

Agriculture Analysis Using Data Mining And Machine Learning Techniques

Submitted to the department of Computer Science

By

MCA IV Semester Student

B.VINAY REDDY [Reg.no :2251926004]

Under the Supervision of

DR.G.RAMA KRISHNA , P.hD
Asst. Professor.



**DEPARTMENT OF COMPUTER SCIENCE &
ENGINEERING**

COLLEGE OF ENGINEERING

Dr.B.R.AMBEDKAR UNIVERSITY,SRIKAKULAM

ETCHERLA-532410

2022-2024

**DEPARTMENT OF COMPUTER SCIENCE AND
COLLEGE OF ENGINEERING
DR.B.R.AMBEDKAR UNIVERSITY,SRIKAKULAM**



CERTIFICATE

This is to certify that the project entitled, “**Agriculture Analysis Using Data Mining And Machine Learning Techniques**” is a bonafide work of B.Vinay Reddy (2251926004) MCA IV semester bearing , submitted to the Department of Computer Science,Dr B.R Ambedkar University, Srikakulam for the academic year 2022-2024.

SUPERVISOR

Head of the Department

Dr.G.Rama Krishna , PhD.
Asst.professor

Sri.R.Sridhar,M.Tech,PhD.
Department of Computer Science
Dr.B.R.Ambedkar university
Srikakulam. Etcherla-532410

External Examiner

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING

DR.B.R.AMBEDKAR UNIVERSITY ,SRIKAKULAM



DECLARATION

I hereby declare that the project entitled “**Agriculture Analysis Using Data Mining And Machine Learning Techniques**” which is submitted by me in partial fulfilment of the award of degree of MASTER OF COMPUTER APPLICATIONS, College of Engineering, **Dr.B.R.Ambedkar University Srikakulam** under the guidance of **Dr.G.RamaKrishna, P.hD** Asst Professor, Dr.B.R.Ambedkar university, srikakulam. This project is original has not been submitted for any degree of any other University.

Srikakulam
Date:

Signature of Candidate
B.Vinay Reddy
Regd.no(2251926004)

ACKNOWLEDGEMENT

It gives me an immense pleasure to express my sincere thanks to my research supervisor **Dr. G.RamaKrishna, P.hD** for his continuous support from the inception of this work. But for his involvement my dream of completion of the project work would not have materialized and I feel highly indebted to his for encouraging me when I am down in spirit feeling the burden of research work at times.

I express my sincere thanks to Head of the Department Assistant Professor **Sri.R.sridhar,Mtech,P.hD** and all other faculty members for extending their support all through my work. I feel grateful to each and everyone in the teaching Nonteaching and lab staff for their timely help whenever I needed.

I express my sincere thanks to principal **Prof.ch.Rajashekhar rao,P.hD** for her great support all through my work. I take this opportunity to extend my sincere thanks to Assistant Principal, **Sri.P.Ramakrishna,P.hD** college of Engineering, Dr.B.R.Ambedkar University and for there support during the submission of this project work.

I extend my sincere thanks to teaching faculty of the Department of Computer science and Engineering for their direct or indirect support for helping us in completion of this project work.

Finally, we would like to thank all of our friend and family members for their continuous help and encouragement.

ABSTRACT

Agriculture is an important application in India. The modern technologies can change the situation of farmers and decision making in agricultural field in a better way. Python is used as a front end for analysing the agricultural data set. Jupyter Notebook is the data mining tool used to predict the crop production. Agriculture is the major source of the Indian Economy. Day by day, the population increases. So the demand of food increases. To get rid of these situations farmers, agricultural scientists, and researchers are trying for better crop yield.

LIST OF FIGURES

Figure No.	Contents	Page No.
Figure 1.2.1	K-NN Representation	14
Figure 1.2.2	SVM Representation	16
Figure 3.1	Proposed Framework	32
Figure 3.2	Architecture For Proposed System	36
Figure 3.3	Proposed Architecture using Jupyter	37
Figure 4.1	Windows Executable	47
Figure 4.2	Python Folder under C	48
Figure 4.3	Python Execution File	48
Figure 4.4	Python installation wizard	49
Figure 4.5	Choose the location	49
Figure 4.6	Python installation Successfully	50
Figure 4.7	Jupyter Notebook installation Started	50
Figure 4.8	Opening Jupyter Notebook	51
Figure 4.9	Jupyter Notebook On Browser	51
Figure 4.10	Open Notebook	52

Figure 4.11	Hello World in jupyter Notebook	52
Figure 4.12	Bar Plot	57
Figure 4.13	Heat map	59
Figure 4.14	Bar plot of each parameter	60
Figure 4.15	K-Nearest Neighbours error rate	63
Figure 4.16	Accuracy for Logistic Regression	64
Figure 4.17	Accuracy for Decision Tree	66
Figure 4.18	Accuracy For Naïve Bayes	69
Figure 4.19	Accuracy for Support Vector Machine	71
Figure 4.20	Accuracy for Random Forest	74
Figure 4.21	Confusion Matrix For K-Nearest Neighbours	77
Figure 4.22	Confusion Matrix For Decision Tree	78
Figure 4.23	Confusion Matrix For Random Forest	79

LIST OF TABLES

Table No.	Contents	Page No.
Table 2.1	Survey on Agriculture data	28-29
Table 4.1	Description of Data set	41
Table 4.2	Descriptive of the table	54
Table 4.3	Data set of first five rows	54
Table 4.4	Data set of last five rows.	55
Table 4.5	Size of the dataset	56
Table 4.6	Shape of the dataset	56
Table 4.7	Parameters of the dataset	57
Table 4.8	Unique Parameters	57
Table 4.9	Value Counts	58

LIST OF ACRONYMS

Acronym	Description
KNN	K Nearest Neighbour
SVM	Support Vector Machine
LR	Logistic Regression
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
ACC	Accuracy
DT	Decision Tree
RF	Random Forest

CHAPTER-1: INTRODUCTION	10-18
1.1 Introduction	11-12
1.2 Algorithms	13-18
1.2.1 K-NN	13
1.2.2 LR	14
1.2.3 DT	15
1.2.4 NB	15
1.2.5 SVM	16
1.2.6 RF	16-18
CHAPTER-2: LITERATUE SURVEY	19-29
2.1 Introduction	21
2.2 Survey on agriculture analysis data	22-29
CHAPTER-3: PROPOSED METHODOLOGY	30-37
3.1 Introduction	32
3.2 Existing System	32
3.3 Motivation	32-33
3.4 Proposed framework	33-34
3.5 Description	35-36
3.5.1 Flow Chart	35
3.5.2 System Achitecture	36

CHAPTER-4:EXPERIMENTAL SETUP &RESULTANALYSIS	38-82
4.1 Simulation Environment	40
4.2 Experimental Setup	40
4.2.1 Data Set	40
4.2.2 Data Set Information	41
4.2.2.1 Data Fields	42
4.3 Performance Measures	42
4.3.1 Classification Accuracy	42
4.3.2 Confusion Matrix	43
4.3.3 Precision (positive predicted value)	43
4.3.4 Recall	44
4.3.5 F1-Score	44
4.4 Tools Used	45
4.4.1 Jupyter Notebook Tool	45
4.4.2 Installation	46-52
4.5 Experimental Result	53-67
4.5.1 Statistical Analysis for machine learning algorithms	53-57
4.5.1.1 Describing Descriptive Statistics	55-56
4.5.1.2 HeatMap(Visualling Statistics)	58-60
4.5.1.3 K-Nearest Neighbours	61-63
4.5.1.4 Logistic Regression	63-65
4.5.1.5 Decision Tree	65-67
4.5.1.6 Naïve Bayes	67-69
4.5.1.7 SVM	69-72
4.5.1.8 Random Forest	72-74

4.6 Confusion Matrix	75-82
4.6.1 K-Nearest Neighbours	76-77
4.6.2 Decision Tree	78-79
4.6.3 Random Forest	79-80
4.6.4 Result Analysis	81-82
4.6.4.1 Analysis on various Classification	83-83
CHAPTER-5 : CONCLUSION AND FUTURE WORK	83 – 87
5.1 Conclusion	87
5.2 Future Work	87
REFERENCES	88-91

CHAPTER 1 : INTRODUCTION

1.1 Introduction

The Jupyter notebook is an open-source, browser-based tool functioning as a virtual lab notebook to support workflows, code, data and visualizations detailing the research process. It is machine and human readable, which facilitates interoperability and scholarly communication. These notebooks can live in online repositories and provide connections to research objects such as datasets, code, method documents, workflows, and publications that reside elsewhere. Jupyter notebooks are one means to make science more open[1].

An attempt has been made to review the research studies on application of data mining techniques in the field of agriculture. Some of the techniques, such as the k nearest neighbor and support vector machines applied in the field of agriculture were presented. Data mining in application in agriculture is a relatively new approach for forecasting / predicting of agricultural crop. This explores the applications of data mining techniques in the field of agriculture. Data mining is the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviours, allowing businesses to make proactive, knowledge-driven decisions[2].

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis. However,

clustering is a difficult problem combinatorially, and differences in assumptions and contexts in different communities has made the transfer of useful generic concepts and methodologies slow to occur. This paper presents an overview of pattern clustering methods from a statistical pattern recognition perspective, with a goal of providing useful advice and references to fundamental concepts accessible to the broad community of clustering practitioners[3]. Clustering is one of the most fundamental and essential data analysis techniques. Clustering can be used as an independent data mining task to discern intrinsic characteristics of data, or as a preprocessing step with the clustering results then used for classification, correlation analysis, or anomaly detection[4]. Data mining is the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviours, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analysis offered by data mining move beyond the analysis of past events provided by retrospective tools typical of decision support systems. There is a growing number of applications of data mining techniques in agriculture and a growing amount of data that are currently available from many resources. This is relatively a novel research field and it is expected to grow in the future. there is a lot of work to be done on this emerging and interesting research field[2]. Clustering algorithms are attractive for the task of class identification in spatial databases. However, the application to large spatial databases rises the following requirements for clustering algorithms: minimal requirements of domain knowledge to determine the input parameters, discovery of clusters with arbitrary shape and good efficiency on large databases. The well-known clustering algorithms offer no solution to the combination of these requirements. In this paper, we present the new clustering

algorithm relying on a density-based notion of clusters which is designed to discover clusters of arbitrary shape[5]. In this analysis, we used some of the common data mining techniques in the field of agriculture. Agriculture is an upcoming research field. Efficient techniques can be developed and used for solving complex agricultural problems using data mining. Future enhancement of this agriculture analysis is to predict the crop yield using these techniques. It is useful for making crop decisions for farmers and government organizations.

1.2 Algorithms

In this project, we have used five different classification algorithms; they are KNN, Logistic regression, Decision Tree, Naïve Bayes and SVM. The problems in the agriculture field can be efficiently solved by using data mining techniques since it anticipate before in hand with the help of raw data's.

1.2.1 K-Nearest Neighbour

Agriculture is full of uncertainty due to climate change, rainfall, soil type and numerous other factors. Crop prediction in agriculture is a very big dilemma and there is huge dataset where farmers find difficult to predict the yield and seed selection. In this current situation due to rise in the population the production of crops and agricultural products needs to be increased simultaneously to meet the demands of the people. These problems could be solved using machine learning algorithms and this paper focuses on these solutions.

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category

that is most similar to the available categories. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

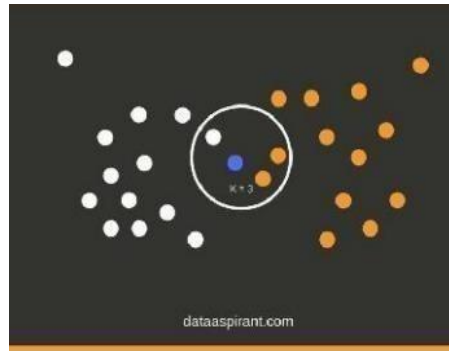


Fig 1.2.1 K-NN representation

1.2.2 Logistic Regression

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. The cost function can be defined as the 'Sigmoid function' or also known as the 'logistic function' instead of a linear function. To map predicted values to probabilities we use the sigmoid

function. Decision boundary on top of our predictions to see how our labels compare to the actual labels.

