

A project report on

CROP PREDICTION BASED ON CHARACTERISTICS OF THE AGRICULTURAL ENVIRONMENT USING VARIOUS FEATURE SELECTION TECHNIQUES AND CLASSIFIERS

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

SINGAREDDY OMSIMHA REDDY (19BCE1214)



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2023

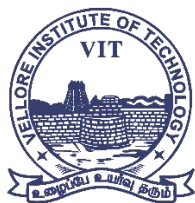
CROP PREDICTION BASED ON CHARACTERISTICS OF THE AGRICULTURAL ENVIRONMENT USING VARIOUS FEATURE SELECTION TECHNIQUES AND CLASSIFIERS

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

SINGA REDDY OMSIMHA REDDY (19BCE1214)



VIT[®]

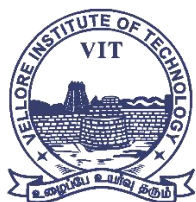
Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April, 2023



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

DECLARATION

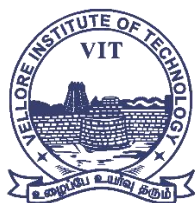
I here by declare that the thesis entitled “ CROP PREDICTION BASED ON CHARACTERSTICS OF THE AGRICULTURAL ENVIRONMENT USING VARIOUS FEATURE SELECTION TECHNIQUES AND CLASSIFIERS” submitted by me, for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Vellore Institute of Technology, Chennai, is a record of bonafide work carried out by me under the supervision of PARKAVI.K.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date:

Signature of the Candidate



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled “CROP PREDICTION BASED ON CHARACTERISTICS OF THE AGRICULTURAL ENVIRONMENT USING VARIOUS FEATURE SELECTION TECHNIQUES AND CLASSIFIERS” is prepared and submitted by **SINGAREDDY OM SIMHA REDDY (19BCE1214)** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** programme is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr./Prof.

Date:

Signature of the Examiner 1

Name:

Date:

Signature of the Examiner 2

Name:

Date:

Approved by the Head of Department
B. Tech. CSE

Name: Dr. Nithyanandam P

Date: 24 – 04 – 2023

ABSTRACT

ABSTRACT

The study of agriculture is gaining popularity. Because soil factors like temperature, humidity, and precipitation have a significant impact, crop prediction is essential in agriculture. In the past, farmers could choose which crop to plant, assess its viability, and choose when to harvest it. Despite this, producers can't imagine taking such a step because of the rapid change in the climate. As a result, machine learning (ML) calculations have recently taken on the role of forecasting. In this study, a variety of these methods were used to figure out agricultural production. Effective feature selection techniques must be utilized to transform unprocessed data into a dataset suitable for machine learning in order to guarantee the accuracy of a particular machine learning (ML) model. Only data characteristics with a significant impact on the model's output ought to be included in order to improve model quality and reduce duplication. The model has just the most fundamental highlights because of the ideal element determination. Our model will be too complex if every characteristic from the original data is added to it without its corresponding value. Additionally, if factors with little impact were included, the model's output would become less accurate. The results show that a troupe method performs better than the current characterization system in terms of prediction accuracy.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr PARKAVI.K professor, SCOPE, Vellore Institute of Technology, Chennai, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Network Security.

It is with gratitude that I would like to extend thanks to our honorable Chancellor, Dr. G. Viswanathan, Vice Presidents, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan and Mr. G V Selvam, Assistant Vice-President, Ms. Kadhambari S. Viswanathan, Vice-Chancellor, Dr. Rambabu Kodali, Pro-Vice Chancellor, Dr. V. S. Kanchana Bhaaskaran and Additional Registrar, Dr. P.K.Manoharan for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dean, Dr. Ganesan R, Associate Dean Academics, Dr. Parvathi R and Associate Dean Research, Dr. Geetha S, SCOPE, Vellore Institute of Technology, Chennai, for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Nithyanandam P, Head of the Department, Project Coordinators, Dr. Abdul Quadir Md, Dr. Priyadarshini R and Dr. Padmavathy T V, B. Tech. Computer Science and Engineering, SCOPE, Vellore Institute of Technology, Chennai, for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staff at Vellore Institute of Technology, Chennai, who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date:

S.OMSIMHA REDDY

TABLE OF CONTENTS

S.NO	CONTENT	PGNO
1	Introduction	1
2	Literature survey	7
3	System analysis 3.1 Existing system 3.1.1 Disadvantages of existing system 3.2 Proposed system 3.2.1 Advantages of proposed system 3.3 Functional requirements 3.4 Non-Functional requirements	13
4	System design 4.1 System architecture 4.2 UML diagrams	17
5	Implementation 5.1 Modules 5.2 Sample code	30
6	Software environment	37
7	System testing 7.1 Testing strategies 7.2 Test cases	45
8	Screenshots	51
9	Appendix	55
10	Conclusion	57
11	References	59

LIST OF FIGURE

FIG.NO	FIG.NAME	PG.NO
4.1.1	System architecture	
4.1.2	Flow diagram	
4.1.3	Dataflow diagram	
4.2.1	Usecase diagram	
4.2.2	Class diagram	
4.2.3	Activity diagram	
4.2.4	Sequence diagram	
4.2.5	Collaboration diagram	
4.2.6	Component diagram	
4.2.7	Deployment diagram	

INTRODUCTION

1. INTRODUCTION

Crop prediction is a complicated process in agriculture for which numerous models have been proposed and evaluated. The issue necessitates the utilization of a variety of datasets due to the fact that crop cultivation is influenced by both biotic and abiotic factors. Biotic factors are elements of the environment that are the result of the direct or indirect influence of living organisms on other living organisms (such as microorganisms, vegetation, animals, parasites, predators, and vermin). Likewise remembered for this classification are anthropogenic variables (preparation, plant insurance, water system, air contamination, water contamination, and soils, and so on.).

These elements might add to the event of various changes in crop yield, causing inward imperfections, morphological deformities, and modifications in the compound structure of plant yield. Abiotic and biotic elements impact the arrangement of the climate as well as the turn of events and nature of plants. Abiotic elements can be isolated into physical, compound, and extra classes. Physical factors include climatic conditions (atmospheric pressure, temperature, humidity, air movements, sunlight), mechanical vibrations (vibration, noise), radiation (such as ionizing, electromagnetic, ultraviolet, and infrared), soil type, topography, soil rockiness, atmosphere, and water chemistry, particularly salinity. Sulfur dioxide and its derivatives, polycyclic aromatic hydrocarbons (PAHs), nitrogen oxides and their derivatives, fluorine and the compounds it contains, lead and the compounds it contains, cadmium and the compounds it contains, nitrogen

fertilizers, pesticides, and carbon monoxide are among the top environmental pollutants. Mercury, arsenic, dioxins and furans, asbestos, and aflatoxins are the others. Abiotic factors, in addition to bedrock, relief, climate, and water conditions, influence its properties. There are many ways that soil-forming factors affect soil formation and agricultural value.

1.1 SOFTWARE REQUIREMENTS

Programming necessities are worried about showing the thing assets and essentials that should be familiar on a PC with give the best demonstration of a program. More often than not, these conditions or fundamentals ought to be presented independently before the program can be presented in light of the fact that they are routinely not piece of the item foundation group.

Platform – A system known as a "registration step" makes it possible to use software in either hardware or software. The design of a computer, its operating system, programming languages, and their runtime libraries are all common stages.

While portraying structure needs, one of the central measures examined is working framework (programming). Even if some reverse similarity is occasionally maintained, newer versions of the same group of operating systems may not permit programming. For instance, Microsoft Windows 98 cannot be used with the majority of Microsoft Windows XP applications; nonetheless, this isn't generally the situation. Programming made with later variants of the bit keeps an eye out won't run or total accurately on Linux dispersions running Piece renditions v2.2 or v2.4.

APIs and drivers – Particular APIs or later device drivers are required for programming that places a heavy reliance on particular hardware, such as high-end display connectors. DirectX, a lot of Microsoft platforms' APIs for overseeing mixed media-related projects, particularly game development, is a striking example.

Web browser – The majority of web applications and programming that comprehensively relies upon Web development use the structure's fundamental default program. No matter what the imperfections in ActiveX controls, Microsoft Web Voyager is a notable piece of programming that unexpected spikes popular for the Windows working system.

1. Node.js with the Version 12.3.1

2. Visual Studio Social class;

3. Python IDEL with Python 3.7

1.2 HARDWARE REQUIREMENTS

The genuine PC property, as a rule suggest as provisions, are extreme winning sythesis of subtleties from some dynamic establishment or figure program. A summary of the fundamental supplies typically comes with a hardware compatibility list (HCL), particularly for difficult-to-work foundations. A HCL describes supplies components that have been evaluated, are practicable, and rarely are not for a specific active foundation or use. The following sections conceal the abundance of essential supplies.

