A Mini Project Report On

**“Prediction of Soil Fertility Using Naïve Bayes Algorithm”**

*Submitted to JNTU HYDERABAD*

*In Partial Fulfillment of the requirements for the Award of Degree of*

### BACHELOR OF TECHNOLOGY IN

#### Artificial Intelligence and Data Science

Submitted By

**G.Vashesha (218R1A7223)**

**B.Divya Sri (218R1A7209)**

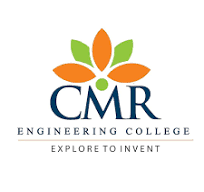
**N.Aravind (218R1A7247)**

**M.Uday Rao (218R1A7243)**

Under the Esteemed guidance of

#### Mrs.K.DURGA

Assistant Professor, Department of CSE(DS)



**Department of Artificial Intelligence and Data Science CMR ENGINEERING COLLEGE**

**UGC AUTONOMOUS**

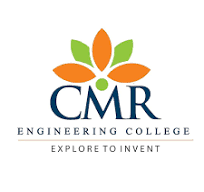
(Approved by AICTE, NEW DELHI, Affiliated to JNTU,Hyderabad) Kandlakoya, Medchal Road, Hyderabad-501 401.

**2024-2025**

# CMR ENGINEERING COLLEGE UGC AUTONOMOUS

*(Accredited by NBA,Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad) Kandlakoya, Medchal Road, Hyderabad-504 401*

**Department of Artificial Intelligence and Data Science**



**CERTIFICATE**

This is to certify that the project entitled **“Prediction of Soil Fertility Using Naïve Bayes Algorithm”** is a Bonafide work carried out by

**G.Vashesha (218R1A7223)**

**B.Divya Sri (218R1A7209)**

**N.Aravind (218R1A7247)**

**M.Uday rao (218R1A7243)**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE** from CMR

Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision.

The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

|  |  |  |
| --- | --- | --- |
| **Internal Guide** | **Mini Project Coordinator** | **Head of the Department** |
| **Mrs.K.Dhurga** | **Mr.R. Srikanth** | **Dr. M. Laxmaiah** |
| Assistant Professor | Assistant Professor | Professor & HOD |
| CSE(DS), CMREC | CSE(DS), CMREC | CSE(DS), CMREC |

**DECLARATION**

This is to certify that the work reported in the present project entitled **Prediction of Soil Fertility Using Naïve Bayes Algorithm** is a record of bonafide work done by us in the Department of Artificial Intelligence and Data Science, CMR Engineering College, JNTU Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

**G.Vashesha (218R1A7223)**

**B.Divya Sri (218R1A7209)**

**N.Aravind (218R1A7247)**

**M.Uday rao (218R1A7243)**

### ACKNOWLEDGEMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. M. Laxmaiah**, HOD, **Department of Artificial Intelligence and Data Science, CMR Engineering College** for their constant support**.**

We am extremely thankful to **Mrs.K.Durga,** Associate Professor, Internal Guide, Department of Artificial Intelligence and Data Science, for his/ her constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if I do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We thank **Mr.R.Srikanth** Mini Project Coordinator for his constant support in carrying out the project activities and reviews.

We express my thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, I am very much thankful to my parents who guided me for every step.

**G.Vashesha (218R1A7223)**

**B.Divya Sri (218R1A7209)**

**N.Aravind (218R1A7247)**

**M.Uday rao (218R1A7243)**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | **TOPIC** |  |
| **ABSTRACT** | | **i** |
| **1.** | **INTRODUCTION** | **1-2** |
| **2.** | **LITERATURE REVIEW** | **4-6** |
|  | 2.1 Existing System | 3 |
|  | 2.2 Proposed System | 4 |
|  | 2.3 Feasability Study | 5 |
|  | 2.4 Hadware Software Requirements | 6 |
| **3.** | **SYSTEM DESIGN** | **7-13** |
|  | 3.0 System Design | 7 |
|  | 3.1 UML diagram | 7 |
| **4.** | **IMPLEMENTATION** | **14-26** |
|  | 4.1 Naïve Bayes Classifier | 14 |
|  | 4.2 Data Mining | 21 |
| **5.** | **SOURCE CODE** | **27** |
| **6.** | **OUTPUT SCREEN** | **33-34** |
| **7.** | **TESTING** | **35-40** |
|  | 7.1 Types of Testing | 36 |
|  | 7.2 Testing Techniques | 39 |
| **8.** | **CONCLUSION** | **41** |
| **9.** | **REFERENCE** | **42** |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Name** | **Page .No.** |
| **1.** | Attribute Description of Soil Data | **23** |
| **2.** | Soil Attribute | **24** |
| **3.** | Chemical Composition of Soil | **25** |
| **4.** | Days to Germinate | **25** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Name** | **Page .No.** |
| **3.1.1** | Use Case Diagram | **8** |
| **3.1.2** | Class Diagram | **9** |
| **3.1.3** | Activity Diagram | **10** |
| **3.1.4** | Sequence Diagram | **11** |
| **3.1.5** | Collaboration Diagram | **12** |

**LIST OF OUTPUT SCREENS**

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Name** | **Page .No.** |
| **6.1** | Home Page | **33** |
| **6.2** | Upload Page | **33** |
| **6.3** | Result Page | **34** |

# ABSTRACT

Data mining involves the systematic analysis of large data sets, and datamining in agricultural soil datasets is exciting and modern research area. The productive capacity of a soil depends on soil fertility.

Achieving and maintaining appropriate levels of soil fertility, is of utmost importance if agricultural land is to remain capable of nourishing crop production .

In this project work, Steps for building a predictive model of soil fertility have been explained. Soil data is received in the form of data sheets and then these are stored for further processing.

A standard dataset is maintained for the purpose of comparison to assess the soil fertility. Data stored is then processed using the naïve bayes algorithm. The resultant data thus obtained is compared with the previously maintained standard dataset.

Based on the accuracy of data with respect to standard data is obtained and then the fertily of soil is concluded to fertile or infertile. This project aims at predicting soil fertility class using naïve bayes algorithm in data mining .

### INTRODUCTION

Data Mining is a very crucial research domain in recent research world. The techniques are useful to elicit significant and utilizable knowledge which can be perceived by many individuals. Data mining programs consists of diverse methodologies which are predominantly produced and used by commercial enterprises and biomedical researchers. These techniques are well disposed towards their respective knowledge domain. The use of standard statisticalanalysis techniques is both time consuming and expensive. Efficient techniques can be developed and tailored for solving complex soil data sets using data mining to improve the effectiveness and accuracy of the Classification of large soil data sets.

A soil test is the analysis of a soil sample to determine nutrient content, composition and other characteristics. Tests are usually performed to measure fertility and indicate deficiencies that need to be remedied. The soil testing laboratories are provided with suitable technical literature on various aspects of soil testing, including testing methods and formulations of fertilizer recommendations. It helps farmers to decide the extent of fertilizer and farm yard manure to be applied at various stages of the growth cycle of the crop. A soil test is the analysis of a soil sample to determine nutrient content, composition and other characteristics. Tests are usually performed to measurefertility and indicate deficiencies that need to be remedied.Soil fertility is a crucial attribute which is considered for land evaluation, also achieving and maintaining necessary levels of fertility isimportant for nurturing crop production, hence this paper includes steps for building an efficient and accurate predictive model of soil fertility with the help of data mining techniques.The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.

Data processing may be a comparatively young and knowledge domain field of computing, is that the method that tries to get patterns in massive knowledge sets. It utilizes strategies at the intersection of AI, machine learning, statistics, and info systems. The goal of the data mining method is to extract information from a knowledge set and rework it into an obvious structure for additional use. A soil take a look at is that the analysis of a soil sample to see nutrient content, composition and different characteristics. Tests area unit sometimes performed to live fertility and indicate deficiencies that require being remedied. During this analysis, soil dataset containing soil take a look at results has been accustomed apply numerous classification techniques in data processing. Soil fertility may be a crucial attribute that is taken into account

for land analysis , additionally achieving and maintaining necessary levels of fertility is vital for nurturing crop production, thus this paper includes steps for building associate degree economical and correct prognostic model of soil fertility with the assistance of J48 rule.

Soil testing plays a vital role in agricultural practices by analyzing soil samples to determine their nutrient content, composition, and other essential characteristics. These tests are crucial for measuring soil fertility and identifying deficiencies that must be addressed to ensure optimal crop growth. Soil testing laboratories are equipped with technical literature detailing various testing methods and fertilizer recommendations, which help farmers make informed decisions regarding the application of fertilizers and organic amendments throughout the crop growth cycle. Maintaining adequate soil fertility is essential for land evaluation and is fundamental to nurturing healthy crop production.

