ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

Crop Recommendation Predictor using Machine Learning for Big Data

[1] Manish Kumar, [2] Dr. (Prof.) Deva Prakash

[1] Research Scholar, Department of Mathematics & Computer Science, Magadh University, Bodhgaya, Bihar, India.
[2] Associate Professor, Head of Department, Department of Mathematics, S.M.D. College, Punpun, Bihar, India.
Corresponding Author Email: [1] maneeshkumarphd@gmail.com, [2] deva.prakashpatna@gmail.com

Abstract*— Agriculture is India's rural population's livelihood. It's also essential for human existence, and it's developed over time to suit the demands of an ever-growing human population. Agriculture is their main source of income in rural area of India. Agriculture is the primary employment of the majority of Indians, as we all know. The majority of Indians rely on agriculture for their living, either openly or implicitly. The great majority of Indian farmers trust on their intuition to determine which crop to plant in a given season. Farmers are accustomed to sowing the same crop, using more fertilizer, and following public opinion. They find comfort in just following historical agricultural traditions and standards, oblivious to the reality that crop yield is highly dependent on current weather, soil, and other variables. The most frequent problem faced by Indian farmers is that they do not select the appropriate crop based on their soil requirements and other factors such as fertilizer s and irrigation patterns. As a result, productivity is impacted. However, a single farmer cannot be expected to take into consideration all of the numerous elements that influence crop development before deciding which one to plant. This problem can be efficiently addressed with machine learning. There have been major advancements in how machine learning may be employed in many businesses and studies during the last several years. As a result, we intend to develop a model/system that will allow machine learning to be utilised in agriculture to benefit farmers.

***Keywords* — Big Data, Big Data Analytics, Machine Learning, Modeling using Machine Learning Algorithms, Rural agricultural systems, Precision Agriculture, R**

I. INTRODUCTION

Small farms are widespread in India. More than 75% of the country's total land capitals are smaller than 5 acres. Agriculture consumes over 80% of the present water supply. Irrigated land makes for roughly 48.8 percent of India's 140 million hectares (mha) of agricultural land. Rainwater is used to irrigate the remaining 51.2 percent.[1] Only around 48.8 percent of the land is irrigated, thus most crops are rain-fed. According to some estimates, agriculture employs around 55 percent of India's overall population. For farmers, deciding what to cultivate and when to sow has always been as much an art as a science. The importance of a country's Gross Domestic Product (GDP) and its ability to feed and equip its own people, as well as exporting raw resources and finished items, is important for a healthy and flourishing economy.

The proportion of GDP growth based on year-over-year data is by no means a reliable indicator of progress. Agriculture and associated industries (subareas include agriculture, livestock, forestry, and fisheries) will always be a major contribution to any economy across the world, hence industrialization cannot be dubbed the pathfinder in an integrated economy. In terms of percentage of GDP, agriculture and related industries accounted for 20.19 percent in 2020-21. [2]

India is one of the world's largest agricultural producers, yet farm productivity remains low. Farmers need to boost productivity so that they may obtain more money from the same amount of land with less labor.

Crop production may be a difficult process that is impacted by soil and other environmental input characteristics.

Agriculture input parameters differ from one field to the next and from one farmer to the next. Collecting such information across a larger area might be a difficult endeavor. Massive information sets like this may be utilized to forecast their impact on main crops in a certain area or location. Precision agriculture makes it possible. Precision farming, as the name indicates, is providing an exact and appropriate amount of fertilizer, soil, and other inputs to the crop at the right moment in order to boost production and yields. Precision agricultural systems aren't all created equal. However, in agriculture, it is critical that the advice offered are exact and precise, as mistakes

can result in significant material and financial loss. Many studies are being conducted in attempt to develop an accurate and efficient crop prediction system. Ensembling is one of the techniques used in these types of studies. Agriculture scientists and researchers throughout the world have created and analyzed a variety of forecasting approaches in the field of agriculture and its allied disciplines.

This paper presents a model that employs machine learning to construct an efficient and accurate model, among the different machine learning approaches that are being employed in this industry.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

II. REVIEW OF LITERATURE

The needs and strategy needed for designing a software model for precision farming are covered in this article [3]. It explores into the fundamentals of precision farming. The authors begin with the fundamentals of precision farming before moving on to constructing a model to support it. This research offers a model that uses Precision Agriculture (PA) concepts to manage variability on small, open farms at the individual farmer and crop level. The model's overall goal is to provide direct advising services to even the tiniest farmer at the level of his or her smallest crop plot, utilizing the most accessible technologies available, such as SMS and email. This model was developed for the situation in Kerala, where the average holding size is significantly lower than the rest of India. As a result, with minor alterations, this model may be used in other parts of India.

The research [4] examines assortment algorithms and how well they work in yield prediction in precision husbandry. These algorithms are used to forecast yield on a soya bean crop using data collected over multiple years. Support Vector Machine, Random Forest, Neural Network, REPTree, Bagging, and Bayes are the methods employed in this article for yield prediction. The result reached at the end is that, among the above-mentioned algorithms, bagging is the best for yield prediction since its error deviation is the smallest, with a mean absolute error of 18985.7864.

The importance of crop selection is covered in the study [5], as well as the elements that influence crop selection, such as production rate, market price, and government policy. This work provides a Crop Selection Method [1] (CSM), which addresses the crop selection problem while also increasing the crop's net yield rate. It recommends a sequence of crops to be planted throughout the course of a season, taking into account weather, soil type, water density, and crop type. The accuracy of CSM is determined by the expected value of important parameters. As a result, a prediction approach with increased accuracy and performance is required.

To categorize the liver disease data set, the study suggests numerous classification approaches. Because accuracy is dependent on the dataset and the learning process, the study underlines the importance of accuracy. To diagnose these disorders and assess their efficacy and correction rate, classification methods such as Nave Bayes, ANN, ZeroR, and VFI were utilized. The accuracy and computing time of the models were compared to see how well they performed. Except for naive bayes, all of the classifiers demonstrated increased prediction accuracy.

In article, the soil records are evaluated and a category is predicted. The crop yield is indicated as a Classification rule based on the expected soil category. Crop yield prediction is done using the Nave Bayes and KNN algorithms. The goal of future study is to develop efficient models employing a variety of classification approaches, such as support vector machines and principal component analysis.

III. OBJECTIVE OF THE PAPER

The article's goals are listed below.

- The goal is to utilize Exploratory Data Analysis (EDA) to figure out how different farming factors impact crop productivity.
- Obtaining an estimate of the harvestable (economic) yield is one of the primary goals of developing operational crop models. Crop management, such as fertilization, cultivation, irrigation, and plant protection, are frequently included in the evaluation of crop development and yield response.
- The model's overall goal is to provide direct advising services to even the tiniest farmer at the level of his or her smallest crop plot, utilizing the most accessible technologies available, such as SMS and email. Models, on the other hand, are rarely employed for early warnings or damage prevention in extreme weather events and natural catastrophes. Governments are now adopting initiatives such as sending SMS messages to farmers.
- It's a recommendation prediction model based on a classifier and classifier tuning. The system suggests crops based on relevant factors. Using a technology-based crop recommendation system in agricultural choices can assist farmers increase crop yields or cultivate appropriate crops based on their land features and climatic conditions.

IV. METHODOLOGY FOR MODEL CREATION

The objectives of this work are to create a machine learning model that can propose crops as part of a crop recommendation system based on technology. The model is designed to be used as a reference tool for farmers, and it will assist them in deciding which crop to plant in order to reduce risk and increase agricultural production.