Architecture – Every PC's dynamic establishment is exclusively worked to fit an interesting PC design. Unambiguous compositional and occupied foundation requirements are adhered to by the majority of register programs. The majority of active, undesignated foundations and projects should be recompiled to accommodate a different manufacturing, despite the fact that there are some. Also, check out a list of popular models and operating systems that help the ruling class.

Processing power – A central processor related with an adequate administration cutback should be connected to a solicitation. A significant portion of the moment of truth gone communicating accompanying capacity in x86-located applications is demonstrated by the PC vendor's model and alarm speed. Transport speed, store, and MIPS are any unique PC essential piece of PC conditions that effect killing breaking point really are reliably neglected. As AMD Athlon and Intel Pentium main processors occasionally have different throughput speeds at comparable alarm speeds, this goal of force is consistently incorrect. Since they have accomplished a critical degree of distinction and are totally examined for participation.

Memory – When a PC is being used, all brand data is stored in its random access memory (RAM). Thought essentials are arranged after prioritizing everything—the solicitations of the solicitation, dynamic establishment, collaborative projects and records, and various dynamic stages. The best presentation of extra minor applications running on a working differentiated undertakings PC establishment is likewise thought about while choosing these models.

Secondary storage – The extent of the program association, the part of brief records brought and got a handle on meanwhile focus on establishment or exercise, and the expected activity of work scope all impact through what much hard plate room vital (self-important RAM is insufficient).

Display adapter – First in class show connectors are quite a bit of decision time driven in the establishment necessities of focus on like plans editors and significant diversion that demand an exhibition going with greater PC drawings than the norm.

Peripherals – Some brand programs need the reduced and furthermore distinguishing exercise of distinguishing element, that demands the better performance or value of those element. These element combine belongings like consoles, directed device, Album ROM drives, and institution designs.

1)Operating System: Only Windows

2) Processor: i5 or later

3) Ram: 4 GB or more

4) 50 GB on a hard drive

LITERATURE SURVEY

2.LITERATURE SURVEY

2.1 Applying naive Bayes classification technique for classification of improved agricultural land soils:

https://www.researchgate.net/publication/309212171_Applying_Naive_Bayes_Data_Mining_Technique_for_Classification_of_Agricultural_Land_Soils

ABSTRACT: A lot of data has been created as a result of personal computers and data storage systems' development. New instruments and strategies, like data extraction, have been made to aid the end of the information partition. The problem has been figuring out how to get information from these raw data. The goal of this study was to test these well-known data mining methods on a soil science instructional index to see if there were any significant correlations. The Division of Soil Sciences and Rustic Science at the S.V. Cultivating School has made various soil informational indexes available. Soil profile estimates from a few locations in the Chittoor Area, Chandragiri Mandal, were included in the data that were gathered. The study employs a variety of data mining methods to determine whether sediments are classified. Additionally, a connection was established between the description of Naive Bayes and the evaluation of the most effective strategy. The findings of the review could be applied to a variety of initiatives related to environmental security, land management, and agriculture.

2.2 Biotic components influencing the yield and quality of potato tubers

http://agronomyaustraliaproceedings.org/images/sampled/2015_Conference/pdf/agronomy2015final00047.pdf

ABSTRACT: Over the past ten years, potato yields in Canterbury have remained relatively constant at 60 t/ha. However, harvests of up to 90 t/ha are predicted by potato development forecasts, which some industrial producers have already achieved. Two years were given to the examination of farming result restrictions by business and scholastic accomplices. 11 harvests were diminished dynamically in anticipation of the basic establishing season. ongoing yield, plant wellbeing, and soil quality assessments) Lower harvests have been credited to soil-borne illnesses, for example, Rhizoctonia stem rankle and Spongospora root contamination, dirt compaction, and insufficient water framework the executives. The effects of Rhizoctonia stem lesions manifested more rapidly (by increasing) in areas where potato crops had never been cultivated before and when vegetation was developing. A controlled field study on a market crop with a high concentration of soil-borne microorganisms was carried out in the second year to ascertain how yield was affected by soil-borne diseases. Flusulphamide (400 ml per ha), in-wrinkle azoxystrobin (1.5 l per ha), and chloropicrin (90, 112, and 146 kg per ha) were utilized as pesticide controls. After treatment (plots splashed with fumigant), soil-borne microbial DNA measures uncovered a humble reduction in Rhizoctonia solani and Spongospora subterranea DNA levels, yet the outcomes were definitely modified. The definitive new result at maximum scale was the mean value of 58 t/ha that the method produced. The azoxystrobin treatment reliably diminished the seriousness of R. solani's assault on underground roots all through the season, contrasted with different medicines.

2.3 Response surface methodology: A retrospective and literature survey

<https://www.tandfonline.com/doi/abs/10.1080/00224065.2004.11980252>

ABSTRACT: To further enhance cycle and item plans, RSM is a method that makes use of both actual planning and numerical normalization techniques. In the 1950s, the first investigations in this field were carried out. From that point forward, they have been generally used, especially in the drug and bike ventures. RSM has undergone significant development and extensive use over the past 15 years. The RSM practices that started in 1989 are the focal point of this rundown. We look at the current study topics and suggest more areas for research.

2.4 Application of response surface methodology for optimization of potato tuber yield

https://www.researchgate.net/publication/281612031_Application_of_Response_Surface_Methodology_for_Optimization_of_Potato_Tuber_Yield

ABSTRACT: The author of this study looks into the functional components that are necessary to increase the production of potato roots in Kenya. Potato producers are forbidden from providing any additional information in this regard. The potato production process was optimized through the use of the reaction surface method and Factorial Plans 2 and 3. Utilizing a reaction surface technique, the combined impacts of water, nitrogen, and phosphorous material improvements were explored and demonstrated. 70.04 percent water structure water, 124.75 kilograms per hectare of urea nitrogen, and 191.04 kilograms per hectare of triple super phosphate phosphorus make up the ideal potato root formulation. The ideal

conditions allow for a yield of 19,36 kilograms of potato roots per allotment that is 1.80 meters by 2.25 meters. Smallholder potato producers in Kenya may see better living conditions and lower costs as a result of increased potato production. In addition, I anticipate that the method used in this potato study will be used in other studies, allowing for a more comprehensive comprehension of agricultural yield.

2.5 Improving potato yield prediction by combining cultivar information and UAV remote sensing data using machine learning

<https://www.mdpi.com/2072-4292/13/16/3322>

ABSTRACT: In precision agriculture, precise significant standard yield proposals are used to identify global yield abnormality patterns, clarify significant factors that influence yield variability, and provide information for site-specific management. If remote monitoring tools are used, changes in the varieties of guaging potato (*Solanum tuberosum* L.) may affect tuber production. This study planned to increment potato yield by coordinating harvest information with machine learning (ML) gauges in light of remote detecting by Unmanned aerial vehicles (UAVs). Numerous cultivars and nitrogen (N) ratios were used in 2018 and 2019 small allotment experiments. UAV photographs were taken on a routine basis throughout the development season. The ML methods Random Forest Regression (RFR) and Support Vector Regression (SVR) were used to incorporate varietal data and particular advancement complements. Unmanned aerial vehicle (UAV) data from the tuber start stage in the early developing season (late June) was found to have a stronger correlation with potato deliciousness than data from later in the season. Each potato variety has its own set of ideal growth indicators

and evaluation intervals for cultivation. The RFR and SVR models performed poorly with only distant separating data ($R^2 = 0.48-0.51$ for recommendation), whereas they performed significantly better with crop data ($R^2 = 0.75-0.79$ for support). Potato output projection is significantly improved over methods that do not use cultivar data when ML algorithms are used to integrate high-spatial-goal UAV images with cultivar data. By integrating more exact harvest data, soil and scene subtleties, regulatory information, and strong ML estimations, almost certainly, extra exploration will be directed to improve potato yield anticipating.

SYSTEM ANALYSIS

3.SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

Calculating the agroclimatic factors that influence the growth of winter plant species in the low temperature zone, particularly cereals, is the primary test. The number and recurrence of days with temperatures over 5 degrees Celsius and days with temperatures somewhere in the range of 0 and 5 degrees Celsius altogether affect winter yield. It is possible to investigate many of these by utilizing publicly available data and long-term recurrence metrics. The need for a state intervention strategy in the cereal market has been evaluated using established models. Meteorological boundary forecasting is required for the creation of precise theories. A specific issue might arise because these parts are unstable. This problem has been attempted by a variety of authorities, with varying degrees of success.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM:

1. Weather and soil factors like temperature, humidity, and precipitation have a significant effect on agricultural harvest forecasts.
2. Farmers have been not able to keep developing because of the unexpected environment changes.