1.2.3 Decision Tree

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

1.2.4 Naïve Bayes

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

1.2.5 Support Vector Machine

“Support Vector Machine” is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space where n is number of features you have with the value of each feature being the value of a particular coordinate. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

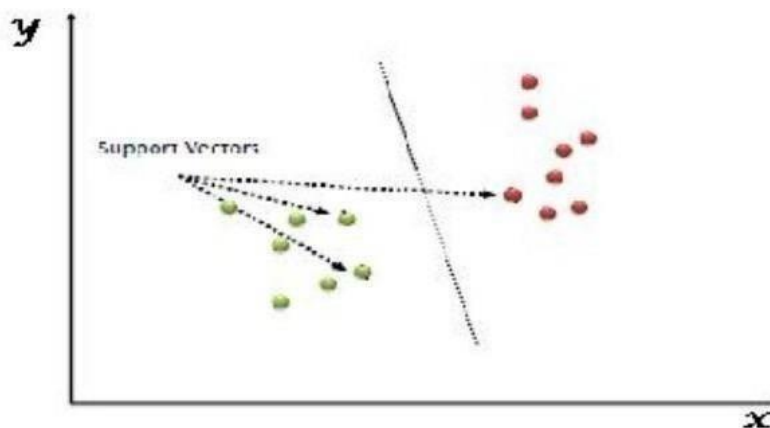


Fig 1.2.2 SVM Representation

1.2.6 Random Forest

Random Forest is a popular machine learning algorithm that belongs

to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity it can be used for both classification and regression tasks. Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of

voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Step 1 – First, start with the selection of random samples from a given dataset.

Step 2 – Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 – In this step, voting will be performed for every predicted result.

Step 4 – At last, select the most voted prediction result as the final prediction result.

CHAPTER 2 : LITERATUE SURVEY

2.1 Introduction

In our survey, we have gone through previous 15 years papers to know about the agriculture data briefly from different corners and also to know about different technologies, methods as well as methodologies used in their own survey.

2.2 Survey on agriculture analysis data using various algorithms

Sally Jo Cunningham et al [7], The ‘mined’ information is typically represented as a model of the semantic structure of the dataset, where the model may be used on new data for prediction or classification. Alternatively, human domain experts may choose to manually examine the model, in search of portions that explain previously misunderstood or unknown characteristics of the domain under study a two-way interaction between the provider of the data and the data mining expert. Both work together to transform the raw data into the final data set(s) input to the machine learning algorithms — with the domain expert providing information about data semantics and ‘legal’ transformations that can be applied to the data, and the data mining expert guiding the process so as to improve the intelligibility and accuracy of the results.

Jharna Majumdar et al [8], In agriculture sector where farmers and agribusinesses have to make innumerable decisions every day and intricate complexities involves the various factors influencing them. An essential issue for agricultural planning intention is the accurate yield estimation for the numerous crops involved in the planning. Data mining techniques are necessary approach for accomplishing practical and effective solutions for this problem. Agriculture has been an obvious target for big data. Environmental conditions, variability in soil, input levels, combinations and

commodity prices have made it all the more relevant for farmers to use information and get help to make critical farming decisions .

NG Yethiraj [9] ,The major reason that data mining has attracted a great deal of attention in information industry in recent years is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. Data mining can be viewed as a result of the natural evolution of information technology. An evolutionary path has been witnessed in the database industry in the development of the following functionalities: data collection and database creation, data management including data storage and retrieval, and database transaction processing, and data analysis and understanding involving data warehousing and data mining. Agriculture and allied activities constitute the single largest component of India's gross domestic product, contributing nearly 25% of the total and nearly 60% of Indian population depends on this profession. Due to vagaries of climate factors the agricultural productivities in India are continuously decreasing over a decade. The reasons for this were studied mostly using regression analysis.

A.Mucherino et al [10] ,In this survey we present some of the most used data mining techniques in the field of agriculture. Some of these techniques, such as the the k nearest neighbor, and support vector machines, are discussed and an application in agriculture for each of these techniques is presented. Data mining in agriculture is a relatively novel research field. It is our opinion that efficient techniques can be developed and tailored for solving complex agricultural problems using data mining. At the end of this survey we provide recommendations for future research directions in agriculture-related fields.

S Mishra et al [11], India is an agrarian country and its economy largely based upon crop productivity. Thus agriculture is the backbone of all business in India. Now India stands in second rank in worldwide in farm production, Agriculture and allied sectors like forestry and fisheries considered for 14.5% of the GDP in 2015 and about 50% of the total manpower. The economy improvement of agriculture towards India's GDP is Agricultural production is mostly affected by environmental factors. Weather influences crop growth and development, causing large intra-seasonal yield variability. In addition, spatial variability of soil properties, interacting with the weather, cause spatial yield variability. Crop agronomic management that is planting, fertilizer application, irrigation, tillage etc., can be used to offset the loss in yield due to effects of weather. As a result, yield forecasting represents an important tool for optimizing crop yield and to evaluate the crop-area insurance contracts.

Niketa Gandhi et al[12], Application of data mining techniques for decision making in agriculture. The paper reports the application of a number of data mining techniques including artificial neural networks, Bayesian networks and support vector machines. The review has outlined a number of techniques that have been used to understand the relationships of various climate and other factors on crop production. This review proposes that further investigations are needed to understand how these techniques can be used with complex agricultural datasets for crop yield prediction integrating seasonal and spatial factors by using GIS technologies.

RB Palepu et al[13], Agriculture is the most basic function to accomplish food demand all over the globe; it is a backbone particularly in the developing countries like India. This paper presents about the role of data mining in perspective of soil analysis in the field of agriculture and also confers about several data mining techniques and their related work by several authors in context to soil analysis

domain. It has been discussed about how data mining techniques are applied in agriculture field. Globally, day to day the requirement of food is escalating; hence the agricultural scientists, farmers, government, and researchers are tiresome to put extra attempt and use numerous techniques in agriculture for improvement in production. As an effect, the data generated in the field of agricultural data enhanced day by day. As the degree of data enlarges, it requires instinctive way for these data to be mined and analyzed when needed. Even at present, a very only some farmers are really using the new methods, tools and techniques in agriculture for better production. Data mining can be used for forecasting the future trends of agricultural processes.

Vinayak A. Bharadi et al [14], Agriculture contributes nearly sixteen percent to total GDP of India and ten percent of the total exports which helps in increasing foreign exchange. The population of India is continuously increasing and to meet the food necessities of this growing population, agricultural yield should be boosted. Knowledge discovered from raw data is useful for many purposes. Data mining techniques are better choices for the same. This paper aims to analyze the agricultural data of India using data mining algorithms and to find useful information from the results of these techniques which would help to improve the agricultural yield. Agriculture is the backbone of Indian economy. Agriculture majorly contributes to the exports of India, directly improving foreign currency exchange. In India, majority of the farmers do not get expected yield due to several reasons. The agricultural yield primarily depends on environmental factors such as rainfall, temperature and geographical topology of the particular region. These factors along with some other influence the crop cultivation.

N.Hemageetha [15] ,By improving the agricultural sector the GDP of the nation could also be improved. The digital era can render its support to the agricultural sector

in wide variety of ways. Data Mining supports decision making process and prediction. Agriculture needs the decision support system in variety of ways such as type of crop to be cultivated. And prediction techniques for rainfall prediction, weather prediction, market price prediction etc., There were many researches going on to support the agriculture using data mining. Analyzing soil provides major contribution to the support of the farmers.

Hetal Patel et al [16] ,However the application of data mining methods and techniques to discover new insights or knowledge is a relatively a novel research area. In this paper we provide a brief review of a variety of Data Mining techniques that have been applied to model data from or about the agricultural domain. The Data Mining techniques applied on Agricultural data include k nearest neighbor, Neural Networks (NN) Support Vector Machine (SVM), Naive Bayes Classifier . Agriculture is the most important application area particularly in the developing countries like India. Use of information technology in agriculture can change the scenario of decision making and farmers can yield in better way. For decision making on several issues related to agriculture field; data mining plays a vital role. In this paper we have discussed about the role of data mining in perspective of agriculture field.

D Ramesh et al [17],Yield prediction is an important agricultural problem. Every farmer is interested in knowing, how much yield he is about expect. In the past, yield prediction was performed by considering farmer's previous experience on a particular crop. The volume of data is enormous in Indian agriculture. The data when become information. Data Mining is widely applied to agricultural problems. Data Mining is used to analyze large data sets and establish useful classifications and patters in the data sets. The overall goal of the Data Mining process is to extract the

information from a data set and transform it into understandable structure for further use.

Kusum Latha et al [18] ,Agriculture field of study requires attention in order to equip the farmers to maximize their output. In our country agriculture is the strength of the economy and growth and more than half of the population is living on the agriculture output. The crop yield is the major factor to decide the farmers earning and governments planning to meet the requirements to ensure the food security. Crop yield prediction will assist the farmers and other stakeholders for better crop planning i.e. selling, warehousing, market prices etc. Mainly data mining techniques for DSS is based on artificial neural networks, Bayesian networks, vector support system etc. The decision support system will help the farmers to cut the losses, improve the crop yield due to proactive planning .

Hooman Fetanat et al [19],Data mining software is one of a number of analytical tools for analyzing data. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

This paper discusses data mining technique such as regression and clustering which is a process model for analyzing data and describes the support that. Cluster analysis or clustering was used is the task of assigning a set of objects into groups so that the objects in the same cluster are more similar to each other than to those in other clusters. In this survey clustering technique divides growth factors into several independent categories. Also, regression technique which was used includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables.

Anshal Savla et al [20], Precision agriculture is the implementation of the recent technology in agriculture. Huge amount of data is collected in agriculture and various techniques of data mining are used to make efficient use of it. In this paper, we have discussed various algorithms related to classification techniques of data mining. These algorithms are implemented on a data set that has been collected over the years for the yield prediction of soybean crop. Further, a comparative analysis is done to show which classification algorithm is best suited for predicting the yield with respect to classification techniques.

Branko Marinkovic et al [21] , Prediction of agricultural yields is a challenging task that demands fusion of knowledge from different areas such as data mining, statistics and agriculture. This paper shows that data mining techniques can be successfully applied to agricultural data analysis. Results that we present are gained on the data set that contains monthly measurements of different environmental parameters and annual yields for maize, soybean. One of the interesting aspects of precision agriculture is the prediction of yields. Data mining offers possibilities to change raw data into valuable information that could be used for making better decisions.

Table 2.1: Survey on Agriculture data

Author Name	Title	Publications Years	Algorithms Me thods	Output Ref
S.Pudumalar et.al	Crop Recommend- -ation System for precision Agriculture.	2016	K-Nearest Neighbor and Naive Bayes	Thus our work would help farmers in sowing the [22] right seed based on soil requirements to increase productivity and acquire profit out of such a technique.
B.Devika et.al	Analysis of Crop Yield Prediction using Data Mining Technique to Predict Annual Yield of Major Crops	2018	K-Nearest Neighbours, Linear Regression.	Power couldalso be obtained by parameters like year, crop, [23] area, production and alternative variables, like climate, agricultural practices and soil characteristics are including.

Suvidha Jambekar et.al	Prediction of Crop Production in India Using Data Mining Techniques.	2018	Logistic Regression Models.	Can be used to predict production of Rice, Wheat [24] and Maize with precision. Accurate forecasts of these parameters would result in accurate production forecast in the future.
M Rajshree et.al	Data Mining Technique For Agriculture and Related Areas .	2011	K-Means clustering, K-Nearest neighbor clustering, Support Vector Machine, Decision Tree.	Using the appropriate technique of data mining, [25] important existing socio- economic pattern in the farmers' data set can be identified.
Ashwin Kowshik et.al	Crop yield prediction based on Indian agriculture using machine learning .	2021	K-Nearest Neighbours, SVM.	[26] Provided the farmer with the yield of a crop based on land area, rainfall, temperature and district using machine learning.

CHAPTER 3 : Proposed Methodology

3.1 Introduction

Machine learning (ML) approaches are used in many fields, ranging from supermarkets to evaluate the behavior of customers to the prediction of customers' phone use. Machine learning is also being used in agriculture for several years . Crop yield prediction is one of the challenging problems in precision agriculture, and many models have been proposed and validated so far. This problem requires the use of several datasets since crop yield depends on many different factors such as climate, weather, soil, use of fertilizer, and seed variety . This indicates that crop yield prediction is not a trivial task; instead, it consists of several complicated steps. Nowadays, crop yield prediction models can estimate the actual yield reasonably, but a better performance in yield prediction is still desirable .

3.2 Existing system

In Agriculture is the most basic function to accomplish food demand all over the globe. It is a backbone particularly in the developing countries like India. The application of data mining techniques in agriculture especially on soils can revise the situation of pledge making and improve cultivation yields in a better way.

