**CROP RECOMMENDATION SYSTEM**

# ABSTRACT

Agriculture is an important field worldwide, where there are many challenges in solving problems in conditionally estimating crops. Many solutions have been proposed regarding this problem using IOT-based services and mechanical technology to reduce manual work. These methods are mainly useful in the case of minimizing manual labor but not in the prediction process. It is necessary to be able to predict the optimal crop to plant based on the soil condition to minimize losses, harm and maximize profits. We build machine learning models to recommend optimal crops to growers based on many parameters and help them make informed decisions before farming. Dataset prepared with values of nitrogen, phosphorus, potassium and soil pH, temperature and rainfall required for a particular crop. In this system, we will provide records approximately the specific traits of the soil and the temperature, humidity, autumn conditions of the region, primarily based totally on which we can make appropriate crop prediction.

**Keywords:** Random Forest, Decision Tree, KNN Neighbors,Logistic Regression

# INTRODUCTION

Agriculture is the basic source of food supply of all the countries of the world—whether underdeveloped, developing or even developed. The world population is estimated to be about 9.7 billion by 2025. This added with unpredictable weather conditions makes it difficult to ensure food sustainability. Fortunately there is a solution for this problem as for many others. Crop Recommendation System takes the N-P-K (Nitrogen, Phosphorous and Potassium) and pH values along with the temperature, humidity values as input and recommends the optimal crop to the farmer, hence ensuring that the farmer takes an informed decision before cultivation. In this system , we train the model using Random Forest , Decision Tree and KNN neighbor . We compare the accuracy of this models and choose the best out of it and store that specific model using pickle module and deploy the machine learning model using Flask. A user can input the various parameters like Nitrogen , Phosphorous , potassium , PH value , Rainfall and Location by interacting with user interface to predict the appropriate crop.

# EXISTING SYSTEM

Existing system involves image analysis that is used for detecting land type based on which further analysis is carried out. Though this method is environmentally safer, it does not provide accurate results as soil conditions are not considered and hence is not feasible. It is a time taking process and also has some added complexity which may not be beneficial and accessible to farmers at all times.

The following are the drawbacks of the Existing System:

* Time Taking Process
* Complicated to use for practical purposes by farmers.  Not accurate

# PROPOSED SYSTEM

We aim to create a system that can accurately tell the farmer the suitable crop to be grown based on the input features of the soil and the temperature conditions of the region. In this system, the models will be trained on a textual data set which will be engineered carefully after performing the feature engineering. The user can interact with the model through a website which takes the necessary inputs and loads the trained model. Based on the input data, the model makes a prediction on the optimum crop to be cultivated. The result is then displayed to the user. Instead of directly taking the temperature and humidity values from the user, the website asks the user for their location. By using a weather API, the system automatically retrieves the temperature and humidity values for that region.

**MERITS-**

* User friendly.
* Very easy to use, as data is submitted using forms.
* Information sharing is easy.

# SYSTEM ANALYSIS

This System Analysis is closely linked to the requirements analysis. It's also "an explicit formal inquiry done to aid someone (referred to as the decision maker) in discovering a better course of action and arriving at a better conclusion than he would have reached on his own." Breaking down the system into distinct components to assess the situation, reviewing project goals, breaking down what needs to be constructed, and engaging users to determine particular requirements are all part of this process

**Functional Requirement Specification**

The System after careful analysis has been identified to be present with the following modules.

**Administration Module:**

The administrator forms the core of the system. Administrator is responsible for training the models on the data and loading the model when desired by the user. When a user submits the input data, the administrator loads the trained model and displays the result. Another responsibility of the Administrator is to fetch the weather data using the weather API.

**User Module:**

This module represents the user of our system. The user can submit the input data to the admin. The data includes the N-P-K, pH values and rainfall. The user then waits for the response from the Administrator module. **Machine Learning Model Module:**

The machine learning model is trained on the dataset. After satisfactory results are obtained, it is saved for later retrieval. It is invoked by the Administrator and performs prediction on the data supplied to it.

