**PREDICTION OF SOIL FERTILITY USING**

**NAÏVE BAYES ALGORITHM**

*A project work submitted to the*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD**

*In partial fulfillment of the requirements for the award of degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

*By*

**B. BALA KRISHNA 14881A0507**

**CH. SHRAVAN KUMAR 14881A0512**

**D. SRIVARDHAN 14881A0516**

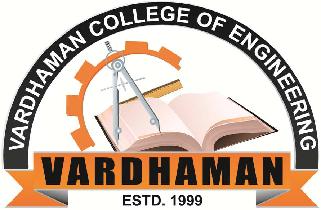
**E. MAHESHWAR 14881A0518**

**Under the Supervision**

**of**

**Dr. M. A. JABBAR**

Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VARDHAMAN COLLEGE OF ENGINEERING**

**(AUTONOMOUS)**

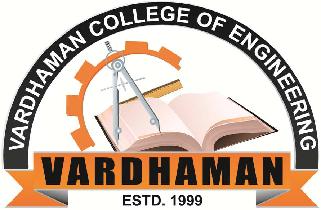
**Shamshabad - 501 218, Hyderabad, T.S.**

**April, 2018**

**VARDHAMAN COLLEGE OF ENGINEERING**

**(AUTONOMOUS)**

**Shamshabad - 501 218, Hyderabad, T.S.**

****

**Department of Computer Science and Engineering**

**CERTIFICATE**

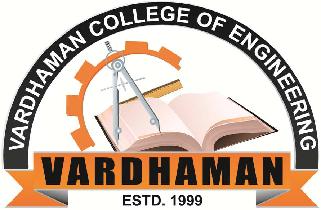
This is to certify that the work embodies in this dissertation entitled **‘*Prediction of Soil Fertility using Naïve Bayes Algorithm*’** being submitted by ‘**B. Bala Krishna 14881A0507, CH. Shravan Kumar 14881A0512, D. Srivardhan 14881A0516, E. Maheshwar 14881A0518’,** in partial fulfillment of the requirement for the award of **Bachelor of Technology** in **Computer Science and Engineering** discipline to Vardhaman College of Engineering(Autonomous), Shamshabad, Hyderabad (T.S) during the academic year 2017-18 is a record of bonafide work, undertaken by him/her the supervision of the undersigned**.**

|  |  |
| --- | --- |
| **Approved and Supervised by:** | **Head of the Department:** |
| **Dr. M. A. Jabbar** | **Dr. H. Venkateshwara Reddy** |
| Professor, | Professor & Head, |
| Dept of Computer Science &Engg, | Dept of Computer Science and Engg, |
| Vardhaman College of Engineering, | Vardhaman College of Engineering, |
| Hyderabad. | Hyderabad. |

Viva-Voce held on………..……………………………………

**­­­­­­­­­­**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Project Coordinator | Internal Examiner | External Examiner |



**VARDHAMAN COLLEGE OF ENGINEERING**

**(AUTONOMOUS)**

**Shamshabad - 501 218, Hyderabad, T.S.**

**Department of Computer Science and Engineering**

**DECLARATION**

**We** ‘**B. Bala Krishna, CH. Shravan Kumar, D. Srivardhan, E. Maheshwar’** , are students of ‘ **Bachelor of Technology in Computer Science and Engineering’, session: 2017 - 18,** Vardhaman College of Engineering, Shamshabad,Hyderabad (T.S.), hereby declare that the work presented in this Project Work entitled ‘***Prediction of Soil Fertility using Naïve Bayes Algorithm*’** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

**Signature of the students**

**B. Bala Krishna 14881A0507**

**CH. Shravan Kumar 14881A0512**

**D. Srivardhan 14881A0516**

**E. Maheshwar 14881A0518**

**Date:**

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We avail this opportunity to express our deep sense of gratitude and hearty thanks to **Dr. TeegalaVijender Reddy,** Chairman and **Sri TeegalaUpender Reddy,** Secretary of VCE, for providing congenial atmosphere and encouragement.

We show gratitude to **Dr. S. SaiSatyanarayana Reddy**, **Principal** for having provided all the facilities and support.

We would like to thank **Dr. H. Venkateshwara Reddy, Professor & Head**, **Computer Science and Engineering** for his expert guidance and encouragement at various levels of our Project.

We are thankful to our guide **Dr. M. A. Jabbar, Professor** for his sustained inspiring Guidance and cooperation throughout the process of this project. His wise counsel and suggestions were invaluable.

We would like to thank **Mr. A.Anil Kumar, Assistant professor,** Department of Computer Science and Engineering for his guidance and support, which helped us to complete this project.

We express our deep sense of gratitude and thanks to all the **Teaching** and **Non Teaching Staff** of our college who stood with us during the project and helped us tomake it a successful venture.

We place highest regards to our **Parents**, our **Friends** and **Well wishers** who helped a lot in making the report of this project.

**B. Bala Krishna 14881A0507**

**CH. Shravan Kumar 14881A0512**

**D. Srivardhan 14881A0516**

**E. Maheshwar 14881A0518**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **TOPIC** |  | **PAGE NO** |
| **ABSTRACT** |  | 8 |
| **CHAPTER - 1** | **INTRODUCTION** | **9** |
| 1.1 | DATA SET COLLECTION | 10 |
| **CHAPTER - 2** | **LITERATURE REVIEW** | **12** |
| 2.1 | EXISTING SYSTEM | 12 |
| 2.2 | DISADVANTAGES OF EXISTING SYSTEM | 13 |
| 2.3 | PROPOSED SYSTEM | 13 |
| 2.4 | ADVANTAGES OF PROPOSED SYSTEM | 14 |
| **CHAPTER – 3** | **NAÏVE BAYES ALGORITHM** | **15** |
| 3.1 | INTRODUCTORY OVERVIEW | 15 |
| 3.2 | USES OF NAÏVE BAYES | 17 |
| 3.3 | BUILDING A NAÏVE BAYES CLASSIFIER | 20 |
| 3.4 | TYPES OF NAÏVE BAYES | 24 |
| 3.5 | NAÏVE BAYES ALGORITHM IN PYTHON | 27 |
| 3.6  **CHAPTER - 4** | CONTEXT  **DATA MINING IN AGRICULTURE** | 35  **39** |
| 4.1 | INTRODUCTION | 39 |
| 4.2 | APPLICATIONS OF DATA MINING | 40 |
| 4.3 | DATA SET COLLECTION | 41 |
| 4.4 | SOIL TESTING | 43 |
| **CHAPTER-5** | **SOIL CLASSIFICATION** | **51** |
|  |  |  |
| 5.1 | SOIL CLASSIFICATION | 51 |
| 5.2  5.3 | A COMPARATIVE STUDY OF SOIL  CONTEXT OF SOIL CLASSIFICATION | 54  55 |

5.4 SOIL TESTING / SOIL ANALYSIS 55

5.5 METHODS OF SOIL TESTING 58

5.6 SAMPLE SOIL TEST REPORT 60

**LIST OF FIGURES**

BLOCK DIAGRAM 10

BUILDING A NAÏVE BAYES CLASSIFIER 22

**LIST OF TABLES**

ATTRIBUTE DESCRIPTION FOR SOIL DATA 11

SOIL ATTRIBUTE 52

CHEMICAL COMPOSITION OF SOILS 52

DAYS TO GERMINATE 53

**UML DIAGRAMS 61**

USE CASE DIAGRAM 61

CLASS DIAGRAM 62

ACTIVITY DIAGRAM 63

SEQUENCE DIAGRAM 64

COLLABORATION DIAGRAM 65

**RESULTS 66**

**CONCLUSION** 74

**BIBLIOGRAPHY** 75

**ABSTRACT**

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

Achieving and maintainingappropriate levels of soil fertility, is of utmostimportance if agricultural land isto remain capable of nourishing crop production .

In this project work, Steps forbuilding a predictive model of soil fertility havebeenexplained.Soil data is received in the form of datasheets and then these are stored forfurtherprocessing.

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

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

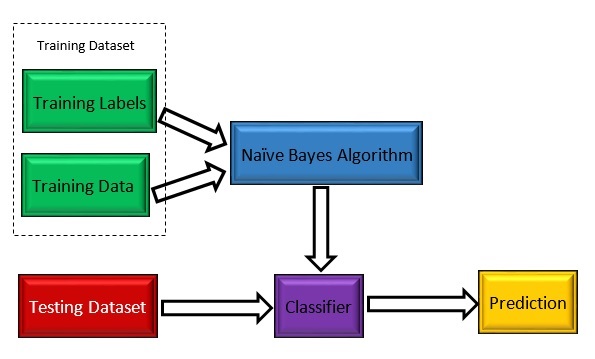
**CHAPTER 1-INTRODUCTION**

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 [1].

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 [2]. 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[4].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.

**BLOCK DIAGRAM OF NAÏVE BAYES ALGORITHM**



* 1. **DATA SET COLLECTION**

Data set needed for this analysis. These datasets contain Varied attributes and their several values of soil samples taken. Dataset has ten attributes and a complete 1988 instances of soil samples.

Table one shows attribute description.

Table1: Attribute Description for Soil Data

|  |  |
| --- | --- |
| Attribute | 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 |

**CHAPTER 2-LITERATURE REVIEW**

**2.1. 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.

**2.2. 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.

**2.3. PROPOSED SYSTEM**

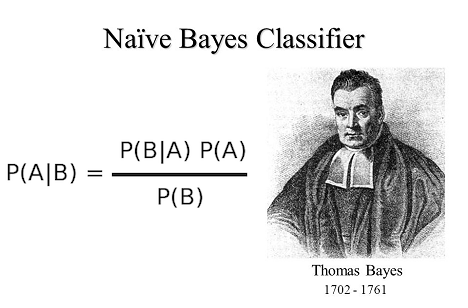
**Naive-Bayes Classification Algorithm::**

1. **Introduction to Bayesian Classification**

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.



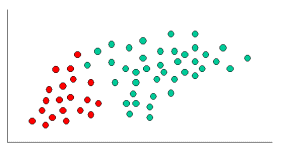
**2.4. ADVANTAGES OF PROPOSED SYSTEM**

* 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.

**CHAPTER 3 –NAÏVE BAYES ALGORITHM**

**3.1. Introductory Overview**

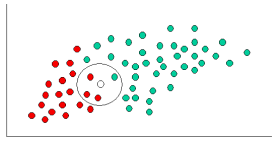
The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods.



