

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Engineering in Computer Science and Engineering

by

GOTTAPU VIJAY KUMAR
(Reg.No : 38110171)

GINKA GOPI AJAY
(Reg.No: 38110162)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **GOTTAPU VIJAY KUMAR (Reg.No.38110171)** **GINKA GOPI AJAY (Reg.No : 38110162)** who underwent the professional training in “**CROP YIELD PREDICTION USING MACHINE LEARNING** ” under my supervision from JUNE 2021 to NOV 2021.

Internal Guide

Dr.N.SRINIVASAN, M.E.,Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Dr. LAKSHMANAN L, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **GOTTAPU VIJAY KUMAR (RegNo:38110171), GINKA GOPI AJAY (Reg.No : 38110162)** hereby declare that the Professional Training Report “**CROP YIELD PREDICTION USING MACHINE LEARNING**” done by me under the guidance of **Dr.N.SRINIVASAN .M.E.,Ph.D** Department of Computer Science and Engineering at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE :

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.SASIKALA, M.E.,Ph.D., Dean,School of Computing** and **Dr.S.VIGNESHWARI,M.E.,Ph.D.,and Dr.L.LAKSHMANAN,M.E.,Ph.D.,Heads of the Department, Dept. of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide, **Dr.N.SRINIVASAN M.E., Ph.D** for his valuable suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the completion of the project.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	ABSTRACT	7
1	INTRODUCTION	
	1.1 AIM OF THE PROJECT	9
	1.2 PROBLEM STATEMENT	9
	1.3 OBJECTIVES	10
	1.4 SCOPE AND LIMITATIONS	11
2	LITERATURE SURVEY	
	2.1 INTRODUCTION	12
	2.2 LITERATURE REVIEW	13
3	TECHNOLOGY OVERVIEW	
	3.1 MACHINE LEARNING	15
	3.2 CLASSIFICATION	15
	3.3 PROPOSED ARCHITECTURE	16
	3.4 PROJECT FLOW CHART	17
	3.5 DATAFLOW DIAGRAM	18
4	DESIGN AND IMPLEMENTATION	
	4.1 DATA COLLECTION AND PREPROCESSING	19
	4.2 CLASSIFIERS USED FOR EXPERIMENTS	20
	4.3 ALGORITHMS USED	21
	4.4 TOOLS AND SYSTEM REQUIREMENTS	24
5	SUMMARY AND CONCLUSION	26
	REFERENCES	27
	APPENDIX	
	A. SOURCE CODE	28
	B. SCREENSHOTS	33

Abstract.

In India, most agricultural products have been adversely affected by the effects of global climate change. In terms of output during the last two decades. It will enable policymakers and farmers to employ appropriate marketing and storage measures in order to estimate agricultural yields sooner in the harvest. This initiative will enable farmers to capture the yield of their crops prior to cultivation in the realm of agriculture, allowing them to make informed decisions.

The implementation of such a method using user-friendly web-based visual software and the machine learning algorithm may subsequently be spread. The farmer is allowed access to the results. Nonetheless, there are several approaches or procedures for such data analytics in agricultural yield prediction, and we can estimate agricultural yield with the help of all of those algorithms. It employs the Random Forest Algorithm. There are no suitable remedies or ideas to the predicament we confront after examining such difficulties and challenges as weather, temperature, humidity, rainfall, and humidity.

As there are numerous sorts of expanding economic growth in nations such as India, even in the agriculture sector. Furthermore, the technique is effective for anticipating agricultural yield output.

Keywords: Data Mining, Naive Bayes, Decision Tree, Random forest, Data Visualization.

Chapter - 1

Introduction

Crop production may also be an advanced development that is influenced by soil and condition input factors. Agriculture process factors differ from region to region and producer to producer. Collecting similar information in a bigger setting may likewise be a daunting issue. The Indian Meteoric Department, on the other hand, tabulates the environmental condition information acquired in the Republic of India at each 1sq.m space in the district's many components. Massive amounts of such data are frequently utilised to forecast their impact on the primary crops of that district or location.

There are several foretelling approaches established and assessed by scholars all over the world in the realm of agriculture or allied disciplines. Agricultural researchers in other nations have revealed that attempts to boost crop production by pro-pesticide state maximisation have been carried out. Driven methods for hazardously high chemical usage have been implemented. These studies[1] have found a link between chemical use and agricultural productivity. Agriculture is a partner trade industry that has benefited considerably from the advancement of detection technology, information science, and machine learning (ML) approaches in recent years.

1.1. Aim of the Project :

The goal of our crop yield prediction project is to determine if a crop should be diagnosed with heart disease or not. We have to find which classification model has the greatest accuracy and identify correlations in our data. Finally, we also have to determine which features are the most influential in our crop yield prediction.

1.2. Problem Statement :

Machine Learning based on prior crop prediction, soil quality analysis to achieve high crop yield through out technology solution. The main objectives of this project is to predict crop-yield which can be extremely useful to farmers in planning for harvest and sale of grain harvest.

Implement a machine learning algorithm that gives better prediction of suitable crop for the corresponding region and crop season in our country.

This project aims to predict yields based on location and weather data.

The aim of this study is to look at the prediction of crops which will offer high yield within the given location considering the climate and soil parameters.

the opportunities to improve care and reduce costs concurrently could apply to as much as 30% of overall crop yield spending. The successful application of data mining in highly visible fields like e-business, marketing and retail has led to its application in other industries and sectors. Among these sectors just discovering is healthcare. The healthcare environment is still „information rich“ but „knowledge poor“. There is a wealth of data available within the healthcare systems. However, there is a lack of effective analysis tools to discover hidden relationships and trends in the data for African genres.

1.3. Objectives

Main Objectives :

Crop yield prediction is **an essential task for the decision-makers at national and regional levels (e.g., the EU level) for rapid decision-making.** An accurate crop yield prediction model can help farmers to decide on what to grow and when to grow. There are different approaches to crop yield prediction.

Specific Objectives :

- Provides new approach to concealed patterns in the data.
- To implement Naïve Bayes Classifier that classifies the disease as per the input of the user.
- Reduce the cost of farmers spendings on crop.

Justification :

Clinical decisions are often made based on doctor's insight and experience rather than on the knowledge rich data hidden in the dataset. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. The proposed system will integrate clinical decision support with computer-based patient records (Data Sets). This will reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as data modeling and analysis tools, e.g., data mining, have the potential to generate a knowledge rich environment which can help to significantly improve the quality of clinical decisions.

