FARMASSIST

by
Divyansh Dhubkarya - 2000290120063
Gagan Gupta - 2000290120064
Anuj Garg - 2000290120034

Under the guidance of
Ms. Shreela Pareek (Assistant Professor CS)

Submitted to the department of Computer Science in partial

fulfilment of the requirements

for the degree of

Bachelor of Technology

in

Computer Science

KIET GROUP OF INSTITUTIONS, Ghaziabad

Dr. A.P.J. Abdul Kalam Technical University, Uttar Pradesh Lucknow

Annexure –III

DECLARATION

We hereby declare that this submission is our own work that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:
Name: Divyansh Dhubkarya
Roll number: 2000290120063
Date:

Signature:
Name: Gagan Gupta
Roll number: 2000290120064
Date:

Signature:
Name: Anuj Garg
Roll number: 2000290120034
Date:

CERTIFICATE

This is to certify that Project Report entitled "**FARMASSIST**" which is submitted by **Divyansh Dhubkarya, Gagan Gupta and Anuj Garg** in partial fulfilment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, formerly Uttar Pradesh Technical University, is a record of the candidate's own work carried out by them under my supervision.

(Supervisor Signature)

Name: Ms. Shreela Pareek

Designation: Assistant Professor CS

Date:

Annexure –IV

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Professor Ms. Shreela Pareek, Assistant Professor, Department of Computer Science for her constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor Dr. Ajay Kumar Shrivastav, Head, Department of Computer Science , KIET Ghaziabad for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:
Name : Divyansh Dhubkarya
Roll No.: 2000290120063
Date :

Signature:
Name : Gagan Gupta
Roll No.: 2000290120064
Date :

Signature:
Name : Anuj Garg
Roll No.: 2000290120034
Date :

ABSTRACT

Agriculture is one of the most important occupations for the majority of people in the world's second largest populated country, India. However, due to a lack of education, accurate information, and India's rapid climate change, farmers frequently cultivate the same crops or the incorrect crops, regardless of whether they are appropriate given the soil, climate, and other elements in that specific area or not. This has had a negative impact on agricultural crop output and performance over the past few decades. Predicting the right crops to grow based on the most important parameters for crop production would help farmers to choose the right crops, improving crop quality, production and yield. To depict the suggestions of diverse Indian crops, a variety of machine learning techniques like "Decision Tree, Naive Bayes, SVM, Logistic Regression, Random Forest" were utilized. These five different categories of artificial intelligence algorithms were the subject of the investigation, with Random Forest and Naive Bayes showing the greatest accuracy with 99.09% .

TABLE OF CONTENTS

CHAPTER 6 CONCLUSION AND FUTURE SCOPE

CHAPTER 7 REFERENCES

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.No	Abbreviations	Full Form
1.	ML	Machine Learning
2.	SVM	Support Vector Machine
3.	KNN	K-Nearest Neighbor
4.	CNN	Convolutional Neural Network
5.	AI	Artificial Intelligence

LIST OF SYMBOLS

+	Addition
%	Percentage
=	Equal

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The power of AI and ML can be applied in the field of farming as well which can help the farmers to become productive and informed. This forms the main objective and motivation of our project **CROP RECOMMENDATION SYSTEM.**

Through this project we plan to achieve the following objectives -

- Bridge the digital gap by providing farmers timely information about crops, government schemes, bank loans etc.
- Helping farmers make smart choices about their crops to reduce the chances of crop failures.
- Digitalizing the whole farming process so that farmers can reap the benefits of their crops.

All nations, whether underdeveloped, developing, or even developed, rely on agriculture as their primary means of supplying their populations with food. By 2025, it is predicted that there will be 9.7 billion people on the planet.
It is challenging to ensure food sustainability when unpredictable weather is added to this. People in the current world are unaware of the importance of planting crops at the proper time and location.
These cultivating methods also alter the seasonal climate, which has a negative impact on basic resources like soil, water, and air and results in food insecurity.
There are no suitable solutions or technologies to deal with the predicament we confront after analysing all these challenges and problems, including weather, temperature, and numerous other elements. There are various approaches to boost agriculture-related economic growth in India. Crop yield and quality can both be increased and improved in a variety of ways.

Fortunately, just like many other issues, this one has a fix.In this project, we aim to simplify the process of crop recommendation by leveraging machine learning algorithms. By considering input parameters such as Nitrogen, Phosphorous, Potassium levels, pH value, and other relevant factors, we can predict the most suitable crop for a given agricultural setting.

Support Vector Machines (SVM), Naive Bayes, Random Forest, and Decision Trees are just a few of the strong algorithms we'll use to do this. These algorithms are suitable for predicting and recommending the best crop selections depending on input factors since they are excellent at analysing and classifying massive datasets.We evaluate the accuracy of various models, pick the best one, and then preserve that specific model.

We can help farmers and agricultural professionals choose the best crops by fusing the knowledge of agricultural science with the power of machine learning. This research has the potential to increase resource efficiency, increase agricultural production, and promote sustainable farming methods.

CHAPTER 2

LITERATURE SURVEY

Previous studies on the Crop Recommendation System have shown the effectiveness of different machine learning algorithms on classifying the types of crops.

Various machine learning algorithms that are previously studied to recommend crops in malignant or benign are as follows:

Naive Bayes

When determining whether a data item falls into a particular category or not, Naive Bayes makes a probabilistic determination. It can be used in text analysis to classify words or sentences as either falling under or not falling under a predetermined "tag" and can be calculated as:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Random Forest

Random Forest is a popular and effective machine learning technique. Multiple decision trees are combined in this ensemble method to produce predictions. Using a randomized subset of the training data and a randomized sample of features, the method builds a large number of decision trees.

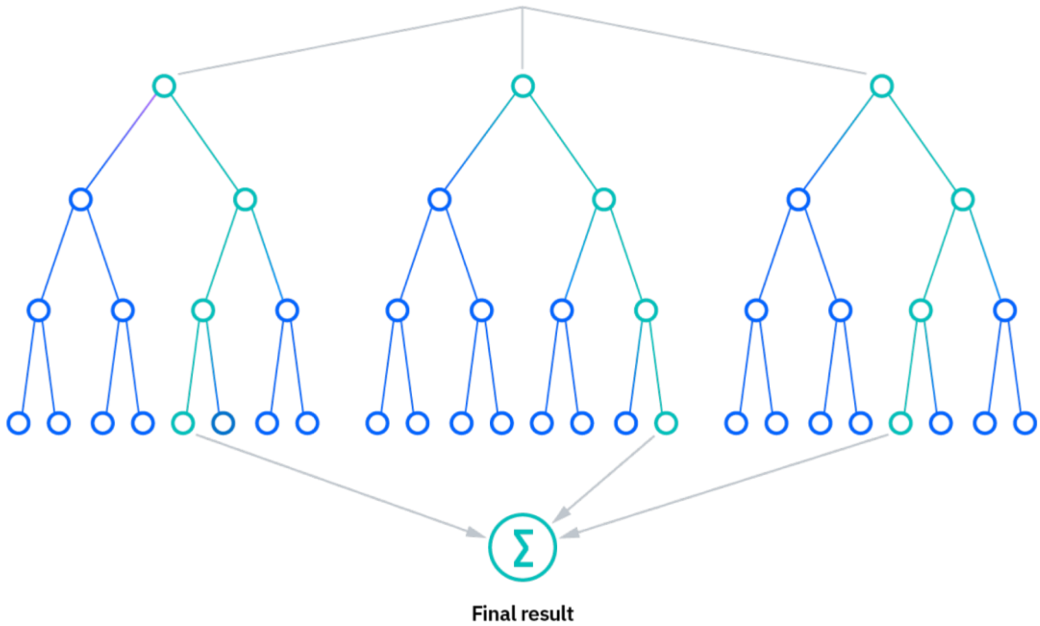

Fig.2.1.1. Random Forest Decision Tree

SVM

Although they can be modified for regression and outlier detection, SVMs are generally employed for classification problems. Finding an ideal hyperplane that maximally separates several classes in the feature space is the central concept of SVM. The closest points to the decision boundary, or the support vectors, are chosen from a subset of the training data points to create this hyperplane.