The primary objective of the data mining process is to extract valuable information from datasets and transform it into a comprehensible structure for further application. In the context of soil testing, this involves analyzing soil datasets to identify patterns and insights using classification techniques. The implementation of data mining in soil fertility analysis enables the development of predictive models that can guide agricultural practices effectively. For instance, using algorithms like J48 allows for the efficient classification of soil fertility levels, thereby facilitating better decision-making in fertilizer application.

Data processing, although a relatively young field, intersects various disciplines including artificial intelligence, machine learning, statistics, and information systems. The goal is to identify patterns within vast amounts of data, ultimately leading to improved understanding and application. In this study, soil datasets containing test results will be utilized to apply various classification techniques, specifically focusing on building an accurate and efficient predictive model of soil fertility. This approach not only aids in understanding soil health but also supports sustainable agricultural practices by ensuring that farmers can make informed decisions to optimize their yields.

By integrating data mining techniques into soil fertility analysis, we can enhance our understanding of soil health and empower farmers with the knowledge they need to maintain and improve crop production effectively.

### LITERATURE REVIEW

* 1. **EXISTING SYSTEM**

Soil fertility is taken into account be one among the essential attributes for deciding cropping pattern specially space. During this section, results of assorted call tree algorithms on dataset area unit shown. Supported these, the most effective classifier is chosen and any used for standardization its performance. The subsequent section explains call tree algorithms like J48 and Simple Cart briefly.

### J48 (C4.5)

J48 is associate degree open supply Java implementation of the C4.5 algorithmic data processing tool. C4.5 may be a program that makes a call tree supported a collection of labeled input file. This algorithmic rule was developed by Ross Quinlan. The choice trees generated by C4.5 will be used for classification, and for this reason, C4.5 is usually noted as a applied math classifier.

#### K – Nearest Neighbour

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non- parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

##### DISADVANTAGES OF EXISTING SYSTEM

* + 1. **Tree structure prone to sampling –** While Decision Trees are generally robust to outliers, due to their tendency to overfit, they are prone to sampling errors. If sampled training data is somewhat different than evaluation or scoring data, then Decision Trees tend not to produce great results.
    2. **Tree splitting is locally greedy –** At each level, tree looks for binary split such that impurity of tree is reduced by maximum amount. This is a greedy algorithm and achieves local optima. It may be possible, for example, to achieve less than maximum drop in impurity at current level, so as to achieve lowest possible impurity of final tree, but tree splitting algorithm cannot see far beyond the current level. This means that Decision Tree built is typically locally optimal and not globally optimal or best.
    3. **Optimal decision tree is NP-complete problem –** Because of number of feature variables, potential number of split points, and large depth of tree, total number of trees from same input dataset is unimaginably humongous. Thus, not only tree splitting is not global, computation of globally optimal tree is also practically impossible.

### PROPOSED SYSTEM

##### NAIVE-BAYES CLASSIFICATION ALGORITHM:

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems.

This Classification is named after Thomas Bayes ( 1702-1761), who proposed the Bayes Theorem.

Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data.

##### ADVANTAGES OF PROPOSED SYSTEM

* Very simple, easy to implement and fast.
* If the NB conditional independence assumption holds, then it will converge quicker than discriminative models like logistic regression.
* Even if the NB assumption doesn’t hold, it works great in practice.
* Need less training data.
* Highly scalable. It scales linearly with the number of predictors and data points.
* Can be used for both binary and mult-iclass classification problems.
* Can make probabilistic predictions.
* Handles continuous and discrete data.
* Not sensitive to irrelevant features.

### 2.3 FEASABILITY STUDY

###### Technical Feasibility

**Data Acquisition**: Collection of soil data from various sources (e.g., agricultural research institutions, government databases).

**Data Storage**: Utilizing databases or cloud storage solutions for data management. **Data Preprocessing**: Cleaning, normalizing, and transforming data to prepare it for analysis.

**Algorithm Implementation**: Using programming languages (Python, R) with libraries (e.g., scikit-learn) for the Naïve Bayes algorithm.

**Model Evaluation**: Implementing metrics (accuracy, precision, recall) to compare the model’s predictions against the standard dataset.

###### Economic Feasibility

**Cost Analysis**: Initial costs include software licenses, data acquisition, and computational resources. Potential savings through improved crop yield and reduced input costs.

**Funding Opportunities**: Grants from agricultural bodies or research institutions focused on sustainability and precision agriculture.

###### Operational Feasibility

**Skills Required**: Data scientists with expertise in data mining and machine learning. **Training and Resources**: Training for staff on data analysis tools and machine learning algorithms.

**User Acceptance**: Collaboration with agricultural experts to ensure model applicability in real-world scenarios.

### 2.4 HARDWARE AND SOFTWARE REQUIREMENTS

###### Hardware Requirements

* Processor **:** Intel i5
* RAM **:** 8 GB
* Storage **:** 256 GB
* Hard disk **:** 20 GB
* Network Requirements **:** Reliable internet connection

###### Software Requirements

* Operating System **:** Windows 10/11, macOS
* Programming Languages **:** Python
* Front end **:** HTML/ CSS
* BACK end **:** Python

### SYSTEM DESIGN

System Design is the core concept behind the design of any distributed systems. System design is defined as the process of creating an architecture for different components, interfaces, and modules of the system and providing corresponding data helpful in implementing such elements in the system.

The proposed system design for a soil testing application utilizing data mining techniques focuses on predicting soil fertility using the Naïve Bayes algorithm. The architecture consists of a front-end web application and a back-end server integrated with a database, ensuring efficient data management and user interaction. The front end will feature a user-friendly interface for inputting soil data, visualizing results, and providing insights into soil fertility classifications, such as fertile or infertile. The back end will handle data processing, model execution, and storage, employing the Naïve Bayes algorithm to analyze soil characteristics and predict fertility.

### UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form ,UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non- software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

### USE CASE DIAGRAM

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.



## Fig. 3.1.1 Use Case Diagram

### CLASS DIAGRAM

A static view of an application is depicted in the class diagram. It displays the properties, classes, functions, and relationships of the software system to provide an overview of the software system. It organizes class names, properties, and functions into a discrete compartment to aid in program development.

The following are the functions of class diagrams:

1. Define the main responsibilities of the system.
2. Serves as a basis for component diagrams and deployment. Use forward and reverse engineering.

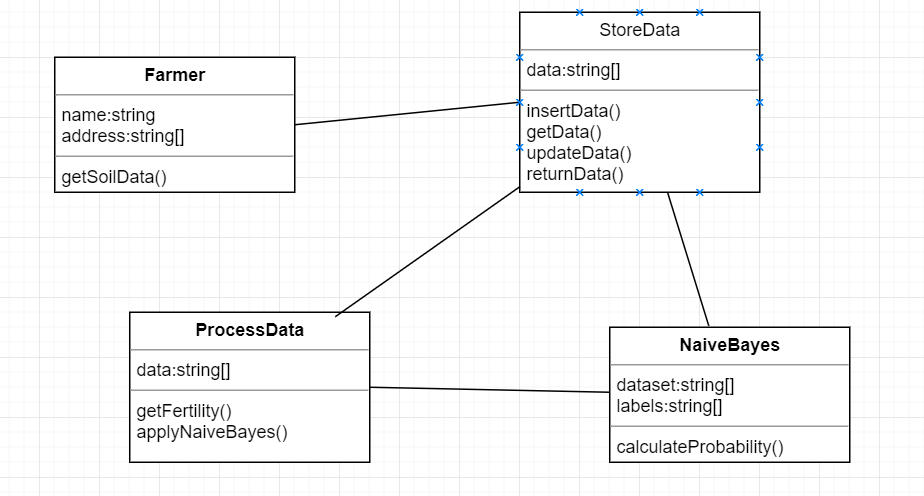


Fig. 3.1.2. Class Diagram

### 3.1.3. ACTIVITY DIAGRAM

An activity diagram is a kind of graphical representation that may be used to depict events visually. It is made up of a group of nodes that are linked to one another by means of edges. They are able to be connected to any other modelling element, which enables the behaviour of activities to be replicated using that methodology. Simulations of use cases, classes, and interfaces, as well as component collaborations and component interactions, are all made feasible with the help of this tool.