3.3 Motivation

Human guess is old traditional methods which often based on assumptions however without having any avoidance to cope with issue. Specifically, plant nutrition, health, disease identification and growth level may need sensitive treatments and remedies for plant health assessment. The implementation of machine learning methods in agricultural science improved the computerized

guessed work for future prediction. The important factor of agriculture field assessment such as soil status prediction, soil salinity, soil pH level, Soil nitrogen level prediction, temperature update, humidity prediction and crop yield prediction and many more application in today world. ML can perform automated decision accuracy without being explicitly programmed in different scenario. Data – driven approaches provided to support obtained data analysis and probabilistic decision making in real life applications.

3.4 Proposed framework

Acquaints in Connection With some of the most important data mining techniques used in agriculture. Mining in agriculture is a innovative groundwork domain. The problems in the agriculture field can be efficiently solved by using data mining techniques. This section describes the outputs obtained after implementation of Data Mining algorithms on the dataset obtained. The data mining techniques like KNN, SVM, Decision Tree, Naïve Bayes, Random Forest, Logistic Regression algorithm where high accuracy can be achieved. So, these technologies can change the situation of farmers and decision making in agricultural field in a better way.

In this work, different states data consisting of varied crops are taken into consideration. Supervised learning is utilized for modelling, which gives the predicted yield and their order of production. The various steps of the proposed framework are discussed in following sub sections.

A. Dataset Collection:

Data is gathered from various sources [9,10] and then analyzed and prepared. This data is utilized for descriptive analysis. The dataset used in this paper consists of various states (Maharashtra, UP, West Bengal, Gujarat etc), different types of crops

(sugarcane, coconut, wheat, gram etc), different seasons (Kharif, Rabi, Whole, Summer etc), different crop years and other parameters such as Rainfall, Temperature, pH, Humidity.

B. Preprocessing the data:

In this module, dataset is preprocessed so as to fill the missing values, the fitting information run and separating the usefulness.

C. Feature Extraction:

Feature extraction ought to streamline the amount of data required to represent a huge dataset. Its goal is to extract useful characteristics from data. The characteristics include high, low and mean temperature, air humidity, soil pH, rainfall.

D.Split dataset into Train and Test set:

This step includes training and testing of the input data. The stacked information is isolated into two sets, such as preparing and testing the data. Training set is mapped with the training set and during the training phase data is to be testing after learning from previous observations. The final data is formed and is processed by machine learning module.

E.Apply Data Mining Techniques:

In our project, different supervised machine learning techniques for prediction of crop yield are used which is given as follows in Figure 3.1.

3.5 Description

We have taken the data set i.e Agriculture data from UCI Repository which is freely available in the internet and for that data set we have various classification algorithms like KNN, SVM, Decision Tree, Naive Bayes, Random Forest, Logistic Regression algorithm. Applying various statistical methods for analyzing the results and generated the final report. For the brief understanding of the proposed system and its methodology as well as to know about what are the methods and algorithms used in it and which approach is followed to give accurate solution.

3.5.1 Flow chart

Below Figure represents the step by step procedure of the proposed system. It clearly explains us about complete view of proposed system that is where we have taken the data and what methods and techniques applied etc. Here we are selecting the data from UCI repository then applying data preprocessing techniques for data cleaning and transformation. After that evaluating the results based on machine learning classification Algorithms.

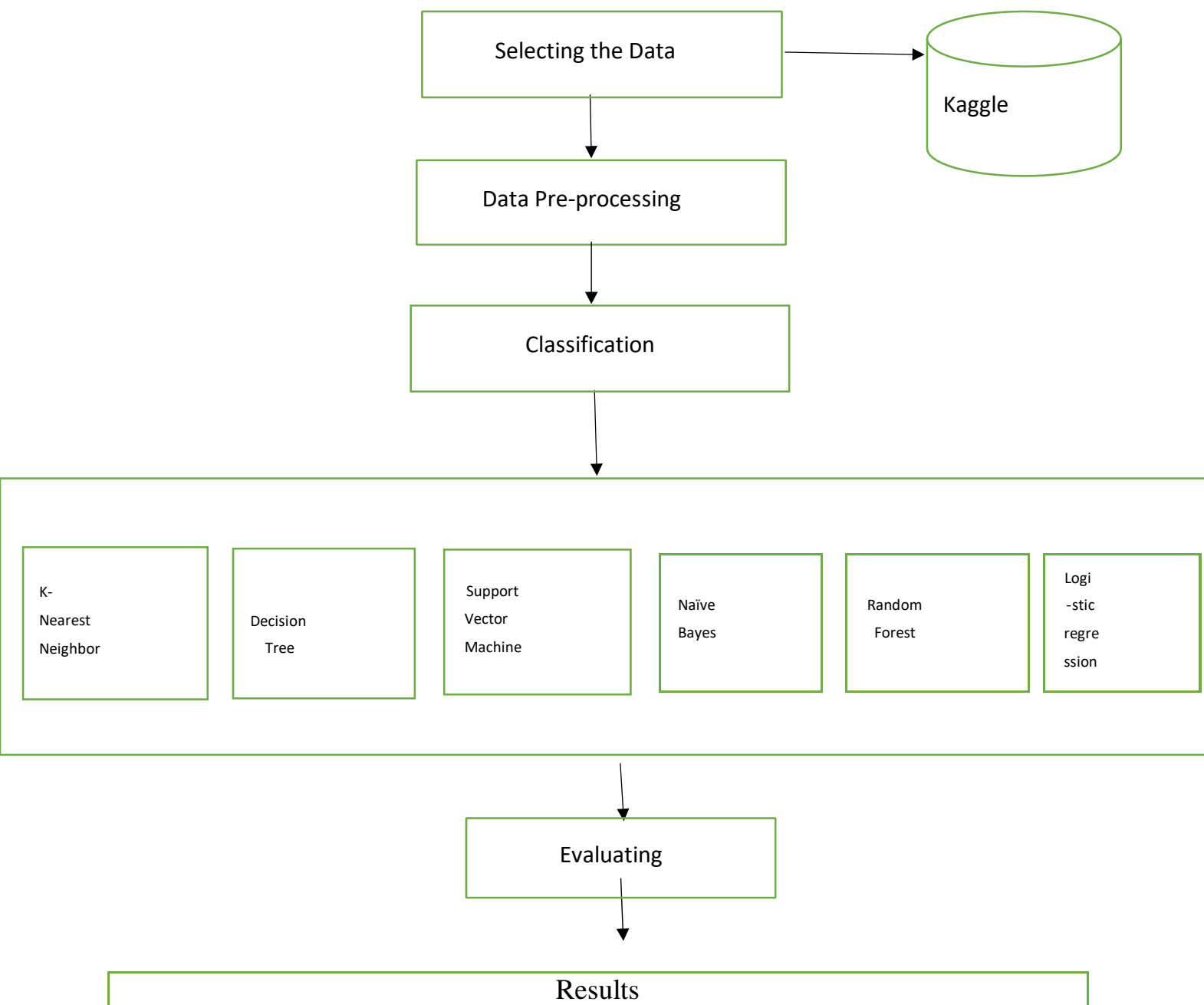


Fig 3.2: Architecture for proposed system

3.5.2 System architecture

Figure 3.2 indicates the architecture of the proposed system by using the Matlab. It is handling three different classification algorithms such as KNearest Neighbor, Support Vector Machine and Logistic Regression.

Choose among various algorithms to train and validate classification models for binary or multiclass problems. After training multiple models, compare their validation errors side-by-side, and then choose the best model. To help you decide which algorithm to use, see Train Classification Models in Classification Learner.

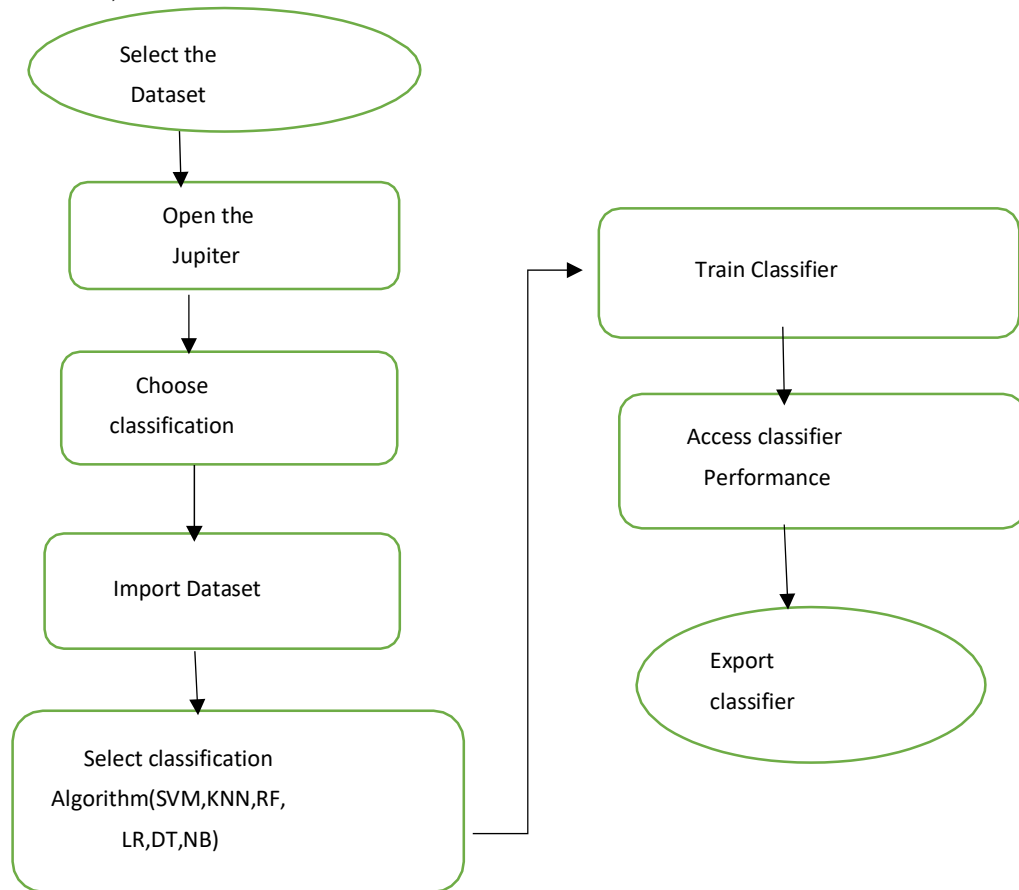


Fig 3.3 Proposed architecture using Jupiter

This flow chart shows a common workflow for training classification models, or classifiers, in the Classification Learner app.

CHAPTER 4 : Experimental Setup and result analysis

Motivation :

Precision agriculture is in trend nowadays. 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 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.

4.1 Simulative environment

The developing environment for the proposed method is jupyter r2018b Version on a system with any Intel or AMD Intel(R) Core i3-1005G1, 1.20GHz , 1.19 GHz RAM and Microsoft windows with family.

4.2 Experimental setup

In Experimental setup we are explaining about the Data Set i.e Agriculture Analysis data set, jupyter.

4.2.1 Dataset

The data used in this project is made by augmenting and combining various publicly available datasets of India like weather, soil, etc. This data is relatively simple with very few but useful features unlike the complicated features affecting the yield of the crop.

The data have Nitrogen, Phosphorous, Potassium and pH values of the soil.

Also, it contains the humidity, temperature and rainfall required for a particular crop

Using this dataset we have analyzed different factors which are mainly leading to

Agricultural Analysis by applying various machine learning and data mining techniques.

Data Set Characteristics:	Multivariate	Number of Instances:	2200	Area:	N/A
Attribute Characteristics:	Real	Number of Attributes:	8	Date Donated	2020- 08-14
Associated Tasks:	Classification	Missing Values?	Yes	Number of Web Hits:	29866

Table 4.1 Description of data set

4.2.2 Data set information

The goal of this research is to change the situation of the farmers and decision making in agricultural field in a better way. Precision agriculture is in trend nowadays. It helps the farmers to get informed decision about the farming strategy.

Dataset which would allow the users to build a predictive model to recommend the most suitable crops to grow in a particular farm based on various parameters. This dataset was build by augmenting datasets of rainfall, climate and fertilizer data available for India.

4.2.2.1 Data Fields

Many risk factors are examined from various areas like:

- 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

4.3 Performance measures

There are some performance measures based on the classification. By using the Jupyter Notebook, we have implemented and analyzed the results of the Logistic Regression, Support Vector Machine and K-Nearest Neighbor ,Decision Tree and Random Forest ,Naïve Bayes by using Jupyter Notebook.

