

NUTRIENTS AND MINERALS BASED CROP SELECTION USING NEURAL NETWORKS

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

By

Shaik Sajid Ali (37110722)

Siddam Sai Praveen (37110715)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

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

**Accredited with Grade "A" by NAAC I 12B Status by UGC
I Approved by AICTE**

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600

119

March – 2021

BONAFIDE CERTIFICATE

DECLARATION

I Shaik Sajid Ali and Siddam Sai Praveen hereby declare that the Project Report entitled Crop Suggestion Using Machine Learning Based on Soil Conditions done by me under the guidance of Dr. /Prof./ Mr./Ms.____(Internal) and ____ (External) at ____ (Company name and address) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Sathyabama Institute of science and technology.

DATE:

**PLACE:
CANDIDATE**

SIGNATURE OF THE

ACKNOWLEDGEMENT

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

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

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr./Mr./Ms MS ROOBINI M.E., Ph.D., for his/her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

India being an agriculture country, its economy predominantly depends on agriculture yield growth and agroindustry products. Data Mining is an emerging research field in crop yield analysis. Yield prediction is a very important issue in agricultural. Any farmer is interested in knowing how much yield he is about to expect. Analyze the various related attributes like location, pH value from which alkalinity of the soil is determined. Along with it, percentage of nutrients like Nitrogen (N), Phosphorous (P), and Potassium (K) Location is used along with the use of third-party applications like APIs for weather and temperature, type of soil, nutrient value of the soil in that region, amount of rainfall in the region, soil composition can be determined. All these attributes of data will be analyzed, train the data with various suitable machine learning algorithms for creating a model. The system comes with a model to be precise and accurate in predicting crop yield and deliver the end user with proper recommendations about required fertilizer ratio based on atmospheric and soil parameters of the land which enhance to increase the crop yield and increase farmer revenue.

TABLE OF CONTENTS

CHAPTER No	TITLE	PAGE
	LIST OF ABBREVIATIONS	Viii
1	INTRODUCTION	1
1.1	About the Project	1
2	LITERATURE SURVEY	3
2.1	Hardware and Software specification	3
2.1.1	Hardware requirements	3
2.1.2	Software requirements	3
2.2	Technologies used	3
2.2.1	Introduction to Python	3
3	METHODOLOGY AND IMPLEMENTATION	12
3.1	Support vector machine(SVM)	11
4	RESULTS AND DISCUSSION	30
4.1	Design and Implementation Constraints	30
4.1.1	Constraints in Analysis	30
4.1.2	Constraints in Design	31

4.1.3	Constraints in Implementation	31
4.2	Other Nonfunctional Requirements	31
4.2.1	Performance Requirements	31
4.2.2	Safety Requirements	32
5	Architecture Diagram	33
6	CONCLUSION	34
	CODING AND TESTING	35
	SOURCE CODE	35
	SCREEN SHOTS	41
	REFERENCES	43

LIST OF ABBREVIATIONS

1. ML - Machine learning
2. SVM - Support vector machine
3. STCR-IPNS - Soil test and yield target based fertilizer prescriptions under Integrated Plant Nutrition System
4. KNN - K nearest neighbour

CHAPTER - 1

REQUIREMENT SPECIFICATIONS

1.1 INTRODUCTION

Agriculture plays an important role in Indian's economy while considering that. the methods used for crops to produce are harsh and not suitable for present conditions. The excessive usage of harmful chemicals and pesticides can cause permanent damage to the field which can result in improper growth in agriculture field. Based on the data acquired from the sample of the soil taken from the field we can determine which crop might be the best option for the soil.

This paper consists of all the detailed methods and working of this project with accuracy. Initially to predict something there has to be a certain valuation, upper limits and lower limits which determine the contents precisely without any complex quantities. Generally, when a farmer consults an agriculture industry, they are some requirements they need to carry such as percentage of potassium (K), phosphorous (P), Iron, Magnesium, calcium etc. This data provides a view over the soil and can show what minerals the soil is lacking for the growth. Instead of consulting a large industry our objective is to create a shorter and faster way to determine what crop is best suitable for agriculture in that field. To do so a web application is created as interface where one can give the data of minerals of their field and comparing that data the machine learning program sorts a better solution for the problem not only on the basis of minerals but also seasonal basis this classification is done by sorting them as Kharif crops and Rabi crops, Kharif season is on the start of summer rainfalls (Autumn), Rabi season typically comes during winter season. Usually, this classification helps to predict crops based on the seasons.

Considering a state such as Tamilnadu, Soil test and yield target based fertilizer prescriptions under Integrated Plant Nutrition System (STCR-IPNS) have been developed for 31 agricultural and horticultural crops and 10 major cropping sequences on 17 soil series covering six agroclimatic zones of Tamil Nadu. These fertilizer

prescriptions can be adopted with recommended secondary and micronutrients along with the entire improved package of practices for various crops in similar and allied soil types occurring in different agro climatic zones of Tamil Nadu[1]. There are around 60 types of different crops in Tamilnadu using a fertilizer prescription equation.

CHAPTER – 2

LITERATURE SURVEY

Hardware and Software Specifications

Hardware Requirements

- Hard Disk : 500GB and Above
- RAM : 4GB and Above
- Processor : I3 and Above

Software Requirements

- ✓ Operating System : Windows 7 , 8, 10 (64 bit)
- ✓ Software : Python 3.7
- ✓ Tools : Anaconda (Jupyter Note Book IDE)

TECHNOLOGIES USED

- Python

Introduction to Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

- Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been backported to Python 2. But in general, they remain not quite compatible.
- Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End Of Life date of January 1, 2020 has been established for Python 2, after which time it will no longer be maintained.
- Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe Monty Python's Flying Circus, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.
- It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python is Interpreted

- Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.
- This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.
- One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.
- In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.
- For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.
- Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.
- Python accomplishes what many programming languages don't: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

Introduction Machine learning

Introduction:

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning tasks:

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. Semi algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.

Classification algorithms and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values true and false. Regression algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object.

In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of "features", or inputs, in a set of data.

Active learning algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used in autonomous vehicles or in learning to play a game against a human opponent. Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

Types of learning algorithms:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

Supervised learning:

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

In the case of semi-supervised learning algorithms, some of the training examples are missing training labels, but they can nevertheless be used to improve the quality of a model. In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.

Unsupervised learning:

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called *clusters*) so that observations within the same cluster are similar according to one or more pre designated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some *similarity metric* and evaluated, for example, by *internal compactness*, or the similarity between members of the same cluster, and *separation*, the difference between clusters. Other methods are based on *estimated density* and *graph connectivity*.

Semi-supervised learning:

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

CHAPTER - 3

METHODOLOGY AND IMPLEMENTATION

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. “Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for either classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well.

Support Vector Machine (SVM)

Introduction:

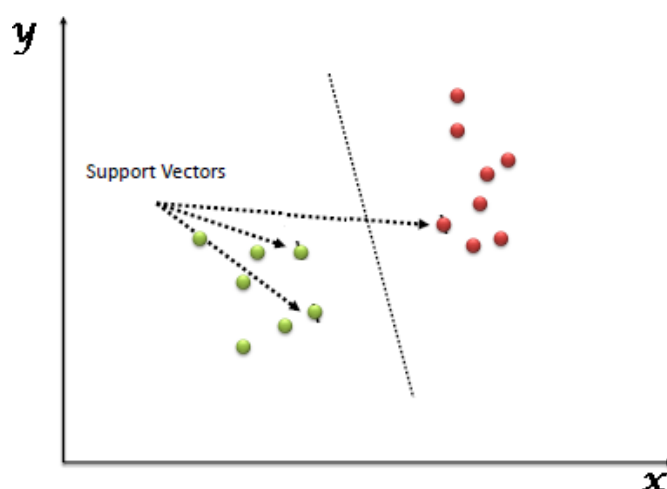
Mastering machine learning algorithms isn't a myth at all. Most of the beginners start by learning regression. It is simple to learn and use, but does that solve our purpose? Of course not! Because, you can do so much more than just Regression!