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.

**Note.**The above probabilities are not normalized. However, this does not affect the classification outcome since their normalizing constants are the same.

**Technical Notes**

In the previous section, we provided an intuitive example for understanding classification using Naive Bayes. In this section are further details of the technical issues involved. Naive Bayes classifiers can handle an arbitrary number of independent variables whether continuous or categorical. Given a set of variables, X = {x1,x2,x...,xd}, we want to construct the posterior probability for the event Cj among a set of possible outcomes C = {c1,c2,c...,cd}. In a more familiar language, X is the predictors and C is the set of categorical levels present in the dependent variable.

Using Bayes' rulewhere p(Cj | x1,x2,x...,xd) is the posterior probability of class membership, i.e., the probability that X belongs to Cj. Since Naive Bayes assumes that the conditional probabilities of the independent variables are statistically independent we can decompose the likelihood to a product of terms.

Using Bayes' rule above, we label a new case X with a class level Cj that achieves the highest posterior probability.

Although the assumption that the predictor (independent) variables are independent is not always accurate, it does simplify the classification task dramatically, since it allows the class conditional densities p(xk | Cj) to be calculated separately for each variable, i.e., it reduces a multidimensional task to a number of one-dimensional ones. In effect, Naive Bayes reduces a high-dimensional density estimation task to a one-dimensional kernel density estimation. Furthermore, the assumption does not seem to greatly affect the posterior probabilities, especially in regions near decision boundaries, thus, leaving the classification task unaffected.

Naive Bayes can be modeled in several different ways including normal, lognormal, gamma and Poisson density functions:

**Note.** Poisson variables are regarded here as continuous since they are ordinal rather than truly categorical. For categorical variables, a discrete probability is used with values of the categorical level being proportional to their conditional frequency in the training data.

**3.2. Uses of Naive Bayes classification:**

1. **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.

1. **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.

1. **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.

1. **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.

**Independence**

**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

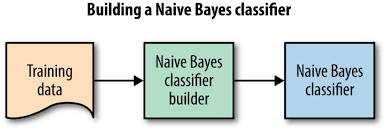
∑ P(A = Vk | B) = 1

K=1

**3.3. 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.

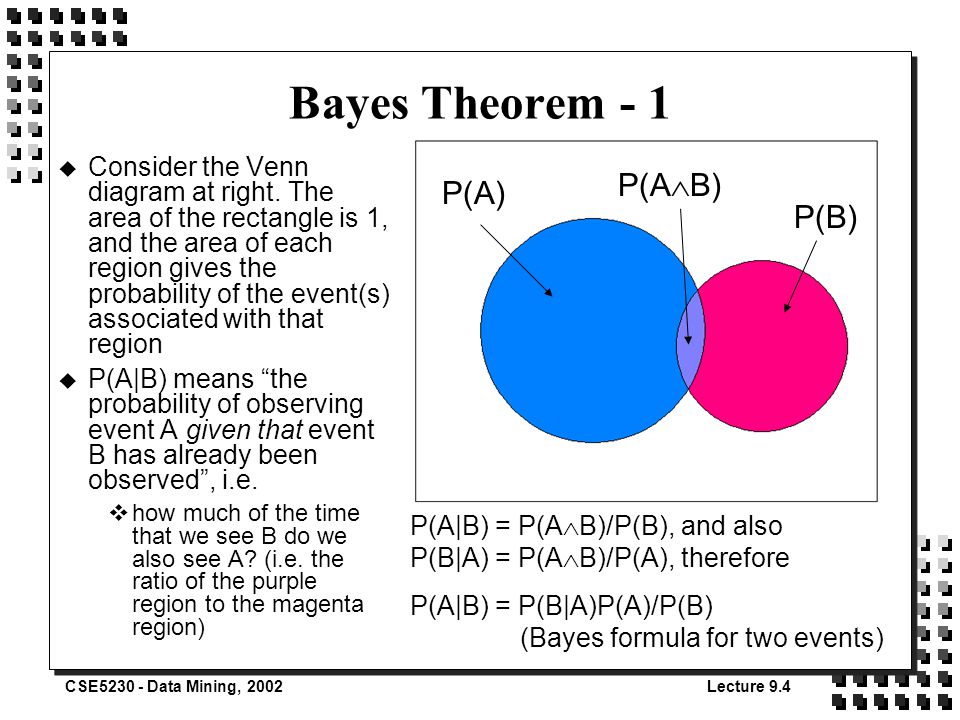
**Introduction**

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. It 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 example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

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.

Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers.[5] Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests.[6]

An advantage of naive Bayes is that it only requires a small number of training data to estimate the parameters necessary for classification.



**Probabilistic model**

Abstractly, naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector x =(x1,…..,xn) representing some n features (independent variables), it assigns to this instance probabilities

p(Ck x1……,xn)

for each of K possible outcomes or classes Ck.

The problem with the above formulation is that if the number of features n is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

p(Ck | x) = p(Ck) p(x | Ck) / p(x)

In plain English, using Bayesian probability terminology, the above equation can be written as

Posterior = (prior x likelihood) / evidence

In practice, there is interest only in the numerator of that fraction, because the denominator does not depend on C and the values of the xi are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

P(Ck, x1,…….,xn )

which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability.

**Constructing a classifier from the probability model**

The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule. The corresponding classifier, a Bayes classifier, is the function that assigns a class label y=Ck for some k as follows:

y = argmaxp(Ck ) II p( xi | Ck ).

**Parameter estimation and event models**

A class's prior may be calculated by assuming equiprobable classes (i.e., priors = 1 / (number of classes)), or by calculating an estimate for the class probability from the training set (i.e., (prior for a given class) = (number of samples in the class) / (total number of samples)). To estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set.

The assumptions on distributions of features are called the event model of the Naive Bayes classifier. For discrete features like the ones encountered in document classification (include spam filtering), multinomial and Bernoulli distributions are popular. These assumptions lead to two distinct models, which are often confused.

* 1. **TYPES OF NAÏVE BAYES**

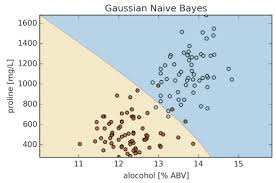
**Gaussian naive Bayes**

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.

When features xi are continuous valued, typically make the assumption they are normally distributed. Then, the probability distribution of v given a class Ck, can be computed by plugging v into the equation for a Normal distribution parameterized by \mu k sigma \_{k}^{2}} sigma \_{k}^{2}}.

That is,

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



Another common technique for handling continuous values is to use binning to discretize the feature values, to obtain a new set of Bernoulli-distributed features; some literature in fact suggests that this is necessary to apply naive Bayes, but it is not, and the discretization may throw away discriminative information.

**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 !)

If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero. This is problematic because it will wipe out all information in the other probabilities when they are multiplied. Therefore, it is often desirable

to incorporate a small-sample correction, called pseudocount, in all probability estimates such that no probability is ever set to be exactly zero. This way of regularizing naive Bayes is called Laplace smoothing when the pseudocount is one, and Lidstone smoothing in the general case.

Rennie et al. discuss problems with the multinomial assumption in the context of document classification and possible ways to alleviate those problems, including the use of tf–idf weights instead of raw term frequencies and document length normalization, to produce a naive Bayes classifier that is competitive with support vector machines.

**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 ))

Wherepki is the probability of class Ckgenerating the term xi. This event model is especially popular for classifying short texts. It has the benefit of explicitly modelling the absence of terms. Note that a naive Bayes classifier with a Bernoulli event model is not the same as a multinomial NB classifier with frequency counts truncated to one.

**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.

This training algorithm is an instance of the more general expectation–maximization algorithm (EM): the prediction step inside the loop is the E-step of EM, while the re-training of naive Bayes is the M-step. 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.

* 1. **NAÏVE BAYES ALGORITHM IN PYTHON**

This algorithm is broken down into the following steps:

1. Handle Data: Load the data from CSV file and split it into training and test datasets.
2. Summarize Data: summarize the properties in the training dataset so that we can calculate probabilities and make predictions.
3. Make a Prediction: Use the summaries of the dataset to generate a single prediction.
4. Make Predictions: Generate predictions given a test dataset and a summarized training dataset.
5. Evaluate Accuracy: Evaluate the accuracy of predictions made for a test dataset as the percentage correct out of all predictions made.
6. Tie it Together: Use all of the code elements to present a complete and standalone implementation of the Naive Bayes algorithm.

**1. Handle Data**

The first thing we need to do is load our data file. The data is in CSV format without a header line or any quotes. We can open the file with the open function and read the data lines using the reader function in the csv module.

We also need to convert the attributes that were loaded as strings into numbers that we can work with them. Below is the loadCsv() function for loading the Pima indians dataset.

**Eg:**

import csv

defloadCsv(filename):

lines = csv.reader(open(filename, "rb"))

dataset = list(lines)

fori in range(len(dataset)):

dataset[i] = [float(x) for x in dataset[i]]

return dataset

**Eg:**

import csv

defloadCsv(filename):

lines = csv.reader(open(filename, "rb"))

dataset = list(lines)

fori in range(len(dataset)):

dataset[i] = [float(x) for x in dataset[i]]

return dataset

We can test this function by loading the pimaindians dataset and printing the number of data instances that were loaded.

**Eg:**

filename = 'pima-indians-diabetes.data.csv'

dataset = loadCsv(filename)

print('Loaded data file {0} with {1} rows').format(filename, len(dataset))

**Eg:**

filename = 'pima-indians-diabetes.data.csv'

dataset = loadCsv(filename)

print('Loaded data file {0} with {1} rows').format(filename, len(dataset))

Running this test, you should see something like:

// Loaded data file pima-indians-diabetes.data.csv rows

Next we need to split the data into a training dataset that Naive Bayes can use to make predictions and a test dataset that we can use to evaluate the accuracy of the model. We need to split the data set randomly into train and datasets with a ratio of 67% train and 33% test (this is a common ratio for testing an algorithm on a dataset).

Below is the splitDataset() function that will split a given dataset into a given split ratio.