4.3.1 Classification accuracy

It is calculated as the total number of correct forecasts divided by the total number of datasets. It works well on balanced data. If it is imbalanced data then errors occur at the

performance. Accuracy is calculated as the number of all correct predictions divided by the total number of the dataset.

$$\text{Accuracy} = (\text{Number of true classified samples}) / (\text{Number of total test data}) \times 100$$

4.3.2 Confusion matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It helps you identify the areas where the classifier has performed poorly. It is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

4.3.3 Precision

Precision helps when the costs of false positives are high. So let's assume the problem involves the detection of skin cancer. If we have a model that has very low precision, then many patients will be told that they have included some misdiagnoses. Lots of extra tests and stress are at stake. When false positives are too high, those who monitor the results will learn to ignore them after being bombarded with false alarms. It is a metric for classification models. Precision recognizes the incidence with which a model was correct when dividing the certain class. High precision describes the low false positive rate.

These are favored in information retrieval positives for the documents that are retrieved in response to a query true positive and are really relevant to the query. y-axis: cision. Precision value is calculated by using the following formula. **Precision=TP/TP+FP**

4.3.4 Recall

The recall is the proportion of correctly divined observations to the total predicted positive observations in real class. Recall helps when the cost of false negatives is high. What if we need to detect incoming nuclear missiles? A false negative has devastating consequences. When false negatives are frequent, you get hit by the thing you want to avoid. A false negative is when you decide to ignore the sound of a twig breaking in a dark forest, and you get eaten by a bear. (A false positive is staying up all night sleepless in your tent in a cold sweat listening to every shuffle in the forest, only to realize the next morning that those sounds .

If you had a model that let in nuclear missiles by mistake, you would want to throw it out. If you had a model that kept you awake all night because you would want to throw it out, too. If, like most people, you prefer to not get eaten by the bear, and also not stay up all night worried about chipmunk alarms, then you need to optimize for an evaluation metric that's a combined measure of precision and recall. Enter the F1 score.

Mathematically, recall value is calculated by using the following formula:-

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

4.3.5 F1-Score

The F-score, also called the F1-score, is a measure of a model's accuracy

on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'. The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall. The F-score is commonly used for evaluating information retrieval systems such as search engines.

$$\text{F1 score} = 2 * (\text{recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

So it's best if we can get a single score that kind of represents both Precision(P) and Recall(R). One way to do that is simply taking their arithmetic mean. i.e $(P + R) / 2$ where P is Precision and R is Recall. But that's pretty bad in some situations.

4.4 Tools used

In this, we used jupyter notebook tool for the classifying the algorithms to visualize the data.

4.4.1 Jupyter Notebook Tool

Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text.

Jupyter Notebook is maintained by the people at Project Jupyter.

The Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, jupyter, comes from the core supported programming languages that it supports: Julia, Python. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

The Jupyter Notebook is not included with Python, so if you want to try it

There are many distributions of the Python language. This article will focus on just two of them for the purposes of installing Jupyter Notebook. The most popular is CPython, which is the reference version of Python that you can get from their website. It is also assumed that you are using Python3.

Using Notebooks is now a major part of the data science workflow at companies across the globe. If your goal is to work with data, using a Notebook will speed up your workflow and make it easier to communicate and share your results. Best of all, as part of the open source Project Jupyter, Jupyter Notebooks are completely free. You can download the software on its own, or as part of the Anaconda data science toolkit.

Although it is possible to use many different programming languages in Jupyter Notebooks, this article will focus on Python, as it is the most common use case. Among R users, R Studio tends to be a more popular choice.

4.4.2 Installation

The easiest way for a beginner to get started with Jupyter Notebooks is by installing Anaconda. Anaconda is the most widely used Python distribution for data science and comes pre-loaded with all the most popular libraries and tools. Some of the biggest Python libraries included in Anaconda include NumPy, pandas, and Matplotlib, though the full 1000+ list is exhaustive.

Anaconda thus lets us hit the ground running with a fully stocked data science workshop without the hassle of managing countless installations or worrying about dependencies and OS-specific (read: Windows-specific) installation issues.

To get Anaconda, simply:

The latest version of Anaconda for Python 3.7.9.

install Anaconda by following the instructions on the download page and/or in the executable.

If you are a more advanced user with Python already installed and prefer to manage your packages manually, you can just use pip:

Pip3 install jupyter

Python is a prerequisite for running a Jupyter notebook, so we need to install python first. Please,

follow this URL and choose right version to install: <https://www.python.org/downloads/>. I have chosen 'Windows x86-64 executable installer' for my Windows 64bit OS. Please choose the version as per your computer Operating system.

Files			
Version	Operating System	Description	MD5 Sum
Gzipped source tarball	Source release		ea132d6f449766623eee886966c7d41f
XZ compressed source tarball	Source release		69e73c49eeb1a853cefd26d18c9d069d
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	68170127a953e7f12465c1798f0965b8
Windows help file	Windows		4403f334f6c05175cc5edf03f9cde7b4
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	5f95c5a93e2d8a5b077f406bc4dd96e7
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	2acba3117582c5177cdd28b91bbe9ac9
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	c9d599d3880dfbc08f394e4b7526bb9b
Windows x86 embeddable zip file	Windows		7b287a90b33c2a9be55fab24a7febbb
Windows x86 executable installer	Windows		02cd63bd5b31e642fc3d5f07b3a4862a
Windows x86 web-based installer	Windows		acb0620aea46edc358dee0020078f228

Figure4.1: Windows Executable

You can download the executable file and save in any location at your computer.

Now next step is to create a 'Python' folder under the C: drive, we will use this folder as installation location at later step.

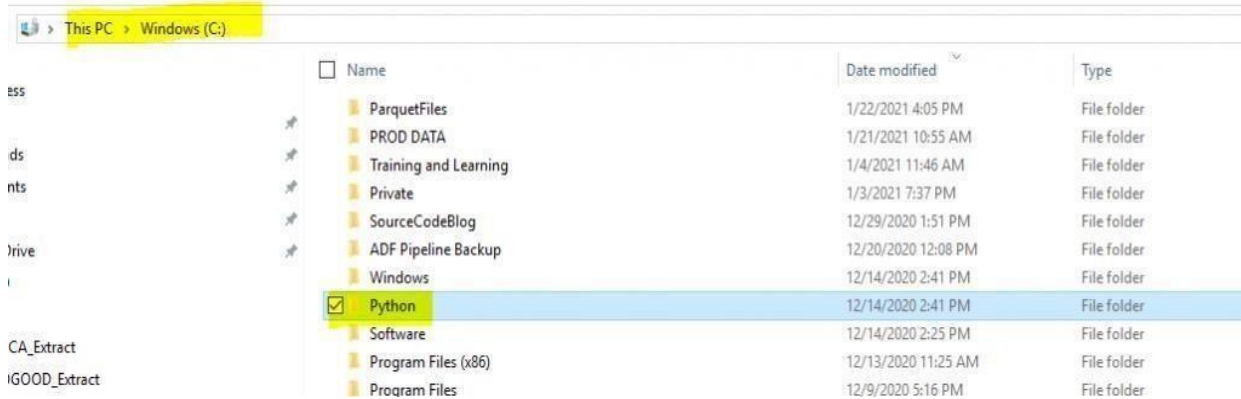


Figure 4.2: Python folder under C

Find out the downloaded executable file, I have saved the executable file under Downloads folder (shown in below figure 4.3). Now double click the executable file To initiate the installation process.

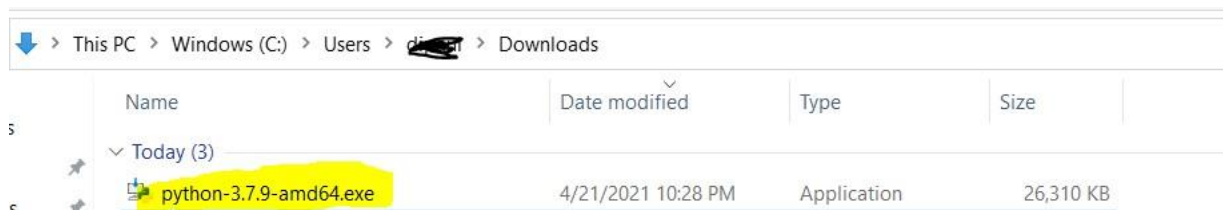


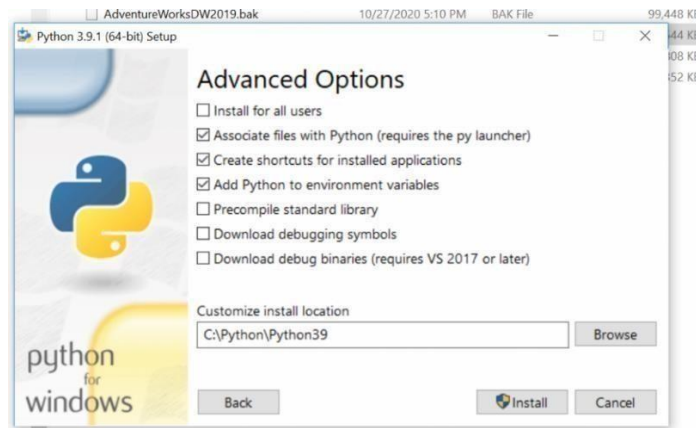
Figure 4.3: Python Execution file

Make sure to choose 'Customize Installation' and check mark 'Add Python 3.9 to PATH' as shown in figure 4.6 followed the customization method to avoid setting up environment.



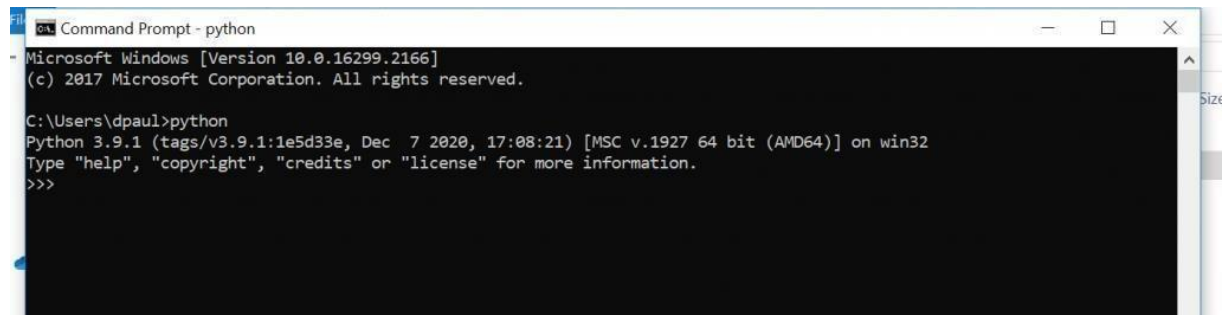
Figúie4.4: Python Installation wizard

As below figure 4.4 shown, the Customize installation location, where make sure you put the installation location folder C:\Python\Python39. We have created 'Python' folder in C drive in earlier step (Fig 2).



Now hit the Install button. Installation will complete in a minute or two.

Let's test if python installed successfully, open command prompt and type "python". If python is installed correctly then you should able to see the python version number and some key help, as shown below in Fig 4.5.



```
Command Prompt - python
Microsoft Windows [Version 10.0.16299.2166]
(c) 2017 Microsoft Corporation. All rights reserved.

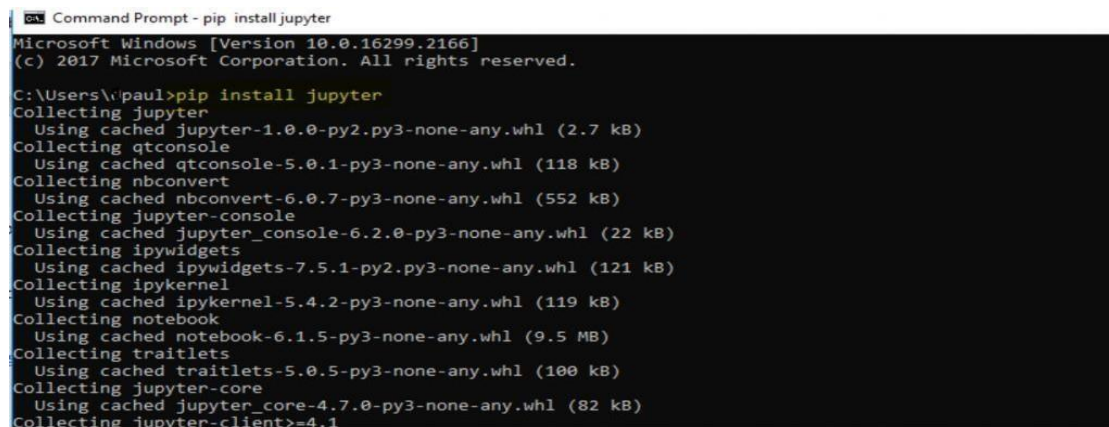
C:\Users\dpaul>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figure 4.6: Python installed successfully

Step 2: Install the Jupyter Notebook

Let's move to the next step, which is to install the Jupyter notebook software. Open command prompt and type the below code:

Pip3 install jupyter