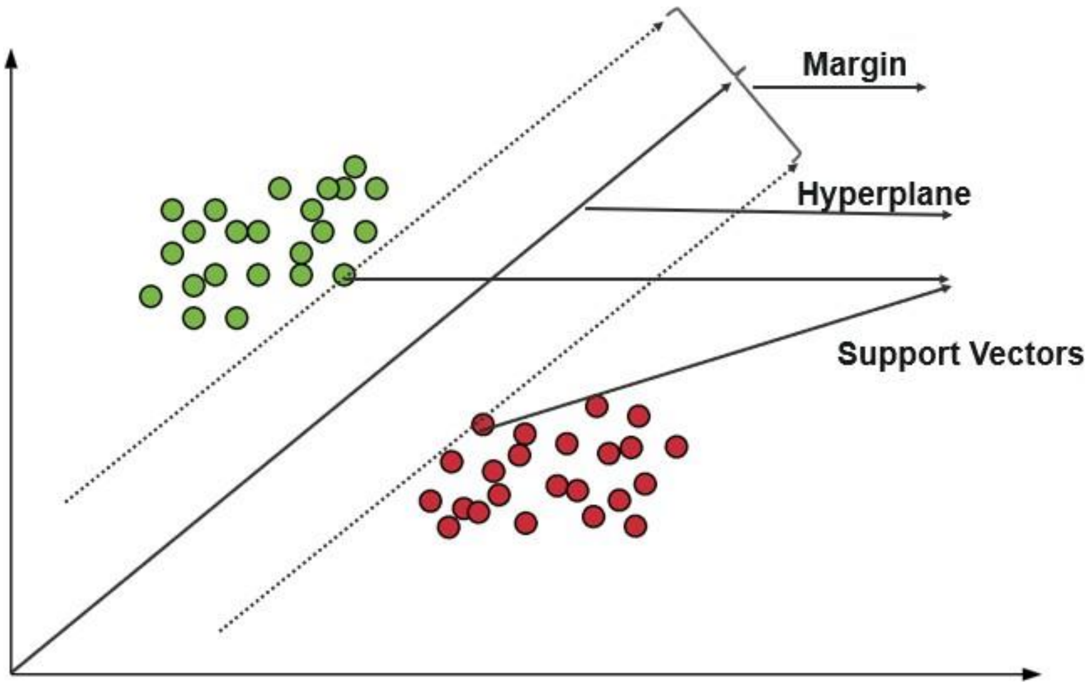

Fig.2.1.2. Support Vector Machine

The research done in this field includes a variety of ideas and implementations which are as follows:

Madhuri Shripathi Rao [2022], Described a dataset encompassing 22 different types of crops was used, the research analyzed various supervised learning algorithms like KNN, Decision Tree, and Random Forest. To identify the crop that will grow the most effectively on a given plot of land, a comparison of the three different supervised machine learning models (KNN, Decision Tree, and Random Forest) was conducted. In summary, it was discovered that the crop forecast dataset revealed Random Forest Classifier had the highest accuracy (99.32%), K-Nearest Neighbor had the lowest accuracy (97.04%), and Decision Tree Classifier's accuracy fell between KNN and Random Forest Classifier.

Rahul Katarya, Ashutosh Raturi, Abhinav Mehndiratta and Abhinav Thapper [2020]. In his paper from page 4, Rahul Katarya discusses the many machine learning methods now used to boost crop output.. In this paper, many artificial intelligence methods, including big data analysis for precision agriculture and machine learning algorithms, are covered. Through the use of KNN, Ensemble-based Models, Neural Networks, etc., they discuss the crop recommender system.

Konstantinos G. Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson ID and Dionysis Bochtis [2018] Describe in detail the research that has been done on the use of machine learning in the agriculture production system. In order to discover new approaches to understand, quantify, and analyze data-intensive processes in farm management, machine learning (ML) was created together with big data technologies, methodologies, methods, and high performance computers. This whitepaper is implemented using Support Vector Machines (SVM).

Thomas van Klompenburg, Ayalew Kassahun and Cagatay Catal [2020], Describe that several publications on crop yield prediction that used deep learning and machine learning were analyzed. According to the study's findings, prediction models with more features didn't necessarily deliver the best performance. Models with fewer and more features should be evaluated to determine the best performing model. Since crop output varies on a wide range of variables, including climate, weather, soil, fertilizers usage, and seed variety, they used a number of datasets. According to the findings, CNN and neural networks are the most popular deep learning and machine learning algorithms, respectively.

Ghadge R, Kulkarni J, More P,Priya RL, in “Prediction of crop yield using machine learning”, Int. Res. state that a Machine Learning Approach to Predict Crop Yield and Success Rate using a multilayer perceptron neural network model was created. The result was initially produced with an accuracy of 45% using the optimizer RMS prop; afterwards, it was improved to 90% by adding layers, altering weight and bias, and switching to the optimizer Adam. In this study, a new crop yield prediction model using a 3 Layer Neural Network is developed. Rectified Linear Activation Unit (Relu) is the activation function that the ANN model issues to create a model for yielding predictions after determining the relationship using a large number of input and output instances. The techniques of forward and backward propagation were employed.

Nigam, Aruvansh, Saksham Garg, Archit Agrawal, and Parul Agrawal. in "Crop yield prediction using machine learning algorithms." state that for predicting crop yield based on meteorological factors, Veenadhari S, Misra B, and Singh CD suggested a machine learning approach. The goal of the study was to create a webpage detailing how climatic factors affected crop yield in particular Madhya Pradesh regions. They created a website that is easy to use, and all of the crops and districts chosen for the study have forecast accuracy rates that are greater than 75%, showing higher prediction accuracy. Any user can utilise the user-friendly website created for crop yield prediction by entering the local climate data for their preferred crop using the C4.5 algorithm.

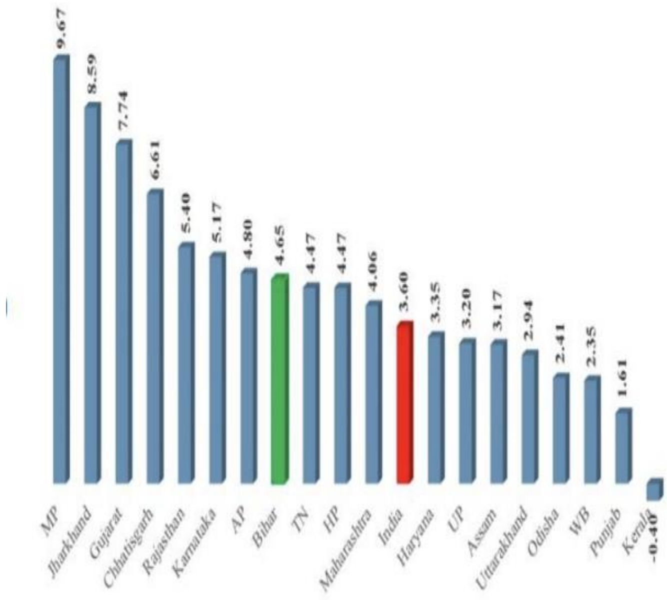

Fig.2.1.3 Production Growth Chart

CHAPTER 3

SYSTEM DESIGN AND METHODOLOGY

3.1 OVERVIEW ON MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that gives systems the ability to automatically learn and evolve from experience without being specially programmed by the programmer. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The main aim of machine learning is to allow computers to learn automatically and adjust their actions to improve the accuracy and usefulness of the program, without any human intervention or assistance. Traditional writing of programs for a computer can be defined as automating the procedures to be performed on input data in order to create output artifacts. Almost always, they are linear, procedural and logical. A traditional program is written in a programming language to some specification, and it has properties like:

- We know or can control the inputs to the program.
- We can specify how the program will achieve its goal.
- We can map out what decisions the program will make and under what conditions it makes them.
- Since we know the inputs as well as the expected outputs, we can be con dent that the program will achieve its goal

Traditional programming works on the premise that, as long as we can define what a program needs to do, we are confident we can define how a program can achieve that goal. This is not always the case as sometimes, however, there are problems that you can represent in a computer that you cannot write a traditional program to solve. Such problems resist a procedural and logical solution. They have properties such as:

- The scope of all possible inputs is not known beforehand.
- You cannot specify how to achieve the goal of the program, only what that goal is.
- You cannot map out all the decisions the program will need to make to achieve its goal.
- You can collect only sample input data but not all possible input data for the program.