There are voluminous records in medical data domain and because of this, it has become necessary to use data mining techniques to help in decision support and prediction in the field of healthcare. Therefore, medical data mining contributes to business intelligence which is useful for diagnosing of disease

1.4 Scope and Limitations

Scope :

Here the scope of the project is that integration of clinical decision support with computer-based crop records could reduce medical errors, enhance crop safety, decrease unwanted practice variation, and improve crop outcome. This suggestion is promising as data modeling and analysis tools, e.g., data mining, have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions

Limitations :

Medical diagnosis is considered as a significant yet intricate task that needs to be carried out precisely and efficiently. The automation of the same would be highly beneficial. Clinical decisions are often made based on proper intuition and experience rather than on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. Data mining have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction :

Nature may be a component of every living organism's life. Farming is the best way to explain nature. Historically, farming demonstrated the soundness of a country's economic structure. Farming is the backbone of any nation's economy. Advancement in the agricultural sector is essential to solve challenges in countries like as India, which has an ever-increasing demand for food as a result of its growing population. Indeed, horticulture has long been regarded as the foundational and hence primary culture practised in India when compared to other nations.

in the United States. Coronary heart disease, Cardiomyopathy crop yield are some categories of heart diseases. The term “cardiovascular disease” includes a wide range of conditions that affect the heart and the blood vessels and the manner in which blood is pumped and circulated through the body. Diagnosis is complicated and important task that needs to be executed accurately and efficiently. The diagnosis is often made, based on doctor’s experience & knowledge. This leads to unwanted results & excessive medical costs of treatments provided to patients. Therefore, an automatic medical diagnosis system would be exceedingly beneficial.

2.2 Literature Review :

Our solution automatically obtains the required meteorological and nutritional status data for a specified link from either a suitable alternative. Another advantage is that their technique has been tested in vast regions and provides a resolution forecast commensurate with a system file's best resolution from the event's soil data. Prior to the beginning of the agricultural season, the accuracy of crop forecasting

The machine learning techniques are used to forecast agricultural yield. The goal of this study is to estimate agricultural yield using the Random Forest algorithm based on existing data. Real data from Tamil Nadu was utilised to build the models, and the models were tested with samples. The Random Forest Algorithm may be used to estimate crop production reliably.

The Agricultural Marketing Gain Evaluation Study is carried out using an airborne system and Smart Farming. Symbiosis International 5th & 6th Floor University, Artur Centre, Gokhale Cross Road, Model Colony, Pune-411016 Symbiosis International 5th & 6th Floor University, Artur Centre, Gokhale Cross Road, Model Colony, Pune-411016 Symbiosis International 5th & 6th Floor University, Artur Centre, Gokhale Cross Road, Model Colony, Pune-411016 Precision agriculture use geographic information system technologies and remote sensors to detect differences in the field and address them using various ways (PA).

Crop stress, irrigation techniques, the presence of pests and diseases, and other factors can all contribute to variability in crop development in an agricultural area. Ensemble Learning is used in this research (EL).

In Global and Regional Crop Yield Prediction Using Decision Trees. Minnesota University's Institute for the Climate, St. Paul, Minnesota, MN 55108. Because of its excellent accuracy and performance, the results suggest that RF is a separate machine-learning approach for regional and worldwide crop production projections. The Journal is constructed using two layers of regression k-nearest neighbour (SVG).

While there are some traditional review papers on crop yield prediction and some SLR papers that do not focus on the application of deep learning in crop yield prediction (e.g. traditional machine learning in crop yield prediction. Here, we must distinguish shallow learning from deep learning), there is no SLR paper that focuses on the use of deep learning in crop yield prediction yet. In this respect, a pioneering effort has been made in the present study representing the way for systematically reviewing the state-of-the-art knowledge on the development of Deep Learning-based methods for crop yield prediction.

Chapter – 3

Technology Overview

3.1. Machine Learning :

Machine learning uses algorithms to identify patterns within data and those patterns are then used to create a data model that can make predictions. With increased data and experience, the results of machine learning are more accurate—much like how humans improve with more practice.

The adaptability of machine learning makes it a great choice in scenarios where the data is always changing, the nature of the request or task are always shifting, or coding a solution would be effectively impossible.

3.2. Classification :

Classification is a process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points.

The classes are often referred to as target, label or categories.

Crop yield prediction can be identified as a classification problem, this is a binary classification since there can be only two classes i.e. which crop. The classifier, in this case, needs training data to understand how the given input variables are related to the class. And once the classifier is trained accurately, it can be used to detect whether crop is there or not for a particular input.

Keywords :

Classifier – It is an algorithm that is used to map the input data to a specific category.

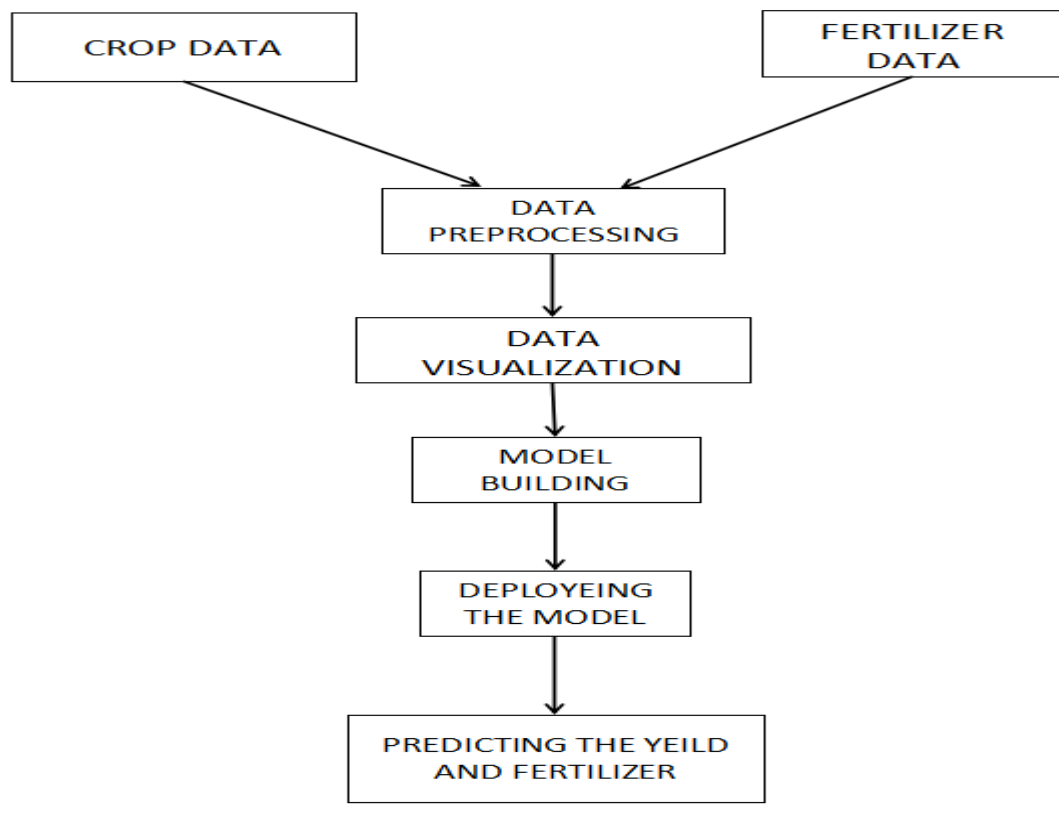
3.3 Proposed Architecture :

In this system we are implementing effective heart attack prediction system using Naïve Bayes algorithm. We can give the input as in CSV file or manual entry to the system. After taking input the algorithm is applied on that input that is Naïve Bayes. After accessing data set the operation is performed and effective heart attack level is produced.