**Performance Requirements**

The output supplied by the application is used to assess performance. The specification of requirements is crucial in the study of a system. It is only possible to develop a system that will fit into the appropriate environment if the requisite specifications are properly supplied. Because they are the ones who will use the system in the end, it is mostly up to the users of the existing system to provide the requirement specifications. This is because the requirements must be known during the early stages of the project so that the system can be designed to meet them. It is difficult to update a system once it has been designed, and developing a system that does not meet the needs of the user is pointless.

The current system is entirely reliant on the user to execute all tasks. for any system can be broadly stated as given below:

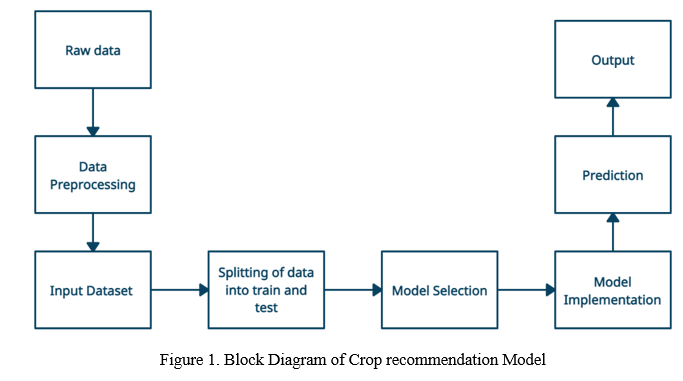
* The system must be able to communicate with the existing system.
* The system should be accurate
* The system must outperform the present system.

 The existing system is completely dependent on the user to perform all the duties.

# SYSTEM DESIGN

The process of establishing the architecture, components, modules, interfaces, and data for a system in order to meet specific criteria is known as systems design. It might be viewed as a product development application of systems theory. The most extensively used approaches for computer system design are object-oriented analysis and design methodologies.

**Architecture=**



The first stage is building the required machine learning model and second stage is deploying it to production. First we load the crop dataset, explore the data and perform the necessary feature engineering. After subjecting our data to proper data cleaning we split it into training and testing sets. We train the model on the training data and test its performance on our test set. After choosing the suitable model for our task, we save the data locally to be accessed by our application when required. The application loads the model that was saved in the previous phase and waits for user input. The weather data is fetched using an API. The input data is transformed into a suitable format. The model that was saved earlier is invoked to get the result. The result is displayed to the user through User Interface.

# SYSTEM IMPLEMENTATION

Any project's implementation stage is a true representation of the defining moments that determine whether a project succeeds or fails. The system or system modifications are installed and made operational in a production environment during the implementation stage. After the system has been tested and accepted by the user, the phase begins. This phase continues until the system meets the defined user requirements and is ready to go into production.

**Sample Code For Our Project:**

**Data Cleaning/Exploration:**

Data Exploration is one of the most important tasks when it comes to tackling a Machine Learning problem. In order to truly understand the problem at hand, we must understand the data that we are dealing with. We can be able to view the first and last five rows of our data set to get an idea of what the features of our data are and how they look.

df=pd.read\_csv("Crop\_recommendation.csv") df.head() df.tail()