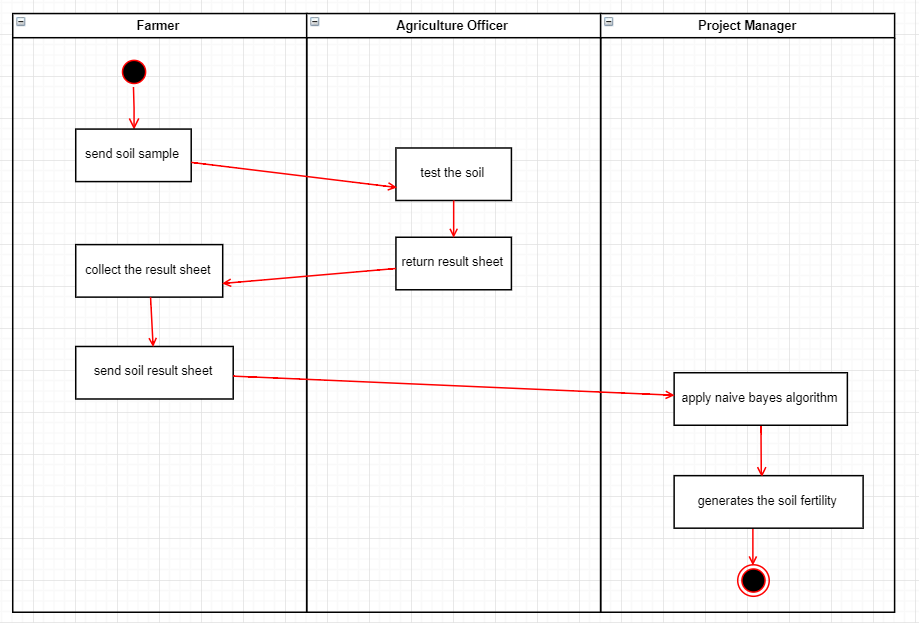


Fig. 3.1.3. Acitivity Diagram

### 3.1.4 .SEQUENCE DIAGRAM

The sequence diagram, also called the event diagram, describes the flow of messages in the system. It helps to visualize various dynamic parameters. He describes the communication between two rescue lines as a series of events arranged in time in which these rescue lines participated during the performance. The lifeline is represented by a vertical bar in UML while the message flow is represented by a vertical dotted line that crosses the bottom of the page. It includes both repetitions and branches

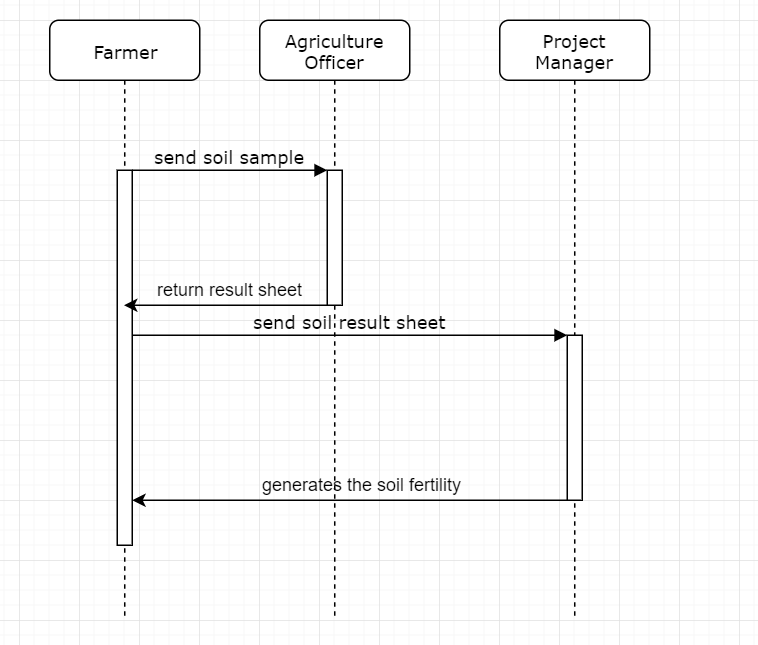


Fig .3.1.4 Sequence Diagram

### 3.1.5.COLLABORATION DIAGRAM

Collaboration diagrams are used in modeling the physical aspects of objectoriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Collaboration diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

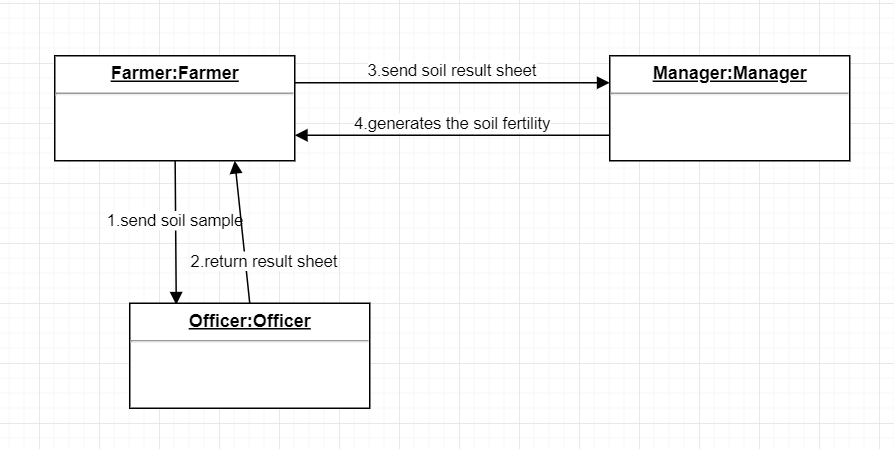
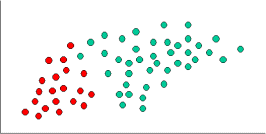


Fig. 3.1.5 Collaboration Diagram

1. **IMPLEMENTATION**
   1. **NAIVE BAYES CLASSIFICATION**
      1. **INTRODUCTION**

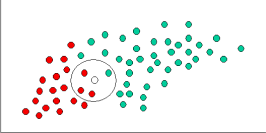
The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods.



To demonstrate the concept of Naïve Bayes Classification, consider the example displayed in the illustration above. As indicated, the objects can be classified as either GREEN or RED. Our task is to classify new cases as they arrive, i.e., decide to which class label they belong, based on the currently exiting objects.

Since there are twice as many GREEN objects as RED, it is reasonable to believe that a new case (which hasn't been observed yet) is twice as likely to have membership GREEN rather than RED. In the Bayesian analysis, this belief is known as the prior probability. Prior probabilities are based on previous experience, in this case the percentage of GREEN and RED objects, and often used to predict outcomes before they actually happen.

Since there is a total of 60 objects, 40 of which are GREEN and 20 RED, our prior probabilities for class membership are:



Having formulated our prior probability, we are now ready to classify a new object (WHITE circle). Since the objects are well clustered, it is reasonable to assume that the more GREEN (or RED) objects in the vicinity of X, the more likely that the new cases belong to that particular color. To measure this likelihood, we draw a circle around X which encompasses a number (to be chosen a priori) of points irrespective of their class labels. Then we calculate the number of points in the circle belonging to each class label. From this we calculate the likelihood.

Although the prior probabilities indicate that X may belong to GREEN (given that there are twice as many GREEN compared to RED) the likelihood indicates otherwise; that the class membership of X is RED (given that there are more RED objects in the vicinity of X than GREEN). In the Bayesian analysis, the final classification is produced by combining both sources of information, i.e., the prior and the likelihood, to form a posterior probability using the so-called Bayes' rule (named after Rev. Thomas Bayes 1702-1761).

Finally, we classify X as RED since its class membership achieves the largest posterior probability.

### USES OF NAIVE BAYES CLASSIFICATION:

###### Naive Bayes text classification

The Bayesian classification is used as a probabilistic learning method (Naive Bayes text classification). Naive Bayes classifiers are among the most successful known algorithms for learning to classify text documents.

###### Spam filtering

Spam filtering is the best known use of Naive Bayesian text classification. It makes use of a naive Bayes classifier to identify spam e-mail.

Bayesian spam filtering has become a popular mechanism to distinguish illegitimate spam email from legitimate email (sometimes called "ham" or "bacn").[4] Many modern mail clients implement Bayesian spam filtering. Users can also install separate email filtering programs. Server-side email filters, such as DSPAM, SpamAssassin, SpamBayes, Bogofilter and ASSP, make use of Bayesian spam filtering techniques, and the functionality is sometimes embedded within mail server software itself.

###### Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering

Recommender Systems apply machine learning and data mining techniques for filtering unseen information and can predict whether a user would like a given resource.

It is proposed a unique switching hybrid recommendation approach by combining a Naive Bayes classification approach with the collaborative filtering. Experimental results on two different data sets, show that the proposed algorithm is scalable and provide better performance–in terms of accuracy and coverage–than other algorithms while at the same time eliminates some recorded problems with the recommender systems.