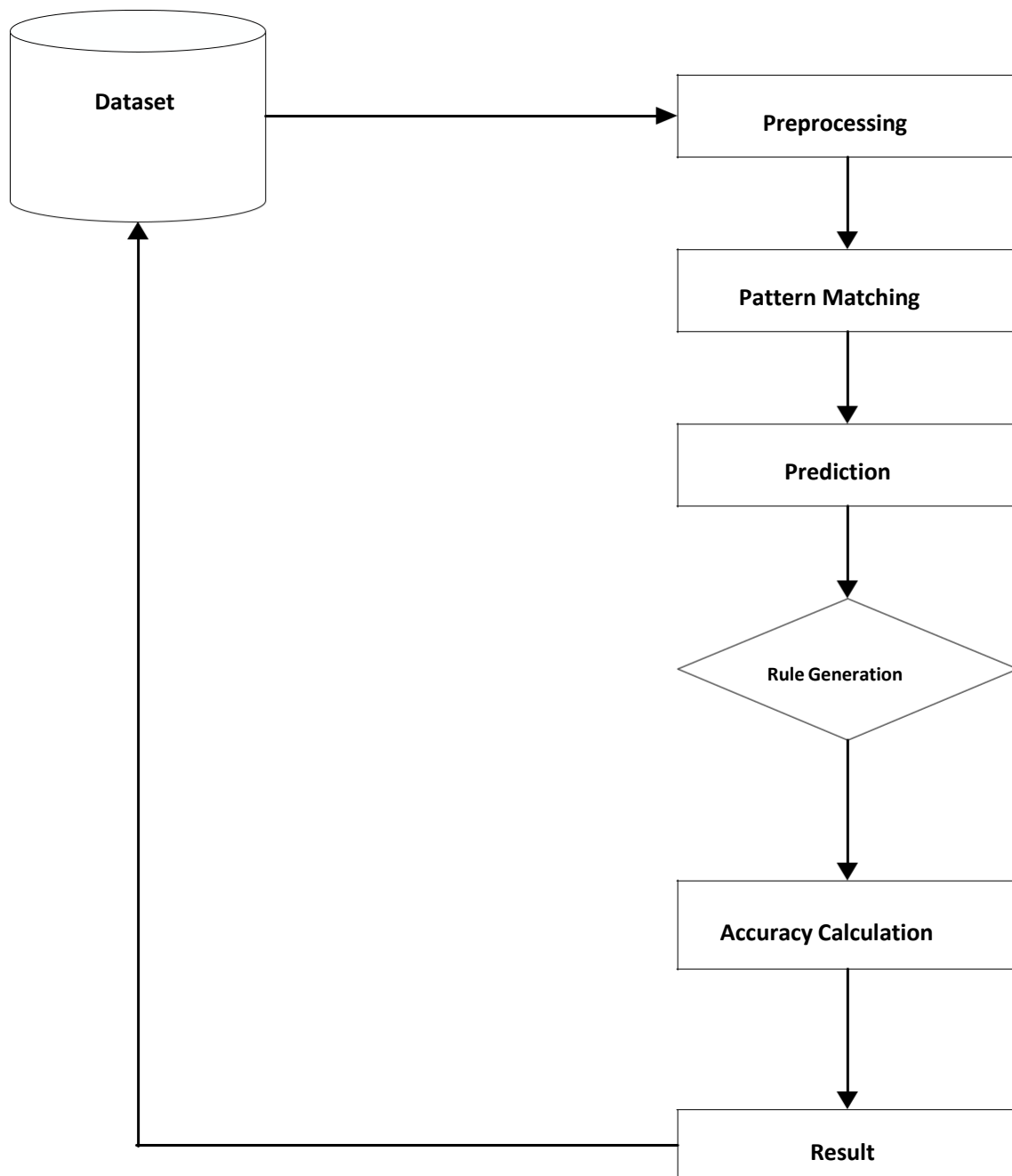
The proposed system will add some more parameters significant to heart attack with their weight, age and the priority levels are by consulting expertise doctors and the medical experts. The heart attack prediction system designed to help the identify different risk levels of heart attack like normal, low or high and also giving the prescription details with related to the predicted result.

3.4 Project Flow Chart :

This will be the proposed flow chart that the system will look like



3.5 Data Flow Diagram :



Chapter – 4

Design and Implementation

4.1 Data Collection and Preprocessing :

The data set for this research was taken from UCI data repository.¹⁴ Data accessed from the UCI Machine Learning Repository is freely available. In particular, the Cleveland and Hungarian databases have been used by many researchers and found to be suitable for developing a mining model, because of lesser missing values and outliers. The data is cleaned and preprocessed before it is submitted to the proposed algorithm for training and testing.

The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms.

The overall objective of our work is to predict more accurately the presence of heart disease. In this paper, UCI repository dataset are used to get more accurate results. Two data mining classification techniques were applied namely Decision trees and Naive Bayes database contains 76 attributes, but all published experiments refer to using a subset of 14 of them.

In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence

(value 0). Attributes with categorical values were converted to numerical values since most machine learning algorithms require integer values

4.2 Classifiers Used for Experiments :

The classifiers used for the experiments are the following

- 1) Nitrogen level
- 2) Phosphorous level
- 3) Potassium level
- 4) Temperature
- 5) Humidity
- 6) Ph value of the soil
- 7) Rainfall in mm of that particular area.

we are undertaking crop forecast at the district level. The main goal is to identify information on climate and soil characteristics such as rainfall, temperature, humidity, soil quality, and so on in a dataset including output details for the previous 12 years. These variables can aid in crop prediction by employing different classifiers on the supplied dataset. As a result, several variables are reviewed, and the ones that significantly contribute to exact crop forecast are evaluated.

