A picture containing diagram

Description automatically generatedLogo

Description automatically generated

*Kingdom of Saudi Arabia*

*Ministry of Education*

*King Faisal University*

***College of Computer Sciences & Information Technology***

**[Crop Recommendation System]**

*A project submitted as a requirement for the course “Artificial Intelligence-CS-411”*

By:

**Group 02- Section 01**

Abdulaziz Alotaibi - 219016732

Abdulrahman Al Majed - 219021871

Ali Alqattan - 219022038

To Laboratory Instructor:

**Mr. Conrado Padua Vizcarra**

**Course Instructor**

**Dr. Abdulelah Algosaibi**

Submission Date: 17/11/2022

Table of Contents

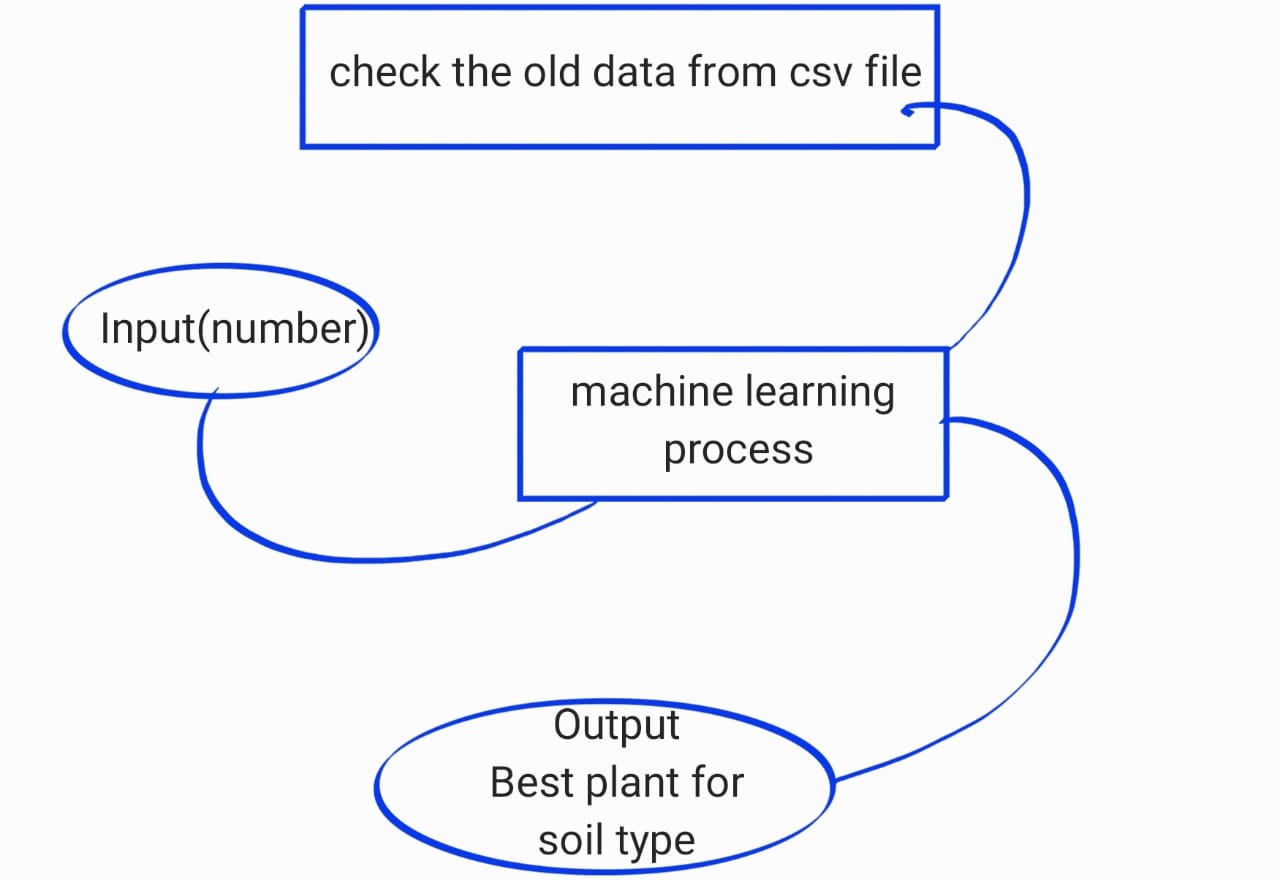
1. Project Abstract 1
2. Algorithm/ Techniques/ Strategy 2
3. Implementation Details and Result 3
4. Prototype3
5. Resources/References 4
6. Tasks Distribution 5
7. Project Abstract

King Faisal University (KFU) identity based on the Kingdom’s vision 2030, aims to be a leading university locally, regionally and internationally in creating a stimulating and future-oriented learning environment. Food security is a vital part for KFU identity. Agriculture is a cornerstone for achieving food security. In this project we will develop a system that finds what is the best plant for the soil. The system will receive a numerical data for a soil and choose the best plant.