3.2 Proposed System:

There are a few problems with this field of study. Although they are currently satisfactory, crop prediction calculations could be improved. This study gives a more exact model to yield estimating that tends to these worries. The prediction method is primarily based on sequence and feature selection (FS) methods. Processes are evaluated before FS techniques are used to balance a collection.

3.2.1 Advantages of proposed system:

1. Only data scores that have a high level of significance in determining the model's final output ought to be included in order to reduce apparent duplication and enhance the accuracy of ML models.
2. As far as gauge accuracy, an outfit technique beats a prior classification methodology.

3.3 FUNCTIONAL REQUIREMENTS

1. Data Collection
2. Data Pre-processing
3. Training and Testing
4. Modiling
5. Predicting

3.4 NON FUNCTIONAL REQUIREMENTS

The status trade name of a point construction is normalized for one NON-FUNCTIONAL REQUIREMENT (NFR). They judge current fashion construction as proved by non-auxiliary models like openness, solace, freedom, convenience, and various rules that are key for current fashion foundation's prospering. The meaningless question of "by means of how quickly does the page load?" is an example Structures that don't fit non-practical models won't meet customer needs if they can't be paid for. Using non-practical requirements, you can limit bureaucracy craftily across a variety of quick surpluses. For instance, the webpage should stack in three seconds precisely when there are more than 10,000 people online right away. Nevertheless, critical as a valuable need is, by all accounts, the representation of nonsensical requirements.

- Convenience necessities, sensibility essentials, recoverability necessities, security requirements, data trustworthiness necessities, limit essentials, openness essentials, scalability necessities, interoperability requirements, constancy necessities, feasibility necessities, authoritative necessities, and normal necessities.

SYSTEM DESIGN

3. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:

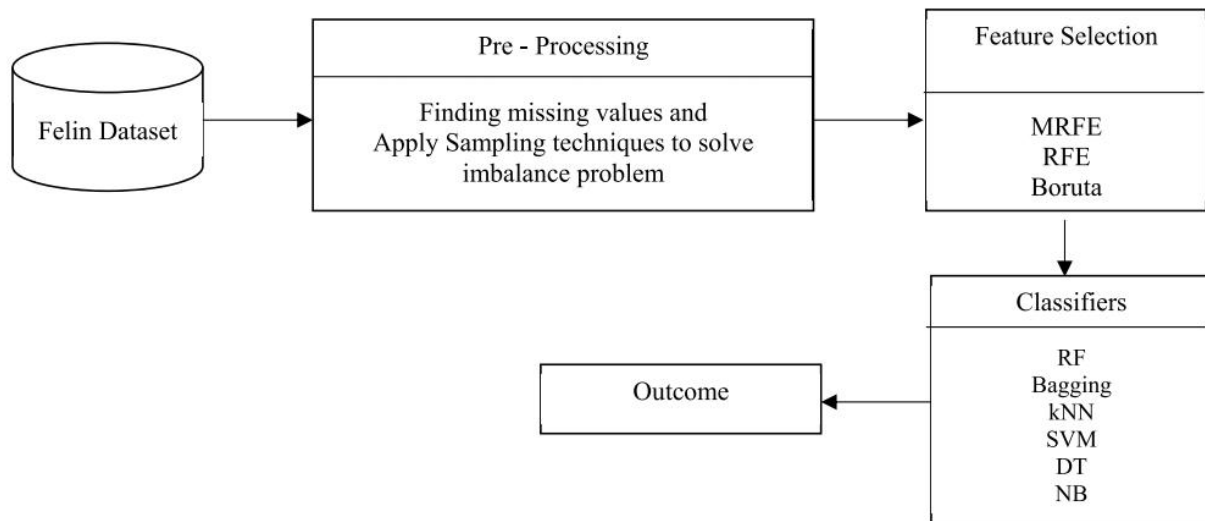


Fig.4.1.1 System architecture

DATA FLOW DIAGRAM:

1. Commonly, the air pocket outline is utilized to show the DFD. To use reasonable graphical behavior, an arrangement can be sent to the level of the document that will be handled within it, the various tasks that have been completed on it, and the document that will be passed as a result.
2. The news stream outline, more commonly known as a DFD, is one of the most important picture gadgets. It is used to make the part models for the establishment. These sections clearly establish the plan's activity, the dossier it uses, the matching confined, and the dossier travel through it.

3. DFD mimics the development of the news through organization and a charming response to various changes. It is a graphical penchant to manage inclined to the dossier stream and real factors refreshes as they progress from obligation to yield.
4. DFD is also known by the name bubble graph. A DFD store should give an establishment some thought. DFD maybe private into stages that are basically comparative from dossier stream and extending useful complicatedness.

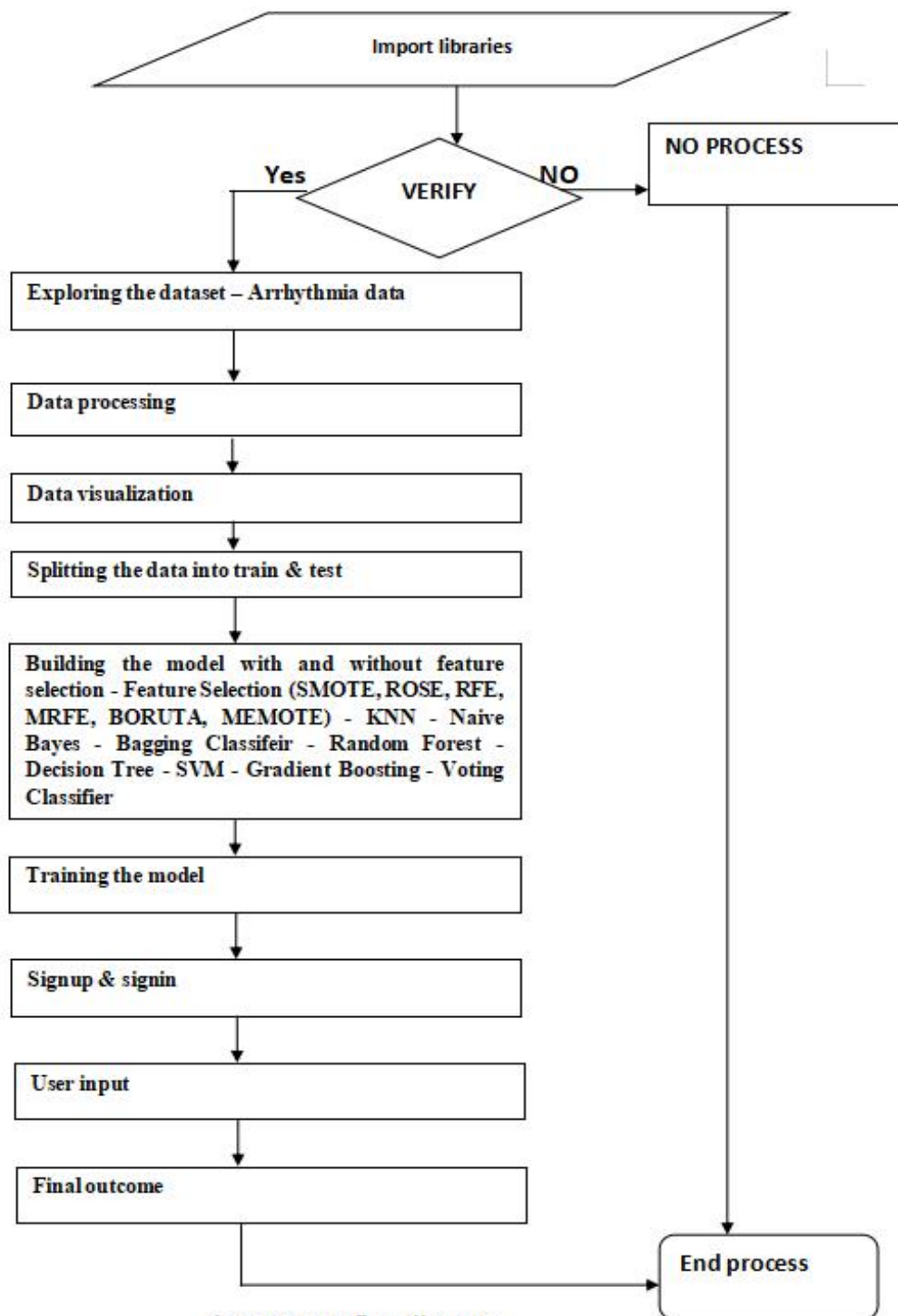


Fig.4.1.3 Dataflow diagram

4.2 UML DIAGRAMS

UML, or the Unified Modelling Language, is its abbreviation. In object-oriented computer programming, UML is a general-purpose, rules-based programming language. The Executive Gathering is liable for coordinating and sustaining the norm.

