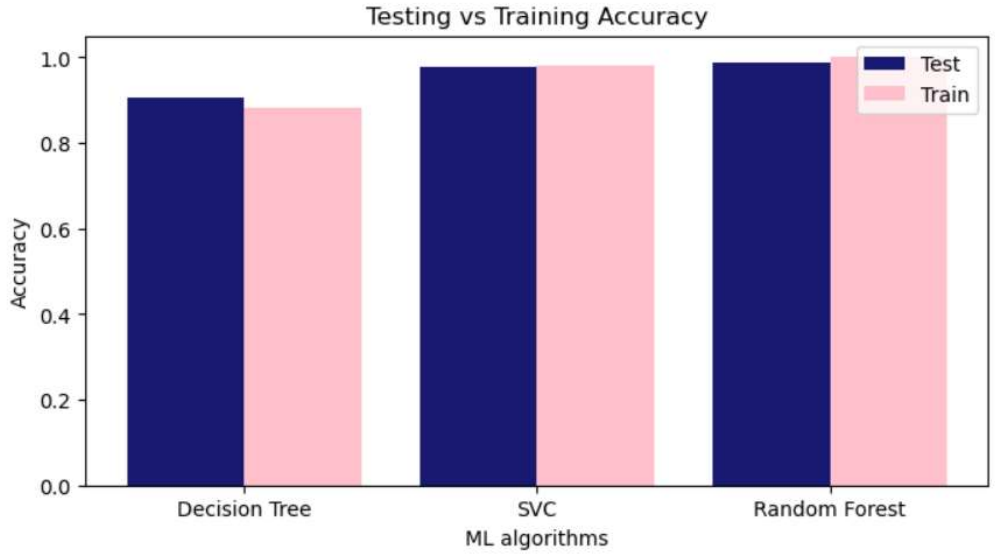
```
Testing Accuracy
Decision tree 90.45454545454545
Naive Bayes 97.72727272727273
Random Forest 99.0909090909091
```

We have tested the accuracy using various models such as Naïve Bayes, Decision Tree, Random Forest. But we are mainly considering Random Forest Classifier as it gives the maximum accuracy

In [51]:
```
plt.figure(figsize=[8,4],dpi = 100, facecolor='white')
plt.title('Accuracy Comparison')
plt.xlabel('Accuracy')
plt.ylabel('ML Algorithms')
sns.barplot(x = acc,y = model,palette='viridis')
plt.savefig('plot.png', dpi=300, bbox_inches='tight')
```



In [52]:
```
label = ['Decision Tree','SVC','Random Forest']
Test = [acc1, acc2,acc3]
Train = [tc1, tc2, tc3]
f, ax = plt.subplots(figsize=(8,4))
X_axis = np.arange(len(label))
plt.bar(X_axis - 0.2,Test, 0.4, label = 'Test', color=('midnightblue'))
plt.bar(X_axis + 0.2,Train, 0.4, label = 'Train', color=('pink'))
plt.xticks(X_axis, label)
plt.xlabel("ML algorithms")
plt.ylabel("Accuracy")
plt.title("Testing vs Training Accuracy")
plt.legend()
plt.show()
```



User Interfaces

User Interface Design

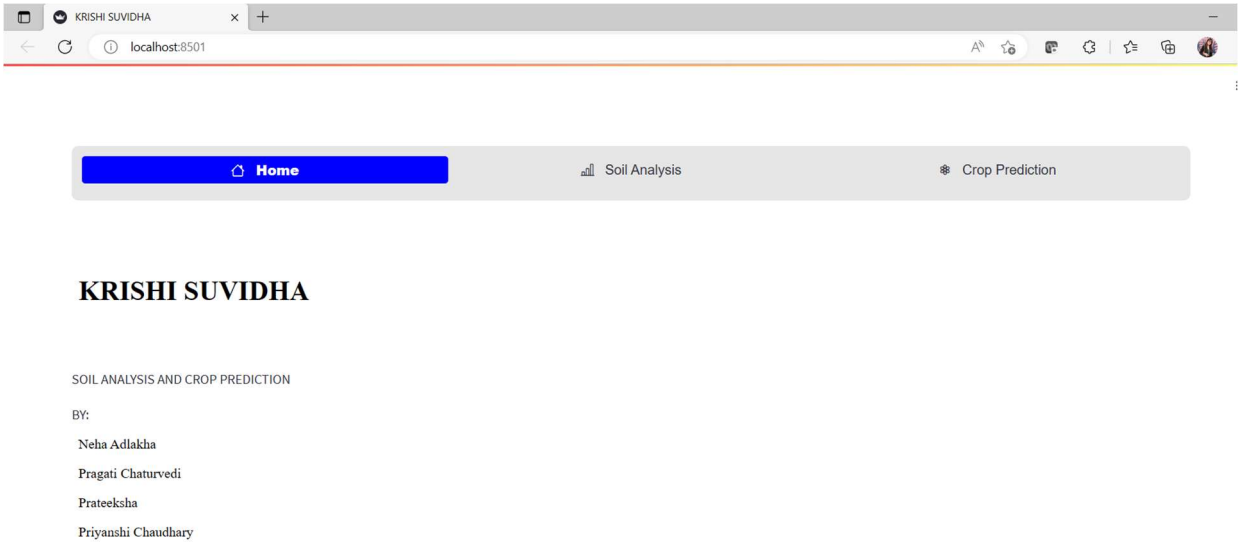
A description of the user interface design of the software is presented.