1. Algorithm/ Techniques/ Strategy

In this project we tried three different algorithms to execute the program. First algorithm is Decision Tree. Decision Tree algorithm is a supervised learning technique and the most preferred for classification problems. Second algorithm is Support Vector Machine (SVM). SVM is the most popular supervised learning algorithm used for classification problems. Last algorithm is Gaussian Naïve Bayes. The last algorithm is the chosen one for predicting the crop because of its accuracy. We will use all these algorithms to solve the crop recommendation problem. A comparison between all algorithms will be calculated to choose one of them to predict a solution for a given soil data.

1. Implementation Details and Result



At the beginning, we entered the data into the program, arranged and separated it in terms of size, and in terms of the number of columns and rows and their size, and they were checked, and the missing data was erased, if any. Then we use three algorithm which is Decision tree, Gaussian Naive Bayes and Support Vector Machine from sklearn library, and we calculate the Accuracy for each algorithm. Based on our application in calculating accuracy, we chose Gaussian Naive Bayes because it is the best in terms of accuracy.

* N - ratio of Nitrogen content in soil
* P - ratio of Phosphorous content in soil
* K - ratio of Potassium content in soil
* temperature - temperature in degree Celsius
* humidity - relative humidity in %
* ph - ph value of the soil
* rainfall - rainfall in mm

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| column  Name | N | P | K | Temp | Humidity | Ph | Rainfall | Output |
| Case 1 | 50 | 30 | 10 | 40 | 55 | 5 | 175 | pigeonpeas |
| Case 2 | 88 | 20 | 40 | 23 | 70 | 6 | 201 | Jute |
| Case3 | 90 | 20 | 50 | 20 | 80 | 4 | 150 | Rice |

Dataset

Graphical user interface, text

Description automatically generated

Import the libraries that we will use for our project

Graphical user interface, text, application

Description automatically generated

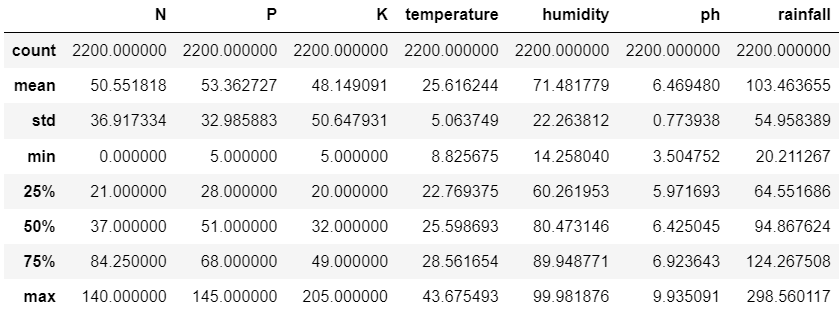
Read the dataset file csv using read\_csv then displays the first five rows of dataset (crop.head)

Table

Description automatically generated



Describe shows a description of the data as shown below





This line checks if a column contains null.

A picture containing chart

Description automatically generated

(crop.size) is rows\*columns. (crop.shape) is rows, columns (2200,8).

(crop.columns) displays names of columns for data set .

Text

Description automatically generated

This line of code displays unique values of label column to see how many plants.



Shows the type of each column

Table

Description automatically generated with medium confidence



Counts number of each label.

صورة تحتوي على منضدة

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص

تم إنشاء الوصف تلقائياً

In this part we are separating the features and target to use the features and

label by one Word. Acc is the variable for the accuracy an model is the variable for the model. Then we Splitting into train and test data.

**Decision Tree Algorithm**



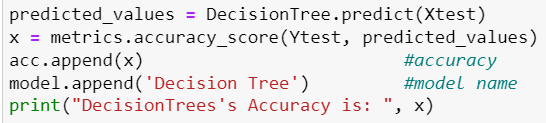
First import the Decision Tree Classifier from sklearn.tree library



Criterion entropy for shannon information gain. The maximum depth of the tree is five.