**Eg:**

import random

defsplitDataset(dataset, splitRatio):

trainSize = int(len(dataset) \* splitRatio)

trainSet = []

copy = list(dataset)

whilelen(trainSet) <trainSize:

index = random.randrange(len(copy))

trainSet.append(copy.pop(index))

return [trainSet, copy]

**Eg:**

import random

defsplitDataset(dataset, splitRatio):

trainSize = int(len(dataset) \* splitRatio)

trainSet = []

copy = list(dataset)

whilelen(trainSet) <trainSize:

index = random.randrange(len(copy))

trainSet.append(copy.pop(index))

return [trainSet, copy]

We can test this out by defining a mock dataset with 5 instances, split it into training and testing datasets and print them out to see which data instances ended up where.

**Eg:**

dataset = [[1], [2], [3], [4], [5]]

splitRatio = 0.67

train, test = splitDataset(dataset, splitRatio)

print('Split {0} rows into train with {1} and test with {2}').format(len(dataset), train, test)

**Eg:**

dataset = [[1], [2], [3], [4], [5]]

splitRatio = 0.67

train, test = splitDataset(dataset, splitRatio)

print('Split {0} rows into train with {1} and test with {2}').format(len(dataset), train, test)

Running this test, you should see something like:

Split 5 rows into train with [[4], [3], [5]] and test with [[1], [2]]

Split 5 rows into train with [[4], [3], [5]] and test with [[1], [2]]

**2. Summarize Data**

The naive bayes model is comprised of a summary of the data in the training dataset. This summary is then used when making predictions.

The summary of the training data collected involves the mean and the standard deviation for each attribute, by class value. For example, if there are two class values and 7 numerical attributes, then we need a mean and standard deviation for each attribute (7) and class value (2) combination, that is 14 attribute summaries.

These are required when making predictions to calculate the probability of specific attribute values belonging to each class value.

We can break the preparation of this summary data down into the following sub-tasks:

1. Separate Data By Class
2. Calculate Mean
3. Calculate Standard Deviation
4. Summarize Dataset
5. Summarize Attributes By Class

**Separate Data By Class**

The first task is to separate the training dataset instances by class value so that we can calculate statistics for each class. We can do that by creating a map of each class value to a list of instances that belong to that class and sort the entire dataset of instances into the appropriate lists.

**Calculate Mean**

We need to calculate the mean of each attribute for a class value. The mean is the central middle or central tendency of the data, and we will use it as the middle of our gaussian distribution when calculating probabilities.

We also need to calculate the standard deviation of each attribute for a class value. The standard deviation describes the variation of spread of the data, and we will use it to characterize the expected spread of each attribute in our Gaussian distribution when calculating probabilities.

The standard deviation is calculated as the square root of the variance. The variance is calculated as the average of the squared differences for each attribute value from the mean. Note we are using the N-1 method, which subtracts 1 from the number of attribute values when calculating the variance.

**Summarize Dataset**

Now we have the tools to summarize a dataset. For a given list of instances (for a class value) we can calculate the mean and the standard deviation for each attribute.

The zip function groups the values for each attribute across our data instances into their own lists so that we can compute the mean and standard deviation values for the attribute.

We can test this **summarize()** function with some test data that shows markedly different mean and standard deviation values for the first and second data attributes.

**Summarize Attributes By Class**

We can pull it all together by first separating our training dataset into instances grouped by class. Then calculate the summaries for each attribute.

We can test this **summarizeByClass()** function with a small test dataset.

**3. Make Prediction**

We are now ready to make predictions using the summaries prepared from our training data. Making predictions involves calculating the probability that a given data instance belongs to each class, then selecting the class with the largest probability as the prediction.

We can divide this part into the following tasks:

1. Calculate Gaussian Probability Density Function
2. Calculate Class Probabilities
3. Make a Prediction
4. Estimate Accuracy

**Calculate Gaussian Probability Density Function**

We can use a Gaussian function to estimate the probability of a given attribute value, given the known mean and standard deviation for the attribute estimated from the training data.

Given that the attribute summaries where prepared for each attribute and class value, the result is the conditional probability of a given attribute value given a class value.

See the references for the details of this equation for the Gaussian probability density function. In summary we are plugging our known details into the Gaussian (attribute value, mean and standard deviation) and reading off the likelihood that our attribute value belongs to the class.

In the **calculateProbability()** function we calculate the exponent first, then calculate the main division. This lets us fit the equation nicely on two lines.

**Calculate Class Probabilities**

Now that we can calculate the probability of an attribute belonging to a class, we can combine the probabilities of all of the attribute values for a data instance and come up with a probability of the entire data instance belonging to the class.

We combine probabilities together by multiplying them. In the calculateClassProbabilities() below, the probability of a given data instance is calculated by multiplying together the attribute probabilities for each class. the result is a map of class values to probabilities.

We can test the **calculateClassProbabilities()** function.

**Make a Prediction**

Now that can calculate the probability of a data instance belonging to each class value, we can look for the largest probability and return the associated class.

The **predict()** function belong does just that.

**4. Make Predictions**

Finally, we can estimate the accuracy of the model by making predictions for each data instance in our test dataset. The **getPredictions()** will do this and return a list of predictions for each test instance.

We can test the **getPredictions()** function.

**5. Get Accuracy**

The predictions can be compared to the class values in the test dataset and a classification accuracy can be calculated as an accuracy ratio between 0& and 100%. The **getAccuracy()** will calculate this accuracy ratio.

We can test the **getAccuracy()** function using the sample code below.

**6. Tie it Together**

Finally, we need to tie it all together.

**Implementation Extensions**

This section provides you with ideas for extensions that you could apply and investigate with the Python code you have implemented as part of this tutorial.

You have implemented your own version of Gaussian Naive Bayes in python from scratch.

You can extend the implementation further.

* **Calculate Class Probabilities**: Update the example to summarize the probabilities of a data instance belonging to each class as a ratio. This can be calculated as the probability of a data instance belonging to one class, divided by the sum of the probabilities of the data instance belonging to each class. For example an instance had the probability of 0.02 for class A and 0.001 for class B, the likelihood of the instance belonging to class A is (0.02/(0.02+0.001))\*100 which is about 95.23%.
* **Log Probabilities**: The conditional probabilities for each class given an attribute value are small. When they are multiplied together they result in very small values, which can lead to floating point underflow (numbers too small to represent in Python). A common fix for this is to combine the log of the probabilities together. Research and implement this improvement.
* **Nominal Attributes**: Update the implementation to support nominal attributes. This is much similar and the summary information you can collect for each attribute is the ratio of category values for each class. Dive into the references for more information.
* **Different Density Function** (*bernoulli* or *multinomial*): We have looked at Gaussian Naive Bayes, but you can also look at other distributions. Implement a different distribution such as multinomial, bernoulli or kernel naive bayes that make different assumptions about the distribution of attribute values and/or their relationship with the class value.
  1. **CONTEXT**

In data collection step we are collecting data from different web sites like news sites, blogs, social media, RSS feeds etc. The collected data is stored into database for further process. Since the collected data is unstructured data we use Mongo DB. Crime data is an unstructured data since the no of field, content, and size of the document can differ from one document to another the better option is to have a schema less database.

Also the absence of joins reduces the complexity. Other benefits of using an unstructured database is that:

 Large volumes of structured, semi-structured, and unstructured data.

 Object-oriented programming that is easy to use and flexible.

The advantage of NoSQL database over SQL database is that it allows insertion of data without a predefined schema. Unlike SQL database it not need to know what we are storing in advance, specify its size etc. For classification we are using an algorithm called Naïve Bayes which is a supervised learning method as well as a statistical method for classification. Naive Bayes classifier is a probabilistic classifier which when given an input gives a probability distribution of set of all classes rather than providing a single output. The algorithm classifies a news article into a crime type to which it fits the best. From classification what we get is "What is the probability that a crime document D belongs to a given class C?" [2].

The advantage of using Naive Bayes Classifier is that it is simple, and converges quicker than logistic regression. Compared to other algorithms like SVM (Support Vector Machine) which takes lot of memory the easiness for implementation and high performance makes it different from other algorithms. Also in case of SVM as size of training set increases the speed of execution decreases. Using Naive Bayes algorithm we create a model by training crime data related to vandalism, murder, robbery, burglary, sex abuse, gang rape, arson, armed robbery, highway robbery, snatching etc. By training means we have to teach them on particular inputs such that we can test them for unknown inputs. For testing the accuracy of the model we apply test data. Unlike SVM as the size of training data increases accuracy of test set also increases. Another advantage of Naïve Bayes is that it works well for small amount of training to calculate the classification parameters.

Also it fixes the Zero-frequency problem i.e. while estimating probability sometimes while checking a probability P(A) \* P(B/D) \* P(C/D) \* P(E/D) where P(C/D)=0. So the estimated probability result always give zero which leads to uncertainty in results. To avoid this condition we add +1 to the count of every zero value classes to achieve uniform distribution. Test results shows that Naive Bayes shows more than 90% accuracy since it categorise each words as tokens and removing frequent words like “the”, “and”, “of” etc which improves accuracy. A word is automatically terminated if it occurred fewer times or less than 3 times. Figure 2 shows a sample pseudo code of Naïve Bayes algorithm. We are also integrating the concept of Named Entity Recognition(NER) in the crime articles.

NER also known as Entity Extraction finds and classify elements in text into predefined categories such as the person names, organizations, locations, date, time etc[11]. So by using this concept in crime article we can get more details related to crime like victim and offender names, location of crime, date, time etc [6]. A sample result of NER is shown in Figure 3. Also related to crimes like burglary we can extract the list of weapons offender used while committing the crime. We have included a concept called Coreference Resolution to find the referenced entities in a text. In linguistics, Coreference occurs when two or more expressions in a text refer to the same person or thing i.e. if they have the same referent[12].

1. Input:

NAVI MUMBAI: The bike borne chain snatchers targeted two women pedestrians in Sanpada and Panvel on May 6, 2014, Tuesday and robbed their gold ornaments. While, 60-year-old woman’s gold chain worth Rs 20,000 was snatched by the bike’s pillion rider around 3.45 pm, while she was walking on the street near HDFC bank in sector-14, Sanpada, yet another woman from Khalapur was targeted by the pillion rider while she was walking along the road near old Thane naka in Panvel. The thief snatched away her gold necklace worth Rs 67,500. In both the incidents, robbery case under Section 392 and 34 has been registered at Turbhe and Panvel police stations respectively. For example :Seema said she would come i.e. here “she” refers to person “Seema”. Likewise we are extracting all referenced entities in a text. Below example shows the working of Coreference concept. A sample is shown below in Fig 4.