Think of machine learning algorithms as an armory packed with axes, sword, blades, bow, dagger etc. You have various tools, but you ought to learn to use them at the right time. As an analogy, think of 'Regression' as a sword capable of slicing and dicing data efficiently, but incapable of dealing with highly complex data. On the

contrary, 'Support Vector Machines' is like a sharp knife - it works on smaller datasets, but on them, it can be much stronger and powerful in building models.

Support Vector Machine

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for either classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).



Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

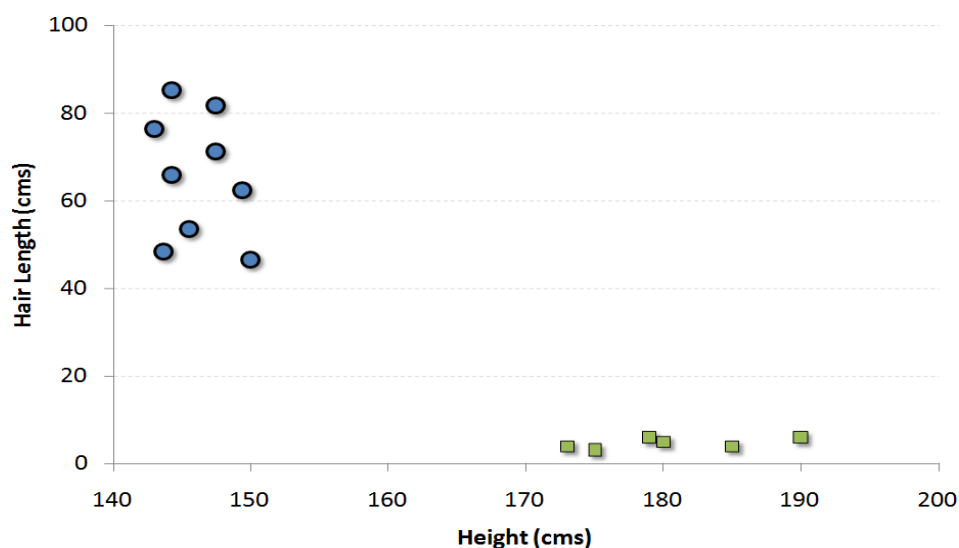
You can look at definition of support vectors and a few examples of its working [here](#).

Support Vector Machine – Simplified

Let's talk about how support vector machine works. **This is suitable for readers who do not know much about this algorithm and have a curiosity to learn a new technique.** In following explore the technique in detail and analyze cases where such techniques are stronger than other techniques.

Classification analysis

Let's consider an example to understand these concepts. We have a population composed of 50%-50% Males and Females. Using a sample of this population, you want to create some set of rules which will guide us the gender class for rest of the population. Using this algorithm, we intend to build a robot which can identify whether a person is a Male or a Female. This is a sample problem of classification analysis. Using some set of rules, we will try to classify the population into two possible segments. For simplicity, let's assume that the two differentiating factors identified are: Height of the individual and Hair Length. Following is a scatter plot of the sample.



The blue circles in the plot represent females and green squares represents male. A few expected insights from the graph are:

1. Males in our population have a higher average height.
2. Females in our population have longer scalp hairs.

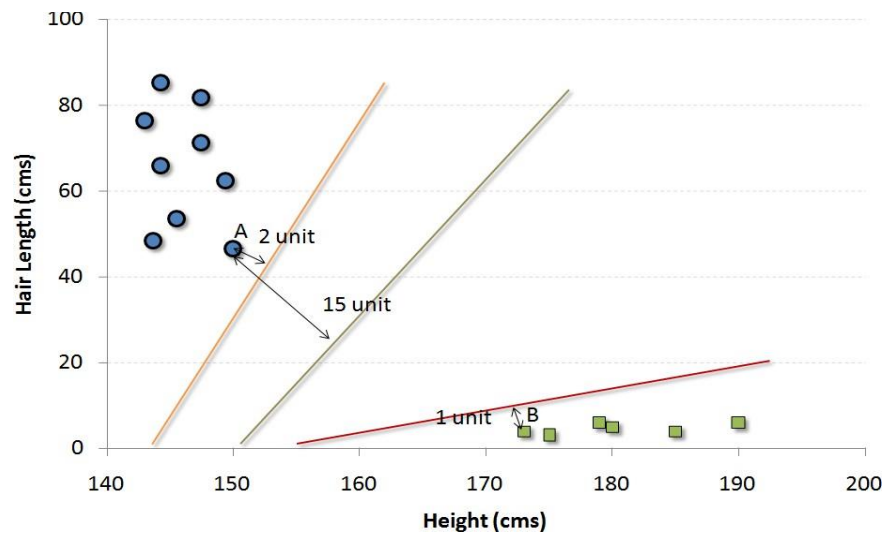
If we were to see an individual with height 180 cm's and hair length 4 cm's, our best guess will be to classify this individual as a male. This is how we do a classification analysis.

A Support Vector and SVM

Support Vectors are simply the co-ordinates of individual observation. For instance, (45,150) is a support vector which corresponds to a female. Support Vector Machine is a frontier which best segregates the Male from the Females. In this case, the two classes are well separated from each other; hence it is easier to find a SVM.

How to find the Support Vector Machine for case in hand?

There are many possible frontiers which can classify the problem in hand. Following are the three possible frontiers.

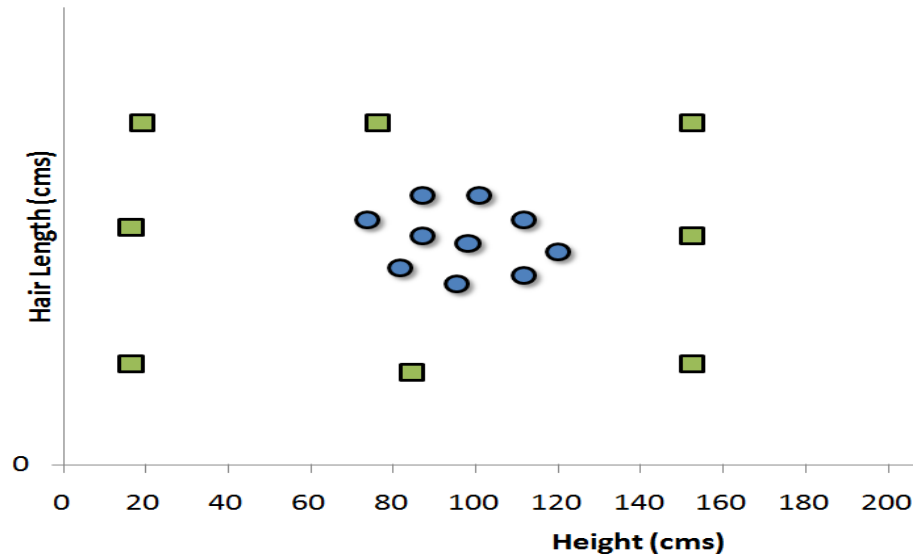


How do we decide which is the best frontier for this particular problem statement?