3.1.1 SUPERVISED AND UNSUPERVISED LEARNING

Machine learning techniques can be broadly categorized into the following types:

Supervised learning takes a set of feature/label pairs, called the training set. From this training set the system creates a generalised model of the relationship between the set of descriptive features and the target features in the form of a program that contains a set of rules. The objective is to use the output program produced to predict the label for a previously unseen, unlabelled input set of features, i.e. to predict the outcome for some new data. Data with known labels, which have not been included in the training set, are classified by the generated model and the results are compared to the known labels. This dataset is called the test set. The accuracy of the predictive model can then be calculated as the proportion of the correct predictions the model labeled out of the total number of instances in the test set.

Unsupervised learning takes a dataset of descriptive features without labels as a train- ing set. In unsupervised learning, the algorithms are left to themselves to discover interesting structures in the data. The goal now is to create a model that finds some hidden structure in the dataset, such as natural clusters or associations. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system does not figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unla- beled data. Unsupervised learning can be used for clustering, which is used to discover any inherent grouping that are already present in the data. It can also be used for association problems, by creating rules based on the data and finding relationships or associations between them.

Semi-supervised machine learning falls somewhere in between supervised and unsuper- vised learning, since they use both labeled and unlabeled data for training, typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring labeled data generally does not require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards.Machine learning algorithms are tools to automatically make decisions from data in order to achieve some overarching goal or requirement. The promise of machine learning is that it can solve complex problems automatically, faster and more accurately than a manually specified solution, and at a larger scale. Over the past few decades, many machine learning algorithms have been developed by researchers, and new ones continue to emerge and old ones modified.

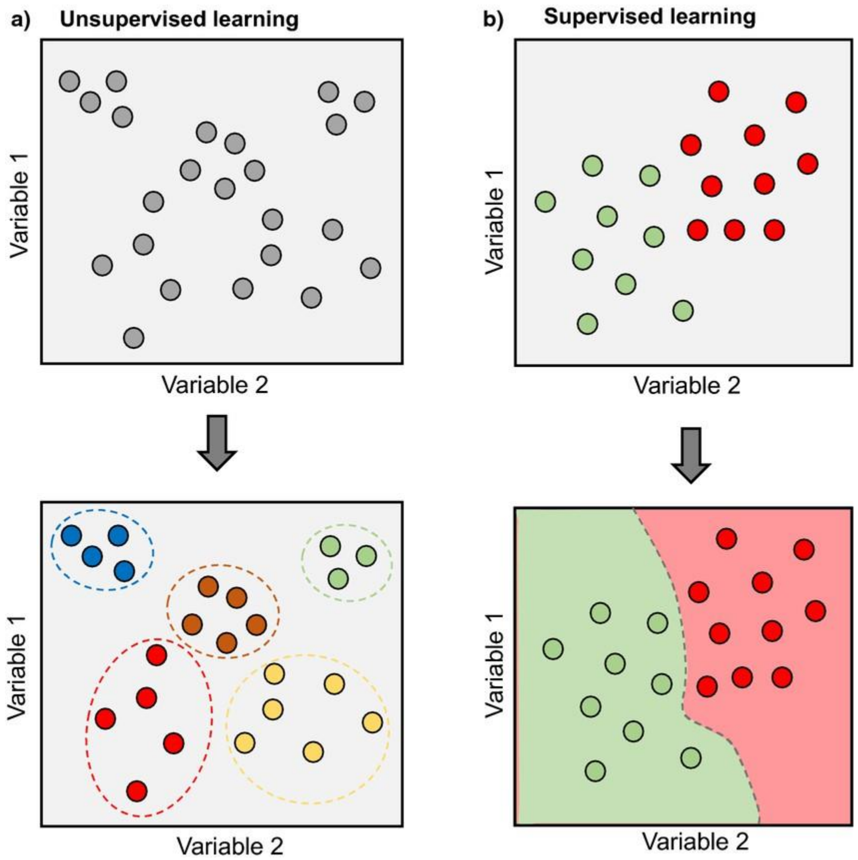

Fig.3.1.1 Supervised and Unsupervised Learning

3.2 ARCHITECTURE

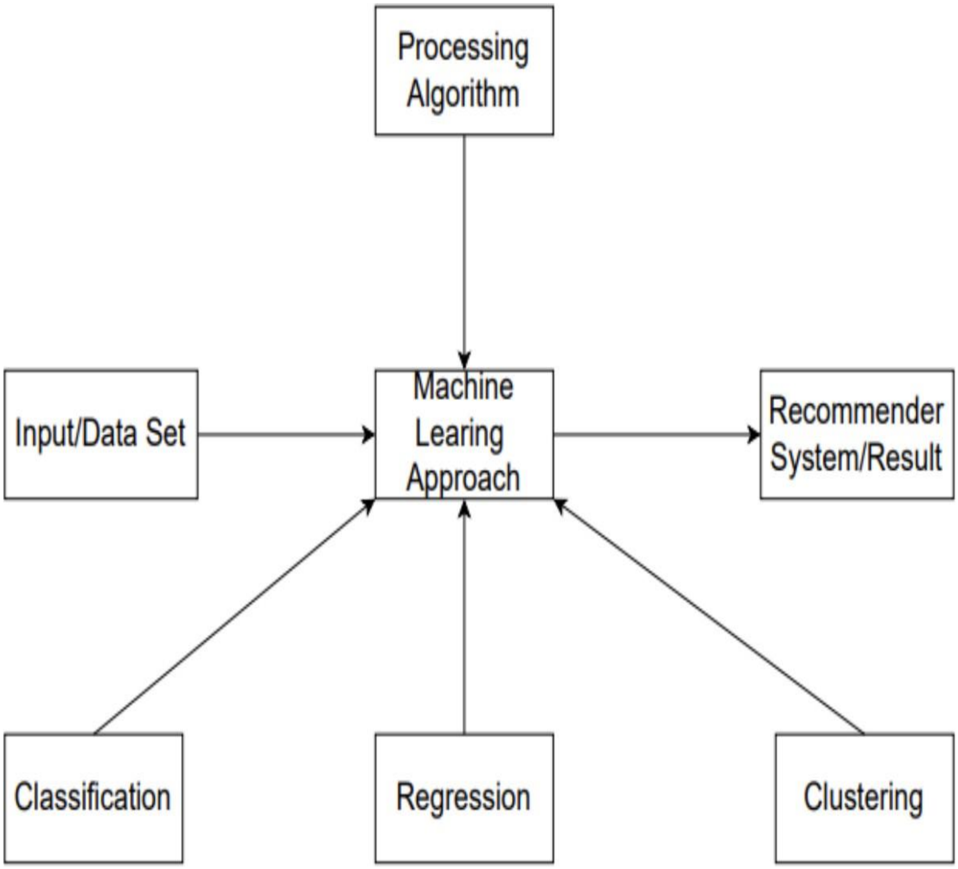

Fig 3.2.1 Architecture Diagram of Model

Fig 3.2.2 Flow Diagram of Model

The above image describes the system architecture of our model.
The steps involved in this process:
1. Uploading the data
2. Splitting the data into training and testing
3. Model Implementation
4. Prediction of crops

3.3 PROPOSED METHODOLOGY

The following steps are proposed in this experiment.

i. First dataset is loaded into the workspace.
ii. Data is cleaned and pre-processed to generate precise and reliable results and then data is splitted into train and test datasets.
iii. Now, the model will be trained on this dataset, first by Naive Bayes and then using different algorithms.
iv. Results are evaluated by all algorithms.
v. Comparison of both algorithms will be given based on accuracy.

3.3.1 Importing the Libraries

It is necessary to import the required libraries in order to build and run our machine-learning model. These significant resources are utilised both in the construction of the model and in providing accuracy. Numerous libraries, including Numpy, Matplotlib, Pandas, Sklearn, etc., are used by our model. These libraries gave us the essential data structures and algorithms, as well as a variety of performance features, to run our model.

Numpy

The goal of NumPy is to offer array objects that are up to 50 times faster than conventional Python lists.Working with ndarray is made incredibly simple by the numerous supporting functions it offers.