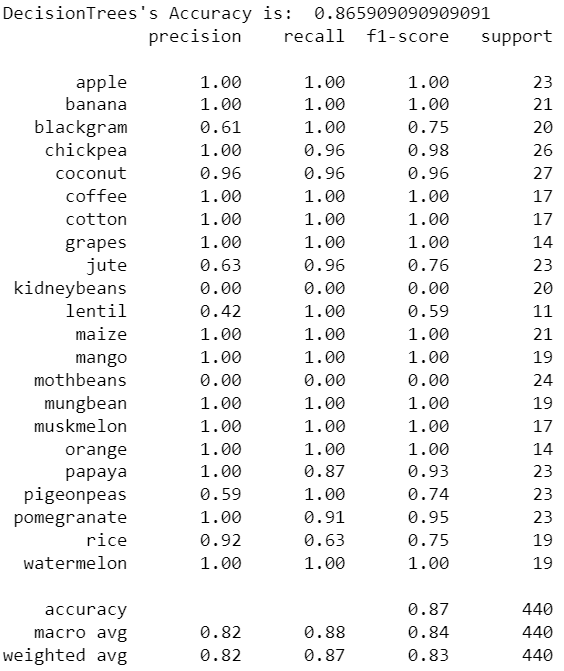
Train Decision Tree Classifier



Apply the Decision Tree algorithm. Then, calculate the accuracy and assign it to variable x. After that append variable (x) “accuracy value” to accuracy array and append model name to model array. We used these two arrays for comparison and demonstration between other algorithms.



Last, print the precision, recall and F1-score values using classification report from sklearn.metrics .

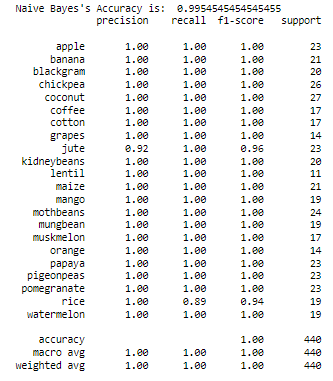


Gaussian Naïve Bayes Algorithm

صورة تحتوي على نص

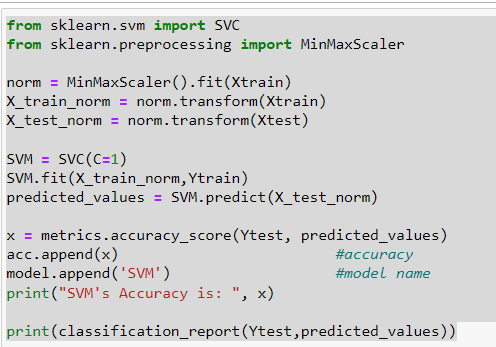
تم إنشاء الوصف تلقائياً

In this part of the code we used the naive bayes library to call its functions to measure the accuracy of this algorithm on dataset. First, let's define and call the sklearn.naive\_bayes and we assign it to variable called Naive Bayes. then we used fit function that take two parameters. the first parameter is the features of the training data and the second parameter is the target or the label of training data. Secondly, we call predict function that take one parameter which is the features of the test data and assign it to new variable called predicted\_values. then we used metrics.accuracy\_score function that takes two parameters which are the true labels and the predicted labels to calculate the accuracy score for a set of predicted labels against the true labels. Then add the result in acc array and add algorithm name in model array to compare it with others algorithm. The last thing we used classification\_report function that takes two parameters which are the true labels and the predicted labels to generate precision, recall, F1 score and support.

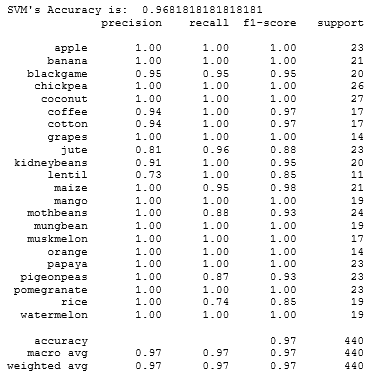


accuracy of the Gaussian naïve bayes

**Support Vector Machine (SVM)** **Algorithm**



Here we use support vector machines to display classification report which include the Accuracy, precision, Recall, f1-score and support. Support vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. From sklearm.svm we import SVC to fit x train which it is the all number (input) and y train is the output which it is the label. X is the accuracy. In the last line we print the accuracy and the classification report.

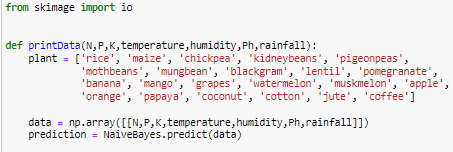
****

صورة تحتوي على نص

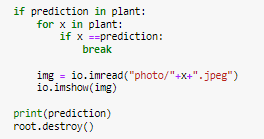
تم إنشاء الوصف تلقائياً

here we display Approximation accuracy of the three algorithms.