###### Online applications

This online application has been set up as a simple example of supervised machine learning and affective computing. Using a training set of examples which reflect nice, nasty or neutral sentiments, we're training Ditto to distinguish between them.

Simple Emotion Modelling, combines a statistically based classifier with a dynamical model. The Naive Bayes classifier employs single words and word pairs as features. It allocates user utterances into nice, nasty and neutral classes, labelled +1, -1 and 0 respectively. This numerical output drives a simple first-order dynamical system, whose state represents the simulated emotional state of the experiment's personification, Ditto the donkey.

###### Example:

Suppose there are two events:

M: Manuela teaches the class (otherwise it’s Andrew)

S: It is sunny

“The sunshine levels do not depend on and do not influence who is teaching.”

###### Theory:

From P(S | M) = P(S), the rules of probability imply: P(~S | M) = P(~S)

P(M | S) = P(M)

P(M ^ S) = P(M) P(S)

P(~M ^ S) = P(~M) P(S)

P(M^~S) = P(M)P(~S)

P(~M^~S) = P(~M)P(~S)

###### Theory applied on previous example:

“The sunshine levels do not depend on and do not influence who is teaching.” can be specified very simply:

P(S | M) = P(S)

“Two events A and B are statistically independent if the probability of A is the same value when B occurs, when B does not occur or when nothing is known about the occurrence of B”

###### Conditional Probability Simple Example:

H = “Have a headache”

F = “Coming down with Flu” P(H) = 1/10

P(F) = 1/40 P(H|F) = 1/2

“Headaches are rare and flu is rarer, but if you’re coming down with ‘flu there’s a 50-50 chance you’ll have a headache.”

P(H|F) = Fraction of flu-inflicted worlds in which you have a headache =

#worlds with flu and headache Area of “H and F” region P(H ^ F) = ------------------

= =

#worlds with flu Area of “F” regionP(F)

###### Theory:

P(A|B) = Fraction of worlds in which B is true that also have A true P(A ^ B)

P(A|B) =

P(B)

Corollary:

P(A ^ B) = P(A|B) P(B)

P(A|B)+P( ¬A|B) = 1

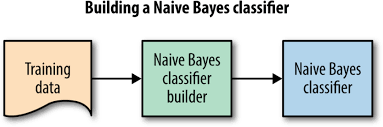
n

∑ P(A = Vk | B) = 1 K=1

##### BUILDING A NAÏVE BAYES CLASSIFIER

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers "based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the text retrieval community in the early 1960s,[1]:488 and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines.[2] It also finds application in automatic medical diagnosis.[3]



Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression,[1]:718 which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

In the statistics and computer science literature, naive Bayes models are known under a variety of names, including simple Bayes and independence Bayes.[4] All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method.

### TYPES OF NAÏVE BAYES

###### Gaussian naive Bayes

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contains a continuous attribute, x. We first segment the data by the class, and then compute the mean and variance of x in each class. Let Mk be the mean of the values in x associated with class Ck, and let sigma \_{k}^{2}} sigma \_{k}^{2}} be the variance of the values in x associated with class Ck. Suppose we have collected some observation value v.

p( x = v | Ck ) = (1 / sqrt(2 pi sigma ^ 2) ) e ^ constant

###### Multinomial naive Bayes

With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial (p1, …..pn) where pi is the probability that event i occurs (or K such multinomials in the multiclass case). A feature vector

{x} =(x\_{1},\dots ,x\_{n})} {x}}=(x\_{1},\dots ,x\_{n}) is then a histogram, with x\_{i}} x\_{i}

counting the number of times event i was observed in a particular instance.

This is the event model typically used for document classification, with events representing the occurrence of a word in a single document (see bag of words assumption). The likelihood of observing a histogram x is given by

p(x | Ck) = ( sigma i x i)! / ( pii x i !)

###### Bernoulli naive Bayes

In the multivariate Bernoulli event model, features are independent booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks,[9] where binary term occurrence features are used rather than term frequencies. If xi is a boolean expressing the occurrence or absence of the i'th term from the vocabulary, then the likelihood of a document given a class Ck is given by

p( x | Ck ) = pi ( p( 1 - pki ) ^ ( 1 – xi ))

###### Semi-supervised parameter estimation

Given a way to train a naive Bayes classifier from labeled data, it's possible to construct a semi-supervised training algorithm that can learn from a combination of labeled and unlabeled data by running the supervised learning algorithm in a loop:

Given a collection D = L U of labeled samples L and unlabeled samples U, start bytraining a naive Bayes classifier on L.

Until convergence, do:Predict class probabilities P( C | x ) for all examples x in D.

Re-train the model based on the probabilities (not the labels) predicted in the previous step.Convergence is determined based on improvement to the model likelihood

{\displaystyleP(D\mid \theta )} P(D\mid \theta ), where theta denotes the parameters of the naive Bayes model.

The algorithm is formally justified by the assumption that the data are generated by a mixture model, and the components of this mixture model are exactly the classes of the classification problem.

### DATA MINING IN AGRICULTURE

**4.2.1. INTRODUCTION**

Data mining in agriculture is a very current research topic. It consists in the application of data mining techniques to agriculture. Current technologies are nowadays able to provide a lot of information on agricultural-related activities, which can then be an examined in order to find important data. A related, but not equivalent term is precision agriculture.

###### Sorting apples by watercores

Before going to market, apples are examined and the ones showing some faults are removed. However, there are also invisible faults that can spoil the apple flavor and look. An example of invisible defect is the watercore.

This is an interior apple disorder that can affect the longevity of the fruit. Apples with slight or mild watercores are pleasant, but apples with medium to harsh degree of watercore cannot be stored for any length of time. Moreover, a few fruits with serious watercore could spoil a whole batch of apples. For this reason, a computational system is under study which takes X-raypictures of the fruit while they run on conveyor belts, and which is also able to inspect (by data mining techniques) the taken pictures and evaluate the probability that the fruit contains watercores.

###### Optimizing pesticide use by data mining

Current studies by agriculture researchers in Pakistan (one of the top four cotton producers of the world) showed that effort of cotton crop yield maximize through pro-pesticide state policies have led to a dangerously high insecticide use. These studies have reported a negative association between insecticide use and crop yield in Pakistan. Hence immoderate use (or abuse) of pesticides is harming the farmers with adverse financial, environmental and social impacts. By data mining the cotton Pest Scouting information along with the meteorological recordings it was shown that how pesticide uses can be optimized (reduced). Clustering of data revealed interesting patterns of farmer practices along with insecticide use dynamics and hence help identify the reasons for this insecticide abuse.

###### Explaining pesticide abuse by data mining

To observe cotton growth, various government departments and agencies in Pakistan have been recording pest scouting, agriculture and metrological information for decades. Bristly estimates of just the cotton pest scouting data recorded stands at around 1.5 million records, and growing. The initial agro-met data recorded has never been digitized, integrated or standardized to give a complete image, and hence cannot support conclusion making, thus requiring an Agriculture Data Warehouse[7]. Creating a novel Pilot Agriculture addition Data Warehouse come behind by analysis through querying and data mining some fascinating discoveries were made, such as pesticides sprayed at the wrong time, wrong pesticides used for the right reasons and secular relationship between pesticide usage and day of the week.

##### APPLICATION OF DATA MINING

There are number of studies which have been carried out on the application of data mining techniques for agricultural data sets. Naive Bayes Data Mining Technique is used to classify soils that analyze large soil profile experimental datasets. [4] Decision tree algorithm in data mining is used for predicting soil fertility. By using clusteringtechniques (Based on Partitioning Algorithms and Hierarchical algorithms) writer inspect the current usage and details of agriculture land disappeared in the past seven years. The overall aim of the study was to determine the land utilization for agriculture and non agriculture areas for the past ten years.

### 4.2.2 DATA MINING METHODOLOGIES

###### The application of k-means algorithm in the field of agriculture:

The k-means algorithm is used for soil grouping using GPS-based technologies. Classification of plant, soil, and residue regions of scrutiny by color images, grading apples before marketing, Monitoring water quality changes, Detecting weeds in accuracy agriculture, the prediction of wine fermentation problems can be performed by using a k-means approach. Knowing in promote that the wine fermentation process could get stuck or be slow can help the enologist to correct it and protect a good fermentation process [6].

###### The k-nearest neighbor application in the field of agriculture:

The k-nearest algorithm is used in imitating daily weather conditions and other weather variables and Estimating soil water parameters and Climate prediction.

###### The applications of neural networks in the field of agriculture:

The neural network is used in forecasting of flowering and maturity dates of soybean and in forecasting of water resources variables.