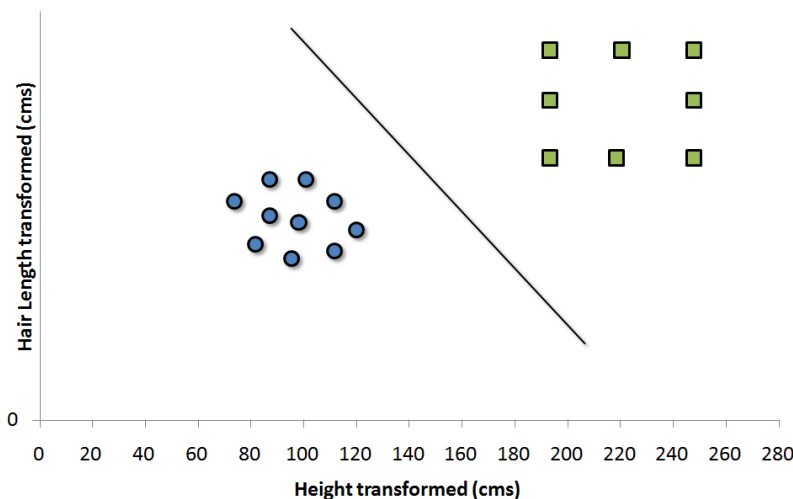
The easiest way to interpret the objective function in a SVM is to find the minimum distance of the frontier from closest support vector (this can belong to any class). For instance, orange frontier is closest to blue circles. And the closest blue circle is 2 units away from the frontier. Once we have these distances for all the frontiers, we simply choose the frontier with the maximum distance (from the closest support vector). Out of the three shown frontiers, we see the black frontier is farthest from nearest support vector (i.e. 15 units).

What if we do not find a clean frontier which segregates the classes?

Our job was relatively easier finding the SVM in this business case. What if the distribution looked something like as follows?



In such cases, we do not see a straight line frontier directly in current plane which can serve as the SVM. In such cases, we need to map these vectors to a higher dimension plane so that they get segregated from each other. Such cases will be covered once we start with the formulation of SVM. For now, you can visualize that such transformation will result into following type of SVM.



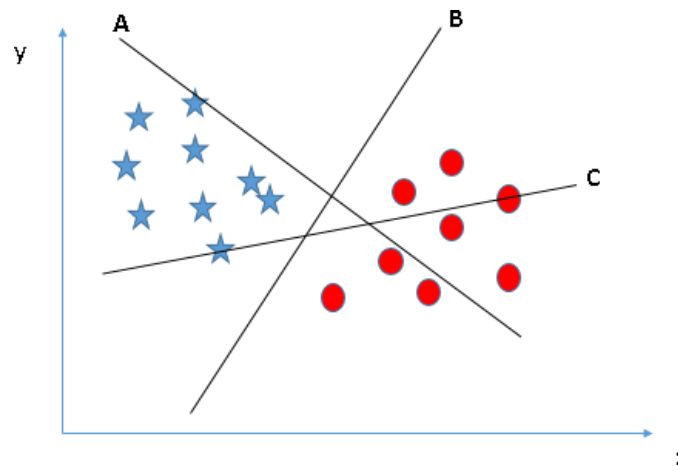
Each of the green squares in original distribution is mapped on a transformed scale. And transformed scale has clearly segregated classes. Many algorithms have been proposed to make these transformations.

How does it work?

Above, we got accustomed to the process of segregating the two classes with a hyper-plane. Now the burning question is “How can we identify the right hyper-plane?” Don’t worry; it’s not as hard as you think!

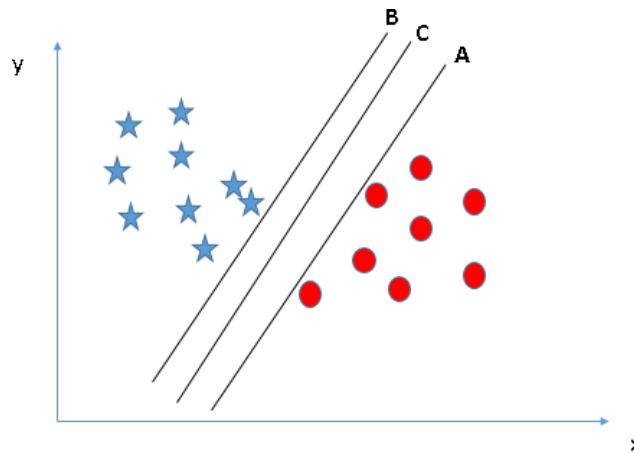
Let’s understand:

- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.

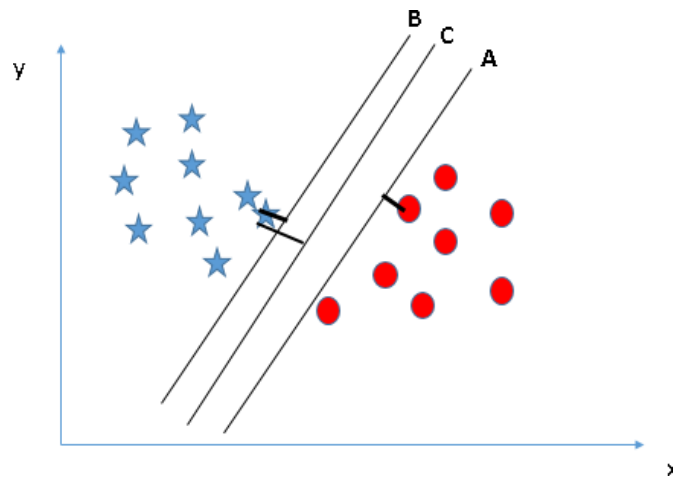


You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.

- **Identify the right hyper-plane (Scenario-2):** Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, how can we identify the right hyper-plane?

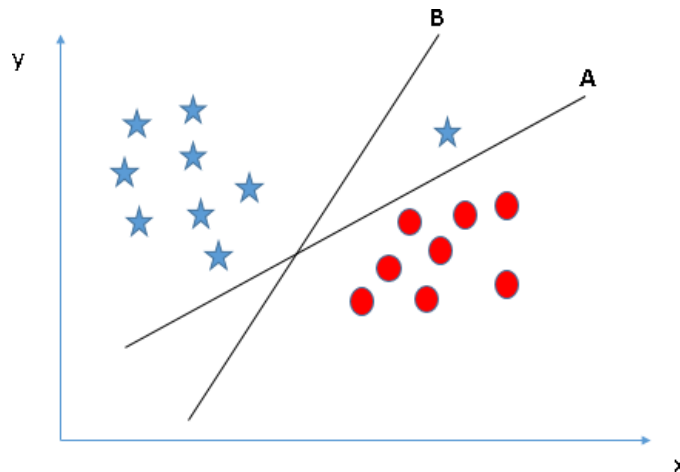


Here, maximizing the distances between nearest data points (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**. Let's look at the below snapshot:



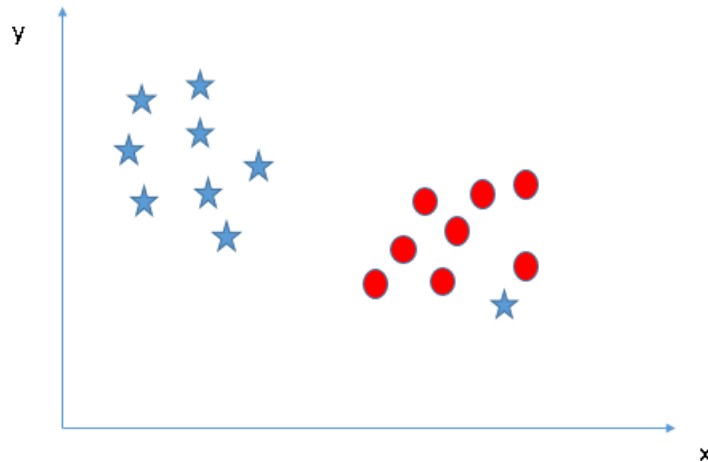
Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

