

# Project Report

on

**TOPIC:** Machine Learning Approach For *Crop*  
RECOMMENDATION SYSTEM

**FARM-EASY**

**Group Members:**

**Aman Kumar Jha –21MCA10142**

**Kaushal Kishor Sharma-21MCA10050**

**Velasiri Muni Raghava-21MCA0008**

**Guide By : ARPITA BARONIA**



**VIT<sup>®</sup>**  
**BHOPAL**  
[www.vitbhopal.ac.in](http://www.vitbhopal.ac.in)

*in partial fulfillment for the award of the degree  
of*

**MASTERS**

in

**COMPUTER APPLICATIONS**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTRIKALAN, SEHORE**

**MADHYA PRADESH - 466114**

# **INDEX**

## 1.Introduction

### 1.1 Overview

### 1.2 Purpose 2

## 2Literature Survey

### 2.1 Existing Problem

### 2.2 Proposed Solution3

## 3.Theoretical Analysis

### 3.1 Block diagram

### 3.2 Hardware / Software designing4

## 4.Experimental Investigations

## 5 Flowchart

## 6 Result

## 7 Advantages & Disadvantages8

## Applications

## 9 Conclusion

## 10 Future scope

## 11 Biblography

## 12 Screenshots of project

## **1. Introduction**

### **1.1 Overview**

Precision agriculture is in trend nowadays. Precision agriculture is a modern farming technique that uses the data of soil characteristics such as (ratio of nitrogen

phosphorus and potassium) 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 the crop failures and will help the farmers to take informed decision about their farming strategy.

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.

### **1.2 Purpose**

To recommend optimum crops to be cultivated by farmers based on several parameters and help them make an informed decision before cultivation.

## **2 Literature Survey**

### **2.1 Existing Problem**

The effects of climate change affect farmers' ability to grow the food we all need. Increasingly volatile weather and more extreme events – like floods and droughts – change growing seasons, limit the availability of water, allow weeds, pests and fungi to thrive, and can reduce crop productivity.

Soil erosion is reducing the amount of land available for agriculture, and declining biodiversity affects the pollination of crops. At the same time, farmers are under pressure to conserve water and use fewer agricultural inputs.

As they adapt to these changes, farmers also need to mitigate the greenhouse gas emissions contributed by agriculture through adopting climate-smart practices – a new learning journey for many.