The creation of a standard language for displaying object-oriented PC programs is the goal of UML. The meta-model and the documentation are currently the two UML components with the greatest significance. As a methodology or cycle pushing ahead, UML can be integrated or expanded.

A standard language for drawing, imagining, putting together, and recording the parts of non-programming designs with a businesslike appearance and programming frameworks is the Unified Modelling Language.

The UML was created because of joining the best course of action progressions that were equipped for copying enormous, complex frameworks.

The process of improving things and creating structured programming relies heavily on the Unified Modeling Language (UML). The programming project plan is basically passed along with graphical UML documentation.

GOALS:

Coming up next are the rudimentary UML situation points:

1. Make it plausible for clients to make and business goliath models by giving organization an organized to-exploit telling optical style.
2. To support key plans, present plans to business and expandability.
3. not believe a particular prioritize terminology or bettering foundation.
4. Enable a happening to resolve the characteristic terminology.
5. Support the commercialization of OO implements.
6. Animate the exercise of supplementary fault-finding methods for level bettering, like parts, foundations, models, and collected endeavors.
7. Incorporate best choice actions.

Use case diagram:

The result of and depiction of a habit scene attracted evaluation is a distinguishing in a way direct drawing in the UML that is to say allude to as a use case outline. Taking entirety into report, an bettering's serviceableness for entertainers, their goals (influenced to by handle cases), and some friendship 'tween use cases will be described clearly. The fundamental aim of a use case outline search out analyze that entertainers benefit from what or which place building limits. They maybe easily met on by containing their regions in bureaucracy.

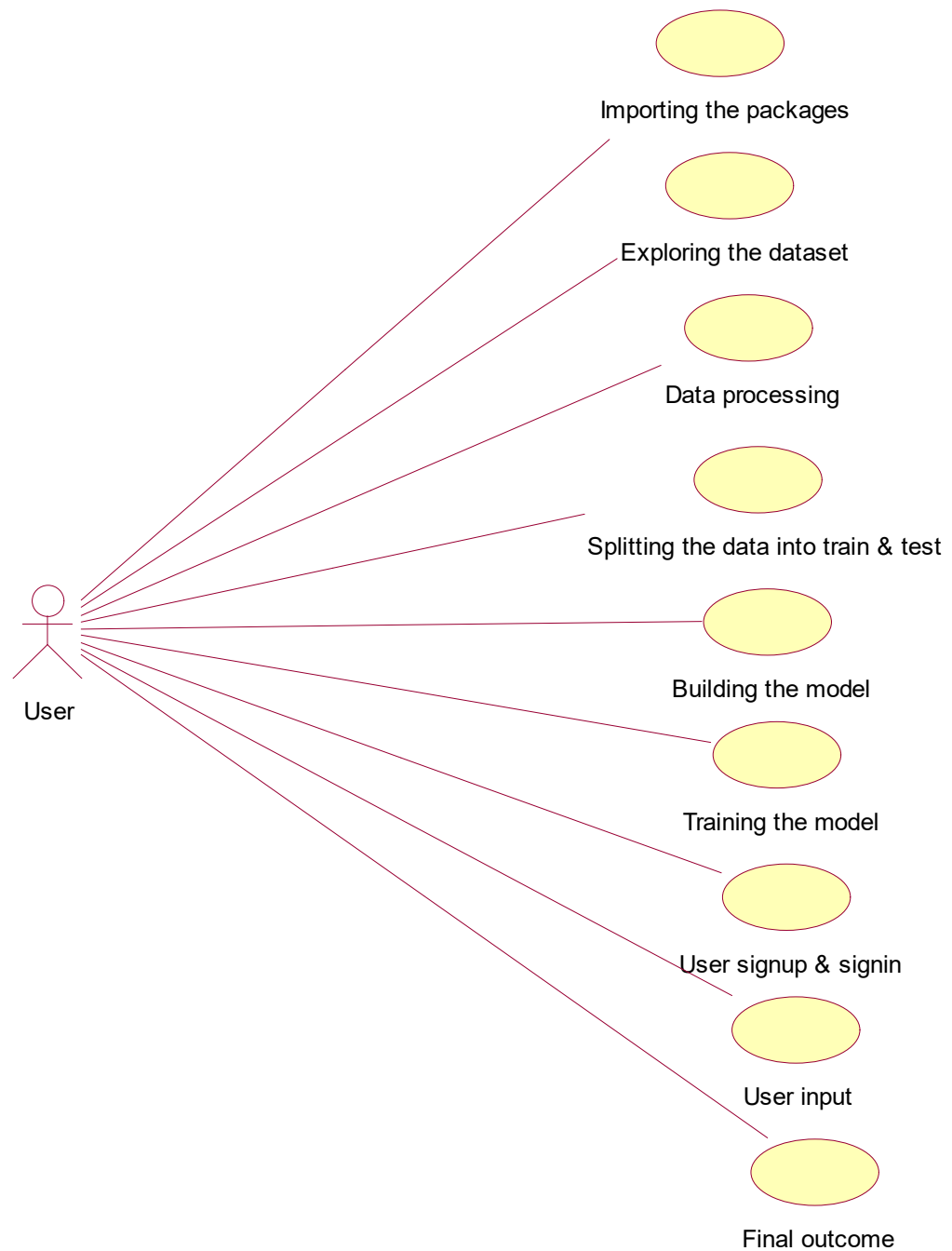


Fig.4.2.1 Usecase diagram

Class diagram:

The utilization case frame is uncluttered and a weighty structure plan is proved taking advantage of the class diagram. Utilizing the class frame, the entertainers in the habit case chart are organized into various pertain classes. A "can't forestall being a" or "has-a" fellowship center from two focuses the classes is possible. In the class frame, different capacity perhaps open each class. The lines of inquiry that the class addresses are its "approaches." In a similar vein, each class oversee receives distinct "credits" from other classes.