- **Identify the right hyper-plane (Scenario-3):**Hint: Use the rules as discussed in previous section to identify the right hyper-plane

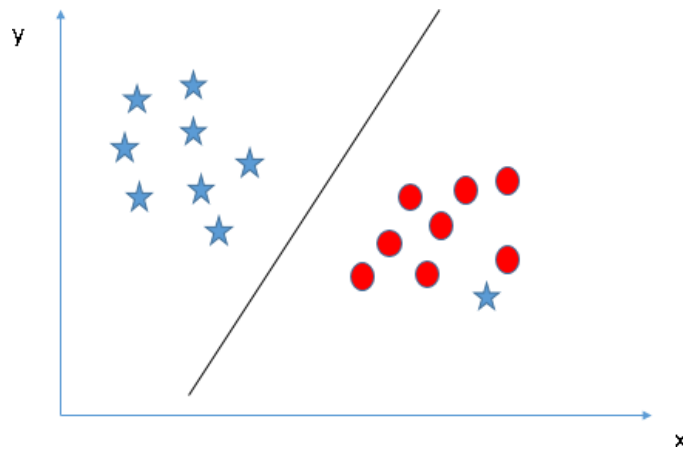


Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A**. But, here is the catch; SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A**.

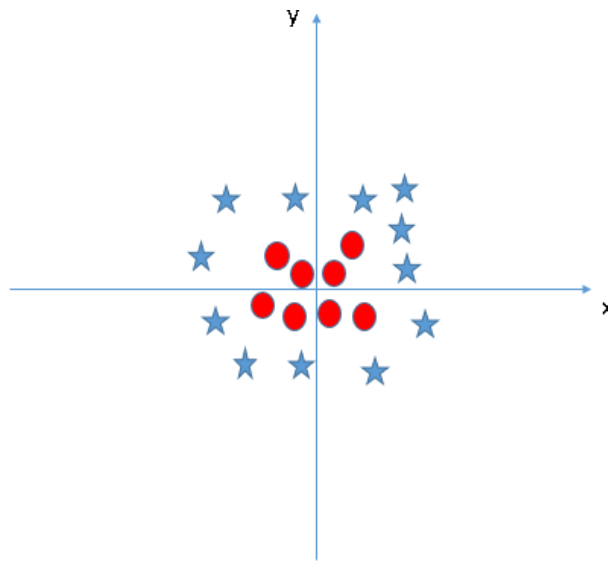
- **Can we classify two classes (Scenario-4)?:** Below, I am unable to segregate the two classes using a straight line, as one of star lies in the territory of other(circle) class as an outlier.



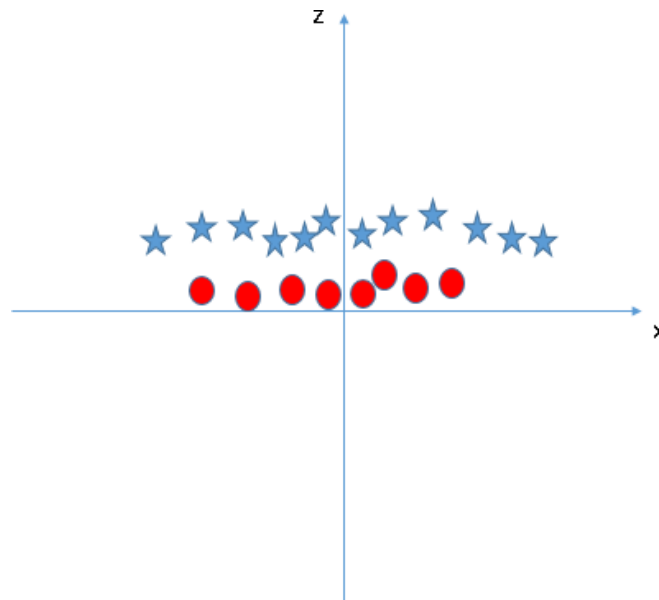
As I have already mentioned, one star at other end is like an outlier for star class. SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin. Hence, we can say, SVM is robust to outliers.



- **Find the hyper-plane to segregate to classes (Scenario-5):** In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.



SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature $z = x^2 + y^2$. Now, let's plot the data points on axis x and z:

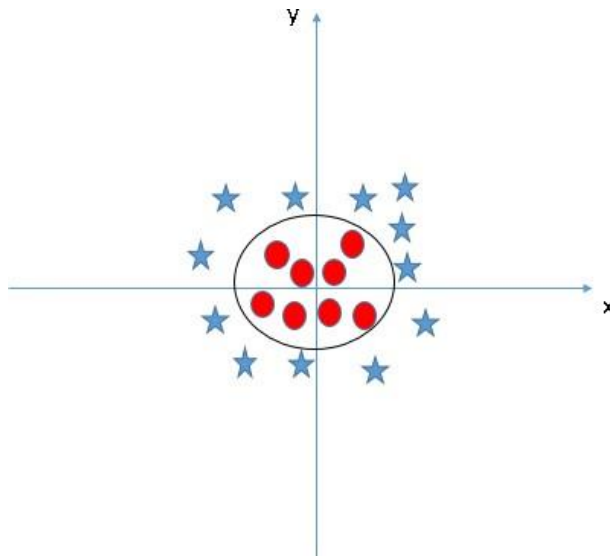


In above plot, points to consider are:

- All values for z would be positive always because z is the squared sum of both x and y
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z .

In SVM, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, SVM has a technique called the **kernel trick**. These are functions which takes low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called kernels. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then find out the process to separate the data based on the labels or outputs you've defined.

When we look at the hyper-plane in original input space it looks like a circle:



Pros and Cons associated with SVM

- **Pros:**
 - It works really well with clear margin of separation
 - It is effective in high dimensional spaces.
 - It is effective in cases where number of dimensions is greater than the number of samples.
 - It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- **Cons:**
 - It doesn't perform well, when we have large data set because the required training time is higher
 - It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping
 - SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library

K-Nearest Neighbors

Introduction

In four years of the analytics built more than 80% of classification models and just 15-20% regression models. These ratios can be more or less generalized

throughout the industry. The reason of a bias towards classification models is that most analytical problem involves making a decision. For instance will a customer attrite or not, should we target customer X for digital campaigns, whether customer has a high potential or not etc. This analysis is more insightful and directly links to an implementation roadmap. In this article, we will talk about another widely used classification technique called K-nearest neighbors (KNN). Our focus will be primarily on how does the algorithm work and how does the input parameter effect the output/prediction.