###### The applications of SVMs in the field of agriculture:

The implementation of support vector machine is the crop Classification and in the analysis of the climate change scenarios.

### 4.2.2 DATASET COLLECTION

Data set required for this analysis. These datasets contain varied attributes and their many values of soil samples taken from literature review. Dataset has ten attributes and a complete 1988instances of soil samples. Table one shows attribute description. The dataset has 9 attributes.

|  |  |
| --- | --- |
| Field | Description |
| Ph | pH value of soil |
| EC | Electrical conductivity, decisiemen per meter |
| OC | Organic Carbon, % |
| P | Phosphorous, ppm |
| K | Potassium, ppm |
| Fe | Iron, ppm |
| Zn | Zinc, ppm |
| Mn | Manganese, ppm |
| Cu | Copper, ppm |

Table 1 : Attribute Description of Soil Data

### 4.2.3 STORAGE, HANDLING, AND MOVING:

Soil chemistry changes over time, as biological and chemical processes break down or combine compounds over time. These processes change once the soil is removed from its natural ecosystem (flora and fauna that penetrate the sampled area) and environment (temperature, moisture, and solar light/radiation cycles). As a result, the chemical composition analysis accuracy can be improved if the soil is analyzed soon after its extraction ȯ usually within a relative time period of 24 hours. The chemical changes in the soil can be slowed during storage and transportation by freezing it. Air drying can also preserve the soil sample for many months.

##### SOIL DATA

The soil data used in this paper consists of 111 instanceswith 8 attributes like (i.e., Depth, Sand, Silt, Clay,Sandbysilt, Sandbyclay, Sandbysiltclay, TextureClass). Thetexture of the Soil data is varied from sand to siltyclayloam where as in sub-surface horizons it varied from sandto clay as in [2]. Table2.Shows the different soil attribute.

|  |  |
| --- | --- |
| **SYMBOL** | **DESCRIPTION** |
| S | Sand |
| Sicl | SiltyClay Loam |
| Sic | Silty Clay |
| C | Clay |
| Sl | Sandy loam |

Table 2.Soil Attribute

#### Chemical composition of soils

The DM content was highest in the negative control of sand (Nc) and lowest in the positive control of compost Pc (Table 3). The soils with high organic matter content were also high in nitrogen. Organic matter was highest in the positive control of compost (Pc) and lowest in Nc. Most of the soils were alkaline (pH >7) except Ll that had a pH of 5.8. Nitrogen content was highest in Pc (1.82 %) and lowest in Nc (0.05 %).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Types of soil** | **Dry matter, (%)** | **Organic matter,**  **(% in DM)** | **pH** | **Nitrogen,**  **(% in DM)** |
| Sand | 96.6 | 4.0 | 7.6 | 0.05 |
| Sub soil sandy | 74.1 | 28.5 | 7.8 | 0.11 |
| Sandy loam | 72.7 | 29.5 | 7.6 | 0.11 |
| Clay soil | 71.9 | 32.0 | 7.4 | 0.11 |
| Sub soil loam | 71.7 | 33.0 | 7.5 | 0.26 |
| Loam with compost | 64.0 | 42.0 | 7.4 | 1.32 |
| Loam with leaf | 61.8 | 52.0 | 5.8 | 0.63 |

Table 3: Chemical composition of soils

#### Germination

Maize germinated faster than rice (P<0.001) but there were no differences due to type of soil (P = 0.21) (Table 4).

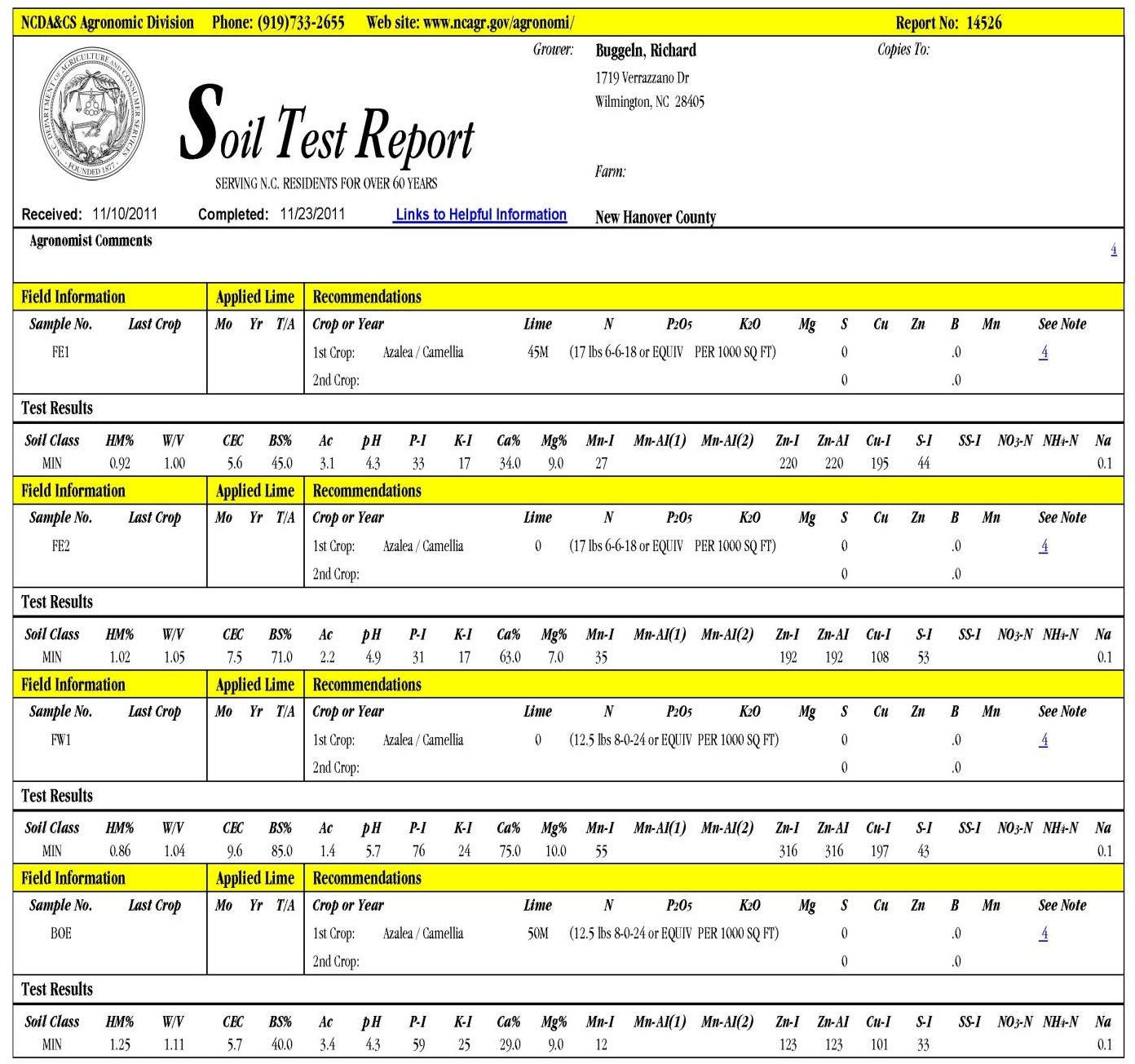
|  |  |  |  |
| --- | --- | --- | --- |
| **Types of soil** | **Maize** | **Rice** | **Average** |
| Sub soil sandy (Ss) | 4.7 | 5.3 | 5 |
| Sandy loam | 3.7 | 4.7 | 4.2 |
| Sand | 4 | 6.7 | 5.3 |
| Clay soil | 2.7 | 5.7 | 4.2 |
| Sub soil loam | 3.3 | 4.7 | 4 |
| Loam with leaf | 3.7 | 5 | 4.3 |

Table 4 : Days to germinate

#### Biomass production

There were differences in the responses of the two indicator plants. Highest biomass yield with maize was in the positive control of compost. In contrast, highest yield of rice was in the loam with leaf compost taken from under the trees (Ll). The lowest yield of maize was in the sub-soil taken from under the sandy loam (Ss); while the lowest yield of rice was in the negative control (Nc).