The approach used for this research is as follows:

a) Collect the data from the public data sources such as www.data.gov.in, www.data.world.

b) Data exploration and analysis.

c) Creating an appropriate input dataset that meets the machine learning algorithms' criteria

d) Gather information for training and testing.

e) Identification of numerous factors previously employed in similar analyses

f) Predictive modelling employing multiple predictors discovered

g) Model evaluation for various predictors and selection of the best model

h) Conclusions and interpretation of the findings

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

V. PROPOSED METHODOLOGY AND CROP RECOMMENDATION MODEL

Following figure depicts the numerous processes involved in this research topic:

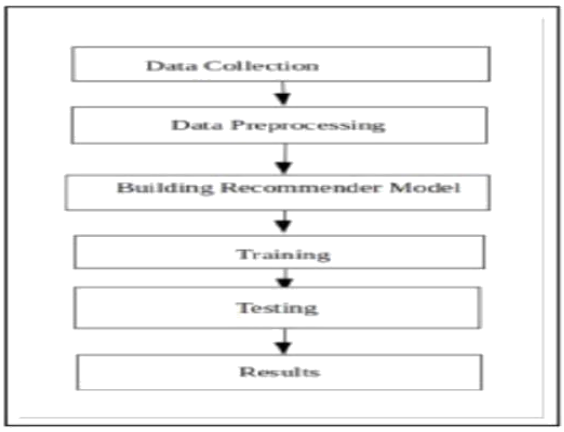


Fig 1: Overview of Methodology

Fig 2: The proposed model for Crop Recommendation

VI. ALGORITHMS APPLIED TO DEVELOP A MODEL

It is decided to explore the use of the following algorithms for the development of the model

- Support Vector Machine (SVM) Linear, Radial and Polynomial
- Gaussian Naive Bayes
- Random Forest
- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)
- Mixture Discriminant Analysis (MDA)
- Classification And Regression Tree (CART)/ Decision Tree
- K- nearest neighbors (KNN)

VII. PROPOSED SYSTEM/PREDICTOR

We apply supervised learning in the proposed system to create a model that predicts crop. Figure 1 shows the steps of the proposed system, which include dataset collection, preprocessing, feature selection, and the application of machine learning algorithms/modules.

A. Dataset Collection: Data is obtained from many sources and organized into data sets. This information is then used in descriptive analysis. Several online abstract sources, such as Kaggle.com and data.gov.in, offer data. Crop recommendation dataset, which incorporates eight variables, including N, P, K, rainfall, ph, humidity, temperature, and label, was utilized in this research.

B. Preprocessing step: This is a crucial phase in the machine learning process. Inserting or removing missing values, selecting the right data range, and extracting functionality are all part of preprocessing. The type of dataset used in the research is crucial.

C. Feature Selection: To constitute a large data set, feature extraction should reduce the amount of data required. The very last set of training is made up of the soil and crop parameters extracted during the pre-treatment phase. The physical and chemical qualities of the soil are among these features. For feature selection, we utilized the Random Forest Classifier algorithm. This approach chooses features based on their values recorded, which means that the characteristic with the highest values recorded is chosen as an essential feature for yield prediction.

D. Split the Dataset into Train and Test Set: This process comprises input data training and testing. With a division ratio of 80 percent or 20 percent, such as 0.8 or 0.2, the loaded data is separated into two groups, such as training data and test data. A classifier is used to construct the available input data in a learning set. To approximate and classify the function, construct the classifier's support data and assumptions in this phase. The data is tested during the test phase. Preprocessing generates the final data, which is then processed by the machine learning module.

E. Applying Machine Learning models: In our research, we employed a variety of machine learning methods to anticipate crop recommendations, as follows:

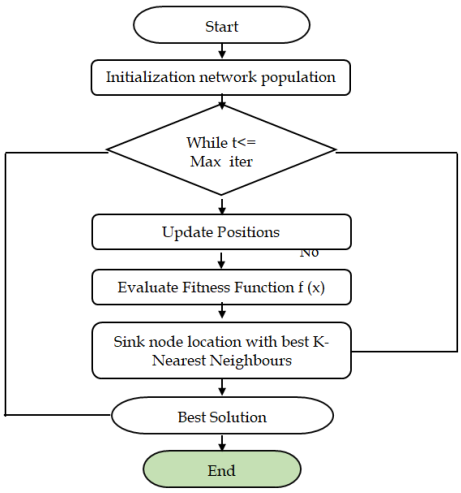
(1) KNN Algorithm

KNN is a nonparametric supervised learning algorithm that divides data points into categories using training sets. The term accumulates information from all educational examples and similarities based on the new case in easy categories. Examine the training for the K examples that are the most similar (neighbors) and forecast the new instance (x) by summing the output variables for these K cases. The class value mode is known as classification (or most commonly). Figure 3 shows a flow diagram of the KNN algorithm.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022



(2). Support Vector Machines

A supervised machine learning approach called Support Vector Machine (SVM) is utilized for both classification and regression. Although we often refer to regression concerns, categorization is the most appropriate term. Finding a hyperplane in an N-dimensional space that clearly classifies the data points is the goal of the SVM method. The number of features determines the hyperplane's size. The hyperplane is essentially a line if there are just two input characteristics. The hyperplane turns into a 2-D plane if there are three input characteristics. Imagining something with more than three characteristics gets challenging. The data is divided into decision surfaces by SVM. The data is further divided into hyper planes of two classes by the decision surface. The supporting vector that defines the hyper plane is defined by the training points. Because of the greater margins, the generalization of classifiers is limited. Probably, a hyper plane with the largest distance to the nearest learning data point has better margins and larger mistakes. Figure 4 depicts the SVM flow chart, which depicts the processes involved in the SVM algorithm.

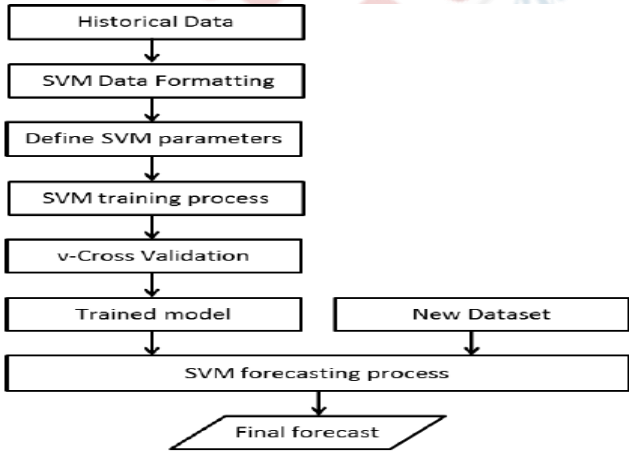


Fig.4: Flow chart for SVM algorithm

(3). Decision Tree

An algorithmic method for dividing a data collection into segments based on several criteria is used to build decision trees. It is among the most popular and useful techniques for supervised learning. A non-parametric supervised learning technique called decision trees is utilized for both classification and regression applications. Classification trees are tree models where the goal variable can take a discrete set of values. Regression trees are decision trees when the target variable can take continuous values (usually real numbers). This is referred to as a Classification and Regression Tree (CART). In each iteration of the process, the entropy H (S) (or information gain IG (S)) of each unused characteristic in set S is computed. After that, choose the property with the lowest entropy value (or the largest information gain). The set S is then subdivided into the qualities that have been chosen. The procedure is repeated in each subgroup, but only those traits that were not previously picked are considered.

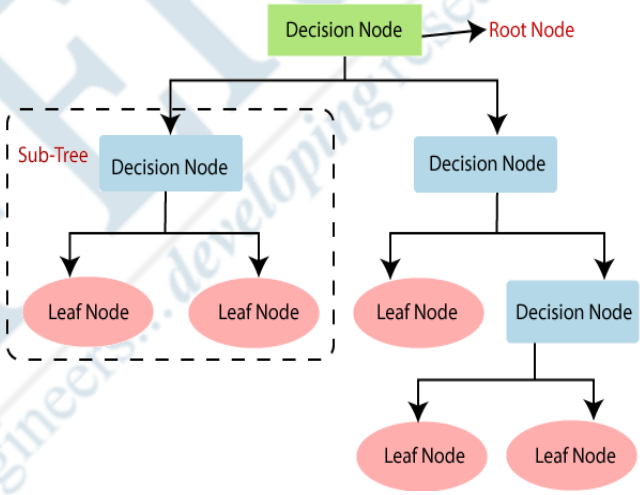


Fig.5: Flow chart for Decision Tree algorithm

4). Naive Bayes Classifier

The Bayes Theorem is used to create a Naive Bayes classifier. It calculates membership probabilities for each class, such as the likelihood that a certain record or data point belongs to that class. The most likely class is defined as the one having the highest probability. Maximum A Posteriori is another name for this (MAP).