Pandas

For data manipulation and analysis in Python, the pandas library is a potent tool. It offers simple-to-use data structures that facilitate the effective processing of structured data, like DataFrame and Series. You may import data into pandas from many different sources, including as CSV files, Excel spreadsheets, SQL databases, and more. Pandas offers a large number of functions and techniques to examine, alter, clean, and summarise the data once it has been loaded. It is a crucial library for data wrangling since it makes it possible to do operations including filtering, sorting, grouping, merging, and altering data.

MatplotLib

Data visualisation using Python is made simple and effective with Matplotlib. Users may easily generate a wide range of plots, charts, and graphs thanks to its complete collection of tools and features. With its adaptable and user-friendly interface, Matplotlib enables users to completely personalise their visualisations, from the colours and labels to the axes and legends. For a variety of uses, such as scientific research, data analysis, and presentations, you may create excellent static, animated, and interactive visualisations with Matplotlib.

Seaborn

Matplotlib was the foundation for Seaborn, a potent Python data visualisation package. It offers a sophisticated user interface for producing educational and aesthetically pleasing statistical visuals. By providing a large selection of predefined styles and colour palettes, Seaborn makes it easier to create visually appealing visualisations. Additionally, it offers specialised tools for building intricate plots, including scatter plots, line plots, bar plots, heatmaps, and more.

SkLearn

Sklearn, also referred to as scikit-learn, is a robust and well-liked Python toolkit for machine learning. For tasks like classification, regression, clustering, dimensionality reduction, and model selection, it offers a wide variety of tools and algorithms. Sklearn smoothly integrates with the scientific Python ecosystem because it is built on top of other well-known Python libraries like NumPy, SciPy, and matplotlib.

3.3.2 Algorithms used

Naive Bayes

It is based on the Bayes' theorem notion, which includes estimating the likelihood of an event based on information or knowledge already known. The "naive" in its name refers to the assumption that each characteristic in a dataset is independent of every other feature. Naive Bayes can nonetheless generate amazingly accurate answers in many real-world applications despite this oversimplifying assumption.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Random Forest

For both classification and regression applications, Random Forest is a popular and effective machine learning technique. Multiple decision trees are combined in this ensemble method to produce predictions. Using a randomized subset of the training data and a randomized sample of features, the method builds a large number of decision trees.

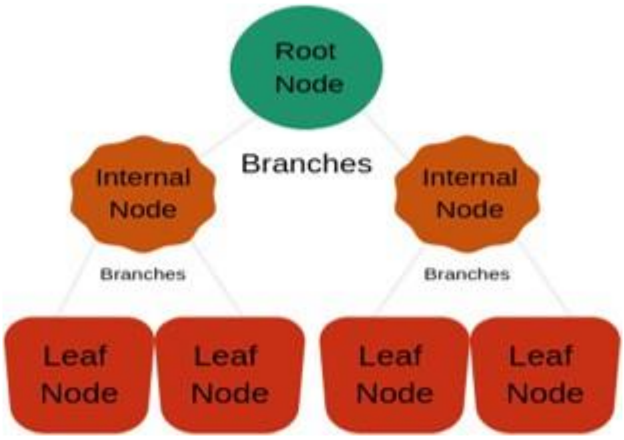

Fig. 3.3.1. Random Forest Diagram

Decision Tree

It is a supervised learning strategy that is quite beneficial for dealing with classification and regression concerns. The decision tree algorithm produces a tree structure by recursively splitting the data in accordance with the features and their values. At each internal node of the tree, the data is divided into several branches according to a decision made based on a specific feature.

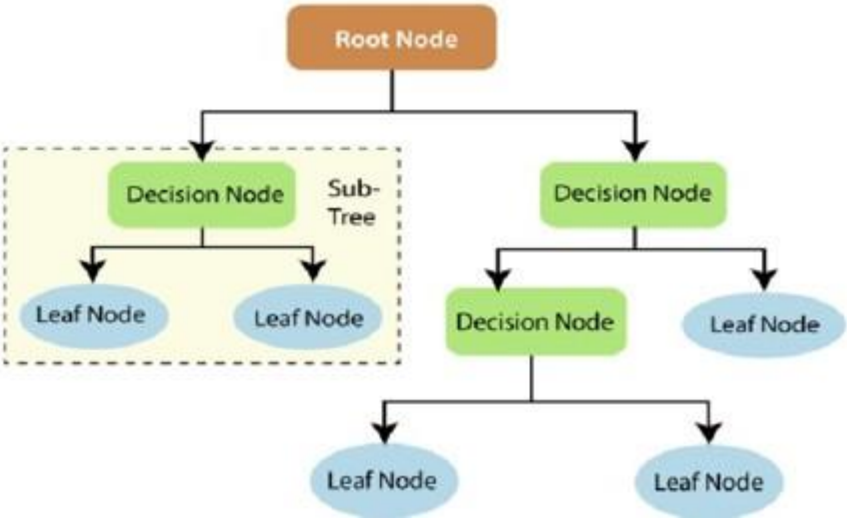

Fig.3.3.2. Decision Tree Diagram

Support Vector Machine (SVM)

Although they can be modified for regression and outlier detection, SVMs are generally employed for classification problems. Finding an ideal hyperplane that maximally separates several classes in the feature space is the central concept of SVM. The closest points to the decision boundary, or the support vectors, are chosen from a subset of the training data points to create this hyperplane.

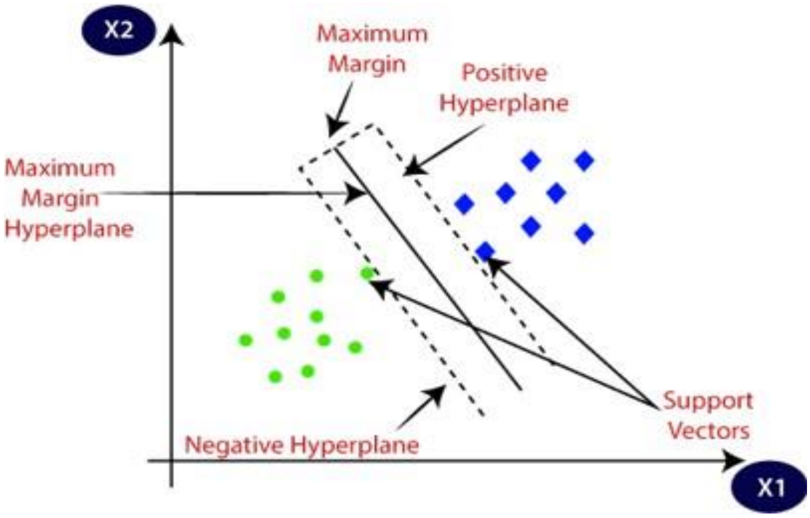

Fig.3.3.3. SVM Graph

Logistics Regression

Its ease of use and interpretability make it a common technique in statistics and machine learning. The dependent variable in logistic regression is binary, i.e., it might have two different potential results. The logistic function used by the method to evaluate the connection between the independent variables and the likelihood of the binary result.