KNN algorithm

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique we generally look at 3 important aspects:

1. Ease to interpret output
2. Calculation time
3. Predictive Power

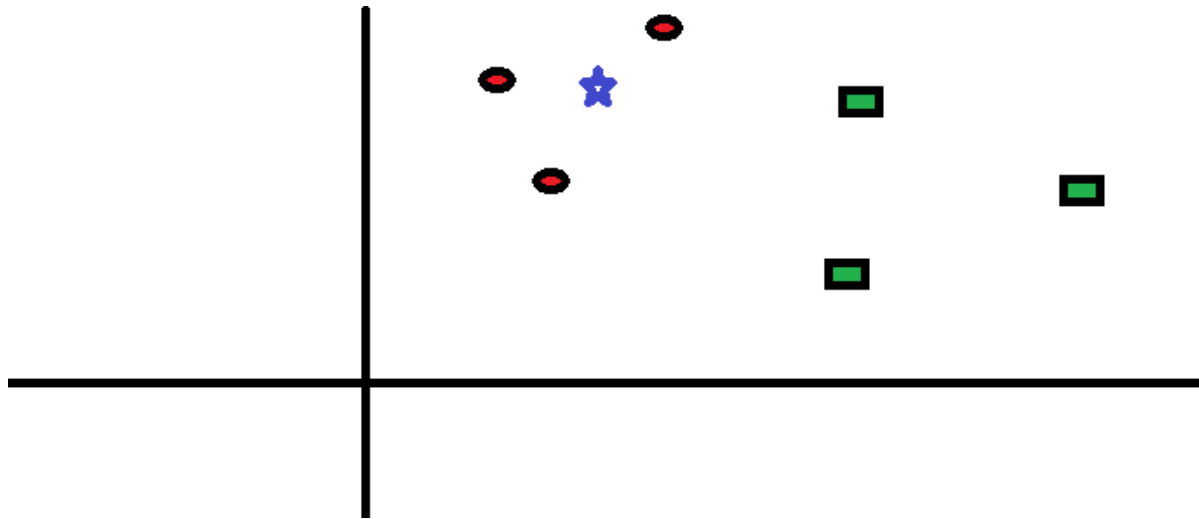
Let us take a few examples to place KNN in the scale:

	Logistic Regression	CART	Random Forest	KNN
1. Ease to interpret output	2	3	1	3
2. Calculation time	3	2	1	3
3. Predictive Power	2	2	3	2

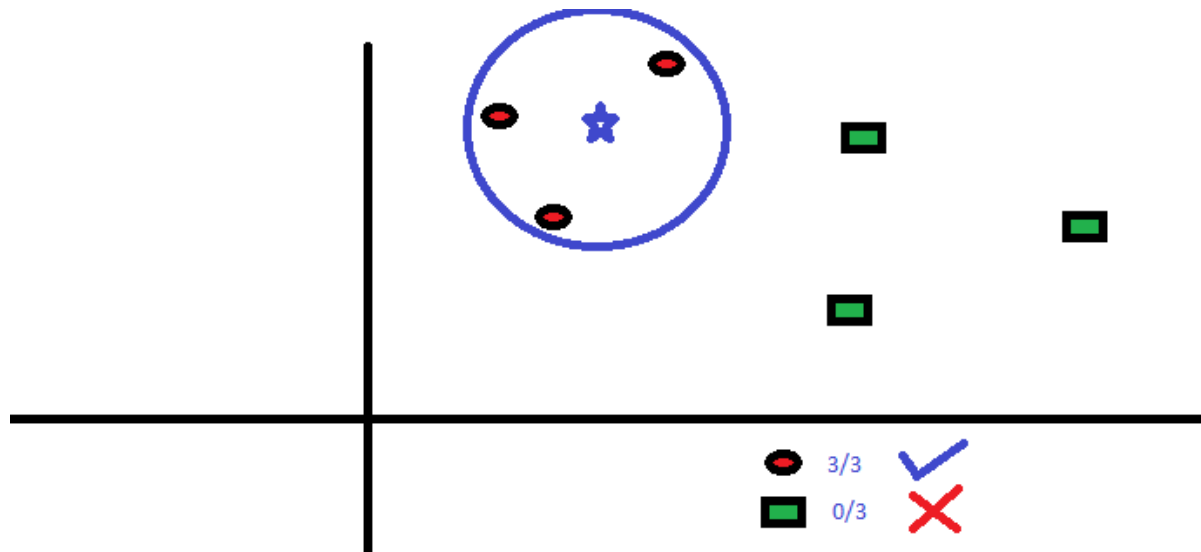
KNN algorithm fairs across all parameters of considerations. It is commonly used for its easy of interpretation and low calculation time.

The KNN algorithm work

Let's take a simple case to understand this algorithm. Following is a spread of red circles (RC) and green squares (GS):



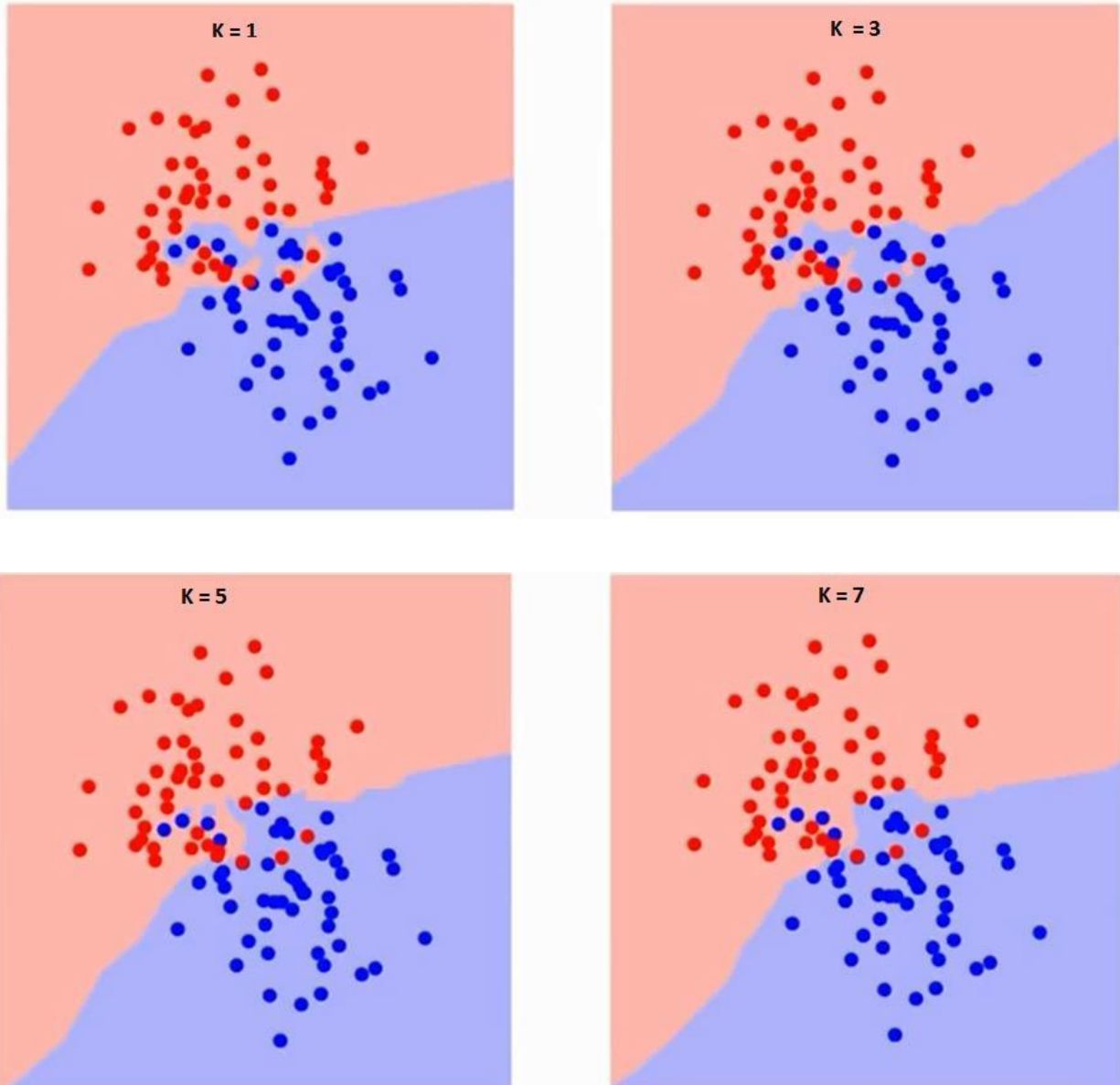
You intend to find out the class of the blue star (BS). BS can either be RC or GS and nothing else. The “K” in KNN algorithm is the nearest neighbors we wish to take vote from. Let's say $K = 3$. Hence, we will now make a circle with BS as center just as big as to enclose only three data points on the plane. Refer to following diagram for more details:



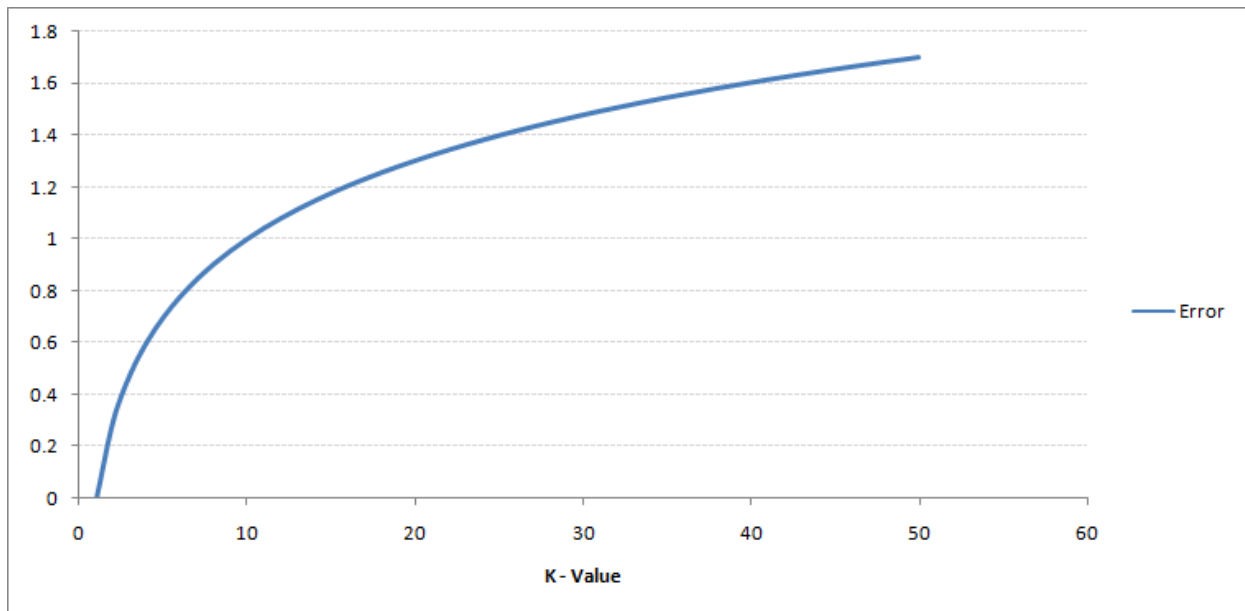
The three closest points to BS is all RC. Hence, with good confidence level we can say that the BS should belong to the class RC. Here, the choice became very obvious as all three votes from the closest neighbor went to RC. The choice of the parameter K is very crucial in this algorithm.

How do we choose the factor K?

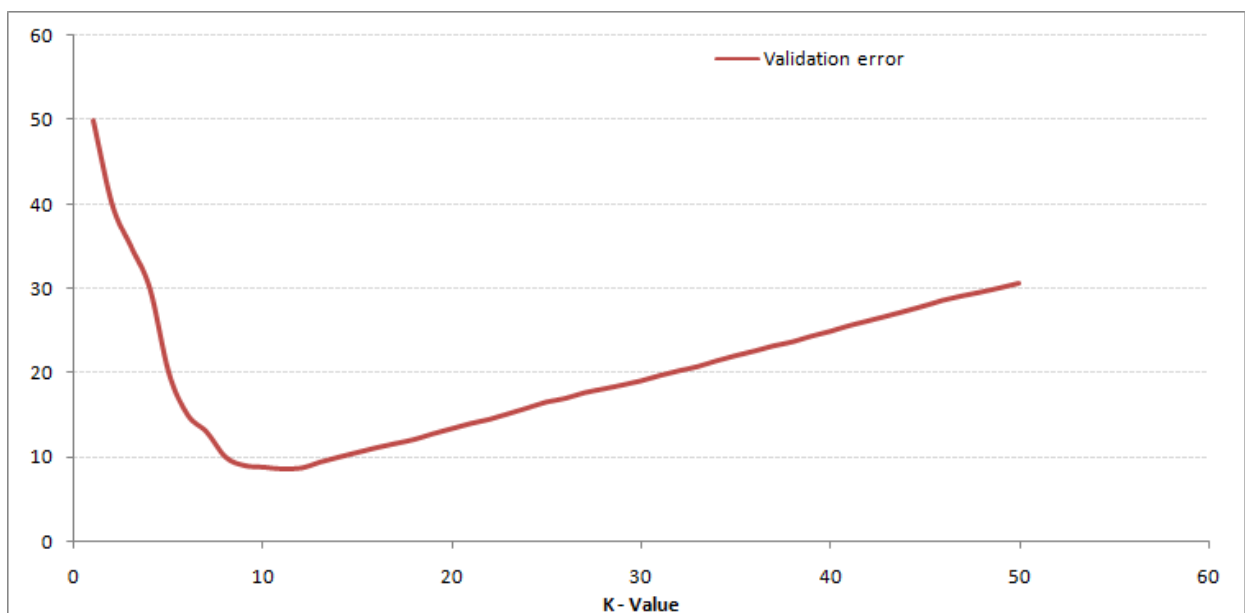
First let us try to understand what exactly does K influence in the algorithm. If we see the last example, given that all the 6 training observation remain constant, with a given K value we can make boundaries of each class. These boundaries will segregate RC from GS. The same way, let's try to see the effect of value "K" on the class boundaries. Following are the different boundaries separating the two classes with different values of K.



If you watch carefully, you can see that the boundary becomes smoother with increasing value of K. With K increasing to infinity it finally becomes all blue or all red depending on the total majority. The training error rate and the validation error rate are two parameters we need to access on different K-value. Following is the curve for the training error rate with varying value of K:



As you can see, the error rate at $K=1$ is always zero for the training sample. This is because the closest point to any training data point is itself. Hence the prediction is always accurate with $K=1$. If validation error curve would have been similar, our choice of K would have been 1. Following is the validation error curve with varying value of K :



This makes the story more clear. At $K=1$, we were over fitting the boundaries. Hence, error rate initially decreases and reaches a minimal. After the minima point, it then increases with increasing K . To get the optimal value of K , you can segregate the training and validation from the initial dataset. Now plot the validation error curve to get the optimal value of K . This value of K should be used for all predictions.

Breaking it down – Pseudo Code of KNN

We can implement a KNN model by following the below steps:

1. Load the data
2. Initialize the value of k
3. For getting the predicted class, iterate from 1 to total number of training data points
 1. Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
 2. Sort the calculated distances in ascending order based on distance values
 3. Get top k rows from the sorted array
 4. Get the most frequent class of these rows
 5. Return the predicted class

Conclusion

KNN algorithm is one of the simplest classification algorithms. Even with such simplicity, it can give highly competitive results. KNN algorithm can also be used for regression problems. The only difference from the discussed methodology will be using averages of nearest neighbors rather than voting from nearest neighbors.