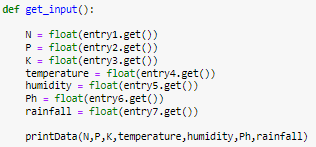
**Graphical user interface (GUI)**

****

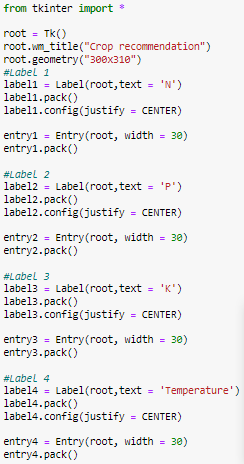
In the part of code we used Skimage library to enable the user to enter the inputs. We created printdata function that takes 7 parameters which are the features of the dataset. then we created an array called "Plant" to store all the plants in dataset. then we used np.array function and pass parameters in the printdata through it and store it in data variable. Then we used NaiveByaes.predict function and gives it "data" that we wanted to predict it.



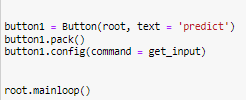
Here we created if condition to check if the prediction exist in the plant array or not. Then we created inner for loop to going through the array to take the plant name to use it in displaying the photo. Then we used imread function to access the image path. Then we used imshow function to display the image. lastly, we printed the plant name and destroy the frame.



Here we created our function called get\_input that will collect all inputs from the user using "get" function, then store each input in the "printdata" function.



Here we import tkinter library. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. wn\_title function for rename the frame and geometry function for the size of the frame. The Label widget is used to provide a single-line caption for other widgets.



The Button widget is used to display buttons in your application. we use Button function with two parameters:

root − This represents the parent window.

text − The name of button

config function has "command" as a parameter

command is function or method to be called when the button is clicked.

tells Python to run the Tkinter event loop. This method listens for events, such as button clicks or keypresses, and blocks any code that comes after it from running until you close the window where you called the method.

1. Prototype

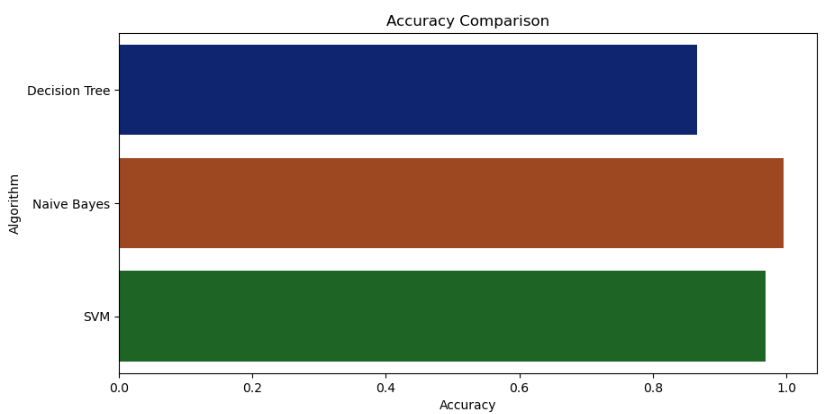


Figure 1: Comparison of the types of models used

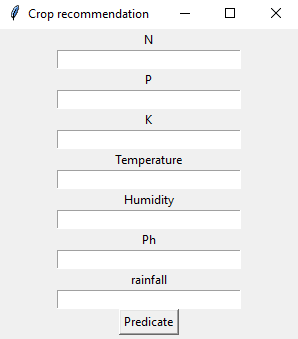


Figure 2: A user interface that enables the user to enter data

**Figures:**

Figure 1 "barplot": A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories. And this figures use seaborn library.

Figure 2 " tkinter ": Python has a lot of GUI frameworks, but Tkinter is the only framework that’s built into the Python standard library. Tkinter has several strengths. It’s cross-platform, so the same code works on Windows, macOS, and Linux. Visual elements are rendered using native operating system elements, so applications built with Tkinter look like they belong on the platform where they’re run. We used some components of this library such as Labels, Button and Entries.

1. **References and resources**

We used Jupiter notebook to implement and execute python code and we use Crop Recommendation Dataset for our project . We used multiple libraries for the program, some of which help to read files and some of which are happy to sort and arrange them, in addition to the libraries of mathematical algorithms to calculate the accuracy and deduce the best result for the data entered through the saved data.

we use Dataset about Crop Recommendation from Kaggle website.

Dataset file link :

<https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset>.

Tkinter:

<https://realpython.com/python-gui-tkinter/>

1. **Tasks Distribution**

|  |  |
| --- | --- |
| **Group Member** | **Tasks Performed** |
| **Abdulaziz Alotaibi** | **Model & GUI** |
| **Abdulrahman Al Majed** | **Model & Dataset & Slides** |
| **Ali Alqattan** | **Model & GUI** |

In the beginning, we divided the project into three main components. First, Choosing the idea and searching for the dataset, Abdulrahman and Ali worked on it. Second, we agreed to choose three types of algorithms: SVM, Decision Tree, and naive bayes. And each of us worked on one of them. Last, Abdulaziz worked on the graphical user interface.