```
Command Prompt - pip install jupyter
Microsoft Windows [Version 10.0.16299.2166]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\dpaul>pip install jupyter
Collecting jupyter
  Using cached jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting qtconsole
  Using cached qtconsole-5.0.1-py3-none-any.whl (118 kB)
Collecting nbconvert
  Using cached nbconvert-6.0.7-py3-none-any.whl (552 kB)
Collecting jupyter-console
  Using cached jupyter_console-6.2.0-py3-none-any.whl (22 kB)
Collecting ipywidgets
  Using cached ipywidgets-7.5.1-py2.py3-none-any.whl (121 kB)
Collecting ipykernel
  Using cached ipykernel-5.4.2-py3-none-any.whl (119 kB)
Collecting notebook
  Using cached notebook-6.1.5-py3-none-any.whl (9.5 MB)
Collecting traitlets
  Using cached traitlets-5.0.5-py3-none-any.whl (100 kB)
Collecting jupyter-core
  Using cached jupyter_core-4.7.0-py3-none-any.whl (82 kB)
Collecting jupyter-client>=4.1
```

Figure 4.7: Jupyter Notebook installation started

When installation is complete, let's run the Jupyter Notebook web application. To do this, you need to go to a cmd prompt and execute this command, as shown in below figure 4.7:

Jupyter notebook

```
Command Prompt - jupyter notebook
Microsoft Windows [Version 10.0.16299.2166]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\ipaul>jupyter notebook
[I 22:13:54.334 NotebookApp] The port 8888 is already in use, trying another port.
[I 22:13:54.337 NotebookApp] Serving notebooks from local directory: C:\Users\dpaul
[I 22:13:54.337 NotebookApp] Jupyter Notebook 6.1.5 is running at:
[I 22:13:54.337 NotebookApp] http://localhost:8889/?token=d66ccc685c2cc2e16246deb058884fcde7e1390c9749649
[I 22:13:54.337 NotebookApp] or http://127.0.0.1:8889/?token=d66ccc685c2cc2e16246deb058884fcde7e1390c9749649
[I 22:13:54.337 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip conf
[C 22:13:54.560 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/dpaul/AppData/Roaming/jupyter/runtime/nbserver-12528-open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=d66ccc685c2cc2e16246deb058884fcde7e1390c9749649
or http://127.0.0.1:8889/?token=d66ccc685c2cc2e16246deb058884fcde7e1390c9749649
```

Figure 4.8: Opening Jupyter Notebook

As soon as you hit the above command button, It will open a browser with jupyter notebook as shown in figure 4.8.



Figure4.9: Jupyter Notebook on browser

Now you can create a Notebook by choosing 'New' and choose Python 3 as show in fig 4.9. This will open a new browser tab where you will write the code.

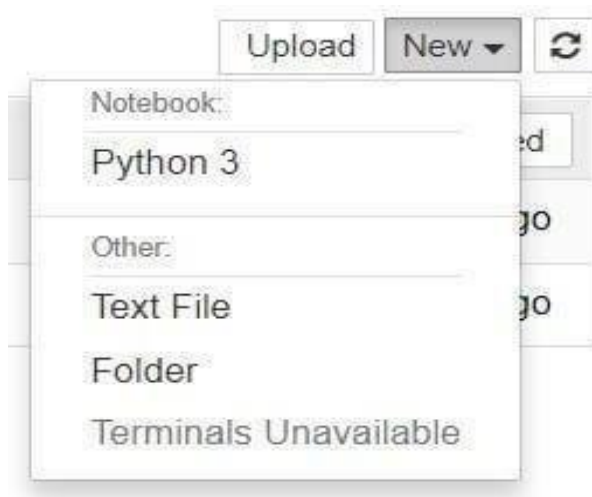
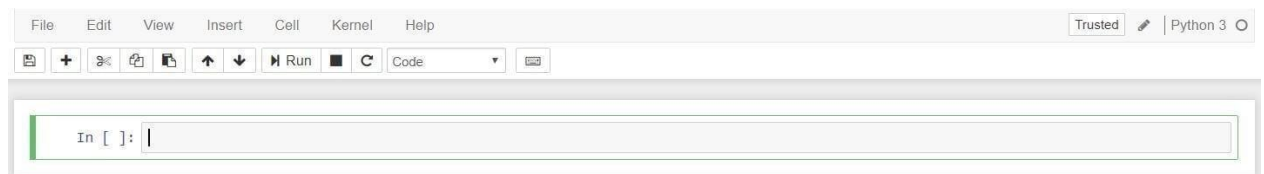


Fig 4.10: Open Notebook

Let's write hello world program in the Jupyter notebook. The browser will look like figure 11 if you enter this code.



`Print('Hello world')`

It is shown after clicking the 'Run' button

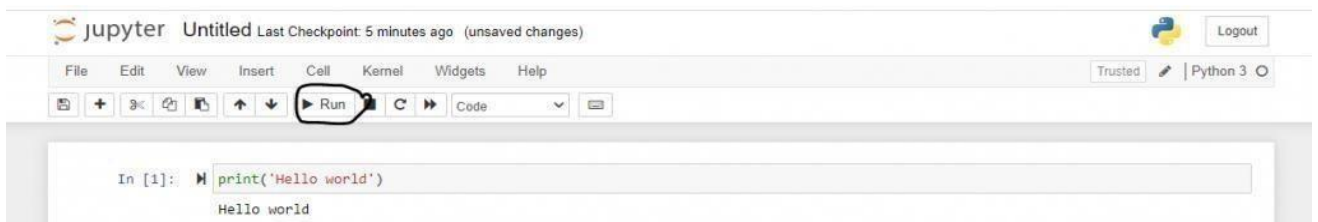


Fig 4.11: Hello world in Jupyter Notebook

Now you can write and run other notebooks.

In this article, we learned how to install python and Jupyter Notebooks and written simple hello world program. There are different ways you can install Jupyter Notebook, but I followed this approach and found simple.

4.5 Experimental result

In experimental analysis, we have taken the result from jupyter Notebook tool performing analysis.

4.5.1 Statistical analysis for machine learning algorithms

We applied various statistical methods for analyzing various classification algorithms that are used in our project such as K-Nearest Neighbor, Support Vector Machine and Logistic Regression, Decision Tree, Naïve Bayes, Random Forest.

4.5.1.1 Describing Descriptive Statisticsa

The dataset contains rows and columns of the data in the below table.

In [23]: df

Out[23]:

	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 4.2 descriptive of the table

Dataset.Head(): The dataset of the first five rows of the data in the below table.

In [25]: df.head()

Out[25]:

	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

Table 4.3 Dataset of first five rows

Dataset.Tail(): The dataset of the last five rows of the data in the below table.

Table 4.4 Dataset of last five rows

```
In [26]: df.tail()
```

```
Out[26]:
```

	N	P	K	temperature	humidity	ph	rainfall	label
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

Table 4.5 Size Of the dataset

```
In [27]: df.size
```

```
Out[27]: 17600
```

Shape of the dataset. The dataset contains 2200 rows and 8 columns and the dataset is name is a object type and remaining all attributes are the float datatype and the dtype is object.

```
In [12]: df.dtypes
Out[12]: N          int64
        P          int64
        K          int64
        temperature float64
        humidity    float64
        ph          float64
        rainfall    float64
        label       object
        dtype: object
```

Table 4.6 Shape Of the dataset

Data type is an attribute associated with a piece of data that tells a computer system how to interpret its value. Understanding data types ensures that data is collected in the preferred format and the value of each property is as expected. It is the most common numeric data type used to store numbers without a fractional component. It is used to store a single letter, digit, punctuation mark, symbol, or blank space. A data type, in programming, is a classification that specifies which type of value a variable has and what type of mathematical, relational or logical operations can be applied to it without causing an error. A string, for example, is a data type that is used to classify text and an integer is a data type used to classify whole numbers.

Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data. The data type defines which operations can safely be performed to create, transform and use the variable in another computation. When a program language requires a variable to only be used in ways that respect its data type, that language is said to be strongly typed.

These are the parameters we are used in our project.

Table 4.7 Parameters of the dataset

```
In [10]: df.columns
```

```
Out[10]: Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label'], dtype='object')
```

We are describing the unique parameters of our project.

Table 4.8 Unique parameters

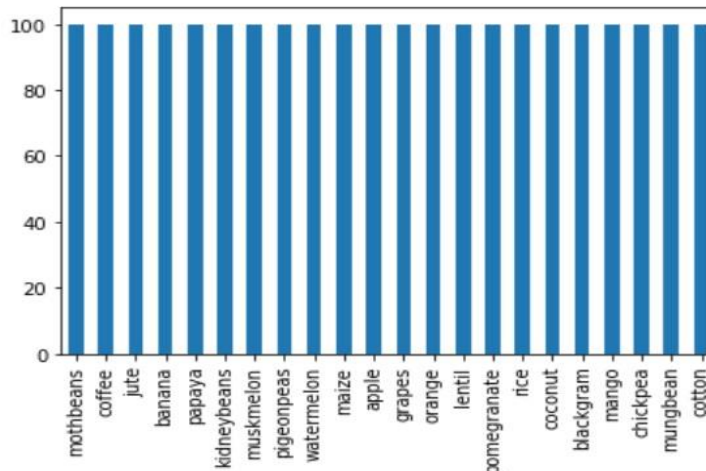
```
In [11]: df['label'].unique()
```

```
Out[11]: 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)
```

Unique Representation Of parameters using Bar Plot

Fig4.12 Bar Plot

```
In [10]: labels = data["label"].unique()  
data["label"].value_counts().plot(kind="bar")  
plt.show()
```



We are describing the value counts of our project.

Table 4.9 Value Counts

```
In [13]: df['label'].value_counts()
```

```
Out[13]: cotton      100  
watermelon  100  
jute        100  
grapes      100  
mungbean    100  
mothbeans   100  
rice        100  
mango       100  
coconut     100  
pomegranate 100  
chickpea    100  
apple       100  
banana      100  
pigeonpeas  100  
lentil      100  
maize       100  
papaya      100  
blackgram   100  
orange      100  
kidneybeans 100  
coffee     100  
muskmelon   100  
Name: label, dtype: int64
```

4.5.1.2 Heat Map (Visualling Descriptive Statistics)

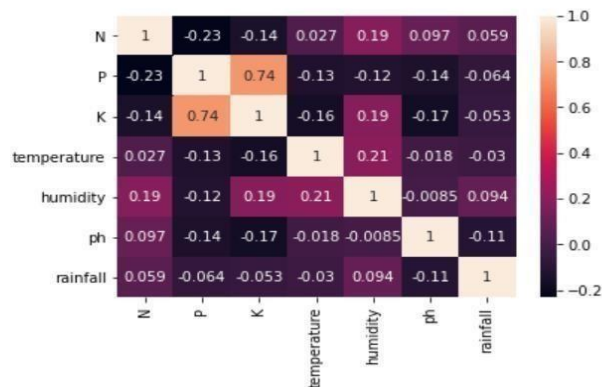
Histogram plot visualization for each attribute will be so difficult because we

have high dimensional columns 8. So better, we can use heat map to find the correlations coefficient values. We will remove the less correlation coefficient columns. We can remove the irrelevant features it will minimize the Accuracy of an algorithm. It will be better if we take relevant features columns then we can achieve to get good accuracy.

Fig 4.13 Heat map

```
In [14]: sns.heatmap(df.corr(),annot=True)
```

```
Out[14]: <AxesSubplot:>
```



```
In [16]: label_dict = {}
for i in range(22):
    label_dict[i] = label_encoder.inverse_transform([i])[0]
label_dict
```