2) Input:

E.g.: A pillion bike rider snatched away a gold mangalsutra worth Rs 85,000 of a 60-year-old woman pedestrian in sector 19, Kharghar on Friday. The victim, ShakuntalaMande, was walking towards a vegetable outlet around 9.40am, when a bike came close to her and the pillion rider snatched her mangalsutra. A robbery case has been registered at Kharghar police station. Third phase is the pattern identification phase where we have to identify trends and patterns in crime. For finding crime pattern that occurs frequently we are using Apriori algorithm. Apriori can be used to determine association rules which highlight general trends in the database. The result of this phase is the crime pattern for a particular place. Here corresponding to each location we take the attributes of that place like VIP presence, weather attributes, area sensitivity, notable event, presence of criminal groups etc. After getting a general crime pattern for a place, when a new case arrives and if it follows the same crime pattern then we can say that the area has a chance for crime occurrence. Information regarding patterns helps police officials to facilitate resources in an effective manner. They can avoid crime occurrence by providing security/ patrolling in crime prone areas, fixing burglar alarms / CCTV etc.

Take a sample list of 100 news for a place and apply Apriori algorithm in it. It will mine the frequent crime patterns for a place. So if there is a pattern in which crime occurred then we assume that if again that pattern occurs in a place then there is probability for crime occurrence in that place. We are considering several attributes for crime pattern detection.

E.g.: For a place Meerut the pattern after mining will be: attribute 1, attribute 2, attribute 3, attribute 4  attribute 1, attribute 3, attribute 4, attribute 5 So the above will be the crime pattern for Meerut. So crime occurs only if the above patterns occur on a day. If any of these patterns occur then only we can say that there is probability for crime occurrence. For prediction we are using the decision tree concept. A decision tree is similar to a graph in which internal node represents test on an attribute, and each branch represents outcome of a test. The main advantage of using decision tree is that it is simple to understand and interpret. The other advantages include its robust nature and also it works well with large data sets. This feature helps the algorithms to make better decisions about variables [4]. Corresponding to each place we build a model. So for getting the crime prone areas we pass current date and current attributes into the prediction software. The result is shown using some visualization mechanisms.

**CHAPTER 4 - DATA MINING IN AGRICULTURE**

**4.1. 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.

**4.2.APPLICATION OF DATA MINING ALGORITHMS IN AGRICULTURE**

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.

**Data mining methodologies and its use in Agriculture domain Methodology**

**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.

**4.3. 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.

Table1 describes data collected for each soil sample.

Table 1 : Attribute Description of Soil Data

|  |  |
| --- | --- |
| 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 |

**Automated System**

Soil classification system is essential for theidentification of soil properties. Expert system can bea very powerful tool in identifying soils quickly andaccurately .Traditional classification systems includeuse of tables, flow-charts. This type of manualapproach takes a lot of time, hence quick, reliableautomated system for soil classification is needed tomake better utilization of technician's time [9].We propose an automated system that hasbeen developed for classifying soils based onfertility. Being rule-based system, it depends on facts, concepts, theories which are required for theimplementation of this system. Rules for soilclassification were collected from soil testing lab.

The soil sample instances were classified into thefertility class labels as: Very High, High, ModeratelyHigh, Moderate, Low, and Very Low. These classlabels for soil samples were obtained with the help ofthis system and they have been used further forcomparative study of classification algorithms.

In agriculture,a soil test commonly refers to the analysis of a soil sample to determine nutrient content, composition, and other characteristics such as the acidity or pH level. Asoil test can determine fertility, or the expected growth potential of the soil which indicates nutrient deficiencies, potential toxicities from excessive fertility and inhibitions from the presence of non-essential trace minerals. The test is used to mimic the function of roots to assimilate minerals. The expected rate of growth is modeled by the Law of the Maximum. Tap water or chemicals can change the composition of the soil, and may need to be tested separately. As soil nutrients vary with depth and soil components change with time, the depth and timing of a sample may also affect results. Composite sampling can be performed by combining soil from several locations prior to analysis. This is a common procedure, but should be used judiciously to avoid skewing results. This procedure must be done so that government sampling requirements are met. A reference map should be created to record the location and quantity of field samples in order to properly interpret test results.

**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.

**4.4. SOIL TESTING**

Without a soil analysis, it’s nearly impossible to determine what a soil needs in order to be productive. Laboratory soil analyses (soil tests) provide information on your soil’s available nutrient-supplying capacity. This information helps you select the correct kind and amount of fertilizer and liming material, which helps you develop and maintain more productive soil and increased crop production. Recommendations in this publication are based on the results of fertilizer experiments, soil surveys, and results obtained by farmers.

**Why should I collect a soil sample?**

Reasons for soil sampling include the following:

• Establish baseline soil nutrient status for new landowners

• Measure change in soil nutrient status over time

• Document soil nutrient management for certification requirements

• Determine nutrient application recommendations prior to planting

• Assess pH and the need for liming

• Avoid excessive nutrient applications or soluble salt accumulation

• Develop a plan for possible variable-rate fertilizing within a field

**When should I collect my soil sample?**

For perennial crops such as orchards, tree plantations, alfalfa, grass seed, and permanent pasture, the most important time to have the soil analyzed is before planting, so that necessary nutrients can be mixed into the soil. This analysis is especially important in acidic soils, which are likely to need liming. Apply lime and mix it with the soil several months before planting (for example, in the fall for spring planting), since it reacts slowly with the soil. Following establishment, then:

• For pastures and legumes, test soils every 3 years after planting.

• For Christmas trees, established fruit and nut trees, berries, and grapes, use annual foliar

tissue analysis instead of soiltesting. Soil samples are recommended every 3 to 5 years

or when the tissue analyses indicate a need.

Do periodic soil tests also for annual crops, particularly when you first cultivate a field or change crops or rotations. For annual crops, especially vegetables, test soil in the fall or winter or just before planting. If you plant successive crops in a single season, you don’t need to test before each planting. Soil samples are recommended every 2 to 3 years.

More information on soil laboratory analysis, soil test interpretation, and crop nutrient recommendations is available in other OSU Extension publications.

**Where should I collect a soil sample?**

The area in which to collect a soil sample may depend on the soil type, crops grown, management history, or all of these. The farm in Figure 1 has Figure 1. Collect a separate soil sample from each of the three areas (A, B, and C). A B C three separate sampling areas: A (orchard), B (pasture), and C (vegetable row crops). In this example, a separate soil sample should be collected from each of the three areas.

**How do I collect my soil sample?**

**Sample where the crop will be planted**

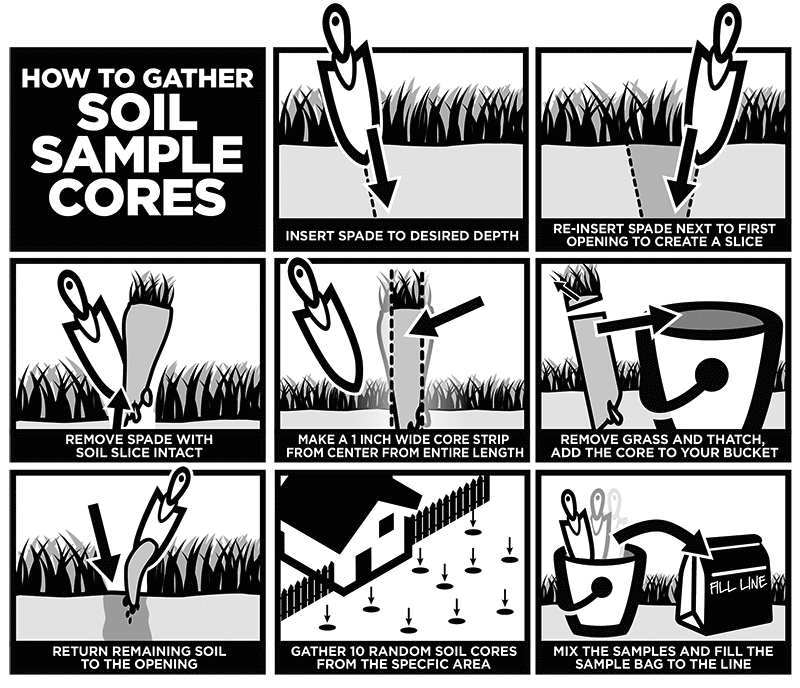
If you are using raised beds, such as for vegetable crops, take your samples in the beds, not in the areas between them.

**Avoid unusual areas**

Avoid sampling in small areas where you know that conditions are different from the rest of the field (for example, former manure piles, fertilizer bands, or fence lines). You often can spot these places by looking for plants growing especially well or very poorly.

**Take 15 to 20 subsamples**

Each sample should consist of subsamples taken from 15 to 20 locations within the sampling area.



**Avoid contaminating the sample**

• Use clean sampling tools (Figure 3), and avoid contaminating the sample during mixing

or packaging. A small amount of fertilizer residue on tools or hands, for instance, can

cause serious contamination of the soil sample.

• Do not include mulch or vegetation in the sample.

• Do not use galvanized metal, brass, or bronze tools to collect samples that will be tested for micronutrients (such as zinc).

**Take the soil sample to the correct depth**

Sample the part of the soil where the plant roots will grow. For most annual and perennial crops, sample from the surface down to about 6 to 8 inches (Figure 4) or to the depth of tillage. For pastures or soils that have limited or no tillage, refer to Evaluating Soil Nutrients and pH by Depth (EM 9014) for more information about collecting your soil sample.

Collect samples at the same depth. For example, if you take initial samples at a 6-inch depth, keep that same depth for all future samples, to get a more accurate comparison.



**Carefully mix the soil sample**

Place all of the soil subsamples from a single sampling area in a clean container and mix thoroughly (Figure 5, page 4). Do not worry about breaking the sample up into tiny particles. Labs have soil grinders to further mix the sample.

**Analyzing my soil sample**

• Find laboratories that perform soil analysis. To search for labs certified by the North

American Proficiency Testing (NAPT) program, go to [www.naptprogram.org](http://www.naptprogram.org)

• Look for a lab that offers a soil test report that you understand.



• Call one or more labs to find out the cost of the soil analysis you need.