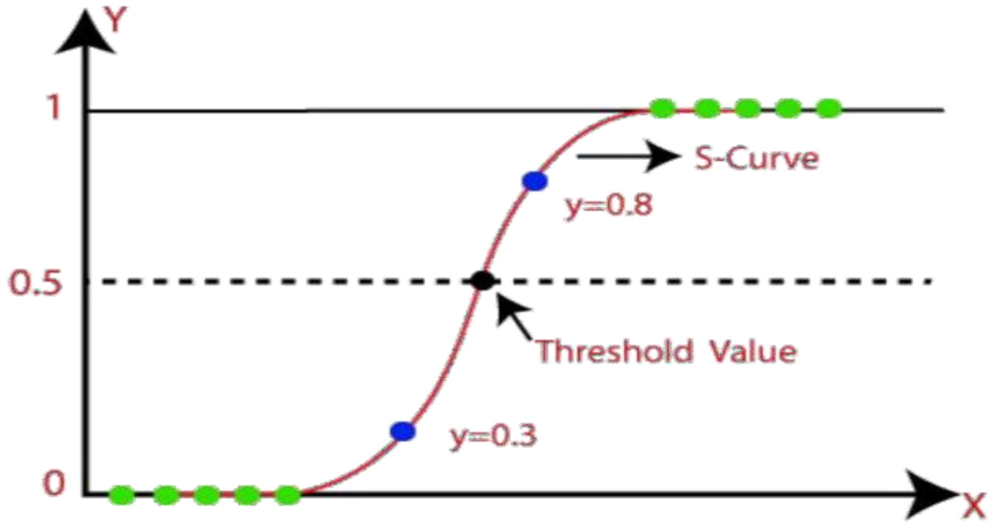

Fig.3.3.4. Logistics Regression Graph

3.4 EXPERIMENTATION

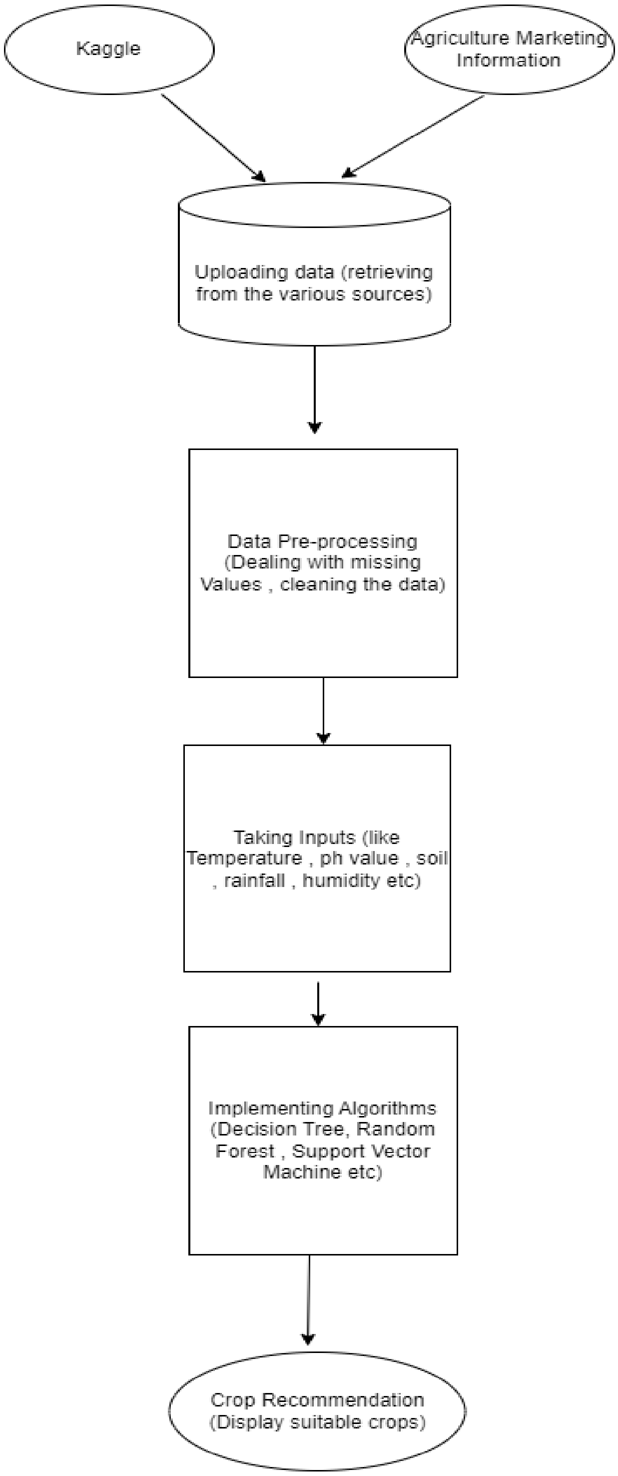

Fig 3 (a). Experimentation Flowchart

3.5 DFD Diagrams

1) 0 Level DFD

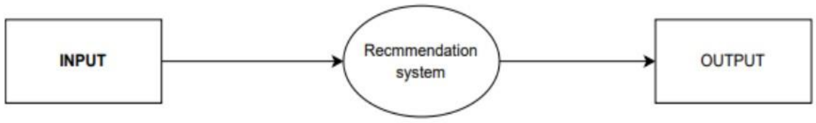

Fig 3 (b) . 0 Level DFD

2) 1 Level DFD

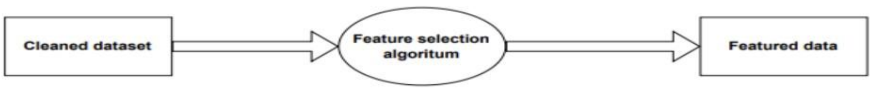

Fig 3 (c) . 1 Level DFD

3) 2 Level DFD

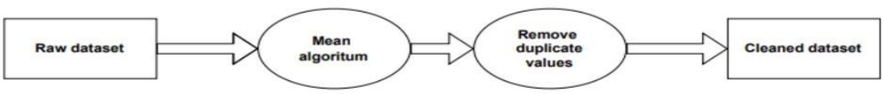

Fig 3 (d) . 2 Level DFD

4) 3 Level DFD

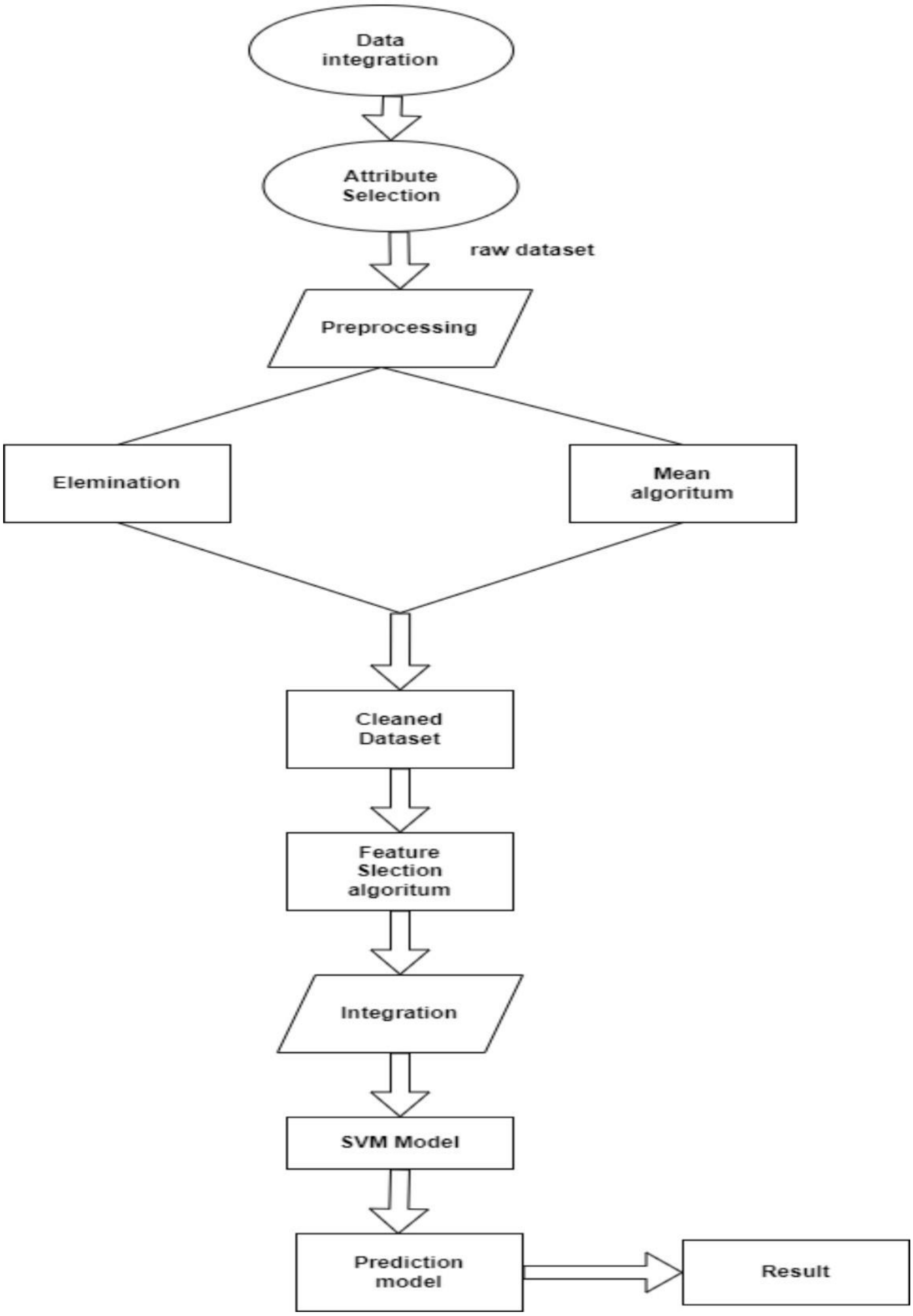

Fig 3 (d). 3 Level DFD

5) Flow Diagram

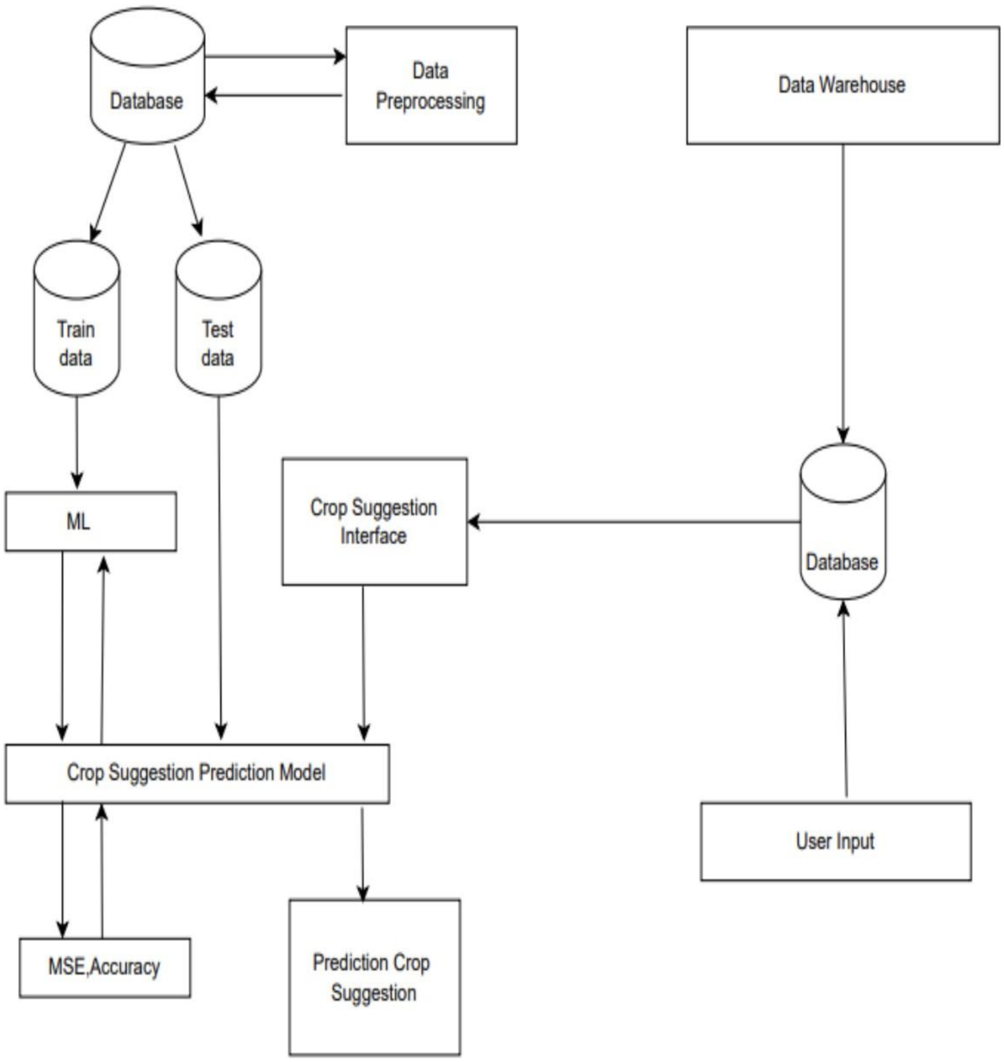

Fig 3 (e). Flow Diagram

3.6 Model and Phases

The waterfall model is a sequential software development process, in which progress is seen as owing steadily downwards (like a waterfall) through the phases of Require- ment initiation, Analysis, Design, Implementation, Testing and maintenance.

Requirement Analysis

This phase is concerned about collection of requirements of the system. This process involves generating document and requirement review.

System Design

Keeping the requirements in mind the system specifications are translated into a software representation. In this phase the designer emphasizes on:- algorithm, data structure, software architecture etc.