```
Out[16]: {0: 'apple',
1: 'banana',
2: 'blackgram',
3: 'chickpea',
4: 'coconut',
5: 'coffee',
6: 'cotton',
7: 'grapes',
8: 'jute',
9: 'kidneybeans',
10: 'lentil',
11: 'maize',
12: 'mango',
13: 'mothbeans',
14: 'mungbean',
15: 'muskmelon',
16: 'orange',
17: 'papaya',
18: 'pigeonpeas',
19: 'pomegranate',
20: 'rice',
21: 'watermelon'}
```

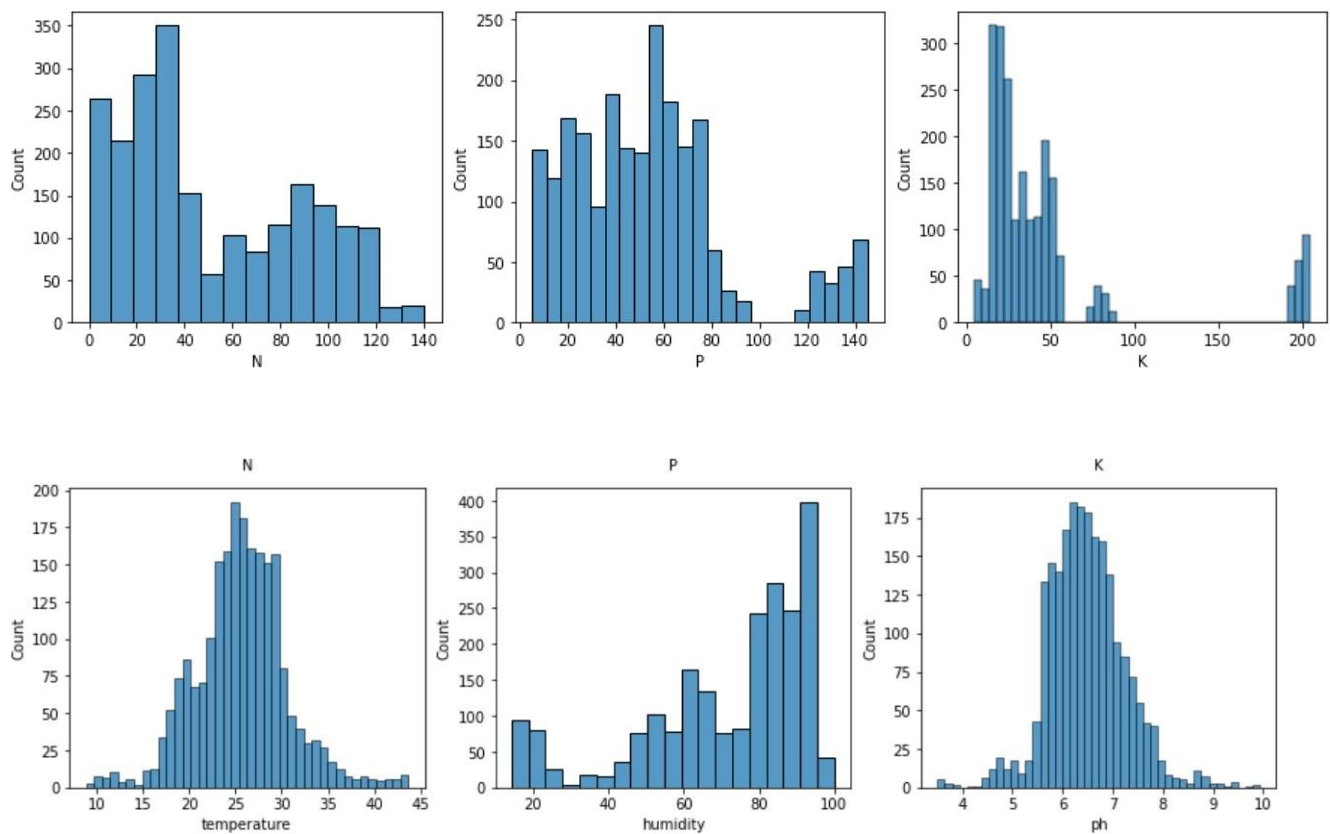
We are defining bar Representation of each parameter

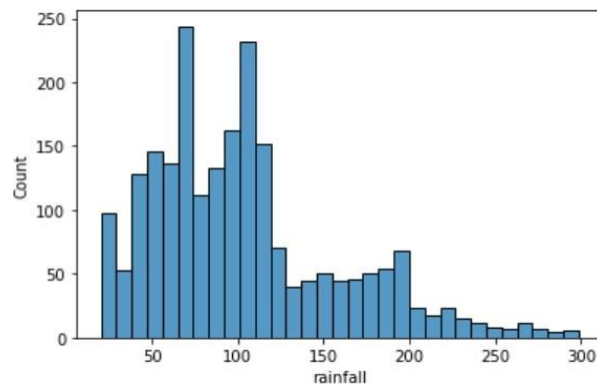
```
In [11]: all_columns = data.columns[:-1]

plt.figure(figsize=(15,13))
i = 1
for column in all_columns[:-1]:
    plt.subplot(3,3,i)
    sns.histplot(data[column])
    i+=1
plt.show()

sns.histplot(data[all_columns[-1]])
plt.show()
```

Fig 4.14 Bar Plot Of Each Parameter





4.5.1.3 K-Nearest Neighbours

K-Nearest Neighbor classifier is one of the introductory supervised classifiers, which every data science learner should be aware of. This algorithm was first used for a pattern classification task which was first used by Fix & Hodges in 1951. To be similar the name was given as KNN classifier. KNN aims for pattern recognition tasks. K-Nearest Neighbor also known as KNN is a supervised learning algorithm that can be used for regression as well as classification problems. Generally, it is used for classification problems in machine learning. KNN works on a principle assuming every data point falling in near to each other is falling in the same class. In other words, it classifies a new data point based on similarity.

Here we do not learn weights from training data to predict output (as in model-based algorithms) but use entire training instances to predict output for unseen data. Model is not learned using training data prior and the learning process is postponed to a time when prediction is requested on the new instance. In KNN, there is no predefined form of the mapping function.

```

In [18]: error_rate = []
        for i in range(1, 50):
            pipeline = make_pipeline(StandardScaler(), KNeighborsClassifier(n_neighbors = i))
            pipeline.fit(X_train, y_train)
            predictions = pipeline.predict(X_test)
            accuracy = accuracy_score(y_test, predictions)
            print(f"Accuracy at k = {i} is {accuracy}")
            error_rate.append(np.mean(predictions != y_test))

plt.figure(figsize=(10,6))
plt.plot(range(1,50),error_rate,color='blue', linestyle='dashed',
         marker='o',markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
print("Minimum error:-",min(error_rate),"at K =",error_rate.index(min(error_rate))+1)

```

Table 4.10 K-NN

```

Accuracy at k = 1 is 0.975
Accuracy at k = 2 is 0.9681818181818181
Accuracy at k = 3 is 0.975
Accuracy at k = 4 is 0.9795454545454545
Accuracy at k = 5 is 0.9772727272727273
Accuracy at k = 6 is 0.9681818181818181
Accuracy at k = 7 is 0.9704545454545455
Accuracy at k = 8 is 0.9704545454545455
Accuracy at k = 9 is 0.9659090909090909
Accuracy at k = 10 is 0.9613636363636363
Accuracy at k = 11 is 0.9590909090909091
Accuracy at k = 12 is 0.9568181818181818
Accuracy at k = 13 is 0.9590909090909091
Accuracy at k = 14 is 0.9590909090909091
Accuracy at k = 15 is 0.9568181818181818
Accuracy at k = 16 is 0.9545454545454546
Accuracy at k = 17 is 0.9545454545454546
Accuracy at k = 18 is 0.95
Accuracy at k = 19 is 0.9522727272727273
Accuracy at k = 20 is 0.9477272727272728
Accuracy at k = 21 is 0.9431818181818182
Accuracy at k = 22 is 0.9409090909090909
Accuracy at k = 23 is 0.9363636363636364
Accuracy at k = 24 is 0.9409090909090909
Accuracy at k = 25 is 0.9318181818181818
Accuracy at k = 26 is 0.9295454545454546
Accuracy at k = 27 is 0.9340909090909091

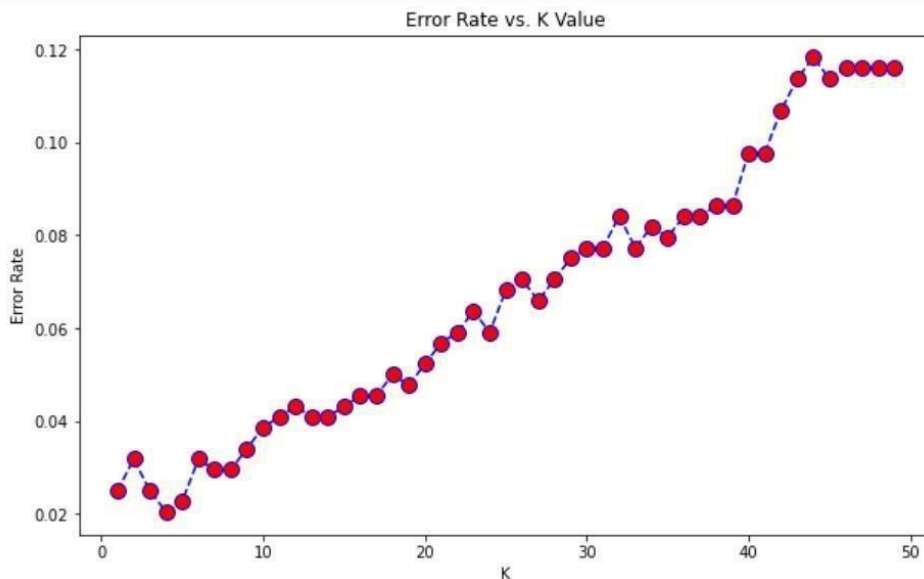
```

```

Accuracy at k = 28 is 0.9295454545454546
Accuracy at k = 29 is 0.925
Accuracy at k = 30 is 0.9227272727272727
Accuracy at k = 31 is 0.9227272727272727
Accuracy at k = 32 is 0.9159090909090909
Accuracy at k = 33 is 0.9227272727272727
Accuracy at k = 34 is 0.9181818181818182
Accuracy at k = 35 is 0.9204545454545454
Accuracy at k = 36 is 0.9159090909090909
Accuracy at k = 37 is 0.9159090909090909
Accuracy at k = 38 is 0.9136363636363637
Accuracy at k = 39 is 0.9136363636363637
Accuracy at k = 40 is 0.9022727272727272
Accuracy at k = 41 is 0.9022727272727272
Accuracy at k = 42 is 0.8931818181818182
Accuracy at k = 43 is 0.8863636363636364
Accuracy at k = 44 is 0.8818181818181818
Accuracy at k = 45 is 0.8863636363636364
Accuracy at k = 46 is 0.884090909090909
Accuracy at k = 47 is 0.884090909090909
Accuracy at k = 48 is 0.884090909090909
Accuracy at k = 49 is 0.884090909090909
Minimum error:- 0.02045454545454545 at K = 4

```

Fig 4.15 :- K-NN Error Rate



4.5.1.4 Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete dataset.

To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

- o Data Pre-processing step
- o Fitting Logistic Regression to the Training set
- o Predicting the test result

```
In [55]: from sklearn.linear_model import LogisticRegression
LogReg=LogisticRegression(random_state=2)
LogReg.fit(train_x,train_y)
predicted_values3=LogReg.predict(test_x)

In [56]: score3=accuracy_score(test_y,predicted_values3)
score3 = score3*100
acc.append(score3)
acc

Out[56]: [90.0, 99.0909090909091, 10.6818181818182, 95.22727272727273]

In [57]: model.append('Logistic Regression')

In [58]: print(classification_report(test_y,predicted_values3))
```

Fig 4.16 Logistic Regression

	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
grapes	1.00	1.00	1.00	18
jute	0.84	0.93	0.88	28
kidneybeans	1.00	1.00	1.00	14
lentil	0.88	1.00	0.94	23
maize	0.90	0.86	0.88	21
mango	0.96	1.00	0.98	26
mothbeans	0.84	0.84	0.84	19
mungbean	1.00	0.96	0.98	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	0.95	0.97	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	1.00	1.00	1.00	17
rice	0.85	0.69	0.76	16
watermelon	1.00	1.00	1.00	15
accuracy			0.95	440
macro avg	0.95	0.95	0.95	440
weighted avg	0.95	0.95	0.95	440

4.5.1.5 Decision Tree

Decision tree as the name suggests it is a flow like a tree structure that works on the principle of conditions. It is efficient and has strong algorithms used for predictive analysis. It has mainly attributes that include internal nodes, branches and a terminal node. Every internal node holds a “test” on an attribute, branches hold the conclusion of the test and every leaf node means the class label. This is the most used algorithm when it comes to Supervised learning techniques. There is no belief that is assumed by the decision tree that is an association between the independent and dependent variables. Decision tree is

a distribution-free algorithm. If decision trees are left unrestricted they can generate tree structures that are adapted to the training data which will result in overfitting. To avoid these things, we need to restrict it during the generation of trees that are called Regularization. The parameters of regularization are dependent on the DT algorithm used.

Fig 4.17 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)

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

Decision tree as the name suggests it is a flow like a tree structure that works on the principle of conditions. It is efficient and has strong algorithms used for predictive analysis. It has mainly attributes that include internal nodes, branches and a terminal node. Every internal node holds a “test” on an attribute, branches hold the conclusion of the test and every leaf node means the class label. This is the most used algorithm when it comes to Supervised learning techniques. There is no belief that is assumed by the decision tree.

DecisionTrees's Accuracy is: 0.9				
	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.59	1.00	0.74	16
chickpea	1.00	1.00	1.00	21
coconut	0.91	1.00	0.95	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	0.74	0.93	0.83	28
kidneybeans	0.00	0.00	0.00	14
lentil	0.68	1.00	0.81	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	0.00	0.00	0.00	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	0.84	0.91	19
pigeonpeas	0.62	1.00	0.77	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.62	0.77	16
watermelon	1.00	1.00	1.00	15
accuracy			0.90	440
macro avg	0.84	0.88	0.85	440
weighted avg	0.86	0.90	0.87	440

4.5.1.6 Naïve Bayes

It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive

Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. The name **naive** is used because it assumes the features that go into the model is independent of each other. That is changing the value of one feature, does not directly influence or change the value .

Again, scikit learn (python library) will help here to build a Naive Bayes model in Python. There are three types of Naive Bayes model under the scikit-learn library: **Gaussian**:-It is used in classification and it assumes that features follow a normal distribution.

Multinomial:-It is used for discrete counts. For example, let's say, we have a text classification problem. Here we can consider Bernoulli trials which is one step further and instead of “word occurring in the document”, we have “count how often word occurs in the document”, you can think of it as “number of times outcome number x_i is observed over the n trials”.

Bernoulli:-The binomial model is useful if your feature vectors are binary (i.e. zeros and ones). One application would be text classification with ‘bag of words’ model where the 1s & 0s are “word occurs in the document” and “word does not occur in the document” respectively.

Fig 4.18 NaiveBayes