* 1. **SAMPLE SOIL TEST REPORT**



# SOURCE CODE

#### Code

import sys

import numpy as np import csv

from os.path import dirname, exists, expanduser, isdir, join, splitext import os

# Gaussian Naive Bayes from sklearn import datasets from sklearn import metrics

from sklearn.naive\_bayes import GaussianNB from sklearn.metrics import accuracy\_score def load\_data(cwd, data\_file\_name):

with open(join(cwd, 'upload', data\_file\_name)) as csv\_file: data\_file = csv.reader(csv\_file)

temp = next(data\_file) n\_samples = int(temp[0]) n\_features = int(temp[1]) target\_names = np.array(temp[2:])

data = np.empty((n\_samples, n\_features)) # print('==>',data)

target = np.empty((n\_samples,), dtype=np.int) for i, ir in enumerate(data\_file):

data[i] = np.asarray(ir[:-1], dtype=np.float64) target[i] = np.asarray(ir[-1], dtype=np.int)

return data, target, target\_names # command line arguments filename = sys.argv[1]

#current working directory cwd = os.getcwd()

# load the iris datasets

data, target, target\_names = load\_data(cwd, filename) # print(data,target,target\_names)

# fit a Naive Bayes model to the data model = GaussianNB() model.fit(data, target)

# make predictions

predicted = model.predict(data) # summarize the fit of the model

# print(metrics.classification\_report(target, predicted)) # print(metrics.confusion\_matrix(target, predicted)) score = accuracy\_score(target, predicted)

print(score)

#### Code

import { Component } from '@angular/core';

import { HttpClient } from '@angular/common/http'; @Component({

selector: 'app-root',

templateUrl: './app.component.html', styleUrls: ['./app.component.css']

})

export class AppComponent { selectedFile: File = null; percentage: number = 0; result: string = null;

constructor(private \_httpClient: HttpClient) {

}

onFileSelected(event) { this.selectedFile = event.target.files[0];

console.log(this.selectedFile);

}

onUpload() {

if (this.selectedFile) {

varfd = new FormData(); fd.append('rawFile', this.selectedFile);

this.\_httpClient.post('http://localhost:3000/upload', fd).subscribe(res => { console.log(res);

this.percentage = parseFloat(res.toString()) 100; if (this.percentage> 75) {

this.result = 'High';

} else if (this.percentage> 50) { this.result = 'Medium';

} else if (this.percentage> 25) { this.result = 'Low';

} else {

this.result = 'Very Low';

}

})

}else{

this.selectedFile = null; this.percentage = 0; this.result = null;

}

}

}

#### Node.js

varmyPythonScriptPath = 'naivebayes.py'; const express = require('express'); constmulter = require('multer');

var storage = multer.diskStorage({ destination: function (req, file, cb) { cb(null, './upload')

},

filename: function (req, file, cb) {

cb(null, Date.now() + '-' + file.originalname);

}

})

var upload = multer({ storage: storage

})

const fs = require('fs'); var input;

const router = express.Router(); const app = express();

router.post('/upload', upload.single('rawFile'), (req, res) => {

if (req.file) {

let file = req.file;

var spawn = require('child\_process').spawn;

var process = spawn('python', ["./naivebayes.py", file.filename]); process.stdout.on('data', (data) => {

console.log(`stdout: ${data}`); res.send(data.toString());

});

process.stderr.on('data', (data) => { console.log(`stderr: ${data}`); res.send(data.toString());

});

process.on('close', (code) => {

console.log(`child process exited with code ${code}`);

});

}

})module.exports = router;