Coding

In this phase the programmer starts his coding in order to give a full sketch of the product. In other words system specifications are only converted into machine code.

Implementation

The implementation phase involves the actual coding or pro- gramming of the software. The output of this phase is typically the library, executa-bles, user manuals and additional software documentation.

Testing

In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.

Maintenance

The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer needs, adapt to accommodate changes in the external environment, correct errors and oversights previously undetected in the testing phase, and enhance the efficiency of the software.

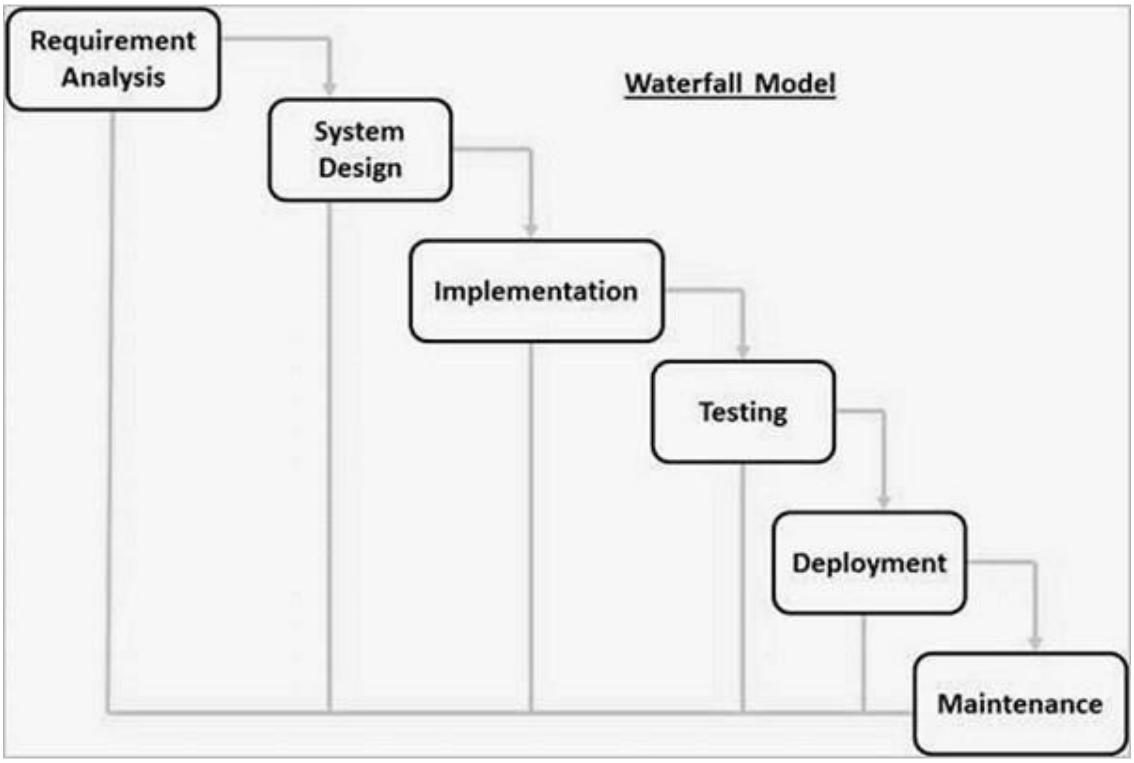

Fig.3.6. Waterfall Model Chart

CHAPTER 4

IMPLEMENTATION

4.1 Software and Hardware Requirements

4.1.1. Hardware Requirements

4.1.1.1. Server-Side Requirements

Processor: Window 8 or above
Hard Disk: 40GB
RAM: 256MB

4.1.1.2. Client-Side Requirements

Processor: Window 8 or above
Hard Disk: 20GB
RAM: 128MB

4.2. Software Requirements

Operating System: Windows 8 or above
Client Script: Python
Component Model: Jupyter Notebook, Kaggle