The Naive Bayes classifier makes the assumption that all of the characteristics are unrelated. The presence or absence of one trait has no bearing on the presence or absence of another. We may take an example from Wikipedia to demonstrate the logic: A fruit is called an apple if it is red, round, and around 4 inches in diameter. A naive Bayes classifier considers all of these traits to independently contribute to the likelihood that this fruit is an apple, even if they are reliant on each other or on the existence of other features.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

We test a hypothesis given numerous pieces of evidence in real-world datasets (feature). As a result, computations become more difficult. To make things easier, the feature independence technique is utilized to 'uncouple' various pieces of evidence and consider them as separate entities.

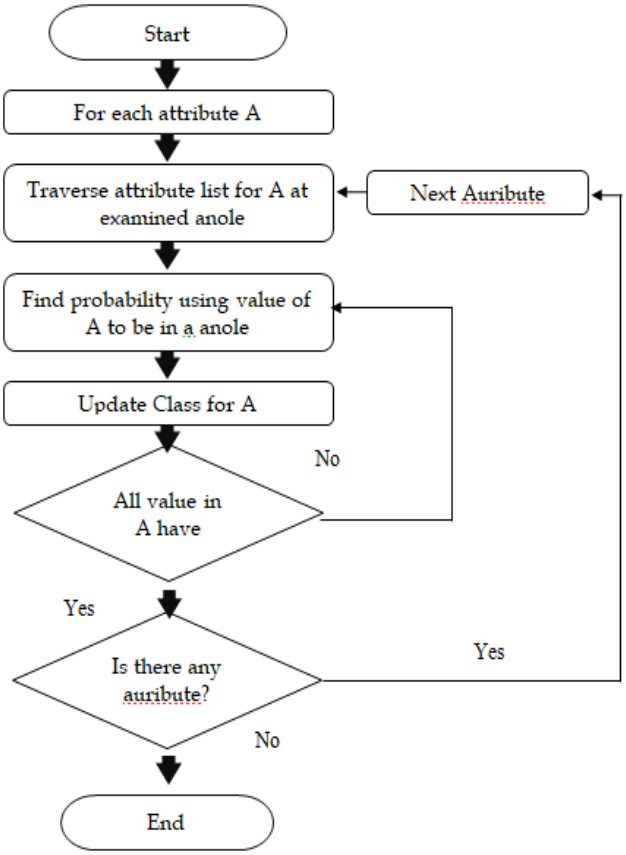


Fig 6: Flow chart for Naïve Bayes algorithm

(5). The Random Forest Classifier

Random forest is a learning method that is supervised. It creates a "forest" out of an ensemble of decision trees, which are commonly trained using the "bagging" approach. The bagging method's basic premise is that combining several learning models improves the final output. As the name indicates, a random forest is made up of a huge number of individual decision trees that work together as an ensemble. Random forest is a versatile, easy-to-use machine learning technique that, in most cases, gives excellent results even without hyper-parameter adjustment. Because of its simplicity and versatility, it is also one of the most often used algorithms (it can be used for both classification and regression tasks). Each tree in the random forest produces a class prediction, and the class with the most votes becomes the prediction of our model.

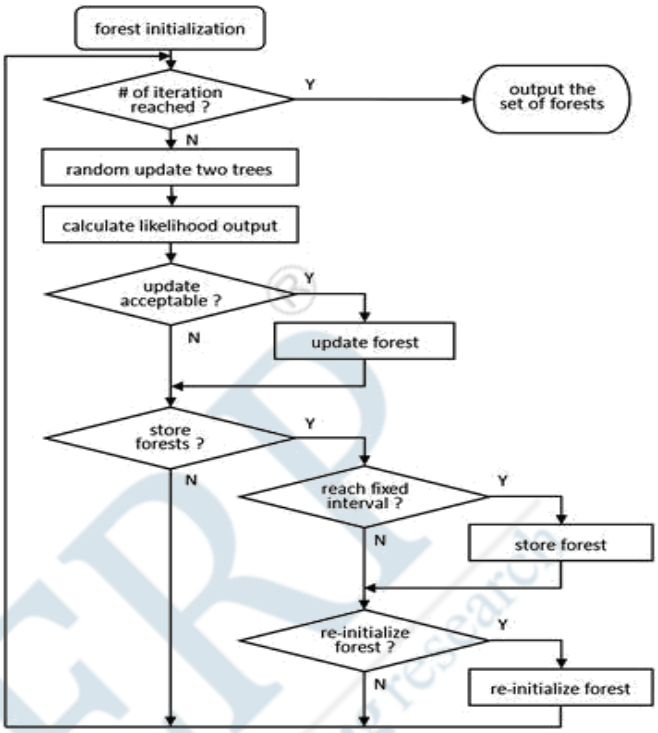


Fig.7: Flow chart for Random Forest Classifier

(6). Linear Discriminant Analysis

LDA (Linear Discriminant Analysis) is a technique for reducing dimensionality. It's utilized in Machine Learning and pattern categorization applications as a pre-processing phase. The purpose of LDA is to project features from a higher-dimensional space onto a lower-dimensional space, avoiding the dimensionality curse while simultaneously saving resources and reducing dimensional costs. Discriminant analysis is a statistical technique for predicting the likelihood of belonging to a certain class (or category) based on one or more predictor factors. It works with predictor variables that are either continuous or categorical.

We previously discussed logistic regression in the context of two-class classification issues, in which the outcome variable has two potential values (0/1, no/yes, negative/positive).

In situations when the outcome variable has more than two classes, discriminant analysis is better than logistic regression at predicting the category of an observation. For multi-class classification situations, it's also more stable than logistic regression.

The following approaches of discriminant analysis will be discussed:

- Linear discriminant analysis (LDA): Predicts the class of a given observation using linear combinations of predictors. Assumes that the predictor variables (p) are normally distributed, and that the classes have the same variances (for univariate analysis, p = 1) or covariance matrices (for multivariate analysis, p > 1).

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

- Quadratic discriminant analysis (QDA) is more versatile than linear discriminant analysis (LDA). The covariance matrix of classes is not assumed to be the same in this case.
- MDA (mixture discriminant analysis) assumes that each class is a Gaussian mixture of subclasses.
- Flexible Discriminant Analysis (FDA): Splines and other non-linear combinations of predictors are utilized.
- Regularized discriminant analysis (RDA): When the number of predictors exceeds the number of samples in the training data, regularization (or shrinkage) increases the estimation of the covariance matrices. As a result, the discriminant analysis is improved.

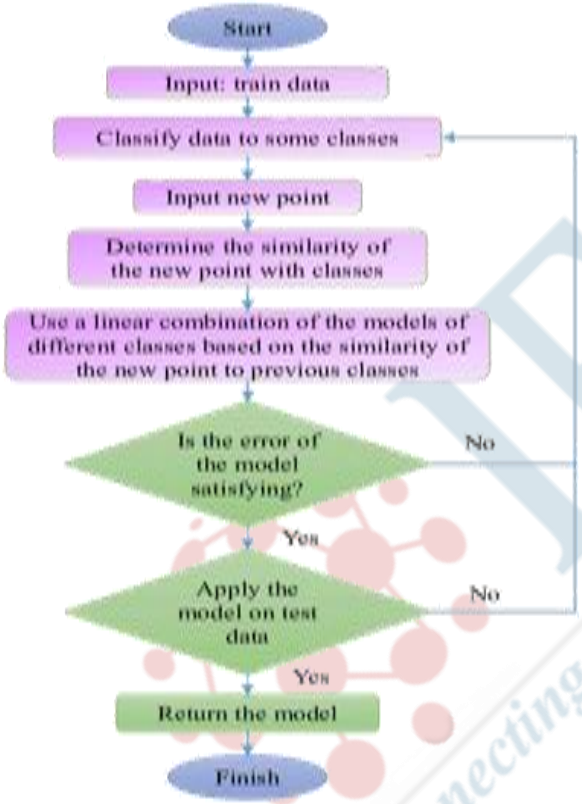


Fig.8: Flowchart of LDA

After the model has been properly trained, it is put to the test on the Testing dataset, which differs from the Training dataset in sample values.