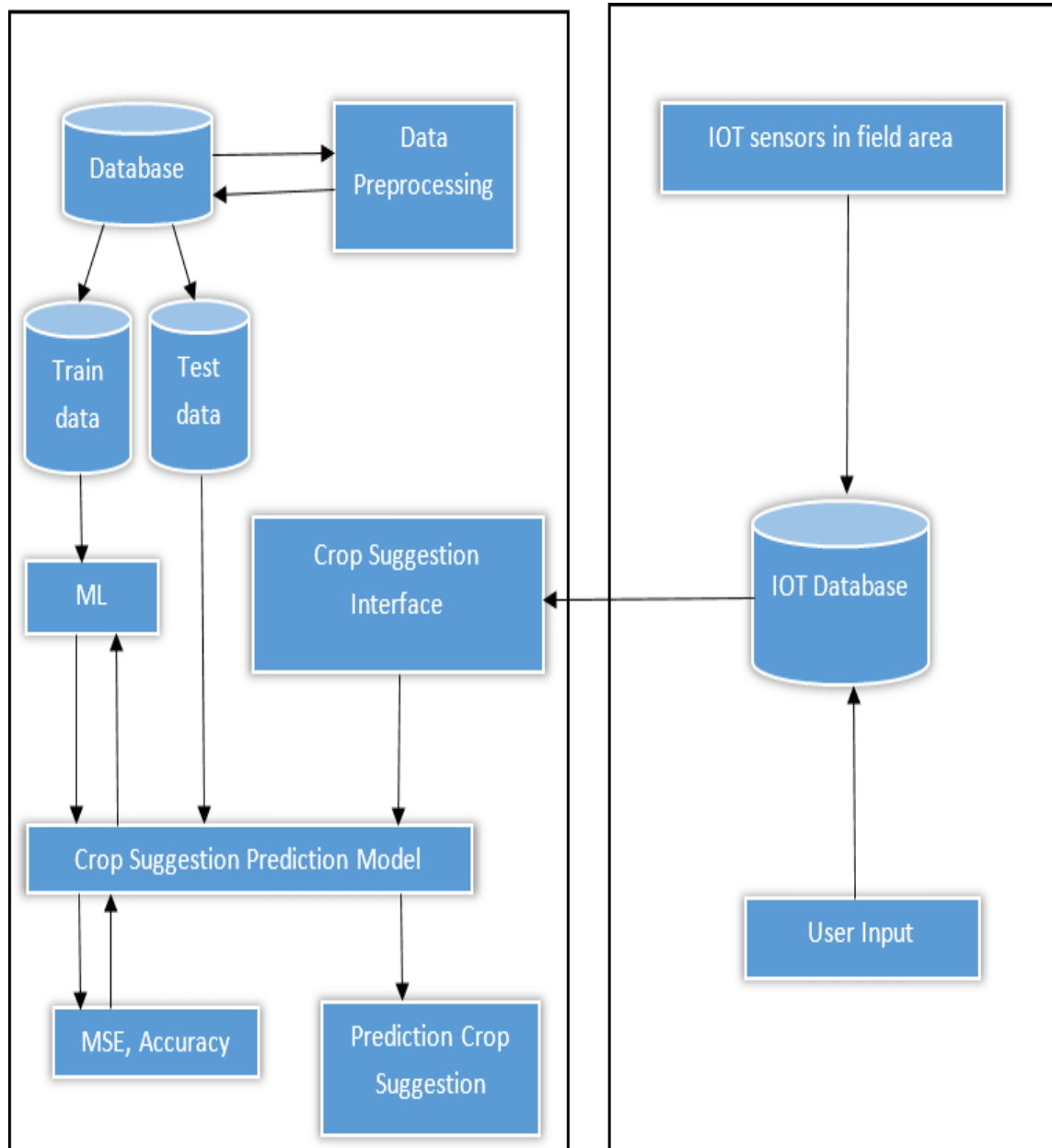
By the report of NCRB the death rate of farmers in India is raised by 18% in 2021 . This is the serious problem for not only India but the whole world.

## **2.2 Proposed Solution**

To solve this problem, we used the available datasets of an crop recommendation with quality of soil , climatic bebhaviour like humidity ph scale and temperature . We used machine learning algorithm called Naïve Bayes to classify which crop should we cultivate in which climatic condition. Then a beautiful interface is made in flask application and styled with the help of css. The predicted output is then send back through an api.