4.3 Algorithms Used :

Naïve Bayes Classifier :

Naïve Bayes classifier is based on Bayes theorem. This classifier uses conditional independence in which attribute value is independent of the values of other attributes. The Bayes theorem is as follows :

Let $X = \{x_1, x_2, \dots, x_n\}$ be a set of n attributes. In Bayesian, X is considered as

evidence and H be some hypothesis means, the data of X belongs to specific class

C. We have to determine $P(H|X)$, the probability that the hypothesis H holds given evidence i.e. data sample X . According to Bayes theorem the $P(H|X)$ is expressed as: $P(H|X) = P(X|H) P(H) / P(X)$.⁴

Algorithm Used	Accuracy	Time Taken
Naïve Bayes	86.77%	609 m
Decision Tree	77.04%	719 m
Random forest	77.04%	1000 m

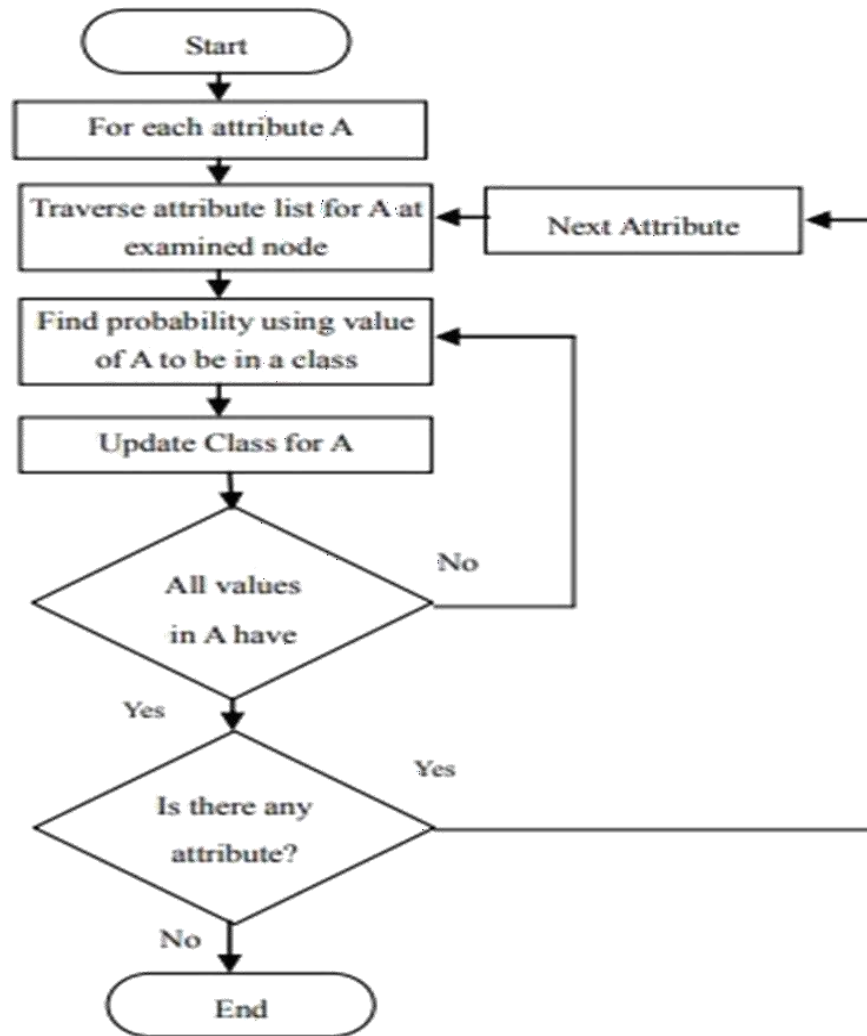
Using Bayesian classifiers, the system will discover the concealed knowledge associated with diseases from historical records of the patients having heart disease. Bayesian classifiers predict the class membership probabilities, in a way that the probability of a given sample belongs to a particular class statistically. Bayesian classifier is based on Bayes' theorem. We can use Bayes theorem to determine the probability that a proposed diagnosis is correct, given the observation. A simple probabilistic, the naive Bayes classifier is used for classification based on which is based on Bayes' theorem. According to naïve Bayesian classifier the occurrence or an occurrence of a particular feature of a class is considered as independent in the

presence or absence of any other feature. When the dimension of the inputs is high and more efficient result is expected, the chief Naïve Bayes Classifier technique is applicable.

The Naïve Bayes model identifies the physical characteristics and features of patients suffering from heart disease. For each input, it gives the possibility of attribute of the expectable state. Naïve Bayes is a statistical classifier which assumes no dependency between attributes. This classifier algorithm uses conditional independence, means it assumes that an attribute value of a given class is independent of the values of other attributes. The advantage of using Naïve Bayes is that one can work with the Naïve Bayes model without using any Bayesian methods. (Brownlee, 2016).

Flowchart Of Naïve Bayes Decision Tree Algorithm :

The classification tree literally creates a tree with branches, nodes, and leaves that lets us take an unknown data point and move down the tree, applying the attributes of the data point to the tree until a leaf is reached and the unknown output of the data point can be determined. In order to create a good classification tree model, we need to have an existing data set with known output from which we can build our model. We also divide our data set into two parts: a training set, which is used to create the model, and a test set, which is used to verify that the model is accurate and not over fitted.



Decision Trees :

The decision tree approach is more powerful for classification problems. There are two steps in this technique building a tree & applying the tree to the dataset. There are many popular decision tree algorithms CART, ID3, C4.5, CHAID, and J48. From these J48 algorithm is used for this system. J48 algorithm uses pruning method to build a tree. Pruning is a technique that reduces size of tree by removing over fitting data, which leads to poor accuracy in predications. The J48

algorithm recursively classifies data until it has been categorized as perfectly as possible. This technique gives maximum accuracy on training data. The overall concept is to build a tree that provides balance of flexibility & accuracy.

Random forest :

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time For regression tasks, the mean or average prediction of the individual trees is returned.

4.4 Tools and System Requirements :

Tools :

For application development, the following Software Requirements are:

Operating System: Windows 7 or any Linux Debian Distro.

Language: Pyython

Tools: RStudio IDE, Microsoft Excel (Optional).Technologies

used: R, Unix, Shiny.

Software requirements :

Operating System the internet	Any OS with clients to access
Network	Wi-Fi Internet or cellular Network
Visio Studio	Create and design Data Flow andContext Diagram
Github	Versioning Control
Google Chrome	Medium to find reference to dosystem testing, display and run shinyApp.

Hardware Requirements :

For application development, the following Software Requirements are:Processor: Intel or high
RAM: 1024 MB
Space on disk: minimum 100mb

For running the application :

Device: Any device that can access the
internetMinimum space to execute: 20 MB

The effectiveness of the proposal is evaluated by conducting experiments with acluster formed by 3 nodes with identical setting, configured with an Intel CORE™ i7-4770 processor (3.40GHZ, 4 Cores, 8GB RAM, running Ubuntu 18.04 LTS with 64-bit Linux 4.31.0 kernel)

Chapter – 5

Result and Conclusion

The current study demonstrated how data mining techniques may be used to estimate crop output based on climatic input factors. The website has been designed to be user-friendly, and the forecast reliability in all of the other gntins and locations picked for the analysis should be more than 75%, suggesting better predictive performance.