CHAPTER – 4

RESULT AND DISCUSSION

The final product provides solution regarding crop selection based on the mineral content in the soil sample these sample data can be obtained from any convenient government agriculture centre who can provide data from sample. Usually, farmers require that data to condition the field for next plantation. Using that data our program can predict a better solution for different crop that is suitable to that field in its present state to regulate the usage of artificial fertilizers. Not only the circumstances of soil conditions but also the seasonal conditions are also to be taken not of for accurate results. In that case this program has rabi and kharif classification of crops on seasonal basis.

This program has two major features:

- 1) Prediction based on the minerals required
- 2) Prediction based on selected crop's required mineral values to the values the sample of soil actually have.

These features can not only suggest a crop for you but also can help you regulate mineral contents for user's desired crop selection.

Design and Implementation Constraints

Constraints in Analysis

- ◆ Constraints as Informal Text
- ◆ Constraints as Operational Restrictions
- ◆ Constraints Integrated in Existing Model Concepts

- ◆ Constraints as a Separate Concept
- ◆ Constraints Implied by the Model Structure

Constraints in Design

- ◆ Determination of the Involved Classes
- ◆ Determination of the Involved Objects
- ◆ Determination of the Involved Actions
- ◆ Determination of the Require Clauses
- ◆ Global actions and Constraint Realization

Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

Other Nonfunctional Requirements

Performance Requirements

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and accessing to the web service;

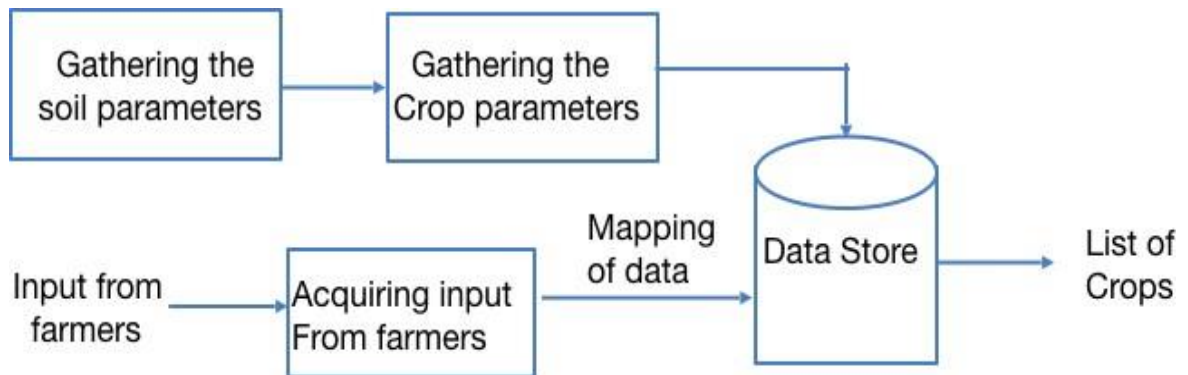
- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.
- Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

Safety Requirements

1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
6. Systems with different requirements for safety levels must be separated.
7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

CHAPTER 5

5.1 Architecture Diagram:



CHAPTER – 6

CONCLUSION

This program in its final stage is a combination of large datasets of minerals and pH values of actually crops collected to be as accurate as possible so that the users aren't supposed to face issue which can cause Hugh loss for an average farmer keeping that in observation there is to be a constant observation on the conditions for the crop for better results and also the nature conditions changing the entities of soil can also cause improper growth in crop to avoid such circumstances constant sample checks to maintain data accurate in order to predict a better solution is necessary. A model is proposed for predicting soil series and providing suitable crop yield suggestion for that specific soil. The model has been tested by applying different kinds of machine learning algorithm. Bagged tree and K-NN shows good accuracy but among all the classifiers, SVM has given the highest accuracy in soil classification. The proposed model is justified by a properly made dataset and machine learning algorithms. The soil classification accuracy and also the recommendation of crops for specific soil are more appropriate than many existing methods. In future, providing fertilizer recommendation is our concern, also data of other districts will be added to make this model more reliable and accurate. Main cause of this program is to suggest a best option. Based on the suggestions it is up to the user's choice to decide whether to rely on suggestion or to go with their own priority but in either cause this program gives it solution as accurate as possible.

APPENDICES

1. Sample Code:

```
from flask import Flask ,render_template,request,jsonify,session

from Crypto.Cipher import AES

from Crypto import Random

from pkcs7 import PKCS7Encoder

import sqlite3 as sql

import base64

import pandas as pd

from sklearn.preprocessing import LabelEncoder

#from flask_bootstrap import Bootstrap

import numpy as np

from sklearn.utils import shuffle

app = Flask(__name__)

app.secret_key = 'any random string'

@app.route('/', methods=['GET', 'POST'])

def home():

    return render_template('index.html')


def validate(username,password):

    con = sql.connect('static/chat.db')

    completion = False

    with con:
```

```

cur = con.cursor()

cur.execute('SELECT * FROM persons')

rows = cur.fetchall()

for row in rows:

    dbuser = row[1]

    dbpass = row[2]

    if dbuser == username:

        completion = (dbpass == password)

return completion

@app.route('/login', methods=['GET', 'POST'])
def login():

    error = None

    if request.method == 'POST':

        username = request.form['username']

        password = request.form['password']

        completion = validate(username,password)

        if completion == False:

            error = 'invalid Credentials. please try again.'

        else:

            session['username'] = request.form['username']

            return render_template('index111.html')

return render_template('index.html', error=error)

```

```

@app.route('/view', methods=['GET', 'POST'])

def view():

    return render_template('index111.html')

@app.route('/cropla', methods=['GET', 'POST'])

def cropla():

    return render_template('cropdetails.html')

@app.route('/about', methods=['GET', 'POST'])

def about():

    return render_template('About.html')

@app.route('/register', methods = ['GET','POST'])

def register():

    if request.method == 'POST':

        try:

            name = request.form['name']

            username = request.form['username']

            password = request.form['password']

            with sql.connect("static/chat.db") as con:

                cur = con.cursor()

                cur.execute("INSERT INTO persons(name,username,password) VALUES
(?,?,?)",(name,username,password))

                con.commit()

                msg = "Record successfully added"

```

```

except:

    con.rollback()

    msg = "error in insert operation"

finally:

    return render_template("index.html",msg = msg)

    con.close()

return render_template('register.html')

@app.route('/list')

def list():

    con = sql.connect("static/chat.db")

    con.row_factory = sql.Row

    cur = con.cursor()

    cur.execute("select * from persons")

    rows = cur.fetchall();

    return render_template("list.html",rows = rows)

@app.route('/crop_predict',methods=['GET','POST'])

def cr():

    lt= []

    lst = []

    data = pd.read_csv(r'dataset\Crop.csv')

    city = pd.read_csv(r'dataset\Ploted_600.csv')

    # data = shuffle(data)

```

```

y = data.loc[:, 'Crop']

labelEncoder_y = LabelEncoder()

y = labelEncoder_y.fit_transform(y)


data['crop_num'] = y

X = data.loc[:, ['N', 'P', 'K', 'pH']].astype(float)

y = data.loc[:, 'crop_num']


# Training Model

from sklearn.neighbors import KNeighborsClassifier

clf = KNeighborsClassifier(n_jobs=10, n_neighbors=10, weights='distance')

clf.fit(X, y)


if request.method == 'POST':

    city_name = request.form['city']

    N = request.form['Nitrogen']

    P = request.form['Phosphorous']

    K = request.form['Potassium']

    pH = request.form['pH']

    if len(city_name) != 0:

        print(city_name)

        npk = city[city['Location'] == city_name]