• After choosing a lab, request any necessary paperwork (such as an information sheet), find out how you should prepare and submit the sample, and get the address where you should send the sample.

• Prepare and submit the sample according to the lab’s instructions. Plastic zipper bags work best; do not use a paper bag unless the lab provides one lined with plastic. Most laboratories ask you to label the sample bag with identifying information and to fill out and include an information sheet with the sample. Don’t forget to include payment in a separate, sealed plastic bag.

• If you are requesting a nitrate nitrogen (NO3-N) test, keep the sample cool and send it

immediately to the lab. Otherwise, you may choose to dry the sample or send it at your convenience.

• Request that the lab provide both a printed report and an electronic spreadsheet format for more flexible recordkeeping.

• Number each sample, record sample depth, and keep a record of the fields and areas you sampled. Take a photo of the labeled sample bags before mailing them, for future reference.

**What analysis should I request?**

• The standard soil analysis from most laboratories measures organic matter, phosphorus (P), potassium (K), calcium (Ca), magnesium (Mg), sodium (Na), and soil pH (acidity).

• For acidic soils, the SMP buffer test is the best way to determine how much lime is needed.

• Certain crops might have higher requirements for specific nutrients. Consult OSU Extension publications (see “Resources,” page 5) to determine whether you should test for nutrients such as sulfur (S), boron (B), or zinc (Zn).

• Nitrate nitrogen (NO3-N) is also commonly reported in standard soil tests. Nitrate Figure 5. Use a clean hand tool to mix the subsamples. nitrogen is not useful to determine soil fertilizer applications in western Oregon, as nitrate is readily leached from the soil profile. To determine a nitrogen application rate for your crop, consult the specific fertilizer guide (see “Interpreting your soil analysis,” page 5). Post-harvest testing for soil nitrate is used in some cropping systems to determine if excessive nitrogen was applied. In arid regions, such as eastern Oregon, soil nitrate nitrogen tests are used in conjunction with nutrient management guides to determine nitrogen applications.

**Sampling over time**

Once you have researched and selected a laboratory, plan to use the same lab for future tests to keep sample analysis consistent and detect changes in soil nutrients. Also, plan to take your soil sample at the same time of year, same depth, and same approximate field location.



**Interpreting your soil analysis**

Once you have received the analysis results for your soil, use the following tools to make decisions:

• Soil Test Interpretation Guide (EC 1478), 2011 version

• OSU Extension Fertilizer and/or Nutrient Management guides. To search for your crop-specific guide, go to the OSU Extension Catalog at http://extension.oregonstate.edu/catalog/ and search by keywords (nutrient management guide, fertilizer guide, and crop).

You can also consult your local OSU Extension Service agent.

Soil testing is often performed by commercial labs that offer a variety of tests, targeting groups of compounds and minerals. The advantage associated with local lab is that they are familiar with the chemistry of the soil in the area where the sample was taken. This enables technicians to recommend the tests that are most likely to reveal useful information.

Laboratory tests often check for plant nutrients in three categories:

•Major nutrients: nitrogen (N), phosphorus (P), and potassium (K)

•Secondary nutrients: sulfur, calcium, magnesium

•Minornutrients: iron, manganese, copper, zinc, boro n, molybdenum, chlorine

Do-it-yourself kits usually only test for the three "major nutrients", and for soil acidity or pH level. Do-it-yourself kits are often sold at farming cooperatives, university labs, private labs, and some hardware and gardening stores. Electrical meters that measure pH, water content, and sometimes nutrient content of the soil are also available at many hardware stores. Laboratory tests are more accurate than tests with do-ityourself kits and electrical meters[8].

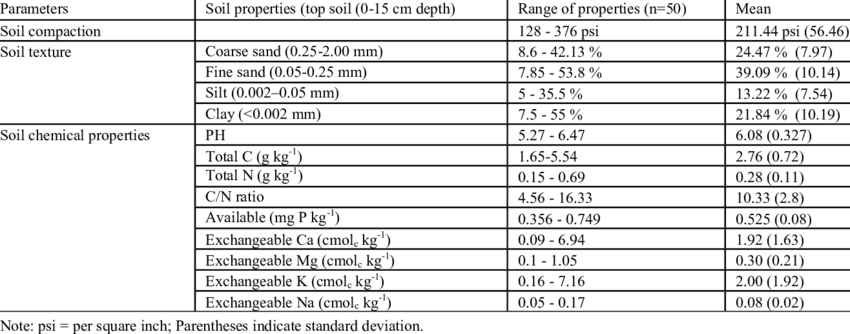
Soil testing is used to facilitate fertilizer composition and dosage selection for land employed in both agricultural and horticultural industries.Prepaid mail-in kits for soil and ground water testing are available to facilitate the packaging and delivery of samples to a laboratory. Similarly, in 2004, laboratories began providing fertilizer recommendations along with the soil composition report.

Lab tests are more accurate, though both types are useful. In addition, lab tests frequently include professional interpretation of results and recommendations. Always refer to all proviso statements included in a lab report as they may outline any anomalies, exceptions, and shortcomings in the sampling and/or analytical process/results.Some laboratories analyze for all 13 mineral nutrients and a dozen non-essential, potentially toxic minerals utilizing the "universal soil extractant" (ammonium bicarbonateDTPA)



**Soil contaminants:**

Common mineral soil contaminants include arsenic, barium, cadmium, c opper, mercury, lead, and zinc.Lead is a particularly dangerous soil component.

****

**CHAPTER 5 - SOIL CLASSIFICATION**

**5.1. Soil classification**

It deals with the systematiccategorization of soils based on distinguishing Characteristics as well as criteria that dictate choices inuse. Soil classification is adynamic subject, from thestructure of the system itself, to the definitions of classes and finally in the application in the field. Soilclassification can be approached from the perspective ofsoil as a material and soil as a resource.The most common engineering classificationsystem for soils is the Unified Soil Classification System(USCS) [6].

The USCS has three major classificationgroups:

(1) Coarse-grained soils (e.g. sands and gravels);

(2) Fine-grained soils (e.g. silts and clays);

(3) Highly

Organic soils (referred to as "peat"). The USCS furthersubdivides the three major soil classes for clarification. Afull geotechnical engineering soil description will alsoinclude other properties of the soil including color, insitumoisture content, in-situ strength, and somewhat moredetail about the material properties of the soil that isprovided by the USCS code.

The soils are classified into different orders, suborders,great groups, sub-groups, families and finally intoseries as per USDA Soil Taxonomy as in [8]. The solidphase of soil can be divided into mineral matter andorganic matter. The mineral particles can be furthersubdivided into classes based on size. The classification of soil particles according to size are Sand, Silt, Clay. Theproposition of Sand, Silt, and Clay present in soil determinesits texture.

**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.

Table 2. Soil Attribute

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

**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 %).

Table 3: Chemical composition of soils

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |

**Germination**

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

Table 4 : Days to germinate

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

**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).

**5.2. AComparative Study of Soil Classification**

The classification of soil was considered criticalto study because depending upon the fertility class ofthe soil the domain knowledge expert’s determineswhich crops should be taken on that particular soiland which fertilizers should be used for the same.The following section describes Naive Bayes, J48,JRip algorithms briefly.

**Naive Bayes**

A naive Bayes classifier is a simpleprobabilistic classifier based on applying Bayes'theorem with strong (naive) independenceassumptions. Depending on the precise nature of theprobability model, naive Bayes classifiers can betrained very efficiently in a supervised learningsetting. An advantage of the naive Bayes classifier isthat it only requires a small amount of training data toestimate the parameters (means and variances of thevariables) necessary for classification[5].

**J48 (C4.5)**

J48 is an open source Java implementationof the C4.5 algorithm in the Weka data mining tool.C4.5 is a program that creates a decision tree basedon a set of labeled input data. This decision tree canthen be tested against unseen labeled test data toquantify how well it generalizes. This algorithm wasdeveloped by Ross Quinlan. It is an extension ofQuinlan's earlier ID3 algorithm. C4.5 uses ID3algorithm that accounts for continuous attribute valueranges, pruning of decision trees, rule derivation, andsoon.The decision trees generated by C4.5 can beused for classification, and for this reason, C4.5 isoften referred to as a statistical classifier [6].

**JRip**

This algorithm implements a propositionalrule learner, Repeated Incremental Pruning toProduce Error Reduction (RIPPER), which wasproposed by William W. Cohen as anoptimizediversion of IREP.In this paper, three classification techniques(naïve Bayes, J48 (C4.5) and JRip) in data miningwere evaluated and compared on basis of time,accuracy, Error Rate, True Positive Rate and FalsePositive Rate. Tenfold cross-validation was used inthe experiment. Our studies showed that J48 (C4.5)model turned out to be the best classifier for soil samples.

**5.3. CONTEXT OF SOIL CLASSIFICATION**

Agriculture is the most significant application area particularly in the developing countries like India. Use of information technology in agriculture can hang the scenario of decision making and farmers can yield in better way. For decision making on overall issues related to agriculture field; data mining plays a vital role. In this paper we have discussed about the role of data mining in outlook of agriculture field. We have also discussed several data mining techniques,application of datamining in agriculture and soil containments.

In this paper, we've got suggested an analysis of the soil information using completely different algorithmsand prediction technique. In this paper we have demonstrated acomparative study of varied classification algorithms i.e. Naïve bayes, J48 (C4.5), JRip with the assistance ofdata mining tool .J48 is incredibly easy classifier to form a decision tree.We have demonstrated a comparative study of various classification algorithms i.e. Naïve Bayes, J48 (C4.5), JRip with the help of data mining tool WEKA. J48 is very simple classifier to make a decision tree. In future, we canplan to build Fertilizer RecommendationSystem which can be utilized effectively by the Soil Testing Laboratories. This System will recommend appropriate fertilizer for the given soil sample and cropping pattern.

**5.4. SOIL TESTING / SOIL ANALYSIS**

Soil testing is used to determine both the amount of each nutrient that is immediately available and the amount that can become available during the life of a crop. Various methods have been developed and the key to success is that the methods must be calibrated. Soil test calibration implies establishing relationship between soil test values and relative crop response (Agboola and Ayodele, 1987). Soil sampling done properly forms the basis of a successful long-term soil and crop nutrient management plan (Potash and Phosphate Institute, 2004).