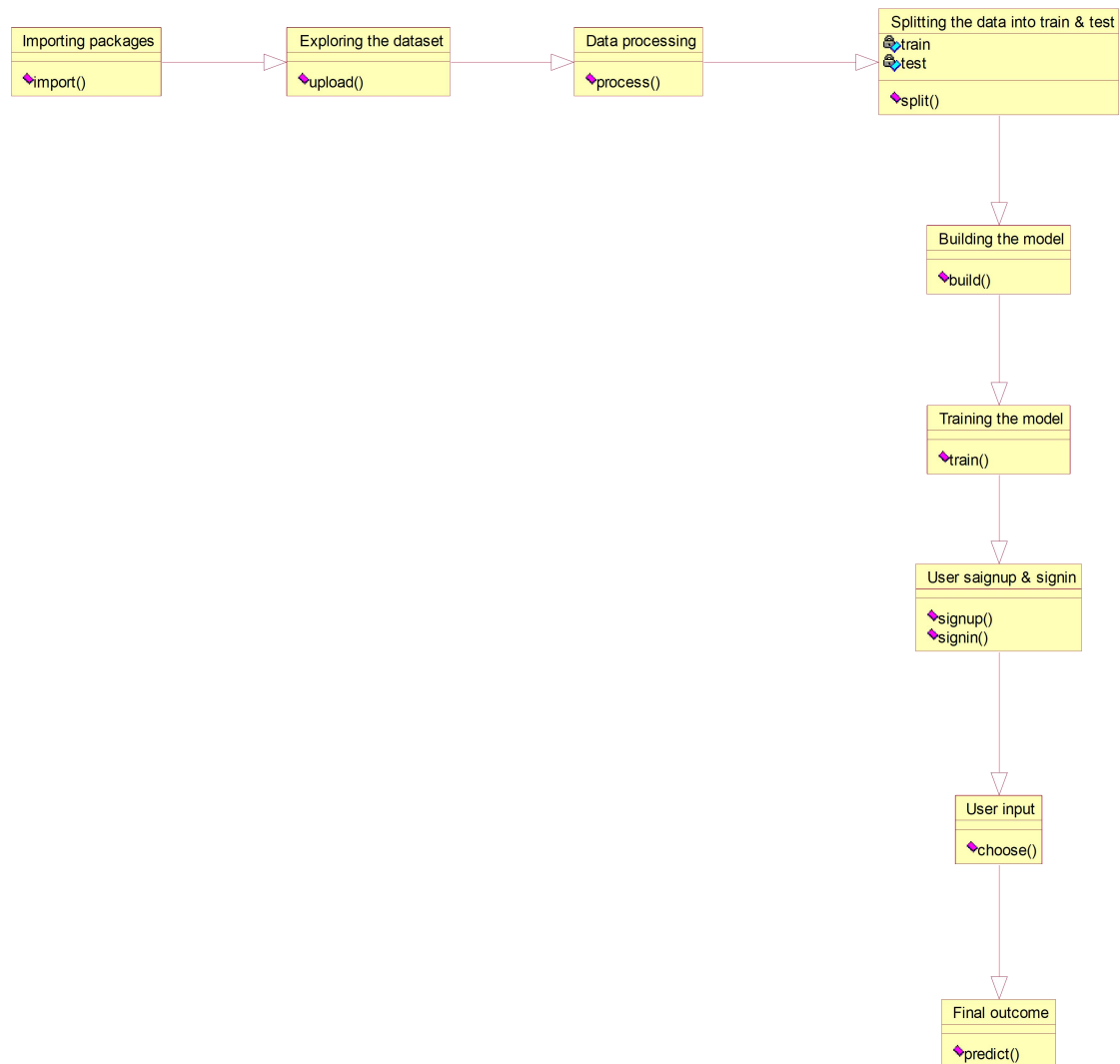


Fig.4.2.2 Class diagram

Activity diagram:

The activity graph depicts the establishment's stage floods. An activity graph has regular parts as a state frame, containing satisfies, satisfies, changes, offset and last states, and gatekeeper conditions.

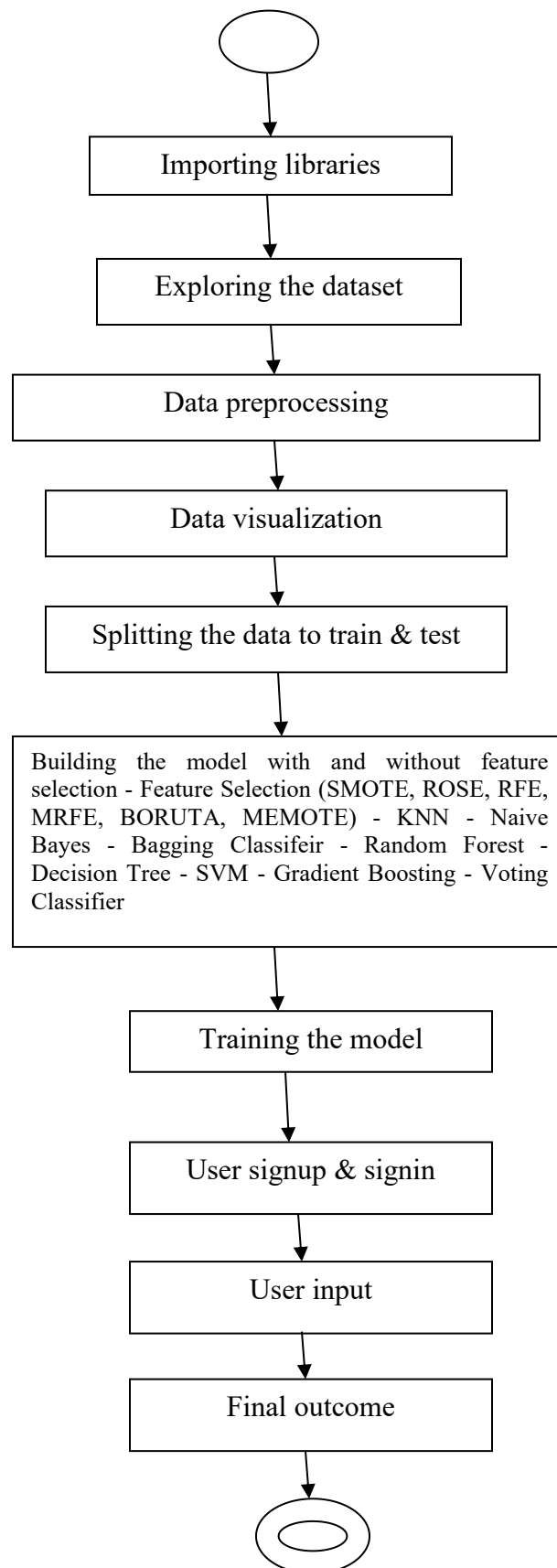


Fig.4.2.3 Activity diagram

Sequence diagram:

The friendships that lie betwixt various parts of the building are imitated in a grouping chart. How a assemblage frame is period-noticed is essential. This exhibits that the exact appeal for latest fashion relationships is imported step by step. The accumulation frame exhibits the "ideas" by which various belongings talk accompanying each one.



Fig.4.2.4 Sequence diagram

Collaboration diagram:

A collaboration frame gathers the correspondences 'tween referring to a specifically known amount of belongings. To manage more easy to understand their appeal, the joint works are popularized as limited in number correspondences. The cooperation outline form it more simple to see by means of what each item take care of buy and sell various belongings.

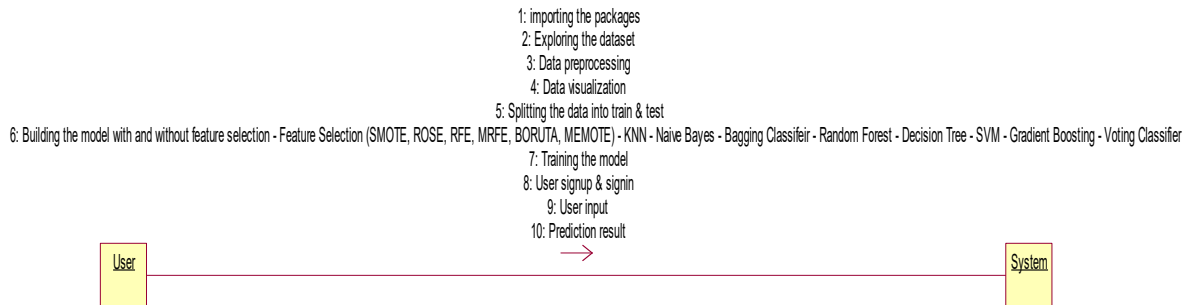


Fig.4.2.5 Collaboration diagram

Component diagram:

The part chart shows the problem with the framework's finding level parts. The components of the structure and the relationships between them are clearly depicted on this blueprint. Any chart shows the parts that have been released after the administration has moved through the development or improvement stage.

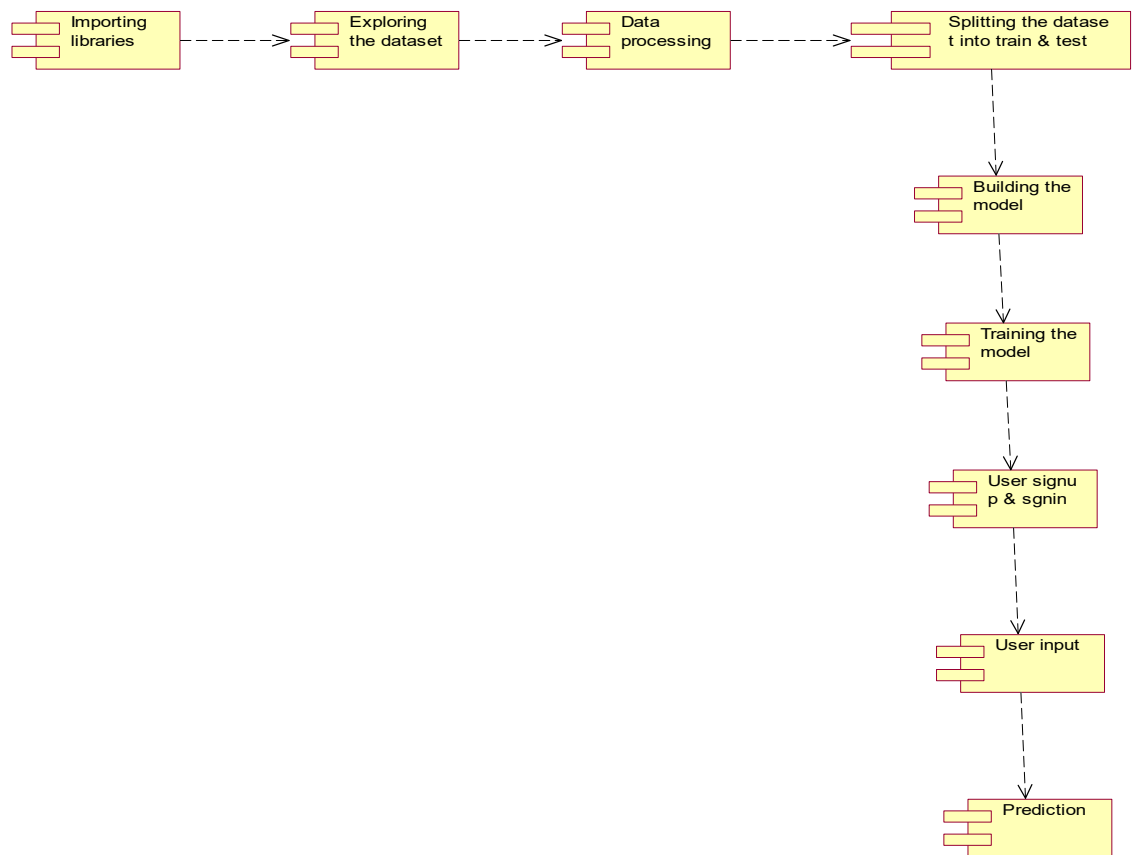


Fig.4.2.6 Component diagram

Deployment diagram:

The organization frame organizes the runtime components of the solicitation. This reasonable is usually strong as soon as a method is finished and ready for transportation.