CHAPTER 5

SNAPSHOTS

5.1 Importing all the libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import tree
```

Fig.5.1.1. Imported Libraries Code Snippet

Here we are showing that we are importing all the necessary files which are required for our ml model.

5.1.1 Data fields

N - ratio of Nitrogen content in soil
P - ratio of Phosphorus content in soil
K - ratio of Potassium content in soil
Temperature - temperature in degree Celsius
Humidity - relative humidity in %
PH - ph value of the soil
Rainfall - rainfall in mm

	N	P	K	temperature	humidity	ph	rainfall	label
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

Fig.5.1.2. Data Set Snippet 1

```
df.dtypes
```

```
N                int64
P                int64
K                int64
temperature    float64
humidity       float64
ph             float64
rainfall       float64
label           object
dtype: object
```

```
[ ] df['label'].value_counts()
```

```
rice           100
maize          100
jute           100
cotton         100
coconut        100
papaya         100
orange         100
apple          100
muskmelon      100
watermelon     100
grapes         100
mango          100
banana         100
pomegranate    100
lentil         100
blackgram      100
mungbean       100
mothbeans      100
pigeonpeas     100
kidneybeans    100
chickpea       100
```

Fig.5.1.3. Data Set Snippet 2

```
df.size
```

```
17600
```

```
[ ] df.shape
```

```
(2200, 8)
```

```
[ ] df.columns
```

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

```
df['label'].unique()
```

```
array(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',
       'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',
       'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',
       'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],
      dtype=object)
```

```
[ ] df.dtypes
```

```
N                int64
P                int64
K                int64
temperature    float64
humidity       float64
ph             float64
rainfall       float64
label           object
dtype: object
```

Fig.5.1.4. Data Set Snippet 3

	N	P	K	temperature	humidity	ph	rainfall	label
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

Fig.5.1.5. Data Set Snippet 4

5.2 Loading, Pre-processing and Splitting the data

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files  No file chosen          Upload widget is only available when the cell has been exe
Saving Crop_recommendation.csv to Crop_recommendation.csv
```

```
[ ] df = pd.read_csv('Crop_recommendation.csv')
```

```
[ ] features = df[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]
    target = df['label']
    #features = df[['temperature', 'humidity', 'ph', 'rainfall']]
    labels = df['label']
```

```
[ ] # Initialzing empty lists to append all model's name and corresponding name
    acc = []
    model = []
```

```
[ ] # Splitting into train and test data

    from sklearn.model_selection import train_test_split
    Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,test_size = 0.2,random_state =2)
```

Fig.5.2. Loading, Pre-Processing and Splitting the Data Code Snippet

Loading a dataset requires reading the data from a file or other source and storing it in memory in order to analyse and model it. CSV, JSON, and Excel are just a few of the formats in which datasets can be saved. Depending on the format of the dataset, various tools and libraries can be used to load the data. With utilities for reading CSV, Excel, and other formats, pandas is a popular package for reading and editing datasets in Python.

After importing the dataset, pre-processing is frequently necessary to prepare it for analysis and modelling. This aids in variable transformation, missing value management, and data formatting and cleaning. It is an important step in the data analysis process because it greatly affects the analysis's results and the model's efficacy.

After preparing the data, it is usual to split the dataset into a training set and a test set. The training set is used to fit the model, and the test set is used to evaluate the model's performance. By segmenting the dataset in this way, data from the training set are included in the test set, enabling an unbiased evaluation of the model. In order to ensure that the training and test sets are accurate representations of the complete dataset, they must have been selected at random.

5.3 Applying Decision Tree and other Algorithms

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

DecisionTree.fit(Xtrain,Ytrain)

predicted_values = DecisionTree.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)

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

```
DecisionTrees's Accuracy is:  90.0
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.59      1.00      0.74        16
    chickpea       1.00      1.00      1.00        21
     coconut       0.91      1.00      0.95        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.74      0.93      0.83        28
 kidneybeans       0.00      0.00      0.00        14
      lentil       0.68      1.00      0.81        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
```


Fig.5.3.1. Applying Decision Tree Code Snippet

Guassian Naive Bayes

```
from sklearn.naive_bayes import GaussianNB

NaiveBayes = GaussianNB()

NaiveBayes.fit(Xtrain,Ytrain)

predicted_values = NaiveBayes.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Naive Bayes')
print("Naive Bayes's Accuracy is: ", x)

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

```
Naive Bayes's Accuracy is:  0.990909090909091
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       1.00      1.00      1.00        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.88      1.00      0.93        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       1.00      1.00      1.00        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
   mothbeans       1.00      1.00      1.00        19
    mungbean       1.00      1.00      1.00        24
```


Fig.5.3.2. Applying Gaussian Naive Bayes Code Snippet

Random Forest

```
from sklearn.ensemble import RandomForestClassifier

RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain,Ytrain)

predicted_values = RF.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)

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

```
RF's Accuracy is:  0.990909090909091
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.94      1.00      0.97        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.90      1.00      0.95        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       1.00      1.00      1.00        23
```

Fig.5.3.3 Applying Random Forest Code Snippet

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

LogReg = LogisticRegression(random_state=2)

LogReg.fit(Xtrain,Ytrain)

predicted_values = LogReg.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Logistic Regression')
print("Logistic Regression's Accuracy is: ", x)

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

```
Logistic Regression's Accuracy is:  0.9522727272727273
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.86      0.75      0.80        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       0.86      0.90      0.88        20
      grapes       1.00      1.00      1.00        18
        jute       0.84      0.93      0.88        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       0.88      1.00      0.94        23
       maize       0.90      0.86      0.88        21
       mango       0.96      1.00      0.98        26
```

Fig.5.3.4 Applying Logistic Regression Code Snippet

Support Vector Machine (SVM)

```
from sklearn.svm import SVC
# data normalization with sklearn
from sklearn.preprocessing import MinMaxScaler
# fit scaler on training data
norm = MinMaxScaler().fit(Xtrain)
X_train_norm = norm.transform(Xtrain)
# transform testing dataabs
X_test_norm = norm.transform(Xtest)
SVM = SVC(kernel='poly', degree=3, C=1)
SVM.fit(X_train_norm,Ytrain)
predicted_values = SVM.predict(X_test_norm)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('SVM')
print("SVM's Accuracy is: ", x)

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

```
SVM's Accuracy is:  0.9795454545454545
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       1.00      1.00      1.00        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      0.95      0.98        22
      cotton       0.95      1.00      0.98        20
      grapes       1.00      1.00      1.00        18
        jute       0.83      0.89      0.86        28
```

Fig.5.3.5 Applying SVM Code Snippet

5.4 Accuracy Comparison

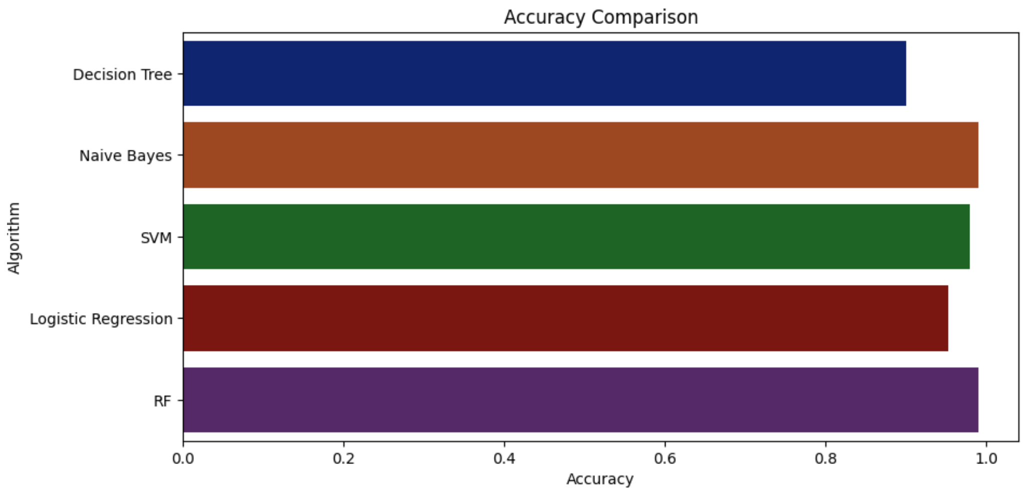

Fig.5.4. Accuracy Comparison

Above image shows the accuracy comparison of all algorithms

.

5.5 Final Results