```

```

val = []

for index, row in npk.iterrows():

    val = [row['N'],row['P'],row['K'],row['pH']]

print(val)

columns = ['N','P','K','pH']

values = np.array([val[0],val[1],val[2],val[3]])

pred = pd.DataFrame(values.reshape(-1, len(values)),columns=columns)

prediction = clf.predict(pred)

# distances, indices = clf.kneighbors(pred, n_neighbors=10)

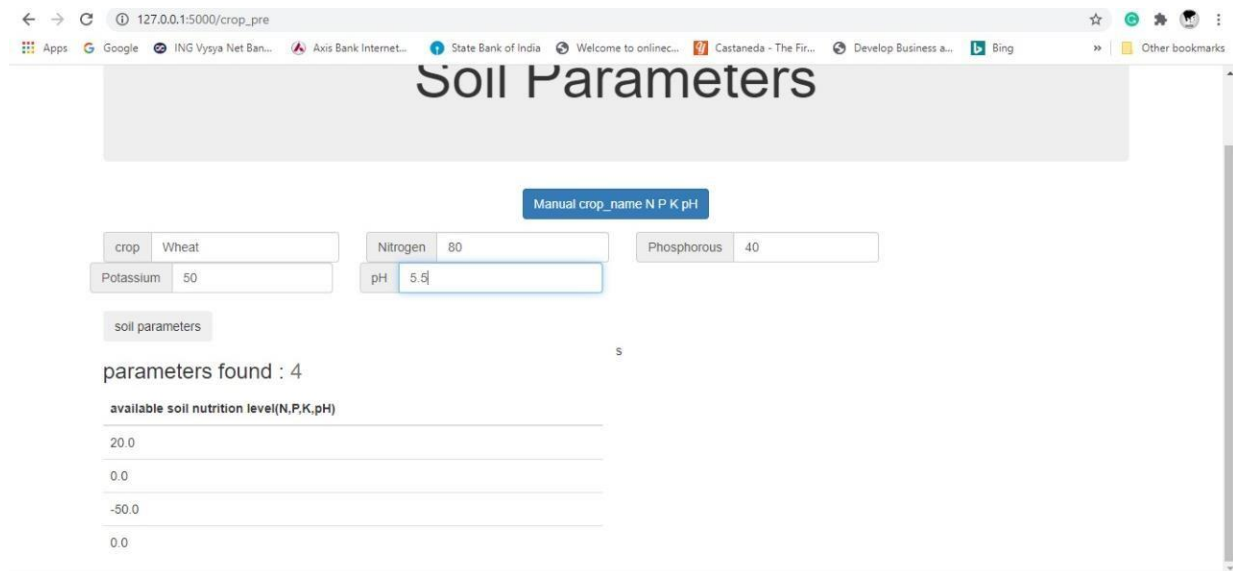
# prediction

real_pred = clf.predict_proba(pred)

print(real_pred)

```

2. SCREEN SHOTS



127.0.0.1:5000/crop_predict

City Wise Manual N P K pH

Vedaranyam

Predict Crop

Crop found : 10

Crop Name

Bajra(Pearl Millet)
Cluster Beans(Gavar)
Horse Gram(kulthi)
Mustard seeds
sesame seed
Cauliflower

127.0.0.1:5000/crop_predict

City Wise Manual N P K pH

Nitrogen 75 Phosphorous 65 Potassium 45 pH 5.5

Predict Crop

Crop found : 10

Crop Name

French Beans(Farasbi)
Fava beans (Papdi - Val)
Black eyed beans(chawli)
Coriander seeds
Jambun(Syzygium cumini)
Tamarind
Tapioca(Suran)
Asafoetida
Cloves

REFERENCES

- Niketa Gandhi et al , " Rice Crop Yield Forecasting of Tropical Wet and Dry Climatic Zone of India Using Data Mining Techniques",IEEE International Conference on Advances in Computer Applications (ICACA) , 2016.
- [2] K.E. Eswari. L.Vinitha. " Crop Yield Prediction in Tamil Nadu Using Baysian Network ", International Journal of Intellectual Advancements and Research in Engineering Computations, Volume-6,Issue-2,ISSN: 2348-2079.
- [3] Shruti Mishra, Priyanka Paygude, Snehal Chaudhary, Sonali Idate "Use of Data Mining in Crop Yield Prediction" IEEE Xplore Compliant – Part Number:CFP18J06-ART, ISBN:978-1-5386-0807-4; DVD PartNumber: CFP18J06DVD , ISBN:978-1-5386-0806-7.
- [4] Anna Chlingaryana, Salah Sukkarieha, Brett Whelanb — Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review , Computers and Electronics in Agriculture 151 (2018) 61-69,Elisver,2018.
- [5] Dakshayini Patil et al,"Rice Crop Yield Prediction using Data Mining Techniques:An Overview", International Journal of Advanced Research in Computer Science and Software Engineering ,Volume 7, Issue 5, May 2017.
- [6] A.T.M Shakil Ahamed, Navid Tanzeem Mahmood, Nazmul Hossain, Mohammad Tanzir Kabir, Kallal Das, Faridur Rahman, Rashedur M Rahman "Applying Data Mining Techniques to Predict Annual Yield of Major Crops and Recommend Planting Different Crops in Different Districts

in Bangladesh” 978-1-4799-8676-7/15/\$31.00 copyright 2015 IEEE SNPD 2015, June 1-3 2015.

[7] K. Menaka, N.Yuvaraj, Indian Journal of Innovations and Developments Vol 5(12), October, 2016 ISSN (online) : 2277-5390 ISSN (Print): 2277-5382

[8] Machine Learning by Tom M. Mitchell, ISBN: 0070428077

[9] Snehal S.Dahikar, Dr.Sandeep V.Rode , "Agricultural Crop Yield Prediction Using Artificial Neural Network Approach ", International journal of innovative and research in electrical, instrumentation and control engineering, volume 2, Issue 2, 2014.

[10] D Ramesh, B Vishnu Vardhan, "Analysis of crop yield prediction using data mining techniques", International Journal of research in engineering and technology, eISSN:2319-1163.

[11] Ramesh A. Medar, "A survey on data mining techniques for crop yield prediction", International Journal of advance in computer science and management studies, ISSN:2231-7782, volume 2, Issue 9, 2014.

[12] Pooja M C et al, "Implementation of Crop Yield Forecasting Using Data Mining", International Research Journal of Engineering and Technology (IRJET), 2018, p-ISSN: 2395-0072

[13] Jig Han Jeong et al, "Random Forests for Global and Regional Crop Yield Predictions", PLOS-ONE DOI:10.1371/journal.pone.0156571 June 3, 2016.

[14] Jharna Majumdar, Sneha Naraseeyappa and Shilpa Ankalaki, Analysis of agriculture data using data mining techniques: application of big data, Springer journal, 2017.

- [15] S.kanaga Subba Raju et al, Demand based crop recommender system for farmers, International Conference on Technological Innovations in ICT For Agriculture and Rural Development, 2017.
- [16] Shriya Sahu et al," An Efficient Analysis of Crop Yield Prediction Using Hadoop framework based on random Forest approach, International Conference on Computing, Communication and Automation , 2017.
- [17] E Manjula and S. Djodkitachoumy International journal of computation Intelligence and informatics, Vol 6:No 4 March 2017