It is most useful before planting to predict lime or fertilizer needs (Reisenauer et al., 1983). Also, it measures levels of specific nutrients in a soil. However, it cannot indicate whether plants growing in that soil are able to take up the nutrients. Soil test are the best way to assess soil pH (Kidder, 1993).

**OBJECTIVES OF SOIL TESTING**

(1) To accurately determine the status of available nutrients in soils (P, K, Mg, pH, Zn, B)

(2) To clearly indicate to the farmer the seriousness of any deficiency or excess that may exist in terms ofvarious crops

(3) To form the basis on which fertilizer needs are determined

(4) To express the results in such a way that they permit an economic evaluation of the suggested fertilizer recommendation

**ROLE OF THE EXTENSION SERVICE IN SOIL TESTING**

The actual analysis of the sample and the making out of fertilizer recommendation is only part of the soil testing service. To a large measure, the efficiency of this service depends upon the care and effort put froth by extension workers and farmers in the collection and dispatch of samples to the laboratory. Its effectiveness also depends upon the proper follow-through of the fertilizer recommendations, including the establishment of result demonstrations on farmer’s fields to induce the farmers to follow the fertilizer recommendations. In this work the staff of the extension service play the most important role, since they are the people directly in contact with the farmers or this reason, the soil chemist in charge of the laboratory must give periodic and through training to the extension staff on these subjects.

**How to sample soils**

It is not possible to move the entire soil of the farm to the laboratory for analysis. Only a small sample is required. A good sample is the first requirement for a reliable soil test. This sample should be a true representative of the farm/plot/field, that is, it should contain all the characteristics of the soil on this farm/plot/field. The proper methods of collecting and handling samples are determined by certain factors (Canadian Society of Soil Science, 2008).

(1) Accuracy and precision

(2) Sample areas that are representative of the farm

(3) Effect of farm size on accuracy

(4) When, how deep and how often to sample

(5) The use to be made of the analyses

(6) The pattern and ease of recognition of soil variability

(7) Previous and proposed management practices.

Although many types of sampling designs exist (Gilbert 1987; Mulla and McBratney, 2000; de Gruijter 2002) only two main types (random and systematic) are commonly used in the soil and earth sciences. Simple Random gives opportunity to all samples to be involved in the final selection while in stratified simple sampling; points are assigned to predefined groups or strata and a simple random sample chosen from each stratum. Stratified sampling (correctly applied) is likely to give a better result than simple random sampling (Williams, 1984).

Soil sampling could be done before or after preparing the land for planting (that is, to estimate pre-plant fertilizer needs). The farm operator must decide what level of detail is relevant to his or her field operations. Are there parts of the field that have different fertility patterns? Are

these areas large enough to be relevant? Does the operator want to engage in site- specific management? Has the operator the ability to vary fertilizer application rates to accommodate the field subsections identified (Canadian Society of Soil Science, 2008)?

The farm should be divided into small units of 1 hectare in size. Each unit should have uniform observableproperties. Subsections of a field would commonly be identified by differences in topography (termed landscape -directed soil sampling), parent material, management history, or yield history. It may be impossible to subdivide a field into smaller units if the farm operator has no prior knowledge of the field, or if there is no obvious topographic or parent material differences (Janzen, 1993). A composite sample consisting of 10 to 20 spot (core)

samples collected using approved sampling techniques is obtained for each unit of the farm.

**Tools**

The best tool is the metal tube called “sampling tube”. If this is not available, cutlass, shovel, hand trowel and auger could be used (Figure 5). Do not use brass, bronze, or galvanized tools because they will contaminate samples with copper and/or zinc.

**Soil test kit:**

This is a compact soil testing equipment with full complement of devices and reagents for the determination of the pH, electrical, conductivity, nitrogen, phosphorus and potassium in soil, fertilizer and water. Carbon, which is a good index of nitrogen content in soil, can also be determined. The equipment is robust and cost saving in terms of laboratory space.

**5.5. Method of soil testing**

**To sample a farm:** A rough map of the farm dividing it into sampling units as shown in Figures 4a and b was made. A composite soil sample is taken from each soil- sampling unit. The farm was then sampled as indicated in the following illustrations:

1. **Use the right sampling tool:** The best tool is a metal tube called a sampling-tube. However, if this is not available, any of the materials illustrated below could be used (Figure 5). These include cutlass, shovel, hand trowel and augur.
2. **The sample:** A composite sample; comprising of 10 to 20 core (single location)

(However 15 sub-samples may be the minimum number required to give sufficiently low variance, especially for P) samples that have been taken randomly in a zigzag fashion across the same soil area (land unit) is obtained (Figure 6). The land unit or soil area from which each composite sample is obtained is about one hectare. Sampling fertilizer bands, terrace channels, dead furrows, roads and other unusual areas should be avoided.

(a) Sample the soil from the surface to about 15 cm (6 inches) depth with a sampling tube, or any of the tools stated above. For a sampling tube, the process is as illustrated in Figure

(b) If the tool available is a spade, a shovel, or a cutlass,proceeds as follows:

(i) Dig a v-shaped hole, 15 cm (6 inches) deep

(ii)Then take one-and a –quarter centimeter (1/2 inch) slice of soil sample from the smooth side of the v-shaped hole illustrated in Figure 8.

1. Put all the core samples taken from the soil area together in a clean plastic bucket, as

a composite sample.

1. Mix the sample well with a clean rod or with your hand in the bucket (Figure 9).
2. Pour the soil sample into a clean plastic bag and tie it securely (Figure 10).
3. Label each plastic bag of sample properly.
4. Fill out the information requested as accurately as possible and send the sample to

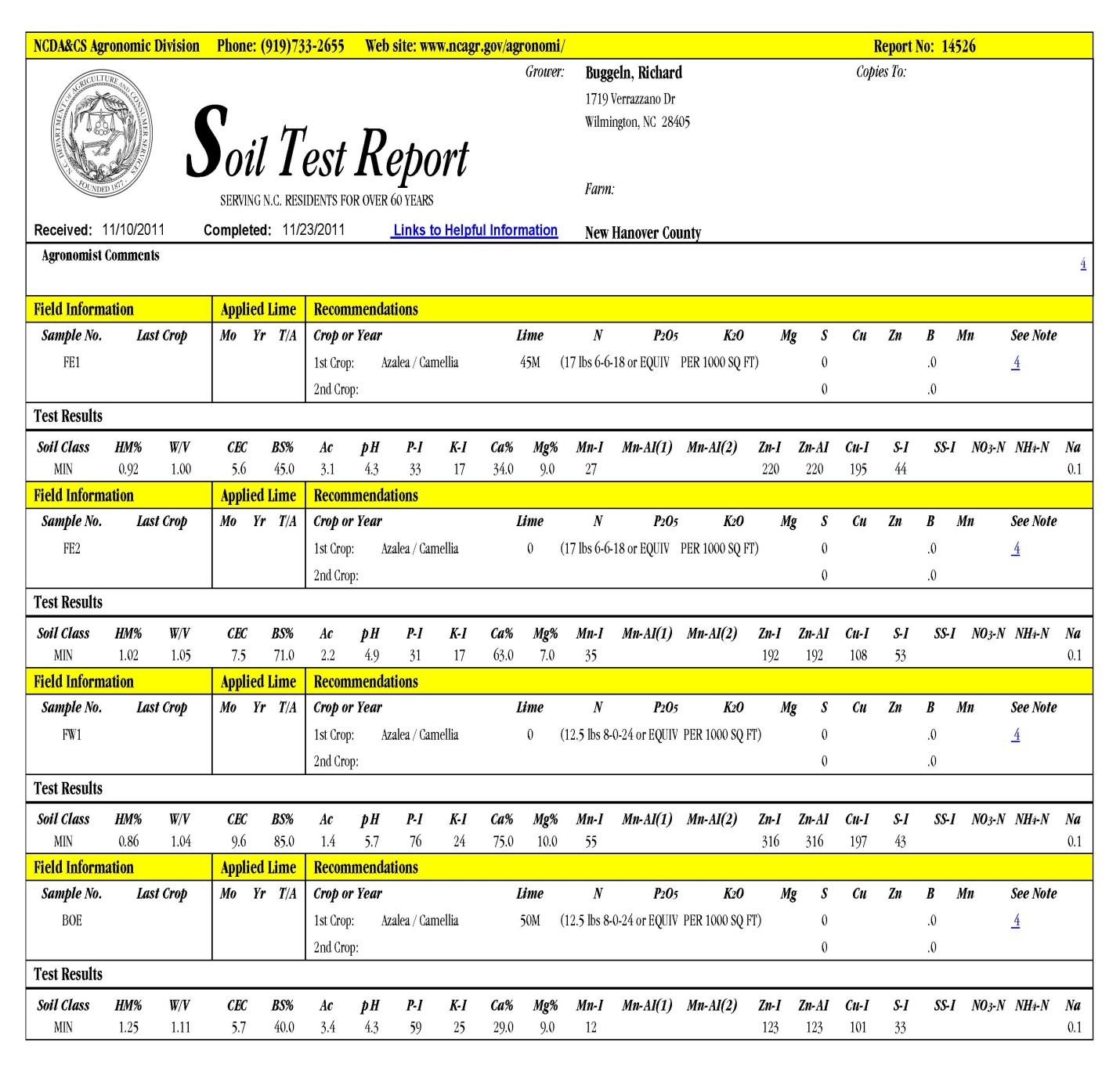
laboratory for analysis (Figure 10).

**Soil test result interpretation**

The figures on the soil analysis report (Figure 11 and Table 4) do not indicate the exact amount of nutrients available to a crop but when interpreted correctly give a description of the soil fertility status. The analytical result is used to suggest how much nutrient should be applied.

The exact amount needed will depend on the crop to be grown and must be modified to suit the conditions under which it is grown.