## **3 Theoritical Analysis**

### **3.1 Block diagram**



### 3.2 Hardware / Software designing

Hardware requirements includes

- i) Processor generation 8 or above
- ii) 4 cores or more and 8 threads or more

iii) Compatible with 4GB but should go for 8 and above

iv) Operating system: windows 10, linuxs

Software requirements includes

i) Python

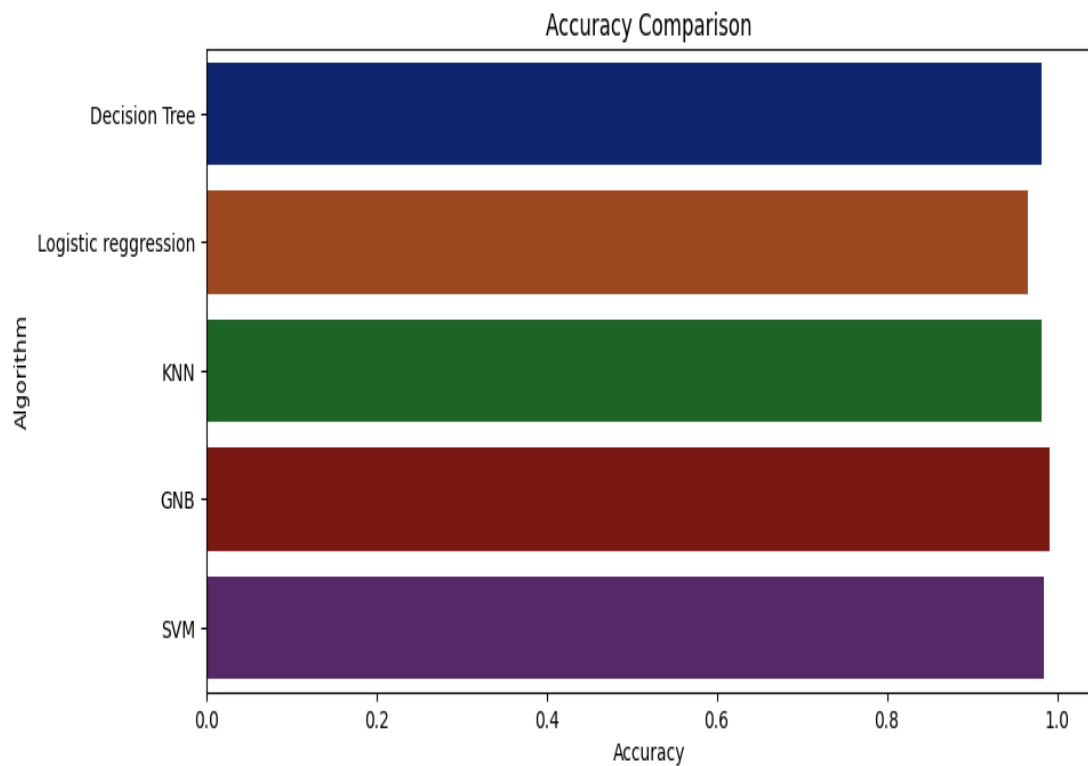
ii) Flask with its modules(render\_template, request)

iii) Machine Learning modules i.e sklearn, numpy, pandas, etc...

iv) Html5 , Css3

## 4 Experimental Investigations

```
from sklearn.metrics import accuracy_score\nfrom sklearn.metrics import confusion_matrix\nfrom sklearn.metrics import classification_report\nfrom sklearn.metrics import roc_auc_score\nfrom sklearn.metrics import roc_curve\nfrom sklearn.metrics import precision_recall_curve\nfrom sklearn.metrics import average_precision_score\nfrom sklearn.metrics import f1_score\nfrom sklearn.metrics import recall_score\nfrom sklearn.metrics import precision_score\nfrom sklearn.metrics import log_loss\nfrom sklearn.metrics import brier_score_function\nfrom sklearn.metrics import jaccard_index_score\nfrom sklearn.metrics import hamming_loss\nfrom sklearn.metrics import cohen_kappa_score\nfrom sklearn.metrics import matthews_correlation_coefficient\nfrom sklearn.metrics import log_likelihood_ratio\nfrom sklearn.metrics import log_loss\nfrom sklearn.metrics import brier_score_function\nfrom sklearn.metrics import jaccard_index_score\nfrom sklearn.metrics import hamming_loss\nfrom sklearn.metrics import cohen_kappa_score\nfrom sklearn.metrics import matthews_correlation_coefficient\nfrom sklearn.metrics import log_likelihood_ratio
```



	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
...	...	...	...	...	...	...	...	...
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

2200 rows × 8 columns

Table: The data set we have used in our project.