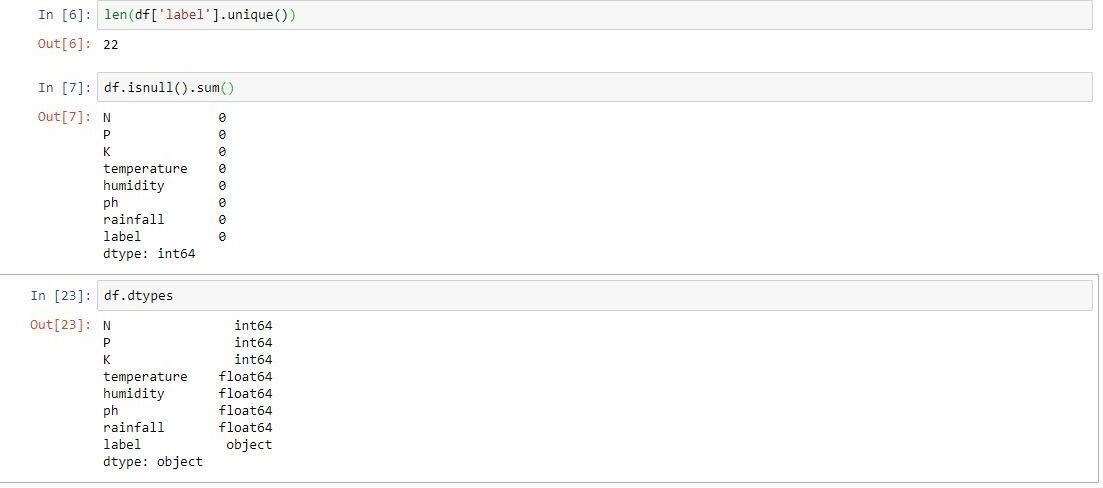
**Figure 2:** Crop Recommendation Dataset

We must also understand the shape of our data set. That is the number of features and the total number of records. We use Pandas shape attribute to do this:

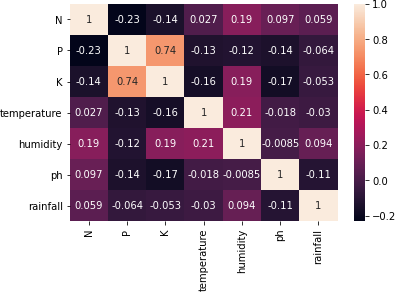
df.shape

(2200, 8)

This dataset consists of 2200 records and 8 columns. Since this is clearly a classification problem, it is necessary to know the number of classes in our target label. It is also necessary to know the total number of record that constitute each class in our data set. We must also check for null values and clean them out if any. To accomplish this, we can use the following code:



One of the most popular ways of exploring the relations between the features in our data is using correlation matrices. Pandas has a built in correlation matrix for a data frame that we can use to truly understand the features in our data. By default the correlation is Pearson’s Correlation. Here the values range between -1 and 1



**Figure 4**: Correlation Matrix Of Dataset Features.

From the figure 4 we can say that most of the features are important. We have assumed a coefficient threshold of 0.8 and hence most of the features are necessary for us. **Splitting The Data Into Training And Testing Datasets:**

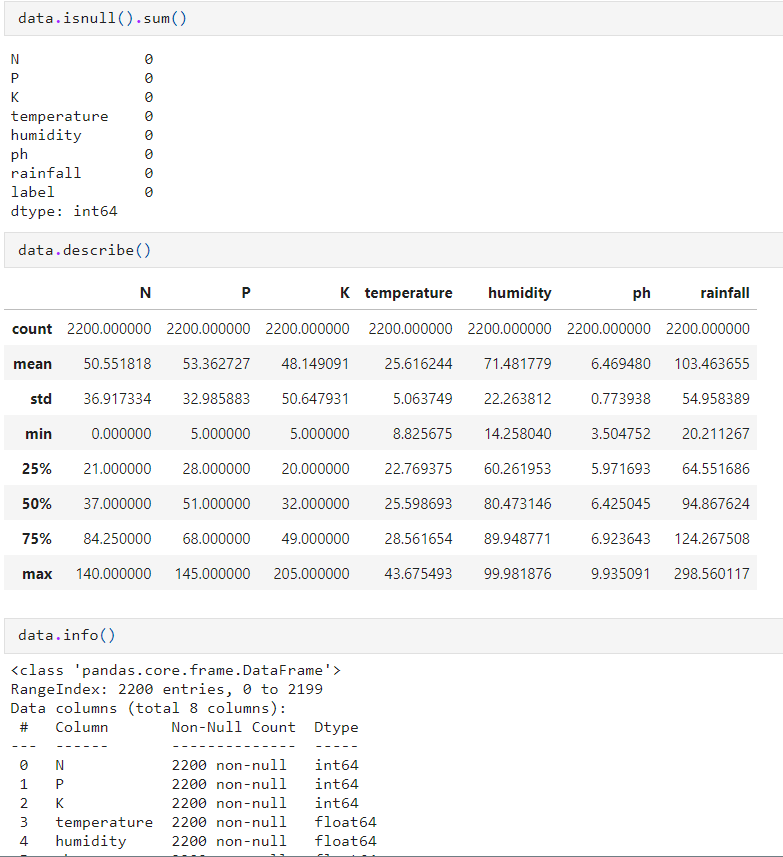
Before we split our data into training and testing sets, we must separate the features and target labels that we want to classify the data into. We do this by:

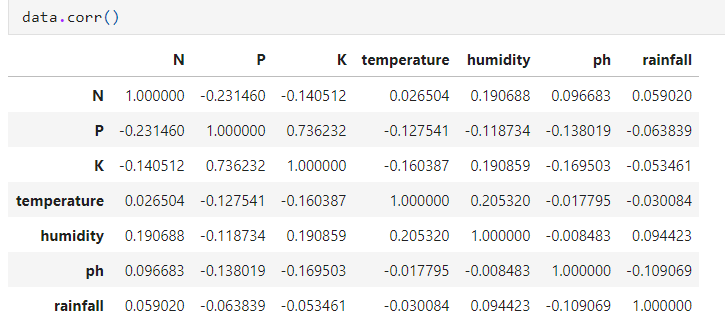
x=df.drop("label",axis=1) #Column wise Drop y=df['label'] print(x) print(y)

In order to solve the classification problem, we must split the data into training and testing datasets. The training data set is the data on which our models will be trained on, and the testing set is the data in which we evaluate the performance of our model using various metrics. We use the sklearn module, which has the necessary implementation of how the split is done and we also import classification\_report of sklearn. This will be useful in order to evaluate our model later on. from sklearn.model\_selection import train\_test\_split from sklearn.metrics import classification\_report,confusion\_matrix x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=2)

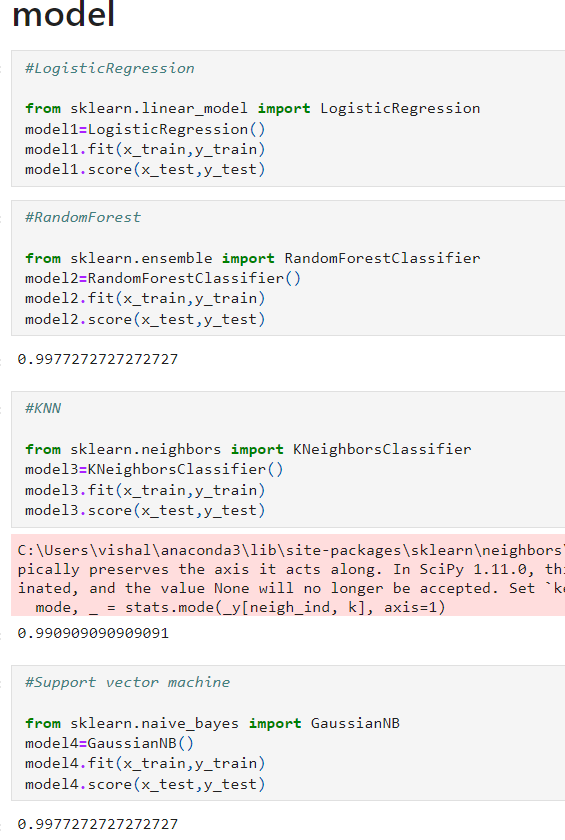
In the above step, we have split the data such that 20% of the dataset is testing data and the remaining 80% of the data is training data. As the split is done in a random order, we set the random state 2.