VIII. MODEL IMPLEMENTATION STEPS

We're working on a real-time model to forecast the crop that has to be planted by the farmer.

• The first stage is to analyze historical data in order to build a machine learning model.

• The second step employs machine learning models to provide farmers with crop estimates.

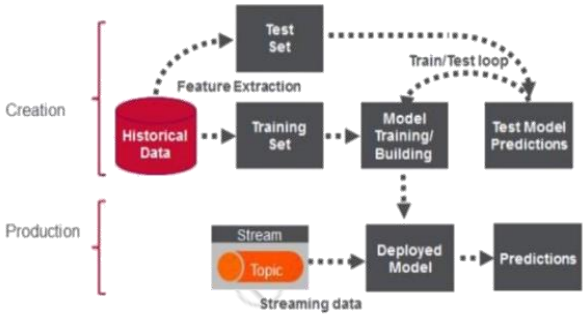


Fig 9: Evaluating System

IX. INTRODCUTION OF R PROGRAMMING LANGUAGE

R is a commonly used statistical software and data analysis tool that is written in an open-source programming language. R comes with a command-line interface by default. R is available on a variety of platforms, including Windows, Linux, and Mac OS X. In addition, the R programming language is the most up-to-date tool.

The R Development Core Team is now working on it, which was conceived by Ross Ihaka and Robert Gentleman at the University of Auckland in New Zealand. The R programming language is a fork of the S programming language. It also works with Scheme-inspired lexical scoping semantics. Furthermore, the project began in 1992, with a first version released in 1995 and a stable beta version issued in 2000.

It's a free, open-source language. This implies that it may be installed in any company without the need for a license. R is a popular programming language for machine learning, statistics, and data analysis. R makes it simple to build objects, functions, and packages. It's a platform-agnostic programming language. As a result, it may be used with any operating system.

R is a programming language that may be used to create statistics as well as interact with other languages (C, C++). As a result, we may readily connect with a wide range of data sources and statistical software.

The R programming language has a large user base that is expanding every day.

R is one of the most popular programming languages in the Data Science employment market, making it the trendiest trend right now.

Features of R Programming Language

i. Statistical Features of R

1. *Basic Statistics:* The mean, mode, and median are the most often used words in basic statistics. "Measures of Central Tendency" is the name given to all of them. As a result, we can simply quantify central tendency using the R programming language.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

2. *2). Static graphics:* R has a lot of tools for producing and developing cool static visuals. Many plot kinds are supported by R, including graphic maps, mosaic plots, biplots, and the list goes on.
3. Probability *distributions:* Probability distributions are important in statistics, and we can easily handle a variety of probability distributions using R, including the Binomial Distribution, Normal Distribution, Chi-squared Distribution, and many others.

ii. Programming Features of R

1). R Packages: R has a large number of libraries available, which is one of its strongest qualities. CRAN (Comprehensive R Archive Network) is a repository for R that contains over 10,000 packages.

Distributed Computing: Distributed computing is a method of improving efficiency and performance by distributing software components over numerous computers. In November 2015, two new R packages for distributed programming, ddR and multidplyr, were published.

iii. Programming in R

1). R is easy to develop and understand since it is syntactically comparable to other commonly used languages. R programme may be developed in a variety of IDEs, including R Studio, Rattle, Tinn-R, and others.

iv. Advantages of R

The statistical analysis package R is the most complete. R is generally the first place where new technologies and concepts arise.

I. R is an open-source programming language. As a result, we can run R at any time and from any location.
II. The R programming language is compatible with both GNU/Linux and Windows.
III. R is a cross-platform programming language that works on any operating system.
IV. Anyone may contribute new packages, bug fixes, and code upgrades to R.

v. Disadvantages of R

• Some packages in the R programming language have a less-than-ideal standard.

• R instructions, on the other hand, put minimal strain on memory management. As a result, the R programming language may use up all of the available RAM.

• In R, there's no one to blame if anything doesn't work.

vi. *Applications of R*

• For data science, we utilize R. It provides us with a wide range of statistics-related libraries. It also serves as a platform for statistical computation and design.

• Many quantitative analysts utilize R as their programming language. As a result, it aids in data import and cleansing.

• R is the most widely used programming language. It's used by a lot of data analysts and research programmers. As a result, it is employed as a basic financial tool.

• R is currently used by tech giants such as Google, Facebook, Bing, Accenture, Wipro, and many others.

X. DESCRIPTION OF DATASET USED IN THE PAPER

The dataset utilized in this work was created by supplementing and integrating numerous publically accessible Indian datasets such as Fertilizer Ratio, Temperature, and Soil Ph Value, among others. Unlike the intricate characteristics determining crop productivity, this data is quite basic, with only a few yet useful features.

The soil's nitrogen, phosphorus, potassium, and pH readings are included in the data. It also includes the humidity, temperature, and rainfall requirements for a given crop.

Data fields

N - ratio of Nitrogen content in soil
P - ratio of Phosphorous content in soil
K - ratio of Potassium content in soil
temperature - temperature in degree Celsius
humidity - relative humidity in %
ph - ph value of the soil
rainfall in mm

Fertilizer Numbers – What Is NPK

We are greeted with a bewildering array of fertilizer alternatives while standing in the fertilizer aisle of a garden or farm store, many of which have a sequence of three digits such as 10-10-10, 20-20-20, 10-8-10, or many other combinations of numbers. "What do the numbers on fertilizer mean?" we might wonder. These are NPK values, which begs the question, "What exactly is NPK?"

What Do Fertilizer Numbers Indicate? The value of the three macro-nutrients consumed by plants is represented by the three numbers on fertilizer. Nitrogen (N), phosphorous (P), and potassium (K), or NPK for short, are the macronutrients in question. The greater the number, the higher the nutrient concentration in the fertilizer.

Fertilizer with the value 20-5-5, for example, has four times more nitrogen than phosphate and potassium. The concentration of all three nutrients in a 20-20-20 fertilizer is double that of a 10-10-10 fertilizer. The fertilizer figures may be used to figure out how much fertilizer is needed to add 1 pound (453.5 g) of the nutrient we're seeking to add to the soil. If the fertilizer's numbers are 10-10-10, we can divide 100 by 10, which tells us that we'll need 10 pounds (4.5 kg) of fertilizer to supply 1 pound (453.5 g) of nutrition to the soil.

If the fertilizer numbers are 20-20-20, we divide 100 by 20

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

to find out how much fertilizer it will take to provide 1 pound (453.5 g) of nutrient to the soil. The other values in a fertilizer containing only one macro-nutrient will be "0." If a fertilizer is 10-0-0, for example, it exclusively includes nitrogen.

These fertilizer numbers, commonly known as NPK levels, should be printed on every fertilizer we buy, whether it's an organic or chemical fertilizer.

What is NPK and Why is it Important to recommend a crop?