```
data = np.array([[104,18, 30, 23.603016, 60.3, 6.7, 140.91]])
prediction = RF.predict(data)
print(prediction)
```

```
['coffee']
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:420: UserWa
  warnings.warn(
```

```
data = np.array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])
prediction = RF.predict(data)
print(prediction)
```

```
['jute']
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:420: UserWa
  warnings.warn(
```

Fig.5.5. Results

In the above Image we have predicted the crop by giving input parameters and selecting the Random Forest Algorithm for prediction of best crop .

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

At present, farmers in India do not have the right information, education, and technology to analyze the right crops according to the crop production affecting parameters which generally leads to selection of the wrong crops. This brings upon lot of losses to the farmer in the form of spoiled harvest and monetary losses, leading to many farmer suicides. To tackle this situation, we successfully proposed and developed an intelligent crop recommendation system that farmers across India may use.

We can use this research to increase the nation's output and revenue. Farmers are thus able to grow the ideal crop, increasing both their output and the overall profitability of the nation. To depict the suggestions of diverse Indian crops, a variety of machine learning techniques like "Decision Tree, Naive Bayes, SVM, Logistic Regression, Random Forest" were utilized. These five different categories of artificial intelligence algorithms were the subject of the investigation, with Random Forest and Naive Bayes showing the greatest accuracy with 99.09% .It is far better than other ML Algorithms.

6.2 FUTURE SCOPE

In the future, it is anticipated that this trained model would use a machine learning method to deliver results from other datasets that are faster and more accurate. It is intended to train this model on huge datasets with thousands of rows and columns in the near future. Additionally, it is anticipated that in the future, this model will be tested with additional machine learning algorithms to determine an effective way to predict the results and that work will be done on the application's user interface to make it easier for end users to use and to do so with the least amount of training or prior knowledge.

CHAPTER 7

REFERENCES

[1] Katarya, R., Raturi, A., Mehndiratta, A., & Thapper, A. (2020). *Impact of Machine Learning Techniques in Precision Agriculture*. https://doi.org/10.1109/icetce48199.2020.9091741

[2] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine Learning in Agriculture: A Review. *Sensors*, *18*(8), 2674. https://doi.org/10.3390/s18082674

[3] Balkrishnan, N., & Muthukumarasamy, G. (2022). Crop production Ensemble Machine Learning model for prediction.
International Journal of Computer Science and Software Engineering (IJCSSE)

[4] Jia, Z. (2009). *An Expert System Based on Spatial Data Mining Used Decision Tree for Agriculture Land Grading*. https://doi.org/10.1109/icicta.2009.751

[5] Khan, M., Islam, Z., & Hafeez, M. (2012). Evaluating the performance of several data mining methods for predicting irrigation water requirements. In *Australasian Data Mining Conference* (pp. 199–207). http://crpit.com/confpapers/CRPITV134Khan.pdf

[6] Dhanya, C. T., & Kumar, D. (2009). Data mining for evolution of association rules for droughts and floods in India using climate inputs. *Journal of Geophysical Research*, *114*(D2). https://doi.org/10.1029/2008jd010485

[7] Rale, N., Solanki, R., Bein, D., Andro-Vasko, J., & Bein, W. W. (2019). *Prediction of Crop Cultivation*. https://doi.org/10.1109/ccwc.2019.8666445

[8] Ramesh, D., & Vardhan, B. V. (2015). ANALYSIS OF CROP YIELD PREDICTION USING DATA MINING TECHNIQUES. *International Journal of Research in Engineering and Technology*, *04*(01), 470–473. https://doi.org/10.15623/ijret.2015.0401071

[9] Mishra, S., Mishra, D., & Santra, G. H. (2016). Applications of Machine Learning Techniques in Agricultural Crop Production: A Review Paper. *Indian Journal of Science and Technology*, *9*(38). https://doi.org/10.17485/ijst/2016/v9i38/95032

[10] Baisali, G. (2016) A Study to Determine Yield for Crop Insurance using Precision Agriculture on an Aerial Platform. Symbiosis Institute of Geoinformatics. Symbiosis International University 5th & 6th Floor, Atur Centre, Gokhale Cross Road, Model Colony, Pune – 411016.

[11] Jeong, J. H., Resop, J. P., Mueller, N. D., Fleisher, D. H., Yun, K., Butler, E. E., Timlin, D., Shim, K., Gerber, J. S., Reddy, V. R., & Kim, S. (2016). Random Forests for Global and Regional Crop Yield Predictions. *PLOS ONE*, *11*(6), e0156571. https://doi.org/10.1371/journal.pone.0156571

[12] Ramesh, D., & Vardhan, B. V. (2015b). ANALYSIS OF CROP YIELD PREDICTION USING DATA MINING TECHNIQUES. *International Journal of Research in Engineering and Technology*, *04*(01), 470–473. https://doi.org/10.15623/ijret.2015.0401071

[13] Yethiraj, N. G. (2012). Applying Data Mining Techniques in the Field of Agriculture and Allied Sciences. *International Journal of Business Intelligence, 001*(002), 40–42. https://doi.org/10.20894/ijbi.105.001.002.004

[14] Ananthara, M. G., Arunkumar, T., & Hemavathy, R. (2013, February). CRY an Improved crop yield prediction model using bee hive clustering approach for agricultural data sets. In 2013 International Conference Pattern Recognition, Informatics and Mobile on Engineering (pp. 473-478). IEEE.

[15] Bang, S., Bishnoi, R., Chauhan, A. S., Dixit, A. K., & Chawla, I. (2019, August). Fuzzy Logic based Crop Yield Prediction using Temperature and Rainfall parameters predicted through ARMA, SARIMA, and ARMAX models. In 2019 Twelfth International Conference on Contemporary Computing (IC3) (pp. 1-6). IEEE.

[16] Gandhi, N., Armstrong, L. J., Petkar, O., & Tripathy, A. K. (2016, July). Rice crop yield prediction in India using support vector machines. In 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE) (pp. 1-5). IEEE.

[17] Gandhi, N., Petkar,O.,& Armstrong, L. J.(2016,July).Rice Crop Yield prediction using Artificial neural Networks.In 2016 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR) (pp 105-110). IEEE

[18] Awan, A. M., & Sap,M. N. M.(2006,April). An intelligent system based on krenel methods for crop Yield Prediction, In Pacific Asia Conference on Knowledge Discovery and Data Mining (PP.841-846). springer, Berlin, Heidelberg.

[19] Manos, B., Bournaris, T., Papathanasiou, J., & Chatzinikolaou, P. (2008). Evaluation of tobacco cultivation alternatives under the EU common agricultural policy (CAP). *Journal of Policy Modeling* 2008; 31 (2): 225-238.

[20] Wu, J., & GE, X. 2012. Optimization Research of Generation Investment Based on Linear Programming Model. *International Conference on Applied Physics and Industrial Engineering*, 1400-1405.

[21] Banasik, A.; Kanellopoulos, A.; Bloemhof-Ruwaard, J.M.; Claassen, G.D.H. Accounting for uncertainty in eco-effificient agri-food supply chains: A case study for mushroom production planning. *J. Clean. Prod.* **2019**, *216*, 249–256. [CrossRef]

[22] Cheraghalipour, A.; Paydar, M.M.; Hajiaghaei-Keshteli, M. A bi-objective optimization for citrus closed-loop supply chain using Pareto-based algorithms. *Appl. Soft Comput.* **2018**, *69*, 33–59. [CrossRef].

[23] Deressa, T. & Hassan R. 2009. Economic Impact of Climate Change on Crop Production in Ethiopia: Evidence from Cross-section Measures. *Journal of African Economies*, August 2009, vol. 18, issue 4, pp.529-554.

[24] Frizzone, J.A., Coelho, R.D., Dourado-Neto, D. & Soliani, R. 1997. Linear Programming model to optimize the water resource use in irrigation projects: an application to the Senator Nilo Coelho Project. *Sci. agric. (Piracicaba, Braz.) [online]*. 1997, vol.54, n.spe, pp. 136-148.

[25] NARDS, National Agricultural and Rural Development Strategy 2007-2013. Ministry of Agriculture, Forestry and Water Economy, 2007.

[26] Kaiser, H. 1991. Climate change and agriculture. *Northeastern Journal of Agricultural and Resource Economics*, volume 20, issue 2, pp.151-163.