Fig.4.2.7 Deployment diagram

IMPLEMENTATION

5.IMPLEMENTATION

MODULES:

1. import shipment: This utility will be used to load all shipments.

The datasets containing cropped data from data exploration will be submitted using this instrument.

3. information purifying and handling: Using this tool, we will receive data for processing.

4. visualization with seaborn and matplotlib: This program outwardly addresses information and data.

5. The data will be divided into train and test sets with this instrument for processing.

6. Optional Functions (SMOTE, ROSE, RFE, MRFE, BORUTA, and MEMOTE)
Voting Classification, Decision Tree, Support Vector Machine, and Gradient Boosting

- 7 Model education: Create the Voting Classifier model for processing and forecasting using the algorithms taught in this module because it is more accurate than other models.

8. For registration and login, Sqlite and the Flask Framework are used: Importing programs is how users can register and log in with this module.

9. The user enters values for features like: Prediction input is provided by the user when this module is used; Predictive processing is carried out on the input.

10. Expectations are based on the taught model: The interface is used to display the anticipated output made possible by this module.

Note: Because it is approximately one hundred percent more precise than other models, the Gradient Boosting Voting Classifier is used as an extension for feature values that are used in forecasting.

ALGORITHMS:

Feature Selection (SMOTE, ROSE, RFE, MRFE, BORUTA, MEMOTE)

KNN: " The abbreviation for "K-Nearest Neighbor" is "KNN." It is a computation regulated by man-made intelligence. Relapse and issue clustering can both be managed with this strategy. The letter "K" addresses the quantity of variables with nearest neighbors that can be anticipated or classified.

Naive Bayes: To characterize, Naive Bayes' probability classification is used. It is demonstrated with the help of suspicions and flexible probability models. Concepts of freedom rarely correspond with reality. As a result, people believe they are stupid.

Bagging Classifier: A bagging classifier is a meta-evaluation that interfaces individual speculations through the utilization of key classifiers to irregular sections of the hidden dataset. By integrating randomization into its development cooperation and afterward building a gathering from it, this sort of meta-assessor is generally used to diminish the hole among itself and a discovery assessor, for example, a choice tree.

RF: The Random Forest For regulated learning, Random Forest is an essential machine learning technique. It can be applied to ML problems like planning and recurrence. It is established on the idea of collection realizing, which joins various orders to handle a difficult issue and improve the model's presentation. The Random Forest predictor, as its name suggests, "takes the mean of multiple decision trees on diverse subsets of the supplied dataset" and "works on the extended exactness of that dataset." Instead of depending just on a solitary chose tree, the scattered woods foresee the result by incorporating estimations from each tree and utilizing the majority of projections.

DT: Decision trees use a variety of criteria to decide whether or not to divide a hub into at least two sub-hubs. The uniformity of the subnodes is enhanced by their appearance. To put it another way, as the center gets closer to the target variable, it becomes more refined.

SVM, or Support Vector Machine,: Backslide and depiction make use of the well-known Unsupervised Learning method known as Support Vector Machine (SVM). Nonetheless, most of its ML applications center around order issues. The goal of the SVM strategy for n-layered space configuration is to decide the ideal line or judgment limit so ensuing information can be added to the right grouping easily. A hyperplane is the most straightforward and logical limit.

Boosting the Gradient: Backslide and arrange applications much of the time utilize slope helping, an ML procedure. A collection of unreliable forecast models, most commonly decision trees, is called an anticipation model. When a decision tree takes on the role of a helpless student, gradient-boosted trees are created. This method does not always outperform random trees. The subsequent development of a gradient-boosted forests model is similar to earlier aid methods, but it surpasses them by allowing for the creation of any observable disaster capacity.

Voting Classifier: A machine learning predictor known as a voting classifier is capable of predicting the outcomes of multiple fundamental models or estimators. Each estimator result's sum parameter could be linked to polling options.

Sample Code :

```
#importing
the
required
libraries

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
#Reading the csv file
data=pd.read_csv('cpdata.csv')
print(data.head(1))
#Creating dummy variable for target i.e label
label= pd.get_dummies(data.label).iloc[:, 1:]
data= pd.concat([data,label],axis=1)
data.drop('label', axis=1,inplace=True)
print('The data present in one row of the dataset is')
print(data.head(1))
train=data.iloc[:, 0:4].values
test=data.iloc[:, 4:].values
#Dividing the data into training and test set
X_train,X_test,y_train,y_test=train_test_split(train,test,test_size=0.3)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
#Importing Decision Tree classifier
from sklearn.tree import DecisionTreeRegressor
clf=DecisionTreeRegressor()
#Fitting the classifier into training set
clf.fit(X_train,y_train)
pred=clf.predict(X_test)
from sklearn.metrics import accuracy_score
# Finding the accuracy of the model
a=accuracy_score(y_test,pred)
print("The accuracy of this model is: ", a*100)
#Using firebase to import data to be tested
```

```

from firebase import firebase
firebase =firebase.FirebaseApplication('https://cropit-eb156.firebaseio.com/')
tp=firebase.get('/Realtime',None)
ah=tp['Air Humidity']
atemp=tp['Air Temp']
shum=tp['Soil Humidity']
pH=tp['Soil pH']
rain=tp['Rainfall']
l=[]
l.append(ah)
l.append(atemp)
l.append(pH)
l.append(rain)
predictcrop=[l]
# Putting the names of crop in a single list
crops=['wheat','mungbean','Tea','millet','maize','lentil','jute','cofee','cotton','grou
nd nut','peas','rubber','sugarcane','tobacco','kidney beans','moth
beans','coconut','blackgram','adzuki beans','pigeon peas','chick
peas','banana','grapes','apple','mango','muskmelon','orange','papaya','watermelon','pom
egranate']
cr='rice'
#Predicting the crop
predictions = clf.predict(predictcrop)
count=0
for i in range(0,30):
    if(predictions[0][i]==1):
        c=crops[i]
        count=count+1
        break;
    i=i+1
if(count==0):
    print('The predicted crop is %s'%cr)
else:
    print('The predicted crop is %s'%c)
#Sending the predicted crop to database
cp=firebase.put('/croppredicted','crop',c)

```

SOFTWARE ENVIRONMENT

6.SOFTWARE ENVIRONMENT

PYTHON LANGUAGE:

Python is an interpreted, object-oriented, irrefutable level programming language. Python is a really semantic, object-coordinated, huge level, unraveled programming language. Considering its sure level express data structures, dynamic making, and dynamic confining, it is particularly beguiling for Practical Application Improvement in like manner to the degree that bid custom as a development or stick piece to consolidate existent parts. The reasonable emphasis placed on understandability in Python reduces the cost of program support. Python's support for collections and modules during projects that remove and reintroduce regulation. The Python negotiator and the total standard athenaeum are honestly distributable and open as beginning or binal strategy each definite foundation. Designs habitually have overwhelming inclinations for Python on account of its unrivaled capability. The modify test-investigate cycle is especially expedient since there is no diagram stage. Because of the way that division dissatisfaction is never brought about by a bug or erroneous information, Python programs are not difficult to research. At the point when the translator finds a misconception, it raises what is going on thinking about everything. In the event that the application doesn't get the special case, the mediator makes the stack proceed. Utilizing a source-level debugger, you can take a gander at clashing articulations, set breakpoints, assess close by and generally speaking factors, go line by line through the code, and check different parts out. Python's ability to do mind is shown by the debugger, which is made in Python. On the other hand, adding a couple of print clarifications to the source code is consistently the most quick procedure to explore a program considering the dupe test-check cycle out.