Now that we understand what the numbers on fertilizer indicate, we must understand why NPK is so crucial to our plants. To thrive, all plants require nitrogen, phosphorus, and potassium. A plant will die if it lacks one or more of these nutrients. Nitrogen (N) - Nitrogen is responsible for the majority of the plant's leaf growth. Phosphorus (P) - Phosphorus is essential for root development as well as flower and fruit development. Potassium (K) - Potassium is a nutrient that aids in the proper functioning of the plant's overall activities. Knowing a fertilizer's NPK levels can assist us in selecting one that is suited for the sort of plant we are cultivating.

If we are producing green vegetables, for example, we may wish to use a fertilizer with a greater nitrogen content to enhance leafy growth. If we're growing flowers, a fertilizer with a larger phosphorus content may be used to produce more blossoms. We should get our soil tested before applying fertilizer to our plant beds. This will also assist us in determining the optimum fertilizer number balance for our garden's soil demands and deficits.

EDA report of dataset used in this paper

With the use of descriptive statistic and graphical representation, exploratory data analysis refers to the critical investigative process of doing first investigations on data in order to uncover patterns, spot anomalies, test hypotheses, and check assumptions. EDA is generally used to investigate what data might disclose outside of formal modelling or hypothesis testing tasks, and to gain a better knowledge of data set variables and their interactions. It can also aid in determining whether the statistical approaches we're contemplating for data analysis are suitable.

Basic statistics about dataset used in this paper

Basic Statistics

Raw Counts

Name	Value
Rows	1,200
Columns	8
Discrete columns	1
Continuous columns	7
All missing columns	0
Missing observations	0
Complete Rows	1,200
Total observations	9,600
Memory allocation	58.3 Kb

Fig10: Basic statistics about dataset used in this paper

(a). Data structure of dataset

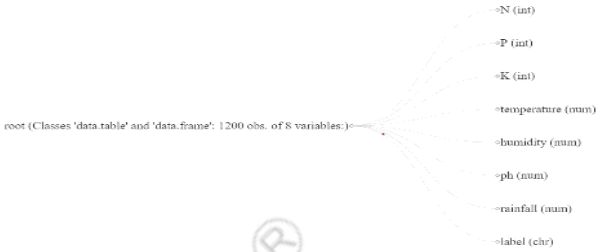


Fig.11: Data structure of dataset

(b). Univariate Distribution of dataset

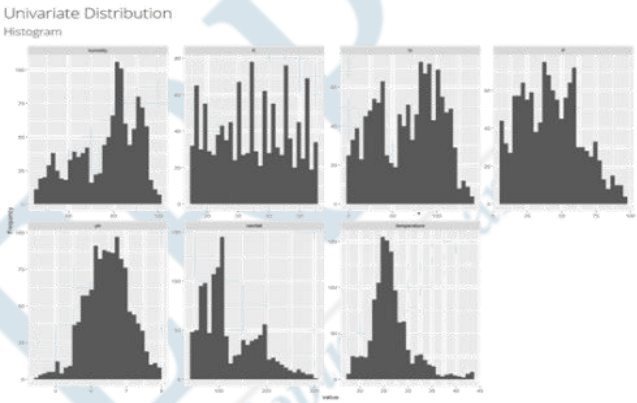


Fig.12: Univariate Distribution of dataset

(c). Bar chart with frequency of target variables

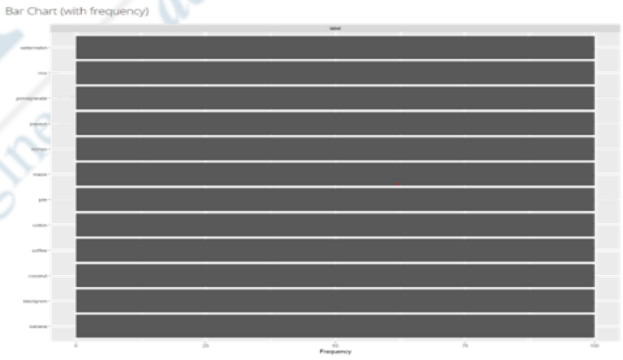


Fig.13: Bar chart with frequency of target variables

(d). QQ Plot

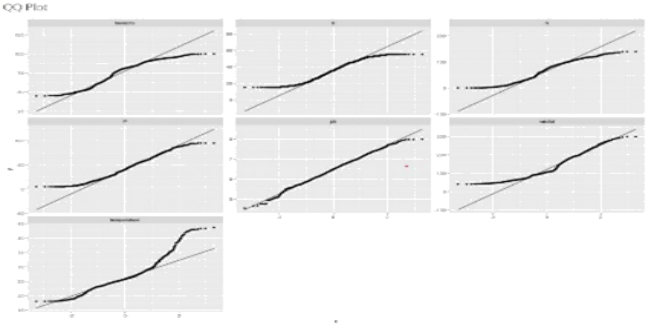


Fig.14: QQ Plot

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

(e). QQ Plot by label (Target Variable)

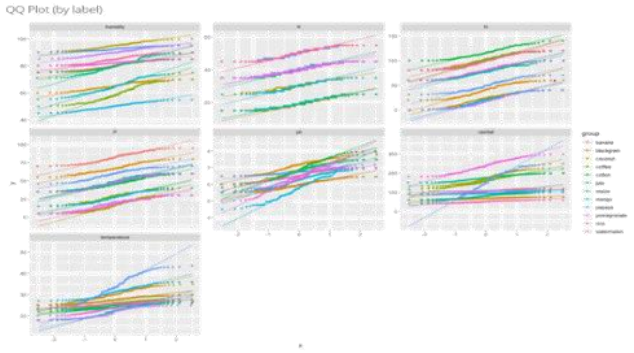


Fig.15: QQ Plot by label

(f). Correlation Analysis of independent and dependent variable

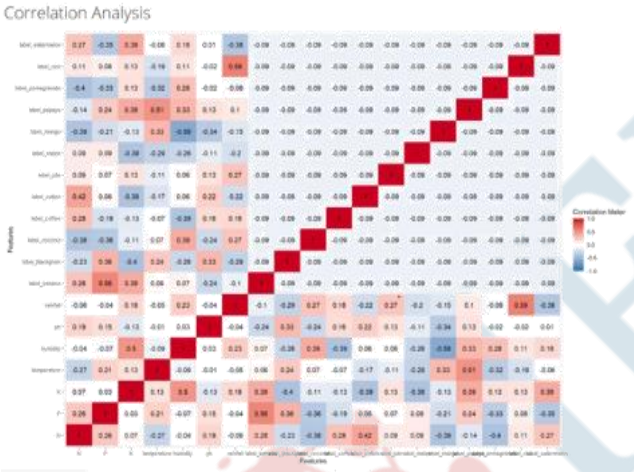


Fig.16: Correlation Analysis of independent and dependent variable

(g). Relative importance with features

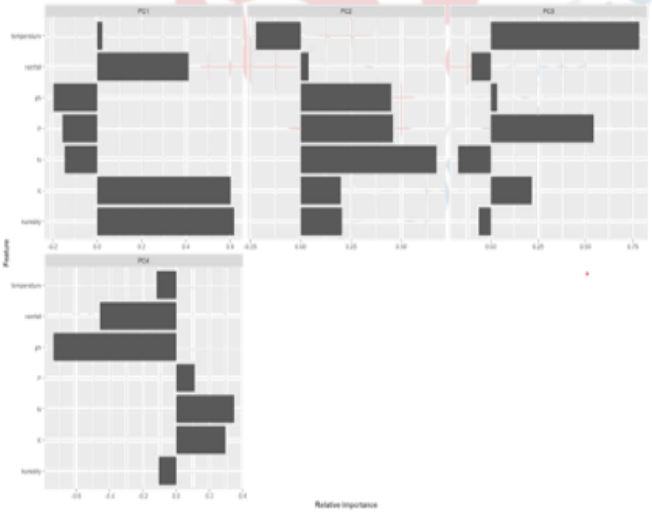


Fig. 17: Relative importance with features

(h). Bivariate Distribution

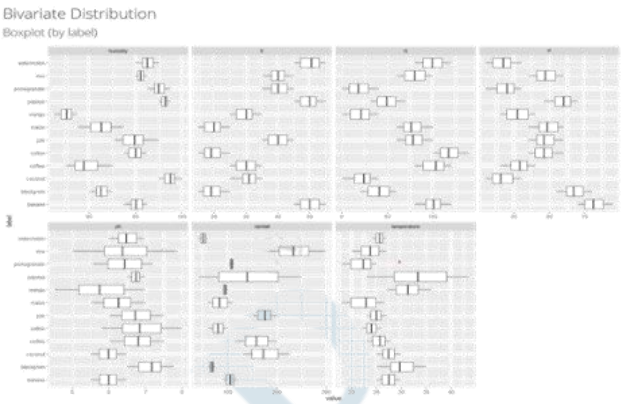


Fig. 18: Bivariate Distribution

XI. SAMPLE OF CODE IN R

The sample code is embedded in this part of the paper. This sample code assists in solving and interpreting the prediction models for variety of data sets.