The net page was created to forecast agricultural yield using data from the tlml region

Accuracy for test set for Naive Bayes = 99.09%

Accuracy for test set for Random Forest = 82%

Accuracy for test set for Decision Tree = 99.09%

Accuracy for test set for SVM = 97.95%

Accuracy for test set for Logistic Regression = 95.22%

Finally, we created a website that accepts four input parameters: District Name, Crop Name, Area in Acres, and Soil Type. As a result, when we provide the inputs, we will receive the YIELD forecast.

For example when we give the followings as input then the Yield is predicted.

District Name:- AMRAVATI

Crop :- Jowar

Area(in acres) :- 2

Soil :- clay

The YIELD predicted is:-15.2 tons

REFERENCES :

- [1] . S.Veenadhari, Dr Bharat Misra, Dr CD Singh.2019."Machine learning approach for forecasting crop yield based on climatic parameters.". International Conference on Computer Communication and Informatics (ICCCI), 2014.
- [2] . P.Priya, U.Muthaiah M.Balamurugan.Predicting yield of the crop using machine learning algorithm. International Journal of Engineering Science Research Technology(IJESRT),2018.
- [3] . D Ramesh ,B Vishnu Vardhan. Analysis Of Crop Yield Prediction Using Data Mining Techniques. IJRET: International Journal of Research in Engineering and Technology.(IJERT),2015
- .
- [4] . F. Baheri, F. Davardoost and V. Ahmadzadeh, "Data mining with learning decision tree and Bayesian network for data replication in Data Grid", Proceedings of the Third International Conference on Contemporary Issues in Computer and Information Sciences, pp. 51-51, 2012..
- [5] . Abdullah, A., Brobst, S, Pervaiz.I., Umer M.,and A.Nisar.2004. Learning dynamics of pesticide abuse through data mining. Proceedings of Australian Workshop on Data mining and Web Intelligence, New Zealand, January,2004.

Appendix

Source Code :

```
import pandas as pd

# import warnings filter
from warnings import simplefilter

# ignore all future warnings
simplefilter(action='ignore', category = FutureWarning)

# import warnings filter
from warnings import simplefilter

# ignore all future warnings
simplefilter(action='ignore', category = FutureWarning)

df=pd.read_csv("D:\DATASCIENCE\FINALYEARPROJECT
\CROPANDFERTILIZER\Dataprocessed\crop_recomme
ndation.csv")
df.head()
df.tail()
df.size
df.shape
df.columns
df['label'].unique()
df.dtypes
df['label'].value_counts()
df.describe()
sns.pairplot(df)
```

```
sns.heatmap(df.corr(),annot=True)
```

```
g = sns.countplot(x='label', data=df)
plt.xlabel('label')
plt.ylabel('Count')
```

```
fig = plt.figure(figsize = (15,15))
ax = fig.gca()
g = df.hist(ax=ax)
```

Seperating features and target label

```
features = df[['N', 'P','K','temperature', 'humidity', 'ph',
'rainfall']]
target = df['label']
#features = df[['temperature', 'humidity', 'ph',
'rainfall']]
labels = df['label']
```

Initializing empty lists to append all model's name
and corresponding name

```
acc = []
model = []
```

Splitting into train and test data

```
from sklearn.model_selection import train_test_split
Xtrain, Xtest, Ytrain, Ytest =
train_test_split(features,target,test_size
0.2,random_state=2)
```

DECISION TREE

```
from sklearn.tree import DecisionTreeClassifier
```

```
DecisionTree =
```

```
DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)
```

```
DecisionTree.fit(Xtrain,Ytrain)
```

```
predicted_values = DecisionTree.predict(Xtest)
```

```
x = metrics.accuracy_score(Ytest, predicted_values)
```

```
acc.append(x)
```

```
model.append('Decision Tree')
```

```
print("DecisionTrees's Accuracy is: ", x*100)
```

```
print(classification_report(Ytest,predicted_values))
```

```
from sklearn.model_selection import cross_val_score
```

```
# Cross validation score (Decision Tree)
```

```
score = cross_val_score(DecisionTree, features,  
target,cv=5)
```

Score

NAIVE BAYES

```
from sklearn.naive_bayes import GaussianNB
```

```
NaiveBayes = GaussianNB
```

```
NaiveBayes.fit(Xtrain,Ytrain)
```

```
predicted_values = NaiveBayes.predict(Xtest)
```

```
x = metrics.accuracy_score(Ytest, predicted_values)
```

```
acc.append(x)
```

```
model.append('Naive Bayes')
```

```
print("Naive Bayes's Accuracy is: ", x)
```

```
print(classification_report(Ytest,predicted_values))
```

```
# Cross validation score (NaiveBayes)
```

```
score =
```

```
cross_val_score(NaiveBayes,features,target,cv=5)
```

```
Score
```

SUPPORTVECTOR MACHINE(SVM)

```
from sklearn.svm import SVC
```

```
# data normalization with sklearn
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
# fit scaler on training data
```

```
norm = MinMaxScaler().fit(Xtrain)
```

```
X_train_norm = norm.transform(Xtrain)
```

```
# transform testing data
```

```
X_test_norm = norm.transform(Xtest)
```

```
SVM = SVC(kernel='poly', degree=3, C=1)
```

```
SVM.fit(X_train_norm,Ytrain)
```

```
predicted_values = SVM.predict(X_test_norm)
```

```
x = metrics.accuracy_score(Ytest, predicted_values)
```

```
acc.append(x)
```

```

model.append('SVM')
print("SVM's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
# Cross validation score (SVM)
score = cross_val_score(SVM,features,target,cv=5)
Score

```

LOGISTIC REGRESSION

```

from sklearn.linear_model import LogisticRegression
LogReg = LogisticRegression(random_state=2)
LogReg.fit(Xtrain,Ytrain)
predicted_values = LogReg.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Logistic Regression')
print("Logistic Regression's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))

# Cross validation score (Logistic Regression)
score = cross_val_score(LogReg,features,target,cv=5)
Score

```

RANDOM FOREST

```

from sklearn.ensemble import RandomForestClassifier

```

```

RF = RandomForestClassifier(n_estimators=20,
random_state=0)
RF.fit(Xtrain,Ytrain)

predicted_values = RF.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values)

# Cross validation score (Random Forest)
score = cross_val_score(RF,features,target,cv=5)
Score