# OUTPUT SCREENS

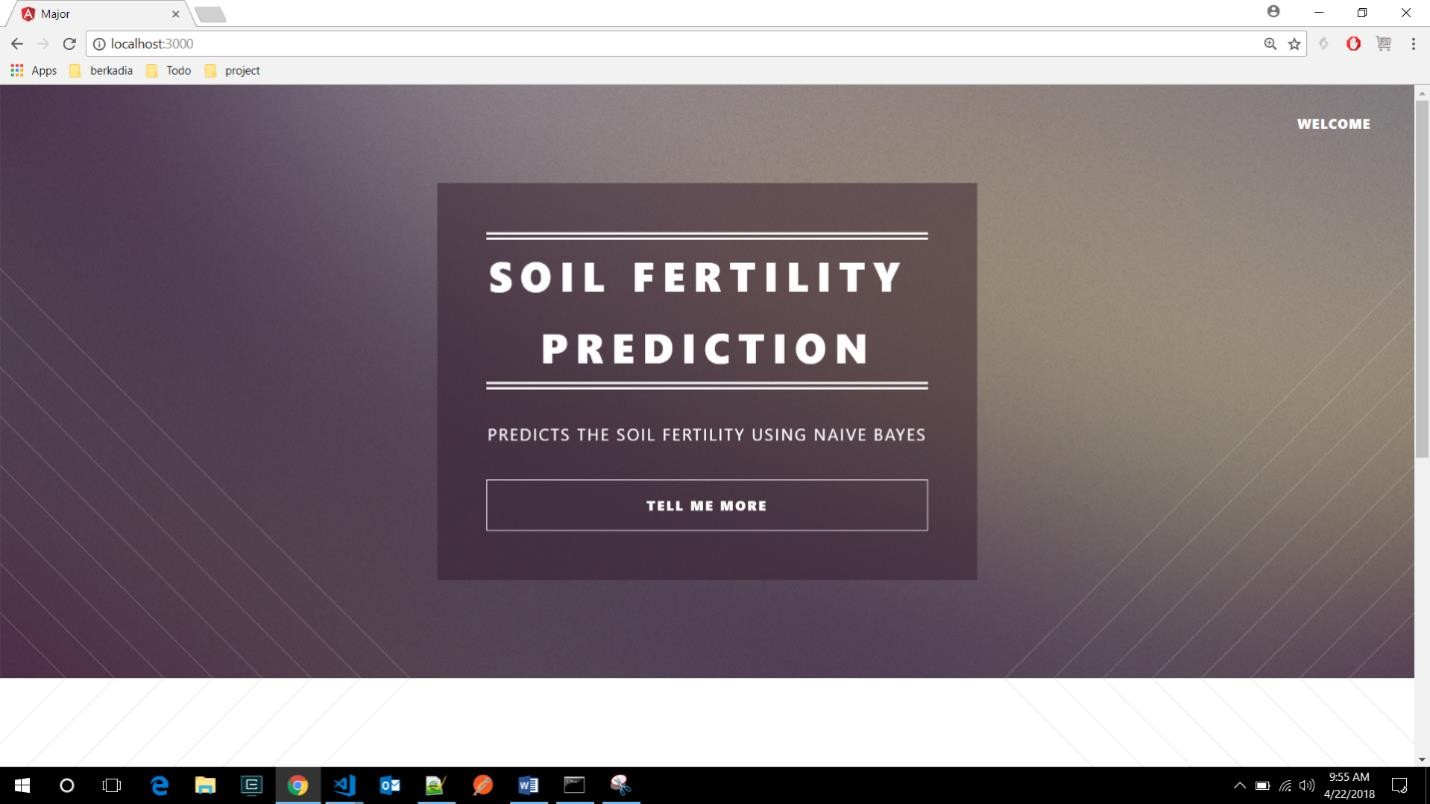


Fig 6.1 Home Page

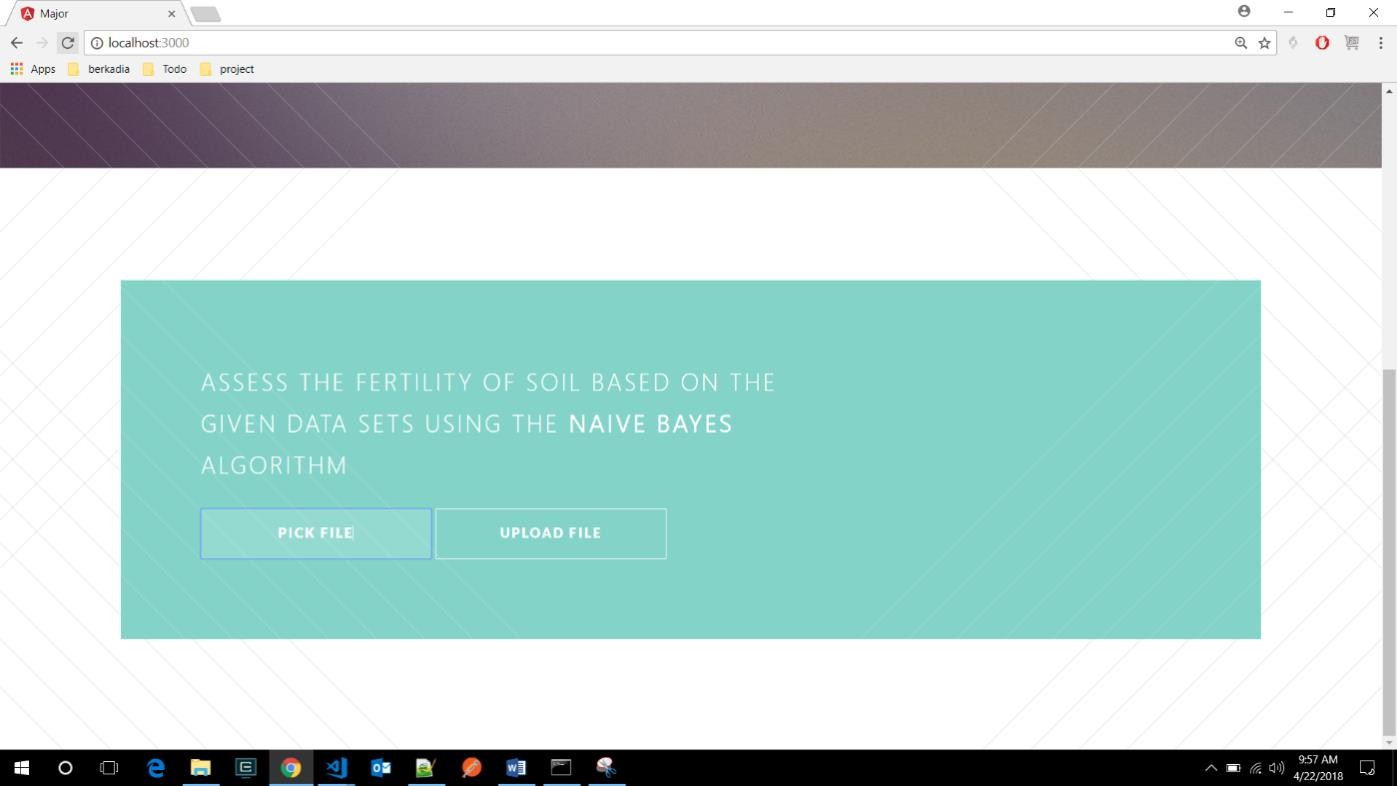
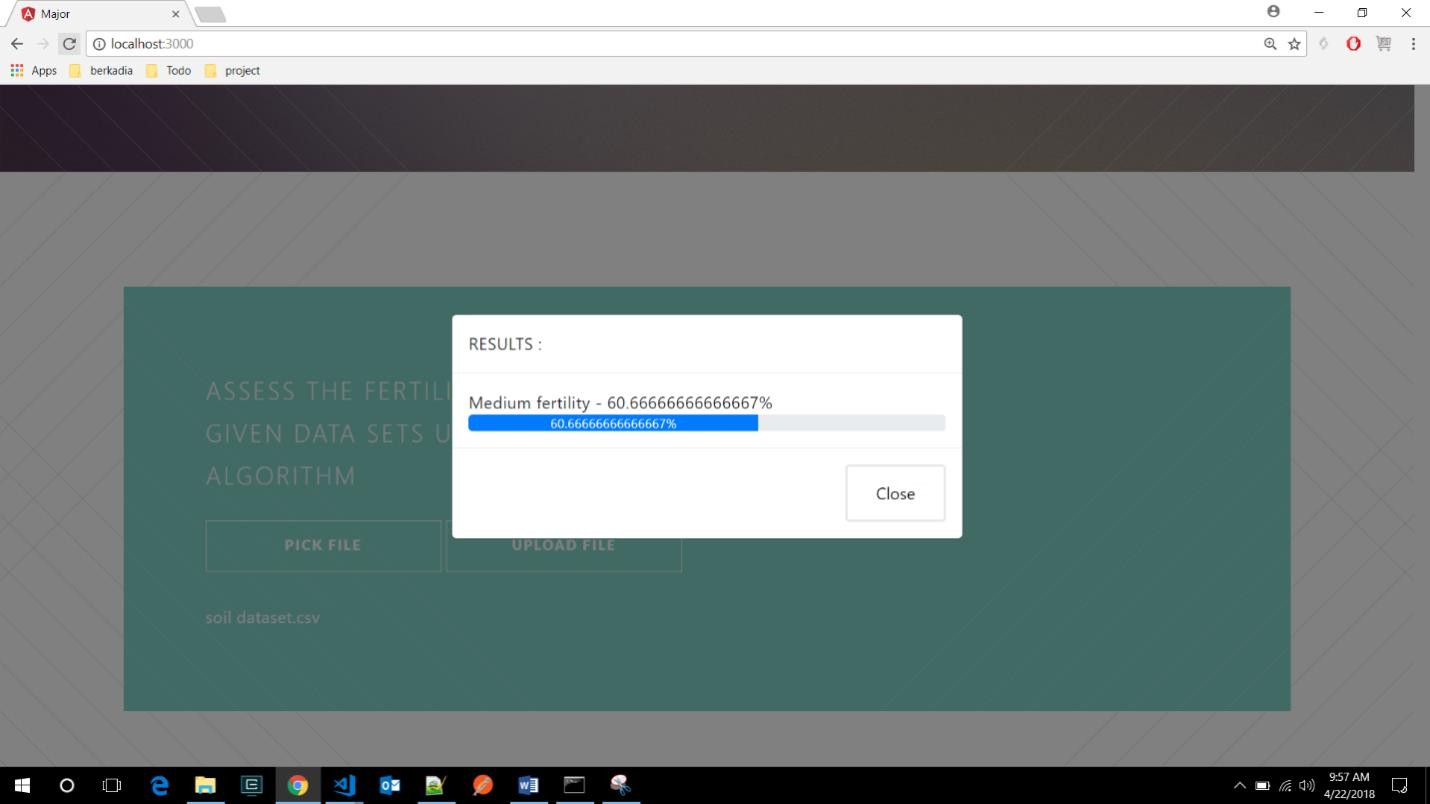


Fig 6.2 Upload Page



## Fig 6.3 Fertility page

### TESTING

Discovering and fixing such problems is what testing is all about. The purpose of testing is to find and correct any problems with the final product. It's a method for evaluating the quality of the operation of anything from a whole product to a single component. The goal of stress testing software is to verify that it retains its original functionality under extreme circumstances. There are several different tests from which to pick. Many tests are available since there is such a vast range of assessment options.

#### Who Performs the Testing:

All individuals who play an integral role in the software development process are responsible for performing the testing. Testing the software is the responsibility of a wide variety of specialists, including the End Users, Project Manager, Software Tester, and Software Developer.

#### When it is recommended that testing begin:

Testing the software is the initial step in the process. begins with the phase of requirement collecting, also known as the Planning phase, and ends with the stage known as the Deployment phase. In the waterfall model, the phase of testing is where testing is explicitly arranged and carried out. Testing in the incremental model is carried out at the conclusion of each increment or iteration, and the entire application is examined in the final test.

#### When it is appropriate to halt testing:

Testing the programme is an ongoing activity that will never end. Without first putting the software through its paces, it is impossible for anyone to guarantee that it is completely devoid of errors. Because the domain to which the input belongs is so expansive, we are unable to check every single input.

### TYPES OF TESTING

There are four types of testing:

#### Unit Testing

The term "unit testing" refers to a specific kind of software testing in which discrete elements of a program are investigated. The purpose of this testing is to ensure that the software operates as expected.

#### Test Cases

The unit testing for the soil testing application encompasses several critical areas to ensure robust functionality. First, data input validation is essential for maintaining data integrity. Test cases include verifying that valid soil parameter formats are correctly accepted, such as ensuring that pH values fall within the acceptable range (0 to 14). Additionally, the system should appropriately handle invalid inputs, such as negative nitrogen content, by returning clear error messages to the user.

Next, the prediction logic of the application requires thorough testing to confirm the accuracy of the Naïve Bayes model. One important test case involves providing known soil parameters and verifying that the model predicts the correct fertility class, such as determining that a soil sample with a pH of 6.5, 30 units of nitrogen, and 50 units of phosphorus is classified as "Fertile." Edge cases should also be tested, such as inputs at extreme pH levels, to ensure the model can handle unusual scenarios and still produce reliable predictions.

Lastly, database interactions must be tested to confirm that data retrieval and insertion functions perform as expected. For instance, a test case should verify that the application can successfully retrieve the correct standard dataset for nitrogen levels, ensuring that comparisons can be accurately made.

#### Integration Testing

The program is put through its paces in its final form, once all its parts have been combined, during the integration testing phase. At this phase, we look for places where interactions between components might cause problems.

#### Test Cases

Integration testing for the soil testing application focuses on verifying the interactions between different components, ensuring that they work together seamlessly to provide accurate predictions and results. One key area of testing involves the integration between the User Interface (UI) and the Web Server. Test cases in this category will check whether the UI correctly submits soil data to the Web Server and handles the responses appropriately. For instance, a test could validate that when a user inputs valid soil parameters, the Web Server processes the request and returns the expected fertility classification, which the UI then displays accurately.