Description of the user interface

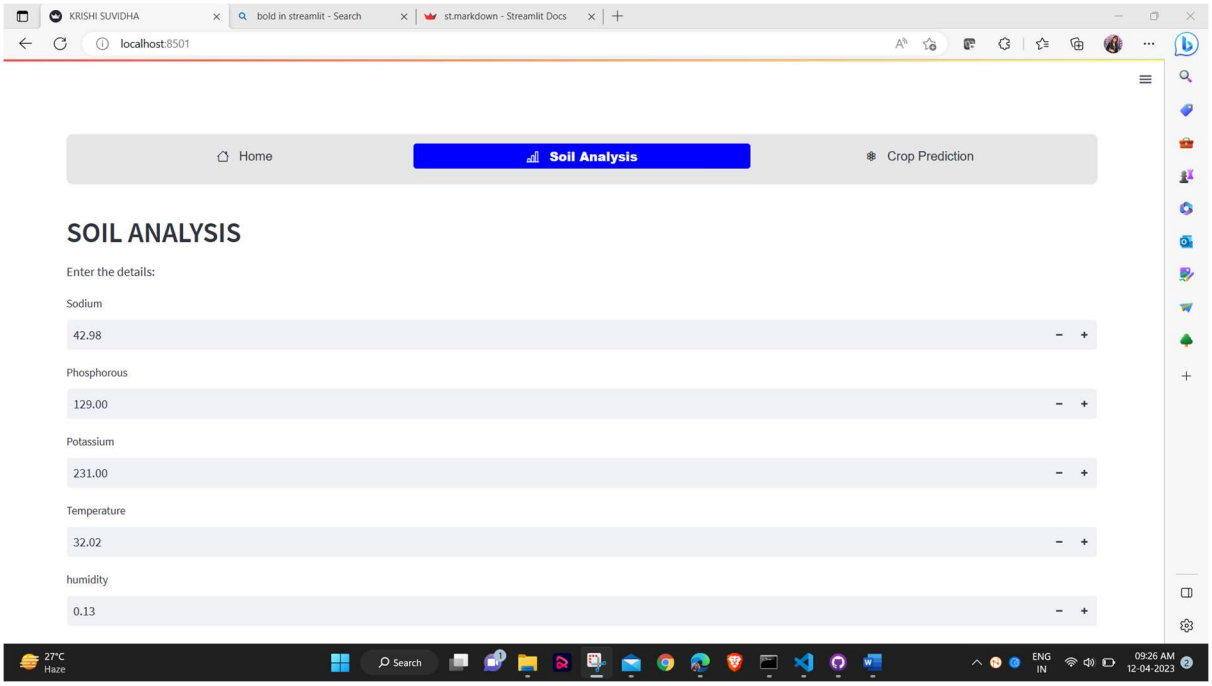
1. First the user will be shown a login page, and he is required to sign up.
2. Then there will be two options i.e., Crop Prediction or Soil Analysis.
3. Upon selecting one of the options, the user will have a set of options. 4. Based on which the result will be shown.

3.Screen Images:

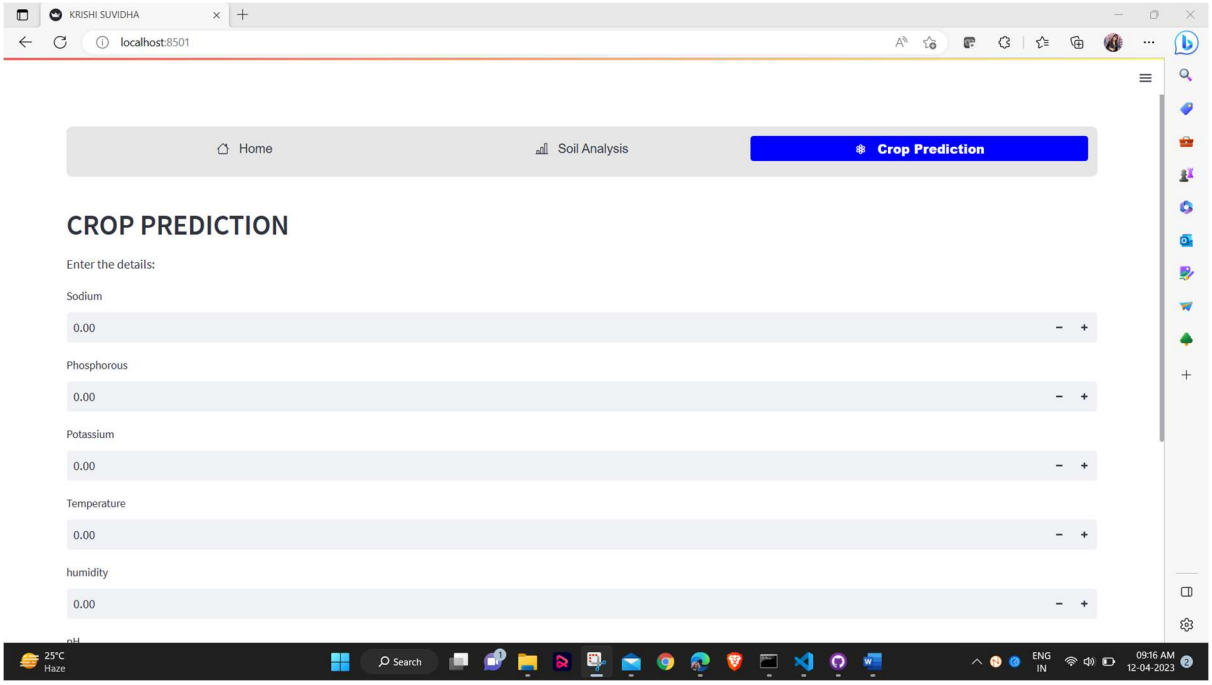
1. Homepage:



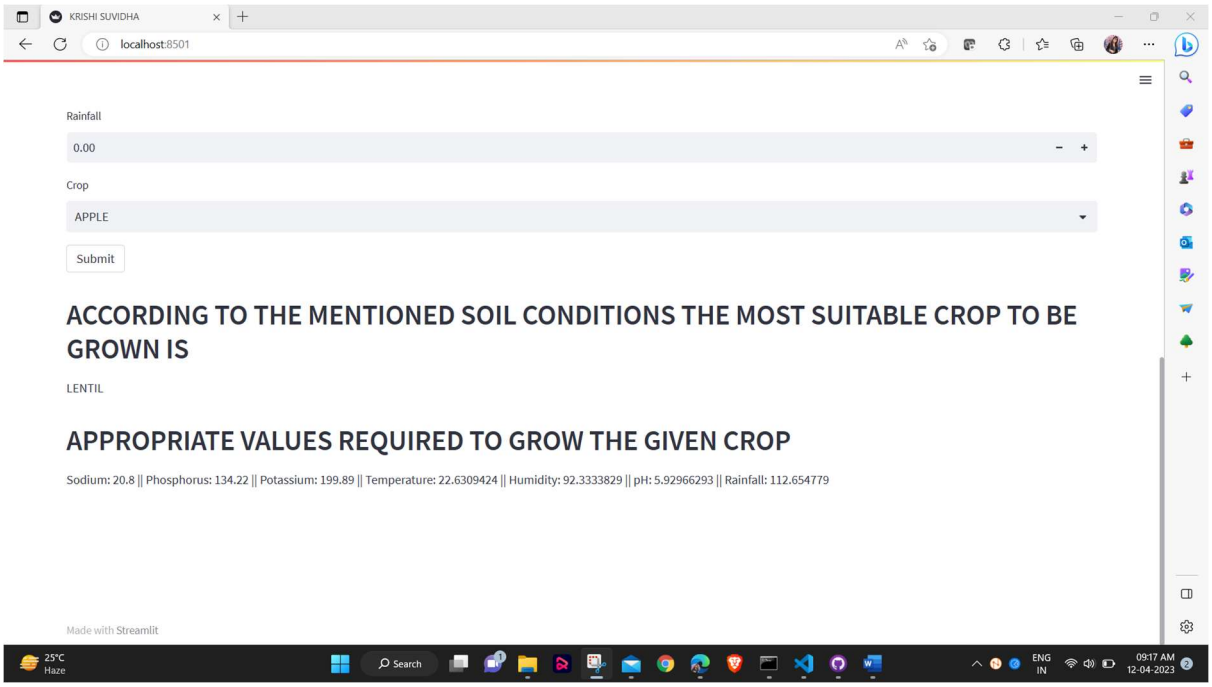
2. Soil Analysis:

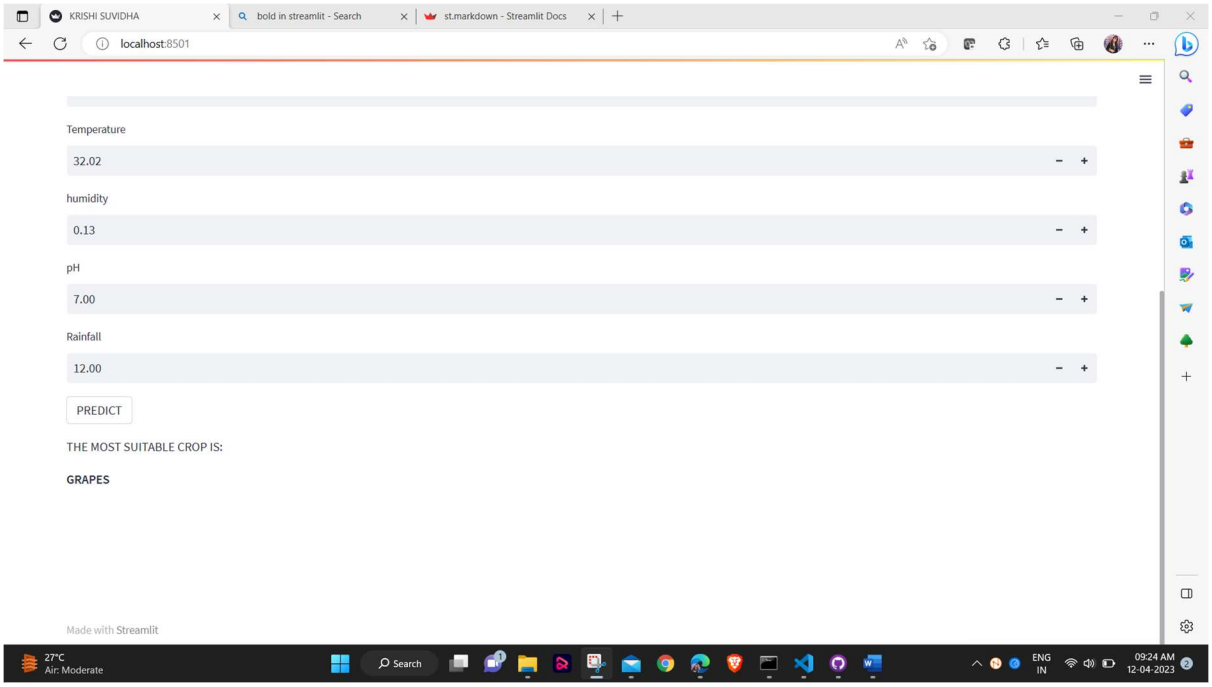


3. Crop Prediction



4. Result


KRISHI SUVIDHA
localhost:8501
Rainfall
0.00
Crop
APPLE
Submit
ACCORDING TO THE MENTIONED SOIL CONDITIONS THE MOST SUITABLE CROP TO BE GROWN IS
LENTIL
APPROPRIATE VALUES REQUIRED TO GROW THE GIVEN CROP
Sodium: 20.8 || Phosphorus: 134.22 || Potassium: 199.89 || Temperature: 22.6309424 || Humidity: 92.3333829 || pH: 5.92966293 || Rainfall: 112.654779
Made with Streamlit
25°C
Haze
Search
ENG
IN
09:17 AM
12-04-2023


KRISHI SUVIDHA
bold in streamlit - Search
st.markdown - Streamlit Docs
localhost:8501
Temperature
32.02
humidity
0.13
pH
7.00
Rainfall
12.00
PREDICT
THE MOST SUITABLE CROP IS:
GRAPES
Made with Streamlit
27°C
Air: Moderate
Search
ENG
IN
09:24 AM
12-04-2023

References

- https://www.tandfonline.com/doi/full/10.1080/13873954.2021.1882505
- https://www.researchgate.net/publication/347382419_Soil_Analysis_and_Crop_Prediction
- https://ieeexplore.ieee.org/document/9725901
- https://www.kaggle.com/code/theeyeschico/crop-analysis-and-prediction/data
- https://www.researchgate.net/publication/347382419_Soil_Analysis_and_Crop_Predict ion
- https://iopscience.iop.org/article/10.1088/1742-6596/2161/1/012033/pdf
- https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3852055
- https://github.com/ankitaS11/Crop-Yield-Prediction-in-India-using-ML/blob/main/Data%20Analysis%20%26%20Visualization%20%20crop%20yield%20dataset.ipynb
- https://www.researchgate.net/publication/221945352_Potential_Uses_and_Limitations _of_Crop_Models