PREPROCESSING OF DATASET IN R

```
Install the package DataExplorer ----------
Install the package tidyverse ----------
Install the package skimr ----------
Install the package flextable ----------

Load the library DataExplore----------
library(DataExplorer)
Load the library tidyverse----------
library(tidyverse)
Load the library skimr ----------
library(skimr)
Load the library flextable ----------
library(flextable)

loading or reading the dataset for crop recommendation ----------
dataset_crop_n <- read.csv("Crop_Recomm_n.csv")
```

#To see every column in a data frame.

```
glimpse(dataset_crop_n)
output
Rows:
2,200
Columns
: 8
                <int> 90, 85, 60, 74, 78, 69, 69, 94, 89, 68, 91, 90, 78,
$ N             93, 94, 60, 85, 91, 77~
                <int> 42, 58, 55, 35, 42, 37, 55, 53, 54, 58, 53, 46, 58,
$ P             56, 50, 48, 38, 35, 38~
                <int> 43, 41, 44, 40, 42, 42, 38, 40, 38, 38, 40, 42, 44,
$ K             36, 37, 39, 41, 39, 36~
$ temperature <dbl> 20.87974, 21.77046, 23.00446, 26.49110,
20.13017, 23.05805, 22.70884, 20.2~ $ humidity <dbl> 82.00274,
80.31964, 82.32076, 80.15836, 81.60487, 83.37012, 82.63941,
82.8~ $ ph <dbl> 6.502985, 7.038096, 7.840207, 6.980401,
7.628473, 7.073454, 5.700806, 5.71~ $ rainfall <dbl> 202.9355,
226.6555, 263.9642, 242.8640, 262.7173, 251.0550, 271.3249,
241.~ $ label <chr> "rice", "rice", "rice", "rice", "rice", "rice",
"rice", "rice", "rice", "r~
```

#To provide summary statistics about variables in data frames, tibbles, data tables and vectors.

```
skim(dataset_crop_n)
output
-- Data Summary ------------------------
                           Values
Name                       dataset_crop_n
Number of rows             1200
Number of columns          8
___________________________
Column type frequency:
  character                1
  numeric                  7
___________________________
Group variables            None
```

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

```
      A.  Variable type: character ------------------------------------------------------------------------
# A tibble: 1 x 8
  skim_variable n_missing complete_rate  min  max empty n_unique whitespace
* <chr>          <int>         <dbl> <int> <int> <int>    <int>      <int>
1 label              0             1     4    11     0       12          0
-- Variable type: numeric ------------------------------------------------------------------------------
# A tibble: 7 x 11
  skim_variable n_missing complete_rate  mean    sd    p0   p25   p50   p75  p100 hist
* <chr>          <int>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1 N                  0             1  67.1  36.1     0    34    73    97   140  ▇▆▆▅▃
2 P                  0             1  42.2  21.5     5    24    41    58    95  ▇▆▅▃▂
3 K                  0             1  35.0  11.8    15    25    35    45    55  ▆▅▇▅▇
4 temperature        0             1  26.5   4.32   18.0  24.0  25.8  28.1  43.7 ▂▇▅▂▁
5 humidity           0             1  77.0  14.0    45.0  66.1  80.9  88.2 100.  ▂▂▃▅▇
6 ph                 0             1   6.47  0.616  4.51  6.07  6.49  6.88  9.94 ▁▃▇▃▁
7 rainfall           0             1 123.   58.0    40.1  77.4 105.   167.  299.  ▇▆▃▂▁

summary(dataset_crop_n)

output
       N                P                K          temperature      humidity
 Min.   :  0.00   Min.   :  5.00   Min.   :15.00   Min.   :18.04   Min.   :45.02
 1st Qu.: 34.00   1st Qu.: 24.00   1st Qu.:25.00   1st Qu.:23.98   1st Qu.:66.14
 Median : 73.00   Median : 41.00   Median :35.00   Median :25.78   Median :80.89
 Mean   : 67.12   Mean   : 42.19   Mean   :34.95   Mean   :26.47   Mean   :76.99
 3rd Qu.: 97.00   3rd Qu.: 58.00   3rd Qu.:45.00   3rd Qu.:28.12   3rd Qu.:88.17
 Max.   :140.00   Max.   : 95.00   Max.   :55.00   Max.   :43.68   Max.   :99.98
       ph           rainfall        label
 Min.   :4.508   Min.   : 40.13   Length:1200
 1st Qu.:6.070   1st Qu.: 77.42   Class :character
 Median :6.488   Median :105.13   Mode  :character
 Mean   :6.469   Mean   :123.17
 3rd Qu.:6.883   3rd Qu.:167.15
 Max.   :7.995   Max.   :298.56
# Generating EDA report ------------------------------------------------------------
create_report(data = dataset_crop_n,
              output_file = 'Crop_Recomm_Report.html',
              output_dir = getwd(),
              config = configure_report(),
              y = 'label')

# Reading, Writing, and Viewing Data ------------------------------------------------------------
# Viewing Data ----------------------------------------------------------------------------------
  head(dataset_crop_n)

output
    N  P  K  temperature humidity       ph     rainfall label
1  90 42 43  20.87974 82.00274 6.502985 202.9355 rice
2  85 58 41  21.77046 80.31964 7.038096 226.6555 rice
3  60 55 44  23.00446 82.32076 7.840207 263.9642 rice
4  74 35 40  26.49110 80.15836 6.980401 242.8640 rice
5  78 42 42  20.13017 81.60487 7.628473 262.7173 rice
6  69 37 42  23.05805 83.37012 7.073454 251.0550 rice

#head(..., ...) returns first n number of rows of the data ------------------------------------
    head(dataset_crop_n, n=10)

output
    N  P  K temperature humidity       ph     rainfall label
1  90 42 43  20.87974 82.00274 6.502985 202.9355 rice
2  85 58 41  21.77046 80.31964 7.038096 226.6555 rice
3  60 55 44  23.00446 82.32076 7.840207 263.9642 rice
4  74 35 40  26.49110 80.15836 6.980401 242.8640 rice

# The tail() function in the R is particularly used to display the last n rows of dataset--------
  tail(dataset_crop)

  output
       N  P  K  temperature humidity       ph     rainfall label
1195  97 35 26  24.91461 53.74145 6.334610 166.2549 coffee
1196 107  34 32  26.77464 66.41327 6.780064 177.7745 coffee
1197  99 15 27  27.41711 56.63636 6.086922 127.9246 coffee
1198 118  33 30  24.13180 67.22512 6.362608 173.3228 coffee
1199 117  32 34  26.27242 52.12739 6.758793 127.1753 coffee
1200 104  18 30  23.60302 60.39647 6.779833 140.9370 coffee

# Returns size of dataframe which is in the form of rows * columns --------
  dim(dataset_crop_n)

  output
[1] 1200    8
# Returns variable name  of dataframe-----------------------------------------------
  dimnames(dataset_crop_n) output
 [1] "N"           "P"           "K"           "temperature" "humidity"    "ph"
 [7] "rainfall"    "label"
# unique returns a vector, data frame or array like x but with duplicates elements/rows removed -
  unique(dataset_crop_n$label)

output
 [1] "rice""maize""blackgram"  "pomegranate" "banana""mango"
 [7] "watermelon"  "papaya"          "coconut"     "cotton"      "jute"        "coffee"
#Displaying data structure of dataset
str(dataset_crop_n)
dataset_crop_matr_n <- as.factor(dataset_crop$label)
str(dataset_crop_matr_n)
```