```

ACCURACY COMPARISION

```

plt.figure(figsize=[10,5],dpi = 100)
plt.title('Accuracy Comparison')
plt.xlabel('Accuracy')
plt.ylabel('Algorithm')
sns.barplot(x = acc,y = model,palette='dark')
accuracy_models = dict(zip(model, acc))
for k, v in accuracy_models.items():
    print (k, '-->', v)
MAKING PRIDECTION

```

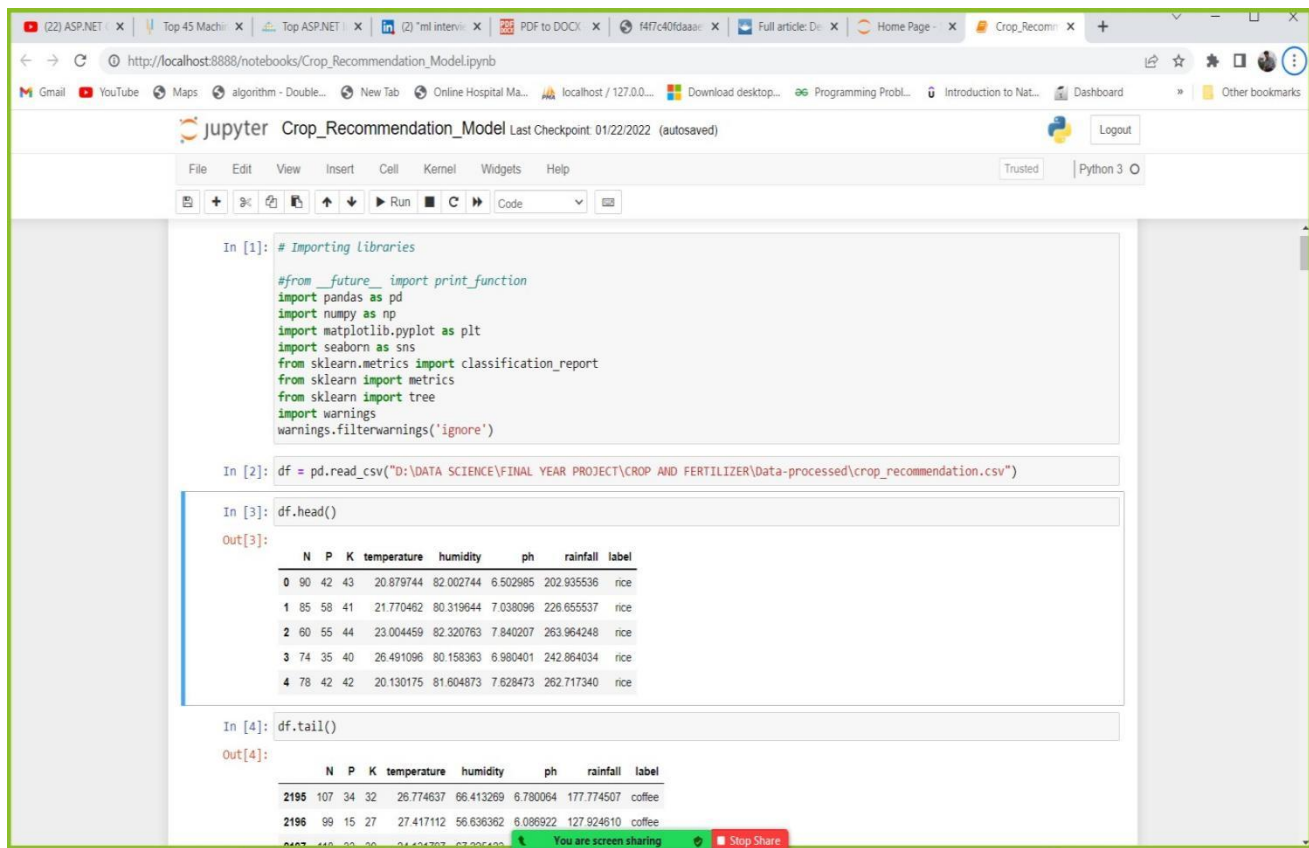
```
data = np.array([[104,18, 30, 23.603016, 60.3, 6.7,  
140.91]])  
prediction = RF.predict(data)  
print(prediction)
```

OUTPUT: COFFEE

```
data = np.array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])  
prediction = RF.predict(data)  
print(prediction)
```

OUTPUT: JUT

SCREENSHOTS :



```
In [1]: # Importing libraries

#from __future__ import print_function
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import tree
import warnings
warnings.filterwarnings('ignore')

In [2]: df = pd.read_csv("D:\DATA SCIENCE\FINAL YEAR PROJECT\CROP AND FERTILIZER\data-processed\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.tail()

Out[4]:
```

	N	P	K	temperature	humidity	ph	rainfall	label
2195	107	34	32	26.774637	66.413289	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	110	30	30	24.424707	67.806420	6.780064	177.774507	coffee

Qeqq

Top ASP.NET | Top 45 Ma... | Top ASP.NET | (2) 'ml inte... | PDF to DO... | 1477c40f... | Full article... | Home Page... | Crop_Reco... | New Tab | +

http://localhost:8888/notebooks/Crop_Recommendation_Model.ipynb

Gmail | YouTube | Maps | algorithm - Double... | New Tab | Online Hospital Ma... | localhost / 127.0.0... | Download desktop... | Programming Probl... | Introduction to Nat... | Dashboard | Other bookmarks

jupyter Crop_Recommendation_Model Last Checkpoint: 01/22/2022 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [1]: # Importing libraries

#from __future__ import print_function
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import tree
import warnings
warnings.filterwarnings('ignore')

In [2]: df = pd.read_csv("D:\DATA SCIENCE\FINAL YEAR PROJECT\CROP AND FERTILIZER\Data-processed\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	68	41	21.770452	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.tail()