**5.6. SAMPLE SOIL TEST REPORT**



**UML DIAGRAMS**

**1. USE CASE DIAGRAM**

A close up of a map

Description generated with very high confidence

**2.CLASS DIAGRAM**

A close up of a map

Description generated with very high confidence

**3.ACTIVITY DIAGRAM**

A screenshot of a cell phone

Description generated with high confidence

**4.SEQUENCE DIAGRAM**

A close up of a map

Description generated with very high confidence

**5.COLLABORATION DIAGRAM**

A screenshot of a social media post

Description generated with very high confidence

**RESULTS**

**Naivebayes python code snippet**

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)

**Angular code snippet**

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 code snippet**

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}`);

});

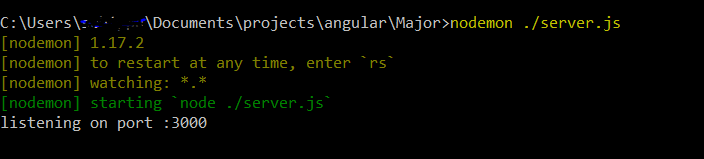
}

})

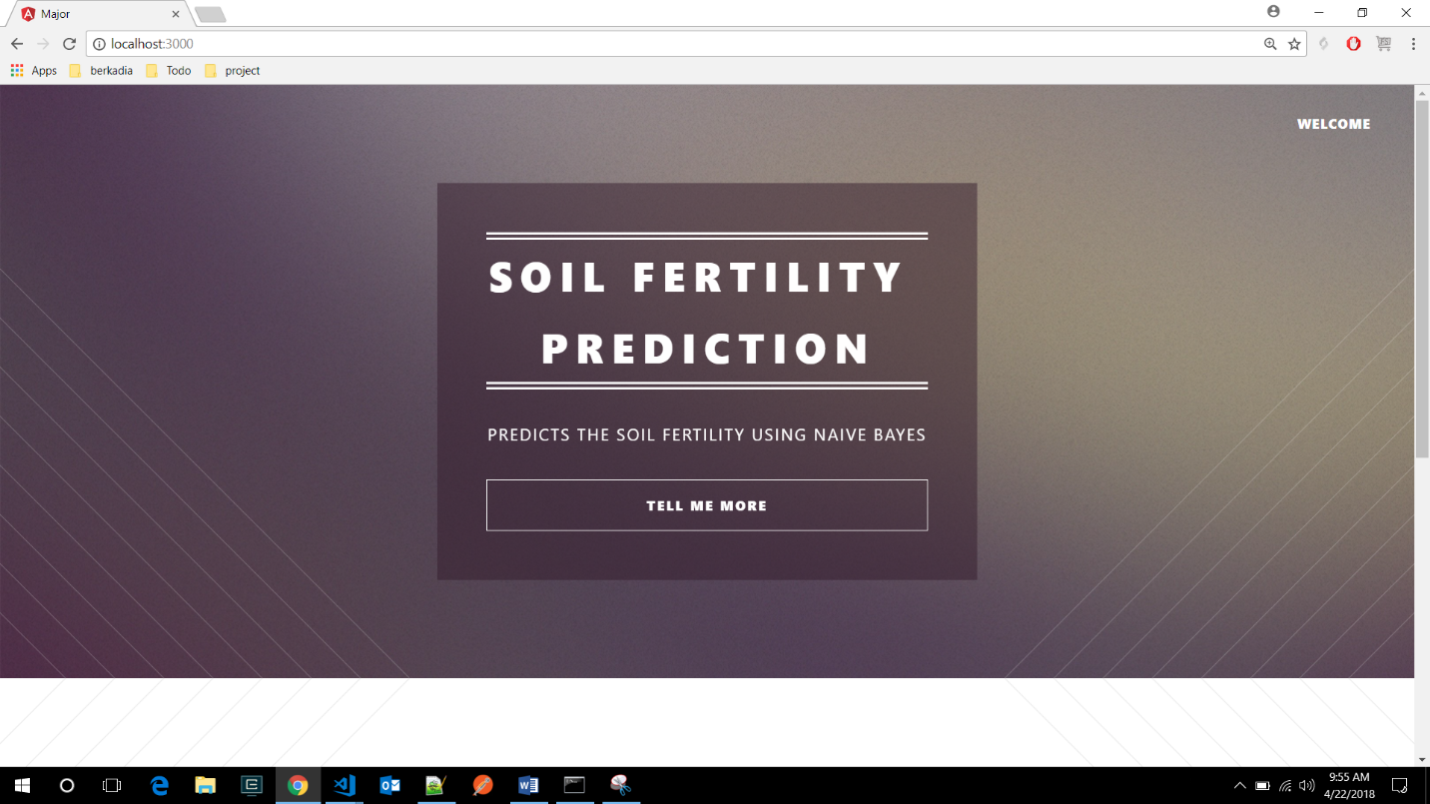
module.exports = router;

**Output**

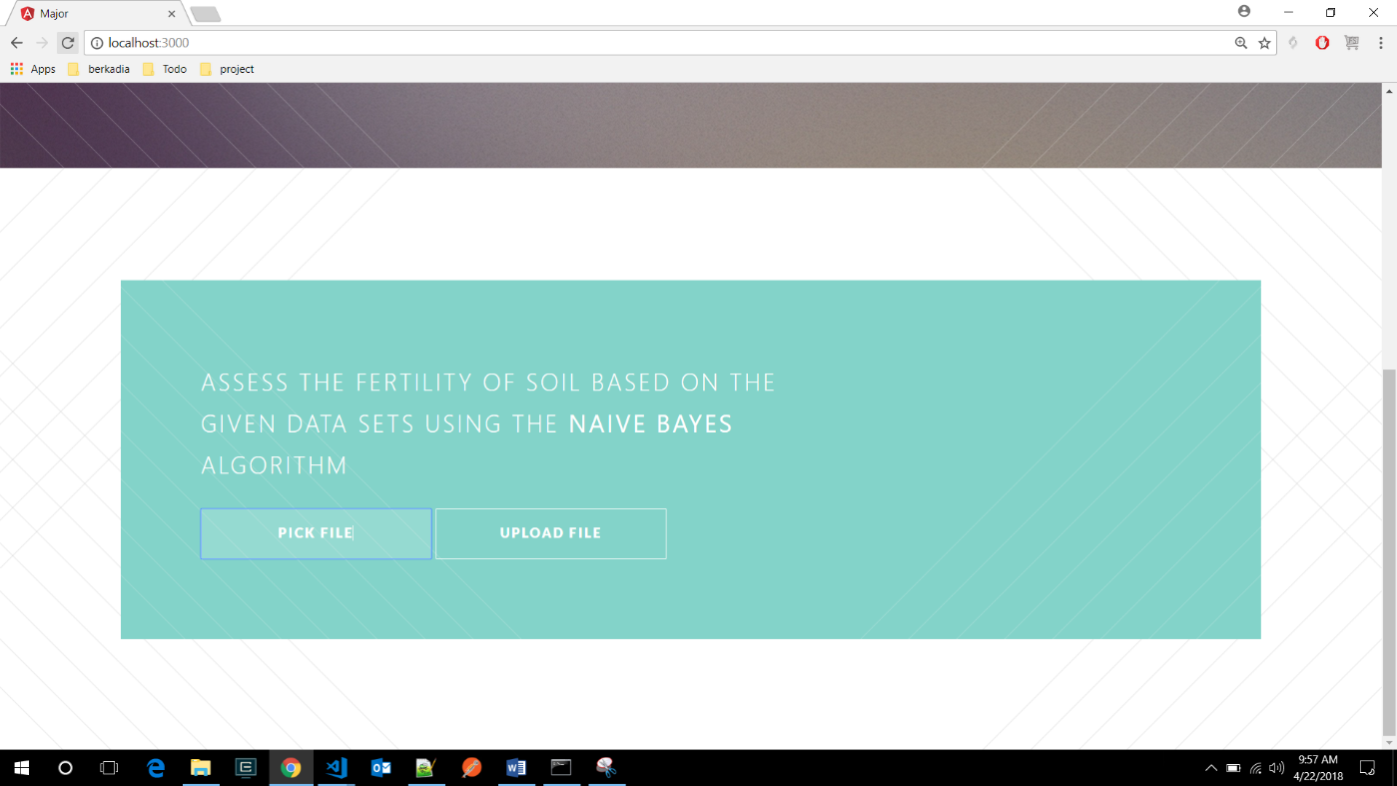
Run the node server with nodemon in the cmd

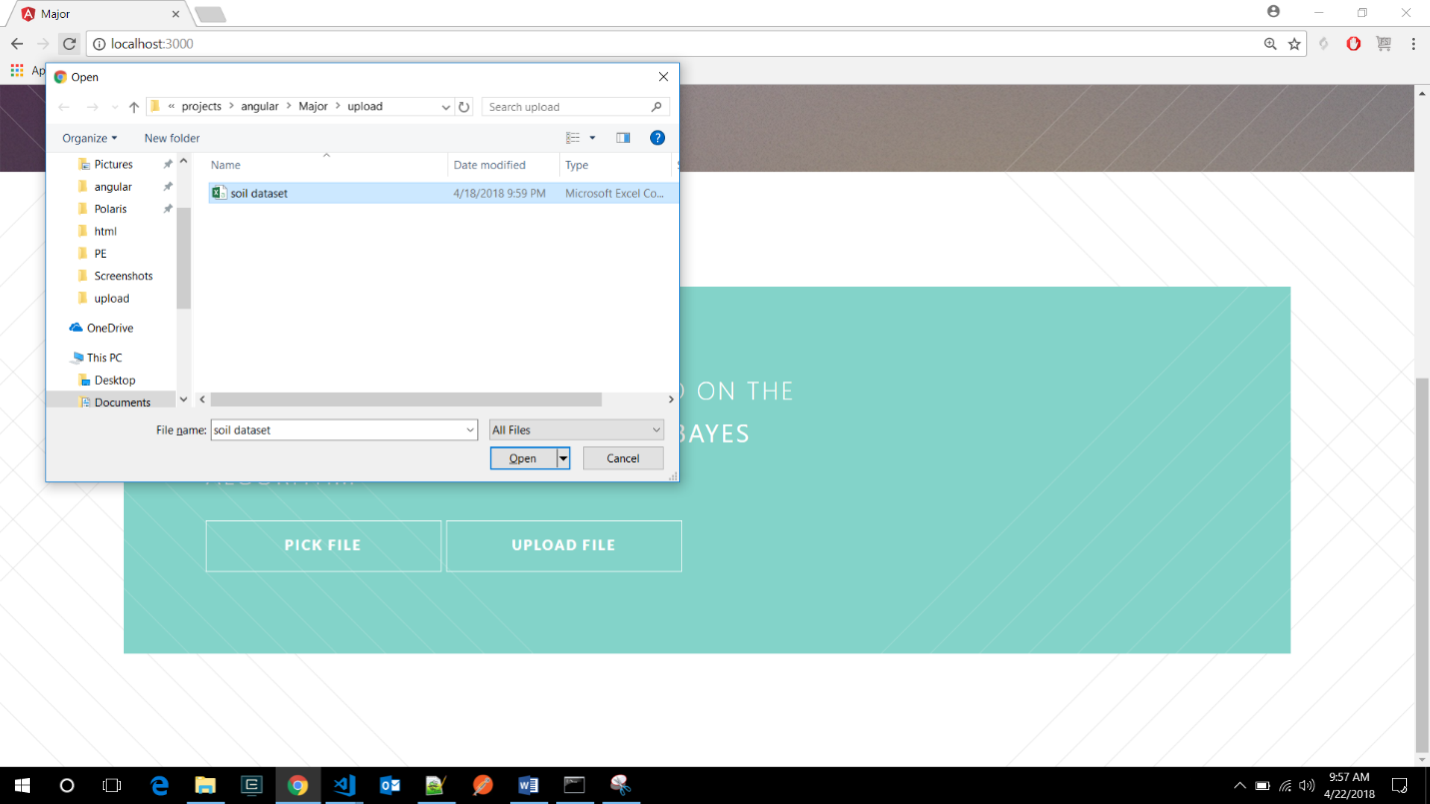


1.Open “localhost:3000” in the Browser.