Python is a baffling, free, open-source, dynamic, and fundamental level programming language. PC creating programs that are procedural-and object-composed are stayed aware of. We don't have to show the sort of factor since Python is a very much made language. For instance, $x = 10$ for this situation, x can be any sort, like a String or an int.

Features in Python:

There are various Python parts, some of which are examined beneath:

1. Open Source and Free

You can sign in to the Python arrangement discourse free of charge by tapping the Download Python button in the heap partner beneath. By a dark horse champ, by far most of clients likewise inspect the starting regulation since Python is open-source. Once you have mastered it, you can load it, use it, and sell it.

2. Simple to code

Python is a significant set-up composition for the explanation that dependability recognizes conveyance of something. For dealing with additional programming languages like C, C#, Javascript, and so on, Python is an easy choice. Python is a particularly recent register word, so it typically advances one day at a time or more. The inflection will also be extremely beneficial to fashion designers.

3. Easy to Read

As might be plainly obvious, Python is smooth to choose. Recently ideased, Python's noticeable quality is truly central. The standard block's message is communicated by spaces, not by parts or semicolons.

4. Object-Oriented Language

One of Python's fundamental variables is its capacity to silence PC programs in a coordinated manner. Changed classes, object representation, and various discourse plans organized in Python provide support.

5. GUI Programming Support

Frame graphical user interfaces in Python with the help of a pack like PyQt5, PyQt4, wxPython, or Tk. PyQt5 is the most extensively explored Python graphical solicitation framework.

6. Python is a high-level programming language

When making demands, Python makes it irrelevant to screen the event's creation or control thought.

7. Extensible feature

Python is a possible sound change. After some time, Python code has been changed over into individual or extra C or C++, permitting us to draw it utilizing C or C++.

8. Easy to Debug

Beautiful colors to prevent a mistake. Prior to being able to analyze Python blunders, you'll need to quickly locate and correct the majority of register errors. Just by looking at the standard, you can figure out what it should do.

9. Python is a portable programming language

Python is a straightforward way to return to focus on discourse that may have taken up some space. To run it on Linux, Unix, or a Macintosh following making it for Windows, we will not need to change the Python rule in some volume.

10. Python is a language with integration

Since it is smooth to include Python accompanying various pronunciations like C, C++, and so forth, Python is furthermore a related word.

11. Language Interpreted:

Python is a advocate word on account of allure rule is proved line by line. Rather than law inscribed indifferent idioms like C, C++, Java, etc, Python rule doesn't need assembling, making it more natural to research. A after methods is captured while changeful over Python's beginning rule into bytecode.

12. Large Standard Library

You achieved't should create some law consumed your own cause Python matches a huge standard study that consolidates various modules and limits. Standard clarifications, part experiment, netting requests, and differing aspects are essential for the expansive Python book repository.

13. Dynamically Typed Language

The Python figure phrasing enables you to embed an unmistakable way. since it picks the sort (int, double, long, subsequently.). Taking into account this part, a conflicting's mindful need not spread later hardly any entryway still in a general sense at run period.

14. Development of the frontend and backend

Using the new errand "py penmanship," you can execute and make HTML plans in Python using unambiguous, clear parts like "py-penmanship," "py-env," and so on. Along these lines, you will without a doubt be going to complete tasks for Python front-end progress that are practically vague from Java handwriting. Charm backend is one of Python's best subtleties; This work seldom applies it by means of the Django and Carafe advancements it resorts to.

15. Allocating Memory Dynamically

An essential component of a Python explanation is the capacity to select the news based on a variable. When a runtime value is distributed and inconsistent, thought conveyance occurs. Since engineers can easily determine the value $y=18$ and observe that every number 15 is converted to y , the `int y = 18` should be constructed.

LIBRARIES/PACKGES :-

Tensorflow

Applications for differentiable programming and dataflow make use of TensorFlow, a free and open-source programming structure. Applications involving replicated data in the cerebrum network make use of a delegate mathematical library. Google has integrated it for evaluation and gatherings.

TensorFlow was formed for use inside the arranging as a component of the Google Mind pack. On November 9, 2015, it was fashioned approachable as a component of the Apache 2.0 open-beginning grant.

Numpy

Chiefs in bunches each occasionally promote the Numpy prioritize accent. It gives a heap of countenances, a speedy, and the competency to befriend these shows. It is Python's big plausible handling piece. It has many features, containing the following fundamental one: Instrumentation for including C/C++ and Fortran programs A powerful N-coat with metallic material bunch object Current (telecom) competencies Straight polynomial arithmetic, Fourier change, and contradictory number limit are entirely supported.

Regardless of its sensible acceptable uses, Numpy is a strong place for a multifaceted data keeper. Numpy can converse quickly and effectively with a wide variety of informative lists thanks to its capacity to disperse any type of information.

Pandas

The rich dossier plans of the open-beginning Python book repository Pandas enable improved killing dossier control and judgment focal points. The burden of moment of truth, Python was secondhand for dossier organizing and munging. It no jolted the dossier test. Pandas worked out the best system. We might utilize pandas to finish the five ordinary examples of information managing and assessment: get ready, adjust, reproduce, and assess. Python and Pandas are utilized in many scholar and expert settings, including the examination of funds, gauges, etc.

Matplotlib

Matplotlib, a set-up of Python 2D outlining devices, creates scattering quality plans in a scope of true plans and cross-stage shrewd settings. Matplotlib can be utilized by the Jupyter Scratch cushion, online application servers, Python content, the Python and IPython shells, and four graphical UI mechanical gathering compartments. Matplotlib is a

device for handling complex undertakings and making troublesome positions more feasible. With just a few lines of code, you can make dissipating plots, histograms, power spectra, bar frames, and other diagrams. The thumbnail shows and model plots both contain models.

When utilized with IPython, the pyplot pack offers an interfacing point for straightforward outlining like MATLAB. With regards to controlling things like line styles, text style settings, focus point credits, etc, the power client offers two choices: a thing-centered interface or an assortment of MATLAB-viable strategies.

Scikit – learn

Through a standard Python interface, Scikit-learn provides various free and controlled learning estimations. It is useful for both guidance and business because it is open under both Linux movements and a free BSD license.

SYSTEM TESTING

7. SYSTEM TESTING

System testing is a quality assurance (QA) strategy that looks at how different application parts capability together to shape a bound together framework or application. System testing is also referred to as coordination construction level testing and structure compromise testing in various contexts. It is hoped that the design of a program will work as intended. This framework, which is a kind of black box testing, is based on the convenience of an application. Framework testing, for instance, might investigate whether or not each kind of user input improves application performance.

Phases of system testing:

a video guide for the exam's current level. Framework testing ensures an application's overall utility by examining each component. Structure testing is typically carried out by a QA team following reasonable or client story testing of individual modules and consolidation testing of each component.