Compare Machine Learning Models

```
# load libraries
library(mlbench)
library(lattice)
library(ggplot2)
library(caret)
library(rpart)
# loading or reading  dataset for crop recommendation ------------------
dataset_crop_n <- read.csv('Crop_Recomm_n.csv')
# prepare training scheme
control <- trainControl(method="repeatedcv", number=10, repeats=3)
```

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

```
# CART
set.seed(7)
fit.cart <- train(label~., data=dataset_crop_n, method="rpart", trControl=control)
# LDA
set.seed(7)
fit.lda <- train(label~., data=dataset_crop_n, method="lda", trControl=control)
#QDA
set.seed(7)
fit.qda <- train(label~., data=dataset_crop_n, method="qda", trControl=control)
#MDA
set.seed(7)
fit.mda <- train(label~., data=dataset_crop_n, method="mda", trControl=control)
#RDA
set.seed(7)
fit.rda <- train(label~., data=dataset_crop_n, method="rda", trControl=control)
# SVM
set.seed(7)
fit.svm <- train(label~., data=dataset_crop_n, method="svmRadial", trControl=control)
# KNN
set.seed(7)
fit.knn <- train(label~., data=dataset_crop_n, method="knn", trControl=control)
# Random Forest
set.seed(7)
fit.rf <- train(label~., data=dataset_crop_n, method="rf", trControl=control)
# Naive Bayes
set.seed(7)
fit.nb <- train(label~., data=dataset_crop_n, method="nb", trControl=control)
# collect resamples
results <- resamples(list(CART=fit.cart, LDA=fit.lda,RDA=fit.rda, MDA=fit.mda,QDA=fit.qda,
                          SVM=fit.svm, KNN=fit.knn, RF=fit.rf, NB=fit.nb))
# summarize differences between
modes summary(results)
output
Call:
summary.resamples(object = results)
Models: CART, LDA, RDA, MDA, QDA, SVM, KNN, RF, NB
Number of resamples: 30
Accuracy
      Min.  1st Qu.  MedianMean  3rd Qu.Max. NA's
CART 0.7250000 0.8333333 0.8333333 0.8405556 0.8833333 0.9166667    0
LDA  0.9416667 0.9666667 0.9750000 0.9744444 0.9833333 0.9916667    0
RDA  0.9750000 0.9833333 0.9916667 0.9897222 1.0000000 1.0000000    0
MDA 0.9583333 0.9750000 0.9833333 0.9816667 0.9895833 1.0000000    0
QDA  0.9750000 0.9833333 0.9916667 0.9897222 1.0000000 1.0000000    0
SVM  0.9416667 0.9770833 0.9875000 0.9852778 0.9979167 1.0000000    0
KNN  0.9416667 0.9666667 0.9750000 0.9766667 0.9916667 1.0000000    0
RF     0.9750000 0.9916667 0.9916667 0.9927778 1.0000000 1.0000000    0
NB     0.9666667 0.9833333 0.9916667 0.9905556 1.0000000 1.0000000    0

Kappa
      Min.  1st Qu.  Median      Mean  3rd Qu.      Max. NA's
CART 0.7000000 0.8181818 0.8181818 0.8260606 0.8727273 0.9090909    0
LDA    0.9363636 0.9636364 0.9727273 0.9721212 0.9818182 0.9909091    0
RDA   0.9727273 0.9818182 0.9909091 0.9887879 1.0000000 1.0000000    0
MDA 0.9545455 0.9727273 0.9818182 0.9800000 0.9886364 1.0000000    0
QDA   0.9727273 0.9818182 0.9909091 0.9887879 1.0000000 1.0000000    0
SVM   0.9363636 0.9750000 0.9863636 0.9839394 0.9977273 1.0000000    0
KNN   0.9363636 0.9636364 0.9727273 0.9745455 0.9909091 1.0000000    0
RF     0.9727273 0.9909091 0.9909091 0.9921212 1.0000000 1.0000000    0
NB     0.9636364 0.9818182 0.9909091 0.9896970 1.0000000 1.0000000    0
# box and whisker plots to compare models
scales <- list(x=list(relation="free"), y=list(relation="free"))
bwplot(results, scales=scales)
```

output

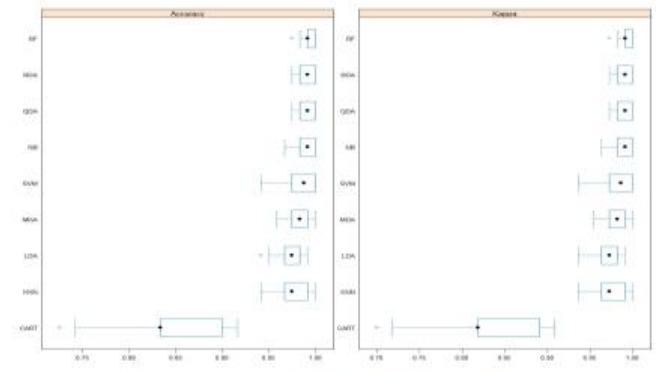


Fig.19: Compare Machine Learning Algorithms in R Box and Whisker Plots
Boxes are ordered from highest to lowest mean accuracy

```
# density plots of accuracy
scales <- list(x=list(relation="free"), y=list(relation="free"))
densityplot(results, scales=scales, pch = "|")
```

output

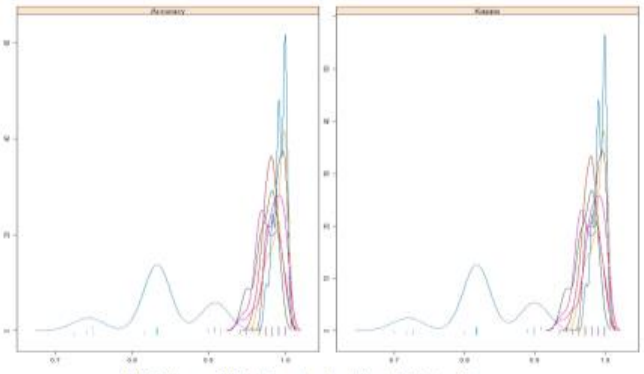


Fig.20: Compare Machine Learning Algorithms in R Density Plots

```
# dot plots of accuracy
scales <- list(x=list(relation="free"), y=list(relation="free"))
dotplot(results, scales=scales)
```

output

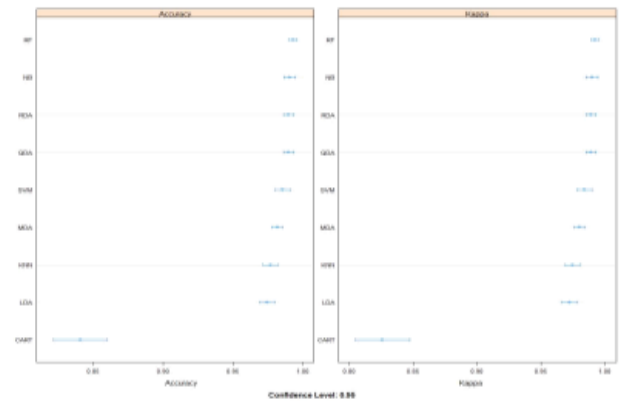


Fig.21: Compare Machine Learning Algorithms in R Dot

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 8, August 2022

Plots # parallel plots to compare models
parallelplot(results)

output
Fig:

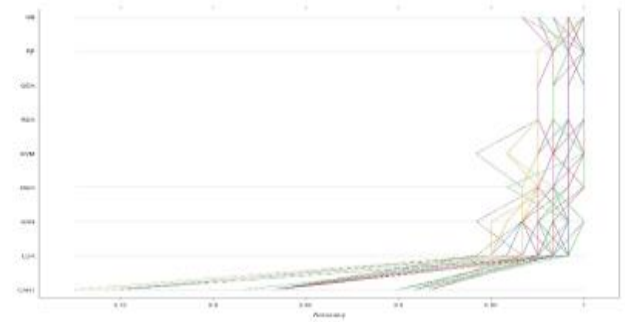


Fig.22: Compare Machine Learning Algorithms in R Parallel Plots

pair-wise scatterplots of predictions to compare models splom(results)
ouput

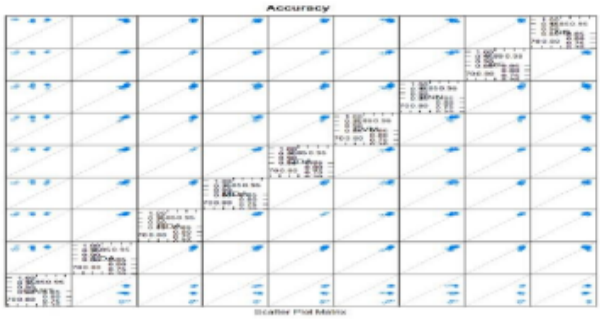


Fig.23: Compare Machine Learning Algorithms in R Scatterplot Matrix

xyplot plots to compare models xyplot(results, models=c("LDA", "SVM")) output
difference in model predictions diffs <- diff(results)
summarize p-values for pair-wise comparisons summary(diffs)
output

Call:
summary.diff.resamples(object = diffs)
p-value adjustment: bonferroni
Upper diagonal: estimates of the difference
Lower diagonal: p-value for H0: difference = 0

Accuracy

	CART	LDA	RDA	MDA	QDA	SVM
CART		-0.1338889	-0.1491667	-0.1411111	-0.1491667	-0.1447222
LDA	1.142e-13		-0.0152778	-0.0072222	-0.0152778	-0.0108333
RDA	6.530e-15	4.328e-08		0.0080556	0.0000000	0.0044444
MDA	1.036e-13	0.1135706				
0.0002273					-0.0080556	-0.0036111
QDA	6.530e-15	4.328e-08	NA	0.0002273		0.0044444
SVM	1.571e-14	4.926e-07	0.7313072	1.0000000	0.7313072	
KNN	1.766e-13	1.0000000	5.614e-05	1.0000000	5.614e-05	0.0445883
RF	6.211e-15	6.136e-08	0.0981103	2.861e-06	0.0981103	0.0237783
NB	6.045e-14	0.0043379	1.0000000	0.0566415	1.0000000	1.0000000

	KNN	RF	NB
CART	-0.1361111	-0.1522222	-0.1500000
LDA	-0.0022222	-0.0183333	-0.0161111
RDA	0.0130556	-0.0030556	-0.0008333
MDA	0.0050000	-0.0111111	-0.0088889
QDA	0.0130556	-0.0030556	-0.0008333
SVM	0.0086111	-0.0075000	-0.0052778
KNN		-0.0161111	-0.0138889
RF	7.759e-06		0.0022222
NB	0.0351897	1.0000000	

Graph for Random Forest

```
#The randomForest package doesn't have any in-built
#way for plotting the trees. We can use the 'party' package.
#It has the required plotting function inbuilt in the package.
library("party")
library(grid)
library(modeltools)
library(stats4)
library(strucchange)
library(zoo)
dataset_crop_nb <- read.csv('Crop_Recomm_n.csv')
dataset_crop_nb[["label"]]= factor(dataset_crop_nb[["label"]])
x <- ctree(label ~ ., data=dataset_crop_nb)
plot(x, type="simple")
```

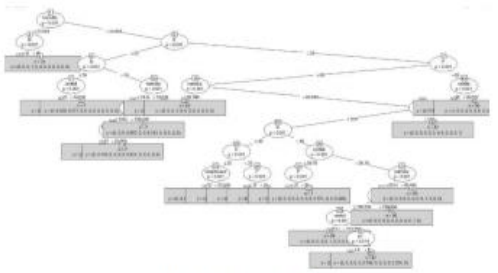
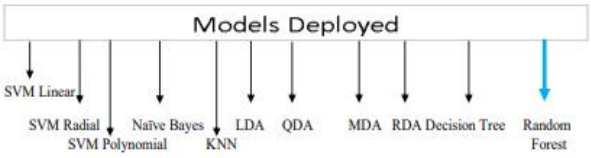


Fig. 24: Graph for Random Forest

Illustration of Evaluation of Model Accuracy

Evaluation Of Model Accuracy										
	SVM Linear	SVM Radial	SVM Polynomial	LDA	QDA	MDA	RDA	Naïve Bayes	KNN	Random Forest	Decision Tree
Train Accuracy	98.23	98.23	98.11	96.12	97.84	98.11	97.12	96.12	97.12	98.25	98.65
Test Accuracy	98.23	97.58	98.30	97.21	96.12	98.23	97.65	97.54	97.86	98.85	97.50

Models Deployed with High Accuracy in Production



XII. CONCLUSION

On an agricultural dataset, various machine learning algorithms were tested to see which one performed the best. This project explains how to use data analysis to analyze a crop dataset. This data collection contains a number of factors that may be used to identify crops for recommendation and to conduct supervisory training on data sets obtained from the agriculture sector in order to divide data into various classes.

The performance of SVM Linear, SVM Radial, SVM Polynomial, naive Bayes, k-nearest neighbor, LDA, QDA, RDA, MDA, Decision Tree, and The Random Forest models on the crop_recomm dataset is investigated in this paper. The motivation for investigating these strategies is owing to their relative easiness, as evidenced by previous research.

This study compares many methods such as SVM Linear, SVM Radial, SVM Polynomial, naive Bayes, k-nearest neighbor, LDA, QDA, RDA, MDA, Decision Tree, and Random Forest model. These algorithms were used to train 0.8 percent of the input data and then tested with the remaining 0.2 percent of the test dataset, with the accuracy of the methods being compared.

As a result, various assessment measures affect how well classifiers work. The Random Forest method provides a higher level of accuracy of 98.85 percent, as well as a lower mean square error. This technology will assist farmers in reducing difficulties and will act as an intermediary in providing them with the information they require to achieve high revenues and maximize profits.

REFERENCES

[1]. S. Veenadhari, Dr Bharat Misra, Dr CD Singh.2019." Machine learning approach for forecasting crop yield based on climatic parameters.".978-1-4799-2352- 6/14/$31.00 ©2014 IEEE

[2]. Anshal Savla, Parul Dhawan, Himtanaya Bhadada, Nivedita Israni, Alisha Mandholia, Sanya Bhardwaj (2015), 'Survey of classification algorithms for formulating yield prediction accuracy in precision agriculture', Innovations in Information, Embedded and Communication systems (ICIIECS).

[3] Rakesh Kumar, M.P. Singh, Prabhat Kumar and J.P. Singh (2015), 'Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique', International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM).

[4]. A.T.M Shakil Ahamed, Navid Tanzeem Mahmood, Nazmul Hossain, Mohammad Tanzir Kabir, Kallal Das, Faridur Rahman, Rashedur M Rahman (2015), 'Applying Data Mining Techniques to Predict Annual Yield of Major Crops and Recommend Planting Different Crops in Different Districts in Bangladesh', (SNPD) IEEE/ACIS International Conference.

[5]. Monali Paul, Santosh K. Vishwakarma, Ashok Verma (2015), 'Analysis of Soil Behaviour and Prediction of Crop Yield using Data Mining Approach', International Conference on Computational Intelligence and Communication Networks