```
In [60]: from sklearn.naive_bayes import GaussianNB
NaiveBayes=GaussianNB()
NaiveBayes.fit(train_x,train_y)
predicted_values1=NaiveBayes.predict(test_x)

In [62]: score1 = accuracy_score(test_y,predicted_values1)

In [63]: score1 = score1*100

In [64]: acc.append(score1)

In [65]: acc
Out[65]: [90.0, 99.0909090909091]

In [66]: model.append('Naive Bayes')

In [67]: print(classification_report(test_y,predicted_values1))
```

	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
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.88	1.00	0.93	28
kidneybeans	1.00	1.00	1.00	14
lentil	1.00	1.00	1.00	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	1.00	1.00	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.75	0.86	16
watermelon	1.00	1.00	1.00	15
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

4.5.1.7 Support Vector Machine

Support vector machines (SVMs) are powerful yet flexible supervised

machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables. An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane .

At first approximation what SVMs do is to find a separating line or hyperplane between data of two classes. SVM is an algorithm that takes the data as an input and outputs a line that separates those classes if possible. Support Vector Machine algorithm is prominent data analysis methodology and it is used for classification and regression techniques. Here the data points have been plotted using n-dimensional space with the value of particular characteristics as the value of a specific coordinate. The classification is done by finding the hyper-plane line that differentiate the classes separately.

Fig 4.19 SVM Accuracy

```
In [72]: from sklearn.svm import SVC  
SVM=SVC(gamma='auto')  
SVM.fit(train_x,train_y)  
predicted_values2=SVM.predict(test_x)
```

```
In [73]: score2=accuracy_score(test_y,predicted_values2)  
score2=score2*100  
acc.append(score2)
```

```
In [74]: acc
```

```
Out[74]: [90.0, 99.0909090909091, 10.6818181818182]
```

```
In [75]: print(classification_report(test_y,predicted_values2))
```

	precision	recall	f1-score	support
apple	1.00	0.23	0.38	13
banana	1.00	0.24	0.38	17
blackgram	1.00	0.19	0.32	16
chickpea	1.00	0.05	0.09	21
coconut	1.00	0.05	0.09	21
coffee	0.00	0.00	0.00	22
cotton	1.00	0.05	0.10	20
grapes	1.00	0.06	0.11	18
jute	1.00	0.07	0.13	28
kidneybeans	0.03	1.00	0.07	14
lentil	0.00	0.00	0.00	23
maize	0.00	0.00	0.00	21
mango	0.00	0.00	0.00	26
mothbeans	0.00	0.00	0.00	19
mungbean	1.00	0.12	0.22	24
muskmelon	1.00	0.30	0.47	23
orange	1.00	0.03	0.07	29
papaya	1.00	0.05	0.10	19
pigeonpeas	0.00	0.00	0.00	18
pomegranate	1.00	0.12	0.21	17
rice	0.50	0.06	0.11	16
watermelon	1.00	0.13	0.24	15
accuracy			0.11	440
macro avg	0.66	0.13	0.14	440
weighted avg	0.66	0.11	0.13	440

4.5.1.8 Random Forest

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Step 1 – First, start with the selection of random samples from a given dataset.

Step 2 – Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 – In this step, voting will be performed for every predicted result.

Step 4 – At last, select the most voted prediction result as the final prediction result.

One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Let's look at random forest in classification, since classification is sometimes considered the building block of machine learning. Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. Sklearn provides a great tool for this that measures a feature's importance by looking at how much the tree nodes that use that feature reduce impurity across all trees in the forest. One of the biggest advantages of random forest is its versatility. It can be used for both regression and classification tasks, and it's also easy to view the relative importance it assigns to the input features. Random forest is also a very handy algorithm because the default hyperparameters it uses often produce a good prediction result. Understanding the hyperparameters is pretty straightforward, and there's also not that many of them.

```

In [88]: from sklearn.ensemble import RandomForestClassifier

In [89]: RF = RandomForestClassifier(n_estimators=20,random_state=0)

In [90]: RF.fit(train_x,train_y)

Out[90]: RandomForestClassifier(n_estimators=20, random_state=0)

In [91]: predicted_values4 = RF.predict(test_x)

In [92]: score4 = accuracy_score(test_y,predicted_values4)
score4 = score4*100
acc.append(score4)
acc

Out[92]: [90.0,
          99.0909090909091,
          10.681818181818182,
          95.22727272727273,
          99.0909090909091]

```

Fig 4.20 Random Forest

```

In [93]: model.append('Random Forest Classifier')

In [94]: print(classification_report(test_y,predicted_values4))

```

	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
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	0.95	0.97	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.81	0.90	16
watermelon	1.00	1.00	1.00	15
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

4.6 Confusion Matrix:-

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the

form of a matrix, hence also known as an error matrix. Some features of Confusion matrix are given below:

4.6.1 K-Nearest Neighbours

```
In [15]: knn_pipeline = make_pipeline(StandardScaler(), KNeighborsClassifier(n_neighbors = 4))
knn_pipeline.fit(X_train, y_train)

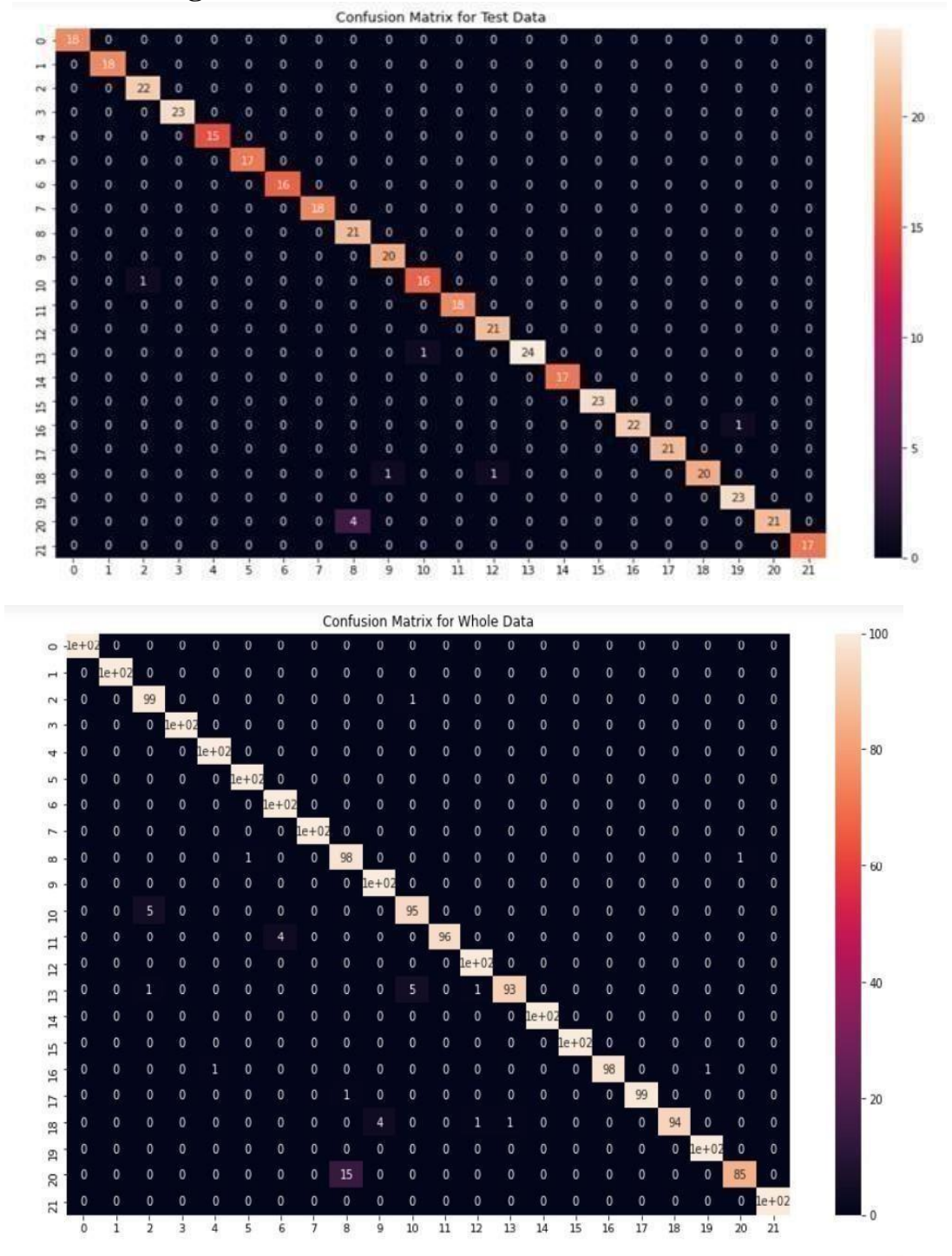
# Test Data Metrics
predictions = knn_pipeline.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
print(f"Accuracy on Test Data: {accuracy*100}%")
plt.figure(figsize = (15,9))
sns.heatmap(confusion_matrix(y_test, predictions), annot = True)
plt.title("Confusion Matrix for Test Data")
plt.show()

print()

# Whole Data Metrics
predictions = knn_pipeline.predict(X.values)
accuracy = accuracy_score(y, predictions)
print(f"Accuracy on Whole Data: {accuracy*100}%")
plt.figure(figsize = (15,9))
sns.heatmap(confusion_matrix(y, predictions), annot = True)
plt.title("Confusion Matrix for Whole Data")
plt.show()
```

Accuracy on Test Data: 97.95454545454545%

Fig 4.21 confusion matrix for K-NN



4.6.2 Decision Tree

```
In [30]: cm=confusion_matrix(Ytest,predicted_values)
```

Fig 4.22 confusion matrix for Decision Tree