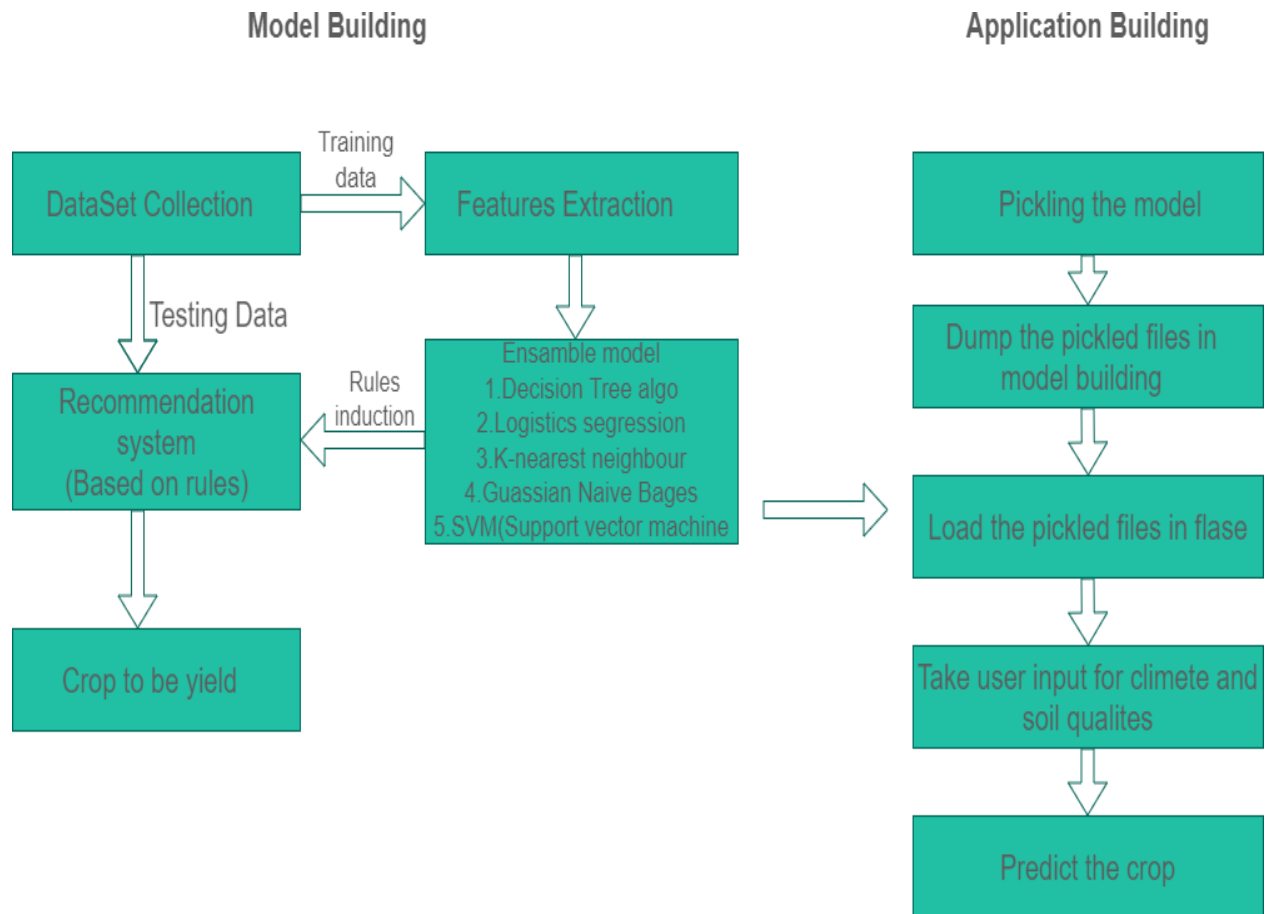
Points to be noted:

- i) Accuracy are given in range of [0,1]. One can multiply it with 100 to get in % percentage.
- ii) Quality of soil input will be input as integer where as rest can be in the form of floating point.

From the table, we can noticed which algorithm scored high in accuracy and took less time in training and testing the data set.

We chose Naïve Bayes algorithm due to the fact, it has high test accuracy as well as it taking really less time on both training and testing set.


## 5 Flowchart



## 6 Result

We were able to achieve 98.42% accuracy on 2200 test instances after training with 20631 training data instances using Naive Bayes Theorem.



 FARM-EASY

AboutData SetModel BuildingContactUs

# FARM-EASY

Used to Recommend optimum crops to be cultivated by farmers based on several parameters and help them make an informed decision before cultivation.