Out[4]:
```

	N	P	K	temperature	humidity	ph	rainfall	label
2195	107	34	32	26.774637	86.413269	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

Top ASP.NET | Top 45 Ma... | Top ASP.NET | (2) 'ml inte... | PDF to DO... | 1477c40f... | Full article... | Home Page... | Crop_Reco... | New Tab | +

http://localhost:8888/notebooks/Crop_Recommendation_Model.ipynb

Gmail | YouTube | Maps | algorithm - Double... | New Tab | Online Hospital Ma... | localhost / 127.0.0... | Download desktop... | Programming Probl... | Introduction to Nat... | Dashboard | Other bookmarks

jupyter Crop_Recommendation_Model Last Checkpoint: 01/22/2022 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
2199 104 18 30 23.603016 60.396475 6.779833 140.937041 coffee

In [5]: df.size
Out[5]: 17600

In [6]: df.shape
Out[6]: (2200, 8)

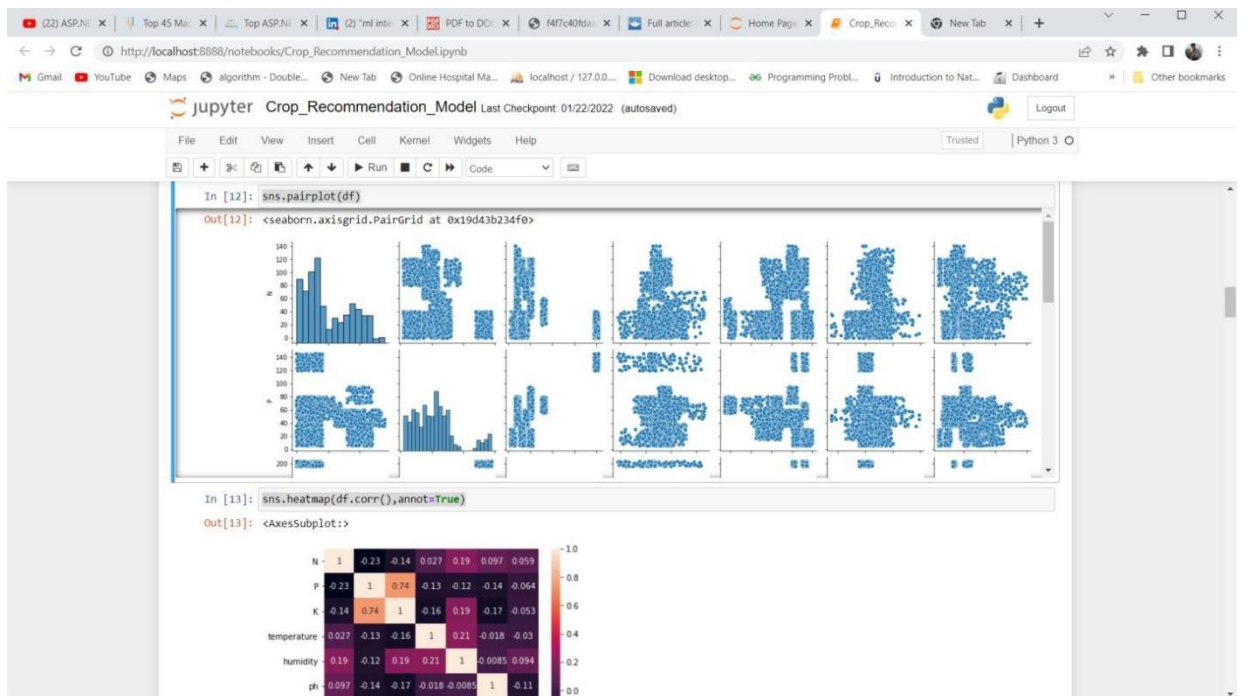
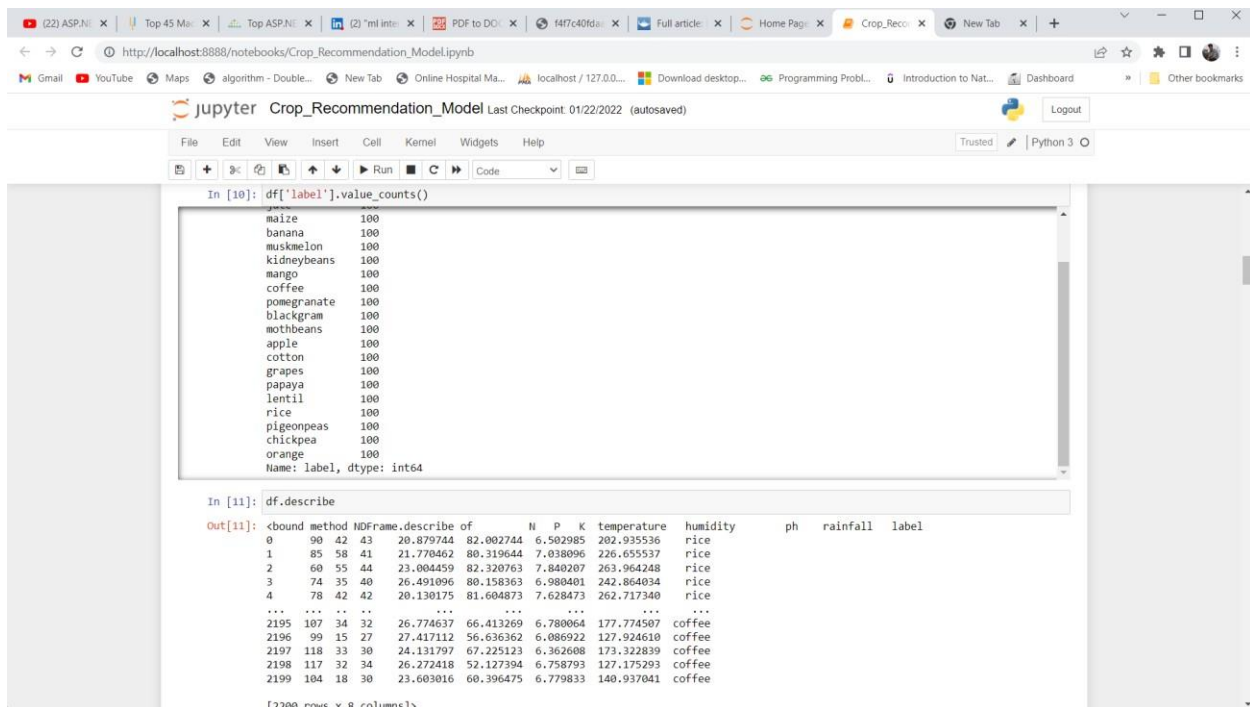
In [7]: df.columns
Out[7]: Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label'], dtype='object')

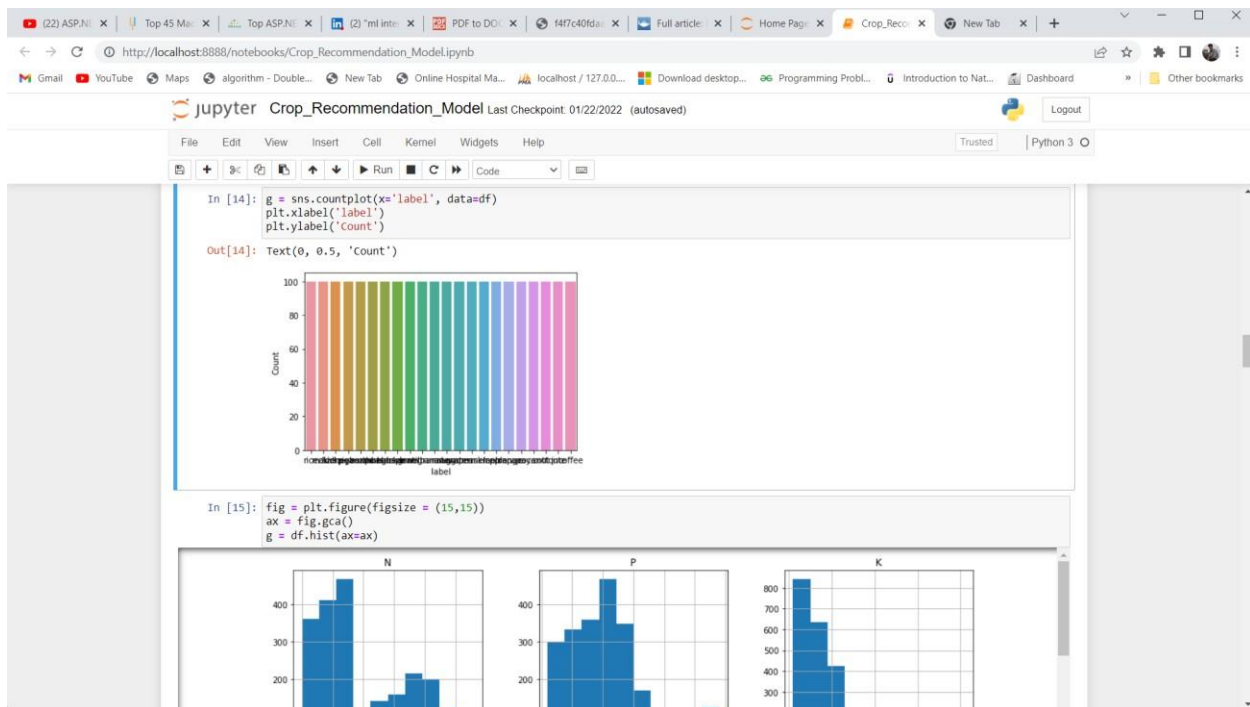
In [8]: df['label'].unique()
Out[8]: array(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',
        'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',
        'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',
        'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],
        dtype=object)

In [9]: df.dtypes
Out[9]:
```