```
In [47]: print(cm)
```

```
[[13  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 21  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 21  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 20  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 18  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  2  0  0  0 26  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0 11  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 23  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 21  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 26  0  0  0  0  0  0  0  0]
 [ 0  0  8  0  0  0  0  0  0  0 11  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 24  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 23  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  0  0]
 [ 0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0 16  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 18  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0]
 [ 0  0  0  0  0  0  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0 10  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 15]]
```

4.6.3 Random Forest

Fig 4.23 Confusion matrix for Random Forest

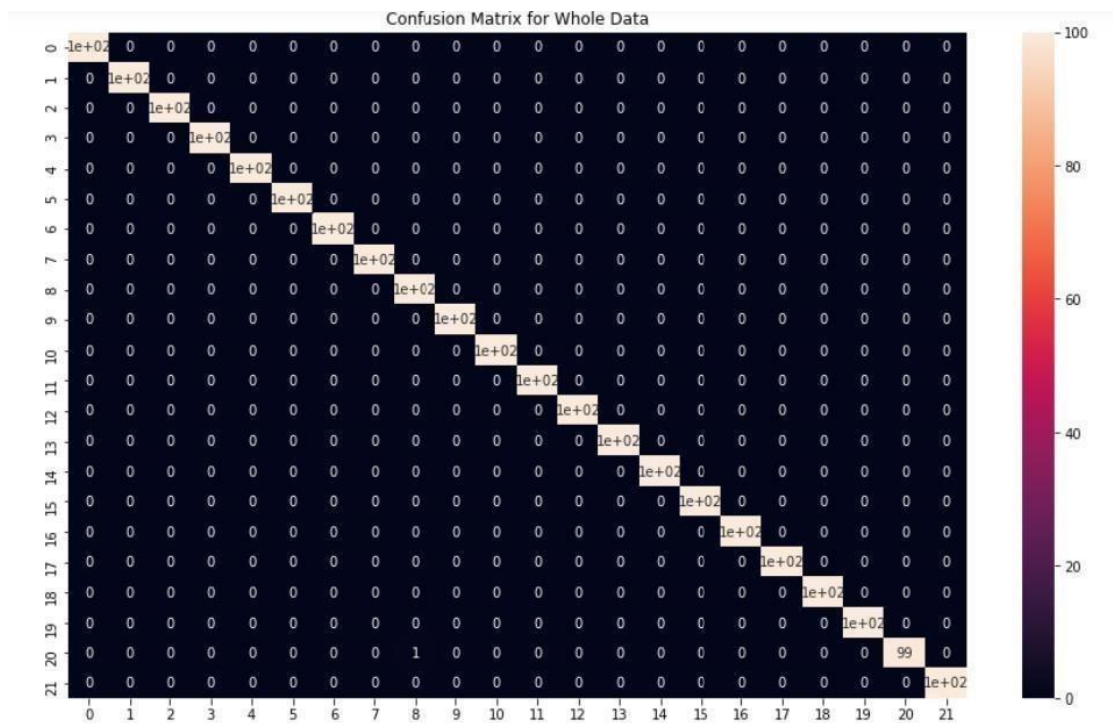
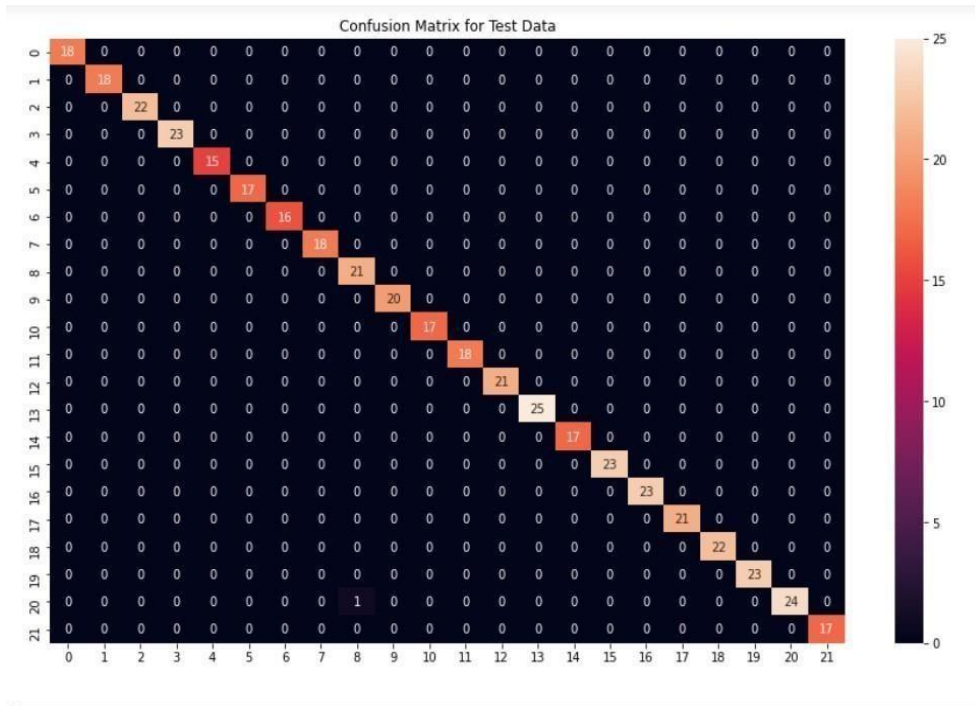
```
In [17]: rf_pipeline = make_pipeline(StandardScaler(), RandomForestClassifier(random_state = 18))
         rf_pipeline.fit(X_train, y_train)

         # Accuray On Test Data
         predictions = rf_pipeline.predict(X_test)
         accuracy = accuracy_score(y_test, predictions)
         print(f"Accuracy on Test Data: {accuracy*100}%")
         plt.figure(figsize = (15,9))
         sns.heatmap(confusion_matrix(y_test, predictions), annot = True)
         plt.title("Confusion Matrix for Test Data")
         plt.show()

         print()

         # Accuray On Whole Data
         predictions = rf_pipeline.predict(X.values)
         accuracy = accuracy_score(y, predictions)
         print(f"Accuracy on Whole Data: {accuracy*100}%")
         plt.figure(figsize = (15,9))
         sns.heatmap(confusion_matrix(y, predictions), annot = True)
         plt.title("Confusion Matrix for Whole Data")
         plt.show()
```

Accuracy on Test Data: 99.77272727272727%



4.6.4 Result Analysis

For computing the result analysis we have taken the result data that are calculated and analyzed at the stage of experimental results. By taking those results we have represented them in a accuracy percentage. For better and clear understanding of the results as well as it is user favorable it means user can easily understand them.

4.6.4.1 Analysis on various classification algorithms

The table 4.9 describes about the comparative analysis of various classification algorithms such as K-Nearest Neighbor, Support Vector Machine, and Logistic Regression, Decision Tree, Random Forest and Naïve Bayes are considered in terms of recall, accuracy rate, precision, F1-score and support. In classification when compared with all the other algorithms, the accuracy rate for Naïve Bayes and Decision tree are maximum. So, we observed that Naives Bayes is better than remaining all algorithms in terms of accuracy rate.

By analyzing these results only we came know accuracy. Based on these statistical measures we analyze that which algorithms is best classifier .

Table 4.11: Analysis of various classification algorithms for Agriculture data

Classification Learner	Accuracy
K-Nearest Neighbours	98.0%
Logistic Regression	95.22%

Decision Tree	90.0%
Support Vector Machine	10.68%
Random Forest	99.9%

CHAPTER 5 : CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this analysis, we used some of the common data mining techniques in the field of agriculture. Some of these techniques, such as the k nearest neighbor, SVM, Naïve Bayes, Decision Tree, Random Forest and logistic Regression are discussed and an application in agriculture for each of these techniques is presented. Data mining in agriculture is an upcoming research field. Efficient techniques can be developed and used for solving complex agricultural problems using data mining. Future enhancement of this agriculture analysis is to predict the crop yield using these techniques. It is useful for making crop decisions for farmers and government organizations. In future, the ANN and NN classification approach can be used for the better classification and improve the classification performance of the crop yield prediction. It could understanding of the high dimensional between complex yearly and seasonal climatic patterns which determine crop yield helps both farmers and other decision makers to be able to predict the effects of drought and other climatic conditions.

5.2 Future work

For future works, we are targeting to improve the Sustainability and reduce risks in Agriculture and Farming. Agriculture and Farming are one of the most important professions in the world. It plays an important role in the economic sector worldwide. Agriculture is a \$5 trillion industry. The future of farming has never been so bright so we are using machine learning and data mining techniques. By using these technologies can change the situation of farmers and decision making in agriculture field in a better way.

REFERENCES

- [1] B. M. Randles, I. V. Pasquetto, M. S. Golshan and C. L. Borgman, "Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study," 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL), 2017, pp. 1-2, doi: 10.1109/JCDL.2017.7991618.
- [2] Veenadhari S, Misra B, Singh CD. Data mining techniques for predicting crop productivity—A review article. In: IJCST. 2011; 2(1).
- [3] Jain A, Murty MN, Flynn PJ. Data clustering: a review. ACM Comput Surv. 1999;31(3):264–323.
- [4] Berkhin P. A survey of clustering data mining technique. In: Kogan J, Nicholas C, Teboulle M, editors. Grouping multidimensional data. Berlin: Springer; 2006. p. 25–72.
- [5] Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise.
- [6] Karthikeya, H. K., Sudarshan, K., & Shetty, D. S. Prediction of Agricultural Crops using KNN Algorithm.
- [7] Cunningham, S. J., & Holmes, G. (1999). Developing innovative applications in agriculture using data mining. In The proceedings of the Southeast Asia regional computer confederation conference (pp. 25-29).
- [8] Majumdar, J., Naraseeyappa, S., & Ankalaki, S. (2017). Analysis of agriculture data using data mining techniques: application of big data. Journal of Big data, 4(1), 1-15.
- [9] Yethiraj, N. G. (2012). Applying data mining techniques in the field of agriculture and allied sciences. International Journal of Business Intelligents ISSN, 2278-2400. [10]

- [10] Mucherino, A., Papajorgji, P., & Pardalos, P. M. (2009). A survey of data mining techniques applied to agriculture. *Operational Research*, 9(2), 121-140.
- [11] Mishra, S., Mishra, D., & Santra, G. H. (2016). Applications of machine learning techniques in agricultural crop production: a review paper. *Indian Journal of Science and Technology*, 9(38), 1-14.
- [12] Gandhi, N., & Armstrong, L. J. (2016, December). A review of the application of data mining techniques for decision making in agriculture. In 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I) (pp. 1-6). IEEE.
- [13] Palepu, R. B., & Muley, R. R. (2017). An analysis of agricultural soils by using data mining techniques. *Int. J. Eng. Sci. Comput*, 7(10).
- [14] Bharadi, V. A., Abhyankar, P. P., Patil, R. S., Patade, S. S., Nate, T. U., & Joshi, A. M. (2017). Analysis and prediction in agricultural data using data mining techniques. *International Journal of Research in Science and Engineering*, 386-393.
- [15] Hemageetha, N. (2016, March). A survey on application of data mining techniques to analyze the soil for agricultural purpose. In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 3112-3117). IEEE.
- [16] Patel, H., & Patel, D. (2014). A brief survey of data mining techniques applied to agricultural data. *International Journal of Computer Applications*, 95(9).

- [17] Ramesh, D., & Vardhan, B. V. (2015). Analysis of crop yield prediction using data mining techniques. *International Journal of research in engineering and technology*, 4(1), 47-473.
- [18] Lata, K., & Chaudhari, B. (2019). Crop Yield Prediction Using Data Mining Techniques and Machine Learning Models for Decision Support System. *Journal of Emerging Technologies and Innovative Research (JETIR)*.
- [19] Fetanat, H., Mortazavifarr, L., & Zarshenas, N. (2015). The analysis of agricultural data with regression data mining technique. *Ciência e Natura*, 37, 102-107.
- [20] Savla, A., Israni, N., Dhawan, P., Mandholia, A., Bhadada, H., & Bhardwaj, S. (2015, March). Survey of classification algorithms for formulating yield prediction accuracy in precision agriculture. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (pp. 1-7). IEEE.
- [21] Marinković, B., Crnobarac, J., Brdar, S., Antić, B., Jaćimović, G., & Crnojević, V. (2009, October). Data mining approach for predictive modeling of agricultural yield data. In *Proc. First Int Workshop on Sensing Technologies in Agriculture, Forestry and Environment (BioSense09)*, Novi Sad, Serbia (pp. 1-5).
- [22] Pudumalar, S., Ramanujam, E., Rajashree, R. H., Kavya, C., Kiruthika, T., & Nisha, J. (2017, January). Crop recommendation system for precision agriculture. In *2016 Eighth International Conference on Advanced Computing (ICoAC)* (pp. 32-36). IEEE.

- [23] Devika, B., & Ananthi, B. (2018). Analysis of crop yield prediction using data mining technique to predict annual yield of major crops. *International Research Journal of Engineering and Technology*, 5(12), 1460-1465.
- [24] Jambekar, S., Nema, S., & Saquib, Z. (2018, August). Prediction of Crop Production in India Using Data Mining Techniques. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)* (pp. 1-5). IEEE.
- [25] Rajshree, M., Arya, S., & Agarwal, R. P. (2011). Data Mining Techniques for Agriculture and Related Areas. *International Journal of Advanced Research in Computer Science*, 2(6), 43-45.
- [26] Kowshik, A., Kishor Gowda, H. K., Rithik Somesh, B. R., Yashas, S., Ramesh, B., & Nithyashree, R. Crop yield prediction based on Indian agriculture using machine learning.
- [27] Pravallika, A. S. K. S., & Saniya, K. R. (2020). A Model To Predict The Crop Based On Soil Composition Using Machine Learning Techniques (Doctoral dissertation, CMR Institute of Technology. Bangalore).