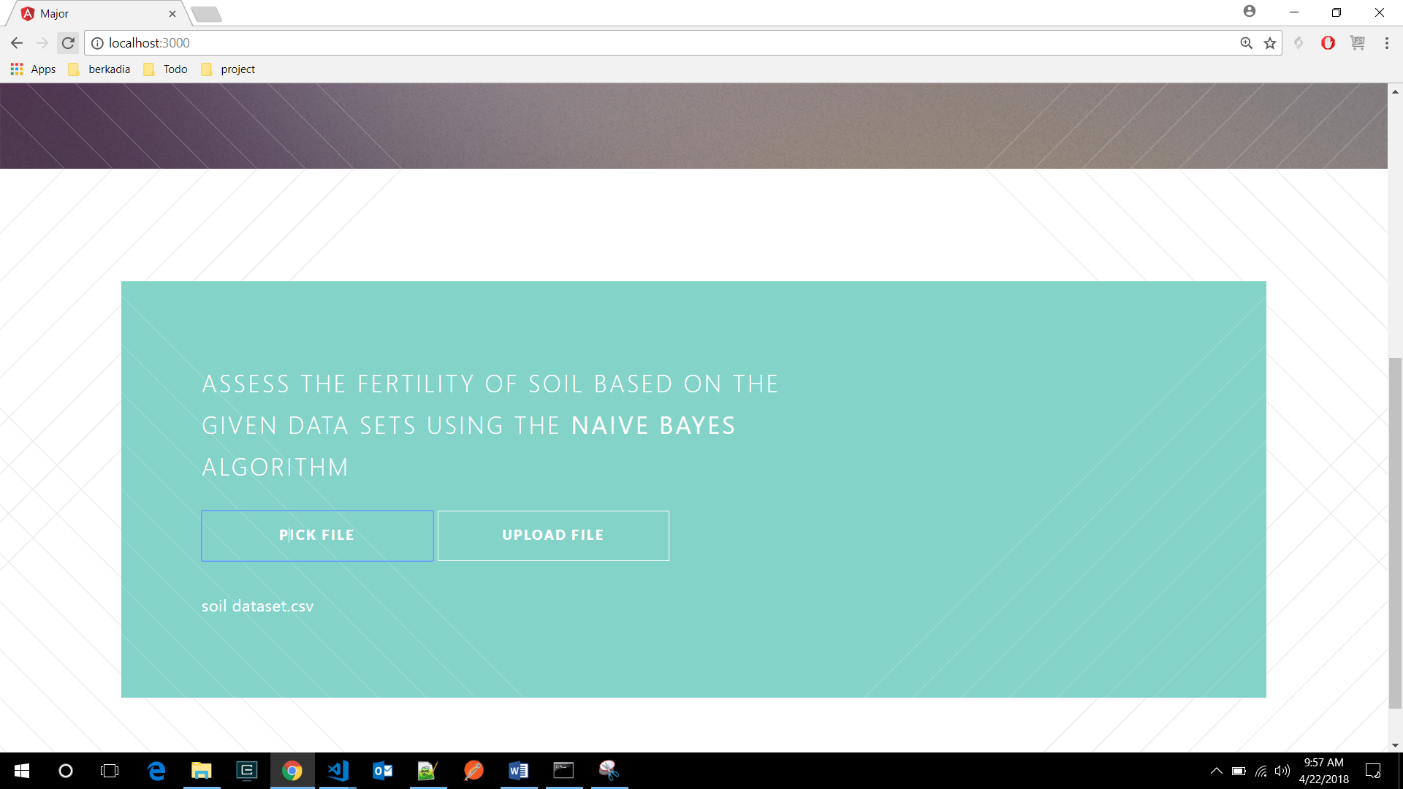


2.Pick a .csv file.

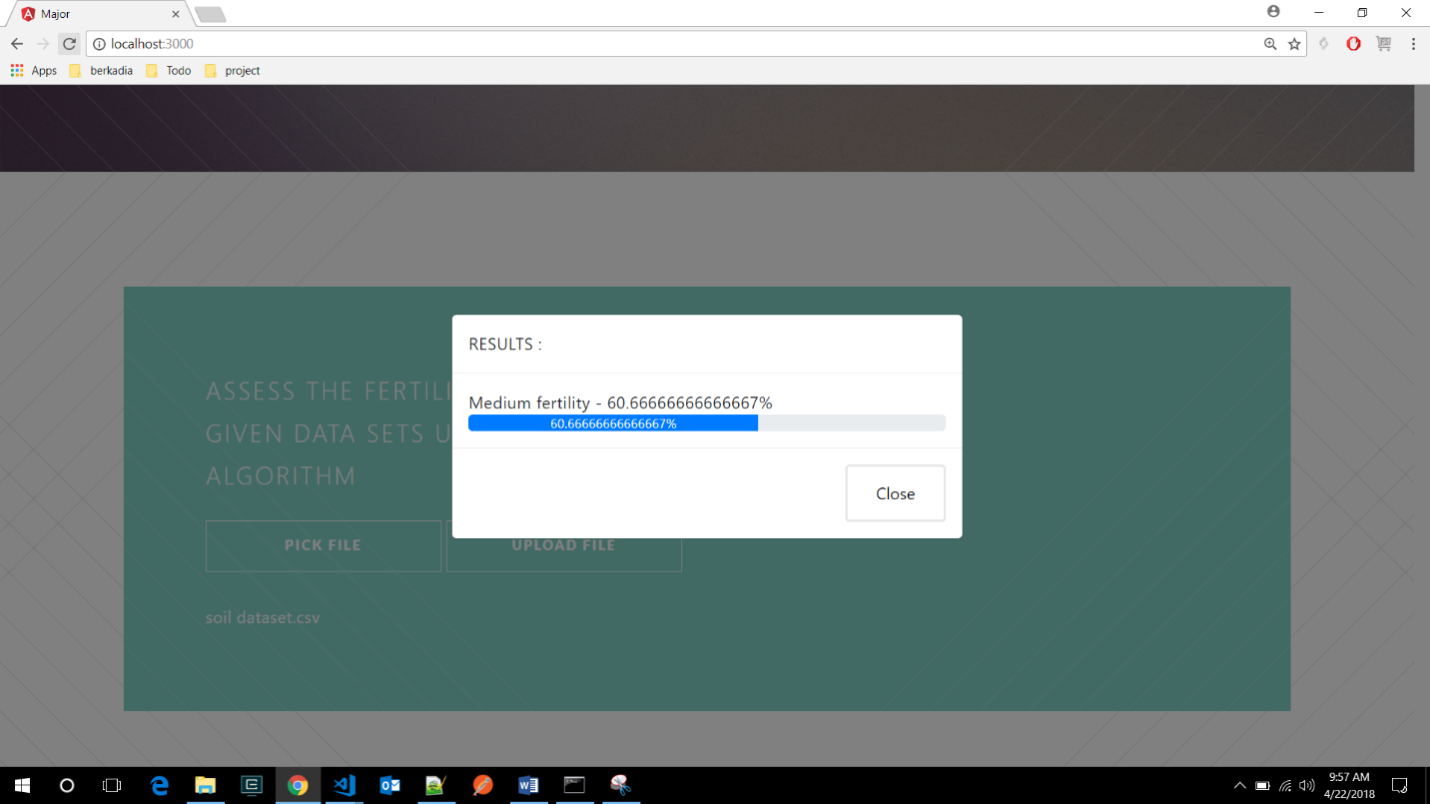




3.Upload the selected .csv file.



4.Fertility is generated.



**CONCLUSION**

Continuous removal of nutrients from the soil via differentmeans requires continuous replacement to maintain productivity. This replacement (fertilization) requiresspecific knowledge in order to truly maximize yield, minimize cost and to reduce adverse effect on soil/crops. Of the methods available, soil test seems to be the easiest to predict fertilizer requirement for Nigerian farmers. At a certain level, the complex tissue analysismay be used as a tool but it must be combined with soil test result.

The large amounts of data that are nowadays virtually harvested along with the crops have to be analyzed and should be used to their full extent.Various decision tree algorithms can be used for prediction of soil fertility. But naïve Bayes is preferred because of its statistical probability Approach.

In simple terms, a Naïve Bayes classifier assumes that the presence (or absence) of a

particular feature of a class is unrelated to the presence (or absence) of any other feature. In Naïve bayes, The mean absolute error is very less compared to other algorithms like J48.

**BIBLIOGRAPHY**

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,

1. “Soil test”, Wikipedia, February 2012
2. L. Armstrong, D. Diepeveen& R. Maddern,(2004), **“**The application of data mining

techniques to characterize agricultural soilprofiles”,

1. “Methods Manual-Soil Testing in India”,Department of Agriculture & CooperationMinistry of Agriculture Government of India,2011
2. “Naïve Bayes”, Wikipedia, February 2012
3. ”C4.5 (J48)”, Wikipedia, February 2012
4. 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,
5. 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.
6. 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