**Data Cleaning=**





**Algorithms=**

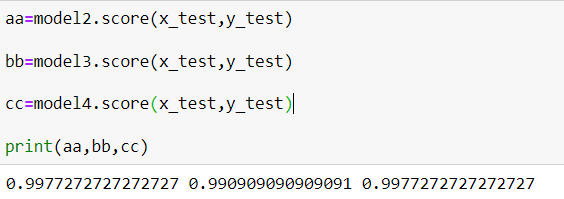


**Accuracy of different models=**

aa= Random Forest

bb= KNeighborsClassifier

cc= Naive\_Bayes

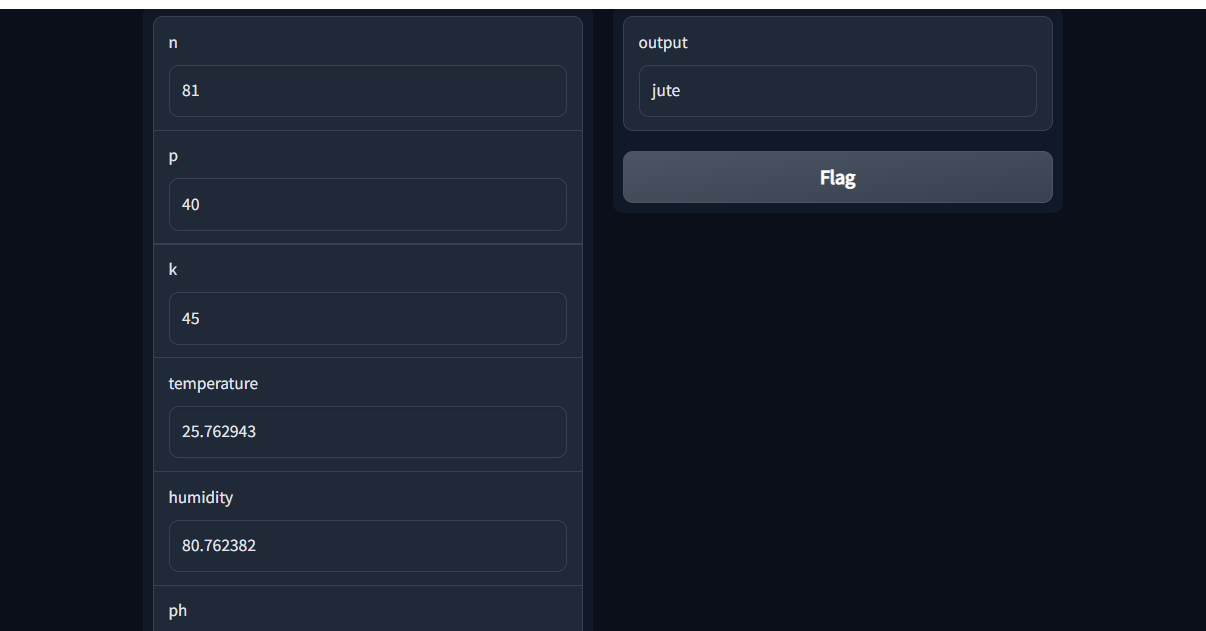


On the basis of above implementation of different types of algorithm we can assume that Random forest is the best algorithm for our crop recomondation system.

**User Interface=**

The Gradio interface can be designed to display the recommended items in a user-friendly and visually appealing manner, such as a grid of movie posters or a list of titles with accompanying descriptions. Users can also provide feedback on the recommended items, which can be used to improve the recommender system over time.

# RESULTS AND DISCUSSION



# CONCLUSION

The Crop Recommendation System is mainly used to recommend the optimal crop to the Farmer. With many industries going digital , it is important that the agricultural sector also take advantage of the various technologies by using them to solve problems faced by the farmers. Using the approach that we discussed, farmers can expect greater yields. Thissystem can be integrated with other smart agricultural systems that already exist. It is cost efficient and helps the farmers make an informed decision.

# REFERENCES

1. Stephen Marsland, ―Machine Learning – An Algorithmic Perspective‖, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.
2. Grady Booch, James Rumbaugh, Ivar Jacobson. The Unified Modeling Language User Guide. Addison- Wesley, Reading, Mass., 1999.
3. https://scikit-learn.org/
4. https://materializecss.com/
5. www.w3schools.com
6. www.wikipedia.org