## FARM-EASY

NITROGEN\_RATIO:

PHOSPHORUS\_RATIO:

POTASSIUM\_RATIO:

TEMPERATURE\_CELCIUS:

HUMIDITY:

PH:

RAINFALL\_MM:

Submit

## FARM-EASY

NITROGEN\_RATIO:

PHOSPHORUS\_RATIO:

POTASSIUM\_RATIO:

TEMPERATURE\_CELCIUS:

HUMIDITY:

PH:

RAINFALL\_MM:

Submit

## **7 Advantages & Disadvantages**

### **7.1 Advantages**

- i) The model had high accuracy on test data set, so it will perform well on unseen datasets.
- ii) The project has an attractive interface for manual prediction which make it easy to use for non-technical people.
- iii) Code modification can be done easy from jupyter notebook if any changes has to be made directly.

### **7.2 Disadvantages**

- iii) The naïve bayes had high accuracy on both training and test dataset. Though the performance of the model is beyond outstanding there could be a case that the dataset was too simple and wasn't drawn from wide area of range.

## **8 Applications**

This project can used for real time manual prediction from crop with proper data input and can be a reference for any hardware maintenance issue research purposes.

## **9 Conclusion**

We would like to humbly concluded that we made an complete web application regarding predicting maintenance issue on Crop recommendation .

## **10 Future scope**

For future, we could make the User Interface(UI) more interactive including dashboards and details happening in behind the scene.

## **11 Bibliography**

- [1] <https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset>
- [2] <https://www.kaggle.com/code/atharvaingle/what-crop-to-grow>
- [3] <https://flask.palletsprojects.com/en/2.1.x/>

## **12 Screenshots of project**

```
from flask import Flask, render_template, request

app = Flask(__name__) #interface between webserver and web application

import pickle
#model1 = pickle.load(open("knnmodel.pkl", "rb"))
#model2 = pickle.load(open("logisticregressionmodel.pkl", "rb"))
model3 = pickle.load(open("naivebayesmodel.pkl", "rb"))
#model4 = pickle.load(open("svmmodel.pkl", "rb"))
#model5 = pickle.load(open("decisiontreemodel.pkl", "rb"))

#We can use any of the model above , here i am using naive bayes theorem for the prediction

@app.route("/") #URL binding

def hello():
    return render_template("index.html")

@app.route("/login", methods = ["POST"]) #URL binding

def user():
    p = request.form['nitro']
    q = request.form['phos']
    r = request.form['potash']
    s = request.form['temp']
```

## Importing the libraries

```
In [1]: 1 import numpy as np
        2 import pandas as pd
```

## Reaading the dataset

```
In [2]: 1 data = pd.read_csv(r"C:\Users\AMAN JHA\Downloads\Crop_recommendation.csv")
```

```
In [3]: 1 data.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

```
<header class="top">
  <div class="logoadjuster">
    <a href=""> </a>
    <span>FARM-EASY</span>
  </div>

  <nav>
    <ul>
      <li> <a href="#aboutpage">About</a> </li>
      <li> <a href="/templates/dataSet.html">Data Set</a> </li>
      <li> <a href="/templates/model.html">Model Building</a> </li>
    </ul>
  </nav>
  <span id="contact"> <a href="#contactus">ContactUs</a> </span>
</header>

<div id="outerContainer">

  <div id="data1PIC">

    <!--  -->
```

```
<header class="top">
  <div class="logoadjuster">
    <a href="/templates/index.html"> </a>
    <span>FARM-EASY</span>
  </div>

  <nav>
    <ul>
      <li> <a href="index.html">Home</a> </li>
      <li> <a href="/templates/dataSet.html">Data Set</a> </li>
      <li> <a href="/templates/model.html">Model Building</a> </li>
    </ul>
  </nav>
</header>

<div id="outerContainer">

  <div id="data1PIC">

    <!--  -->

    <div class="adjuster">

      <h1>FARM-EASY</h1>
```