Another critical aspect of integration testing is the interaction between the Web Server and the Database. Test cases should confirm that the Web Server can successfully retrieve standard datasets for comparison and correctly store new soil data entries in the database. For example, a test might simulate a request to retrieve nitrogen standards and verify that the data returned matches the expected values stored in the database. Additionally, testing should include scenarios where the Web Server attempts to insert new soil data, ensuring that the database processes the insertion correctly and reflects the new entry.

Finally, the integration between the Web Server and the Machine Learning Model must be rigorously tested. This includes validating that the Web Server can send processed soil data to the Naïve Bayes model and receive accurate predictions in return.

#### Functional Testing

One kind of software testing is called functional testing, and it involves comparing the system to the functional requirements and specifications. In order to test functions, their input must first be provided, and then the output must be examined. Functional testing verifies that an application successfully satisfies all of its requirements in the correct manner. This particular kind of testing is not concerned with the manner in which processing takes place; rather, it focuses on the outcomes of processing. Therefore, it endeavours to carry out the test cases, compare the outcomes, and validate the correctness of the results.

#### Test Cases

Functional testing for the soil testing application focuses on verifying that the system operates according to the specified requirements and performs its intended functions correctly. This type of testing includes several key areas, starting with user input functionality. Test cases will ensure that users can successfully enter soil parameters such as pH, nitrogen, and phosphorus levels through the User Interface (UI). For example, a test could involve inputting valid soil data and confirming that the system accepts the input without errors. Additionally, edge cases, such as entering boundary values (e.g., pH values of 0 and 14), should also be tested to ensure the system behaves as expected.

Another crucial area of functional testing is the prediction logic implemented by the Naïve Bayes algorithm. Test cases should evaluate whether the model produces accurate fertility classifications based on various soil parameter inputs. For instance, a test might involve providing a known set of soil parameters and checking that the predicted result matches the expected fertility class, such as determining whether a given sample is classified as "Fertile" or "Infertile." Moreover, the system should also handle invalid inputs gracefully, returning appropriate error messages to guide users.

### TESTING TECHNIQUES

There are many different techniques or methods for testing the software, including the following:

### BLACK BOX TESTING

During this kind of testing, the user does not have access to or knowledge of the internal structure or specifics of the data item being tested. In this method, test cases are generated or designed only based on the input and output values, and prior knowledge of either the design or the code is not necessary. The testers are just conscious of knowing about what is thought to be able to do, but they do not know how it is able to do it.

#### Test Cases

Black box testing for the soil testing application focuses on validating the system's functionality without examining its internal code or structure. This type of testing is primarily concerned with input-output behavior, ensuring that the application meets its requirements from an end-user perspective.

One key area of black box testing involves input validation. Test cases will be designed to provide various types of soil parameter inputs, including valid, invalid, and boundary values. For example, entering a valid pH value (e.g., 6.5) should yield an accepted input response, while entering an invalid value (e.g., - 1.0) should trigger an appropriate error message. Boundary tests will also be conducted by providing edge values, such as a pH of 0 or 14, to verify that the system handles these cases correctly.

Another focus area is the prediction functionality of the Naïve Bayes algorithm. Test cases will be created to input known soil parameter combinations and check the output for expected fertility classifications. For instance, providing a set of

inputs (e.g., pH: 6.0, Nitrogen: 25, Phosphorus: 45) should result in a predictable output, such as "Fertile." Additionally, scenarios with unexpected or malformed input, such as non-numeric values, will be tested to ensure the system handles these gracefully, ideally by returning informative error messages.

### WHITE BOX TESTING

During this kind of testing, the user is aware of the internal structure and details of the data item, or they have access to such information. In this process, test cases are constructed by referring to the code. Programming is extremely knowledgeable of the manner in which the application of knowledge is significant. White Box Testing is so called because, as we all know, in the tester's eyes it appears to be a white box, and on the inside, everyone can see clearly. This is how the testing got its name.

As an instance, a tester and a developer examine the code that is implemented in each field of a website, determine which inputs are acceptable and which are not, and then check the output to ensure it produces the desired result. In addition, the decision is reached by analyzing the code that is really used.

#### Test Cases

White box testing for the soil testing application involves examining the internal logic and structure of the code to ensure that all pathways, conditions, and loops function correctly. This type of testing focuses on verifying the implementation of algorithms, data flow, and control flow within the application.

One primary focus of white box testing is the Naïve Bayes algorithm implementation. Test cases will be created to check the accuracy of the algorithm's calculations, including the probability computations for different soil parameters. For example, tests will ensure that the algorithm correctly calculates the conditional probabilities and applies them to classify soil samples accurately.

### CONCLUSION

Continuous removal of nutrients from soil due to various agricultural practices, such as crop harvesting, can significantly impact soil fertility and overall agricultural productivity. In regions like Nigeria, where agriculture is a primary source of livelihood, maintaining soil health is crucial for ensuring food security and sustainable farming practices. To achieve this, farmers must understand the importance of replacing the nutrients that crops deplete over time. This process, known as fertilization, requires specific knowledge about soil health, crop nutrient needs, and the appropriate types of fertilizers to use.

Among the various methods available for assessing soil fertility, soil testing is the most straightforward and accessible for Nigerian farmers. Soil testing involves analyzing soil samples to determine nutrient levels and deficiencies, allowing farmers to make informed decisions about the types and amounts of fertilizers needed.. This combination allows for a more comprehensive understanding of both soil and plant health, facilitating targeted interventions that enhance agricultural productivity.

In today's data-driven agricultural landscape, the ability to analyze large volumes of data collected during crop production is essential. This data can include soil characteristics, weather patterns, crop yield statistics, and more. Utilizing advanced data analysis techniques helps farmers optimize their practices, leading to improved outcomes. Among these, decision tree algorithms are popular due to their ability to handle complex datasets and provide interpretable results.

By employing the Naïve Bayes classifier, farmers can analyze soil data efficiently and accurately predict soil fertility levels. This information empowers them to make data-driven decisions regarding fertilization, ultimately leading to enhanced crop yields, reduced input costs, and improved soil health. In summary, integrating soil testing with data analysis techniques like Naïve Bayes provides Nigerian farmers with valuable insights, enabling them to adopt more sustainable and productive agricultural practices in the face of evolving challenges.

### REFERENCE

1. Kumar & N. Kannathasan, (2011), “A Surveyon Data Mining and Pattern Recognition Techniques for Soil Data Mining “, IJCSIInternational Journal of Computer Science Issues,Vol. 8, Issue 3,
2. “Soil test”, Wikipedia, February 2012
3. L. Armstrong, D. Diepeveen& R. Maddern,(2004), **“**The application of data mining techniques to characterize agricultural soilprofiles”,
4. “Methods Manual-Soil Testing in India”,Department of Agriculture & CooperationMinistry of Agriculture Government of India,2011
5. “Naïve Bayes”, Wikipedia, February 2012
6. ”C4.5 (J48)”, Wikipedia, February 2012
7. W. Cohen, (1995),” Fast Effective RuleInduction, in Twelfth International Conference onMachine Learning.Witten& F. Eibe, (2005), “Data Mining:Practical Machine Learning Tools andTechniques” 2nd Edition, San Francisco: MorganKaufmann,
8. S. Cunningham & G. Holmes, (1999), ”Developing innovative applications in agriculture using data mining”, Department of Computer Science University of Waikato Hamilton, New Zealand, Technical Report.
9. R. Vamanan& K. Ramar, (2011), “Classification Of Agricultural Land Soils A Data Mining Approach”, International Journal on Computer Science and Engineering, ISSN: 0975-3397, Vol.3
10. Abdul-Raheem, A. M., T. M. Oyedele, and O. O. Obalum (2019). "Soil Fertility Management: A Review of Current Practices and Future Directions." Journal of Agriculture and Environment for International Development, 113(1), 75-91.
11. IITA (2015). "Soil Fertility and Nutrient Management." International Institute of Tropical Agriculture. Available at: [IITA Soil Fertility]
12. Lal, R. (2016). "Soil Health and Sustainability." Sustainability, 8(9), 902.
13. 4. López-Granados, F., M. P. González-Moro, and F. J. R. Zorrilla (2015). "Precision Agriculture for Sustainable Food Production." Sustainability, 7(10), 14482-14509.
14. 5. Mohan, S., B. S. S. Kumar, and T. V. S. Kumar (2020). "Predicting Soil Fertility Using Data Mining Techniques." International Journal of Agriculture and Biology,