N	int64
P	int64
K	int64
temperature	float64
humidity	float64
ph	float64
rainfall	float64
label	object
	dtype: object

```
In [10]: df['label'].value_counts()
```





```
### Separating features and target label

In [16]: features = df[['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall']]
target = df['label']
#features = df[['temperature', 'humidity', 'ph', 'rainfall']]
labels = df['label']

In [17]: # Initializing empty lists to append all model's name and corresponding name
acc = []
model = []

In [18]: # Splitting into train and test data

from sklearn.model_selection import train_test_split
Xtrain, Xtest, ytrain, ytest = train_test_split(features, target, test_size = 0.2, random_state = 2)

Decision Tree

In [19]: from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy", random_state=2, max_depth=5)

DecisionTree.fit(Xtrain, ytrain)

predicted_values = DecisionTree.predict(Xtest)
x = metrics.accuracy_score(ytest, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)

print(classification_report(ytest, predicted_values))

DecisionTrees's Accuracy is: 90.0
precision    recall  f1-score   support
```

Jupyter Notebook interface showing the execution of a Gaussian Naive Bayes model. The notebook is titled "Crop_Recommendation_Model" and is running on a local host at 127.0.0.1.

```

In [20]: from sklearn.model_selection import cross_val_score

In [21]: # Cross validation score (Decision Tree)
score = cross_val_score(DecisionTree, features, target, cv=5)

In [22]: score
Out[22]: array([0.93636364, 0.90909091, 0.91818182, 0.87045455, 0.93636364])

```

Guassian Naive Bayes

```

In [23]: from sklearn.naive_bayes import GaussianNB

NaiveBayes = GaussianNB()
NaiveBayes.fit(Xtrain, Ytrain)

predicted_values = NaiveBayes.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Naive Bayes')
print("Naive Bayes's Accuracy is: ", x)

print(classification_report(Ytest, predicted_values))

```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	1.00	1.00	1.00	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22

Jupyter Notebook interface showing the execution of a Support Vector Machine (SVM) model. The notebook is titled "Crop_Recommendation_Model" and is running on a local host at 127.0.0.1.

```

In [24]: # Cross validation score (NaiveBayes)
score = cross_val_score(NaiveBayes, features, target, cv=5)
score
Out[24]: array([0.99772727, 0.99545455, 0.99545455, 0.99545455, 0.99090909])

```

Support Vector Machine (SVM)

```

In [25]: from sklearn.svm import SVC
# data normalization with sklearn
from sklearn.preprocessing import MinMaxScaler
# fit scaler on training data
norm = MinMaxScaler().fit(Xtrain)
X_train_norm = norm.transform(Xtrain)
# transform testing data
X_test_norm = norm.transform(Xtest)
SVM = SVC(kernel='poly', degree=3, C=1)
SVM.fit(X_train_norm, Ytrain)
predicted_values = SVM.predict(X_test_norm)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('SVM')
print("SVM's Accuracy is: ", x)

print(classification_report(Ytest, predicted_values))

```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	1.00	1.00	1.00	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	0.95	0.98	22
cotton	0.95	1.00	0.98	20

Jupyter interface showing the SVM model training and evaluation results.

```

In [26]: # Cross validation score (SVM)
score = cross_val_score(SVM, features, target, cv=5)
score

Out[26]: array([0.97954545, 0.975      , 0.98863636, 0.98863636, 0.98181818])

In [27]: #Saving trained SVM model

```

Logistic Regression

```

In [28]: from sklearn.linear_model import LogisticRegression
LogReg = LogisticRegression(random_state=2)
LogReg.fit(Xtrain, Ytrain)
predicted_values = LogReg.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Logistic Regression')
print("Logistic Regression's Accuracy is: ", x)
print(classification_report(Ytest, predicted_values))

```

Logistic Regression's Accuracy is: 0.9522727272727273

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.86	0.75	0.80	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	0.86	0.90	0.88	20

Jupyter interface showing the Logistic Regression model training and evaluation results.

```

In [29]: # Cross validation score (Logistic Regression)
score = cross_val_score(LogReg, features, target, cv=5)
score

Out[29]: array([0.95      , 0.96590909, 0.94772727, 0.96590909, 0.94318182])

```

Random Forest

```

In [30]: from sklearn.ensemble import RandomForestClassifier
RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain, Ytrain)
predicted_values = RF.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)
print(classification_report(Ytest, predicted_values))

```

RF's Accuracy is: 0.9909090909090909

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.94	1.00	0.97	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.90	1.00	0.95	28
kidneybeans	1.00	1.00	1.00	14
lentil	1.00	1.00	1.00	23