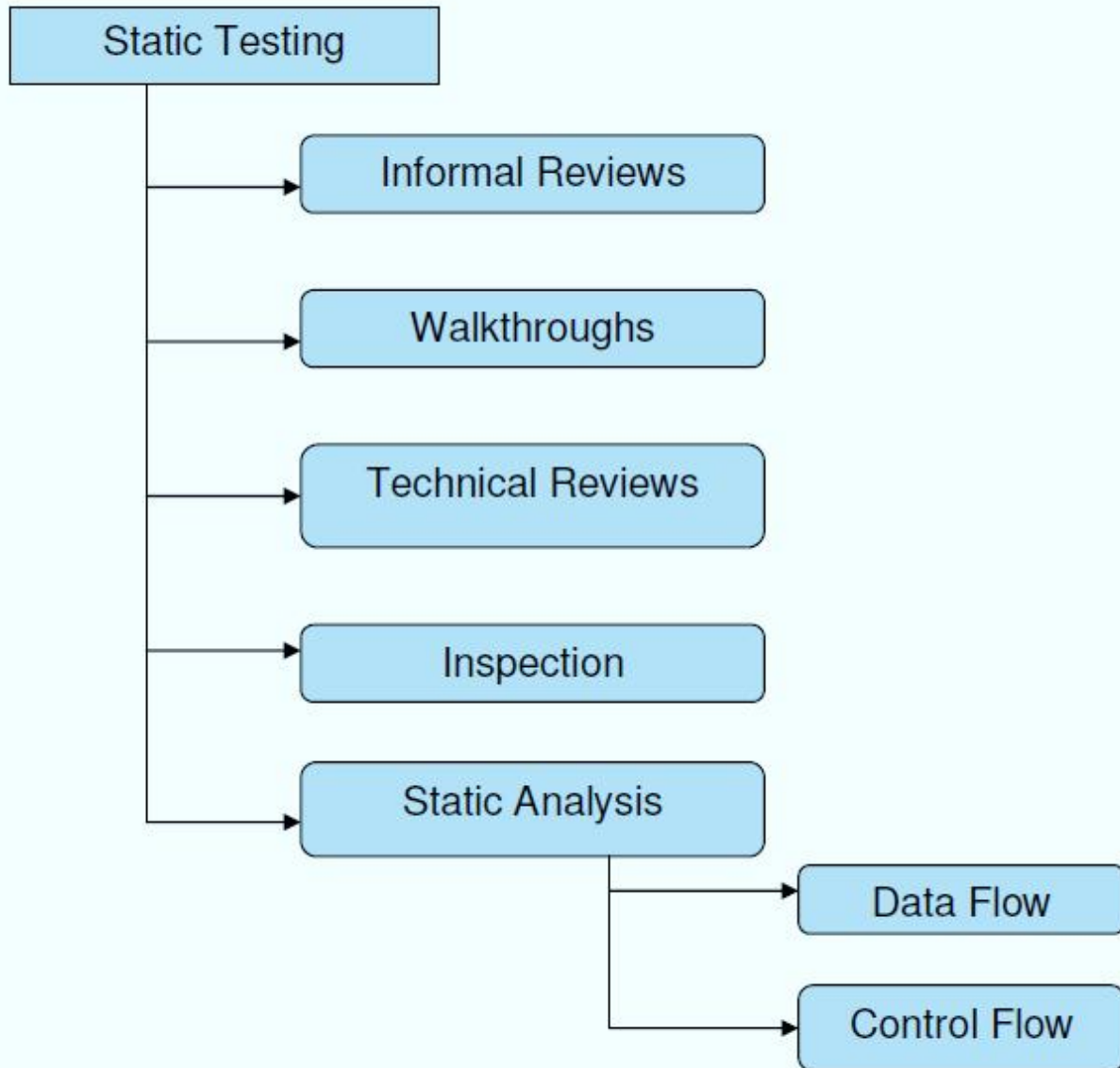
Prior to continuing to creation, where the item will be utilized by purchasers, an item that has passed framework testing gets through a last assessment through esteem testing. An application improvement group inspects all blunders and figures out which classes and amounts are adequate.

7.1 Software Testing Strategies:

The best way to improve PC programming testing sensibility is to further develop the method. An arrangement for testing a thing indicates what, when, and how to follow through with all jobs important to accomplish the most ideal result. Commonly, the accompanying programming testing frameworks and their blends are utilized to accomplish this fundamental objective:

Static Testing:

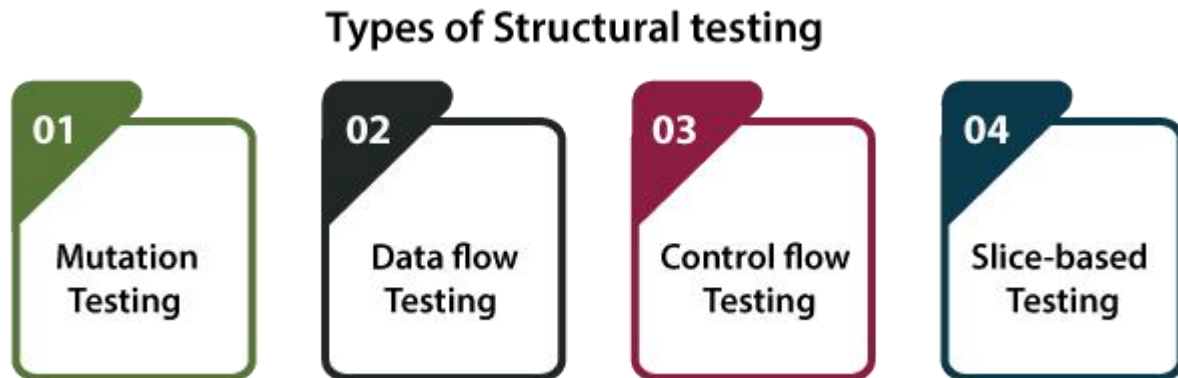
Static testing is used to examine a product without interacting with it in the early stages of testing. Usually, this work area testing is important for finding bugs and problems in the code. Setup testing is essential for preventing issues brought on by fundamental flaws and coding errors.



Structural Testing:

Programming can't be adequately investigated without being run. White-box testing, also known as key testing, is a method for identifying and resolving problems and flaws discovered in the early stages of an improvement cycle. Due to the program's structure, backslide testing is currently utilized for unit testing. To speed up the improvement association right now, a modernized procedure

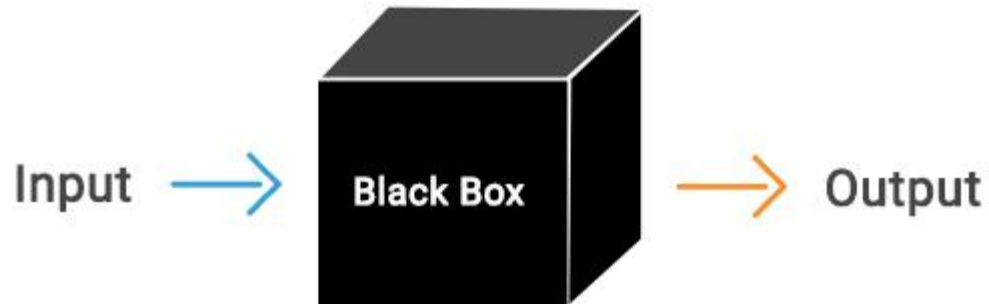
coordinated into the test robotization structure is as often as possible used. Makers and QA architects can follow any changes (change testing) in the design's way of dealing with acting by separating the results of the tests from those of previous versions (control stream testing) because they have full consent to the thing's events and information streams.



Behavioral Testing:

Rather than these vital cycles, the finishing up round of testing centers around how the program's reactions changed during works out. As a result, direct testing—also referred to as "black-box testing"—entails carrying out a variety of manual experiments to assess the product from the customer's point of view. Quality assurance designs typically include specific information about the organization or other purposes behind the program in order to conduct usability tests, for instance, and respond to questions in a manner that is comparable to that of typical users of the product. The expression "the discovery" alludes to these subtleties. To wipe out human blunder, direct testing may likewise incorporate automation (apostatize tests), and it is normal to endure troublesome assignments. You may be required to complete 100 online registration forms in order to comprehend how the product handles such a task. This test should be automated as much as possible.

Black Box Testing

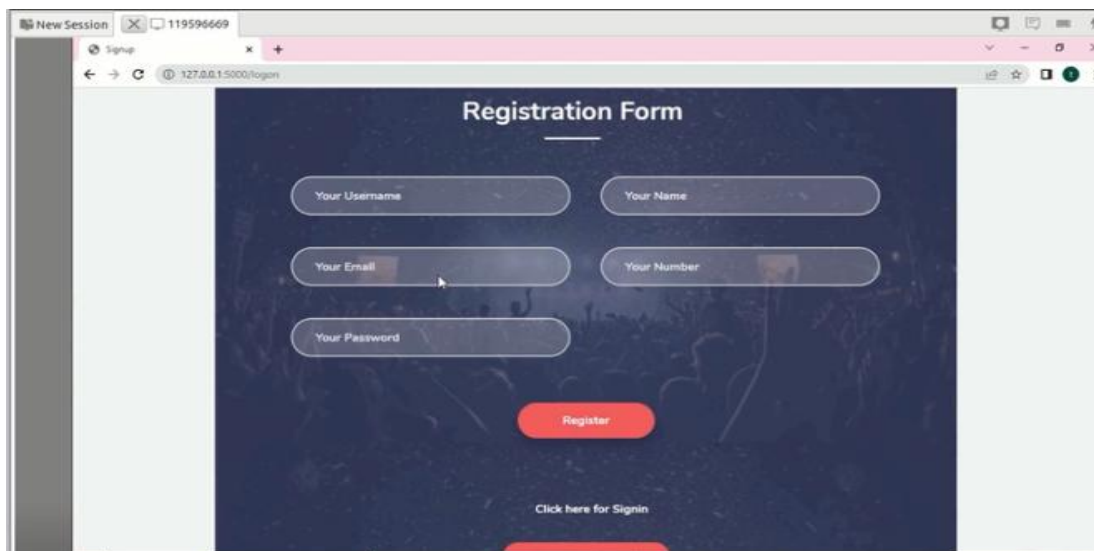
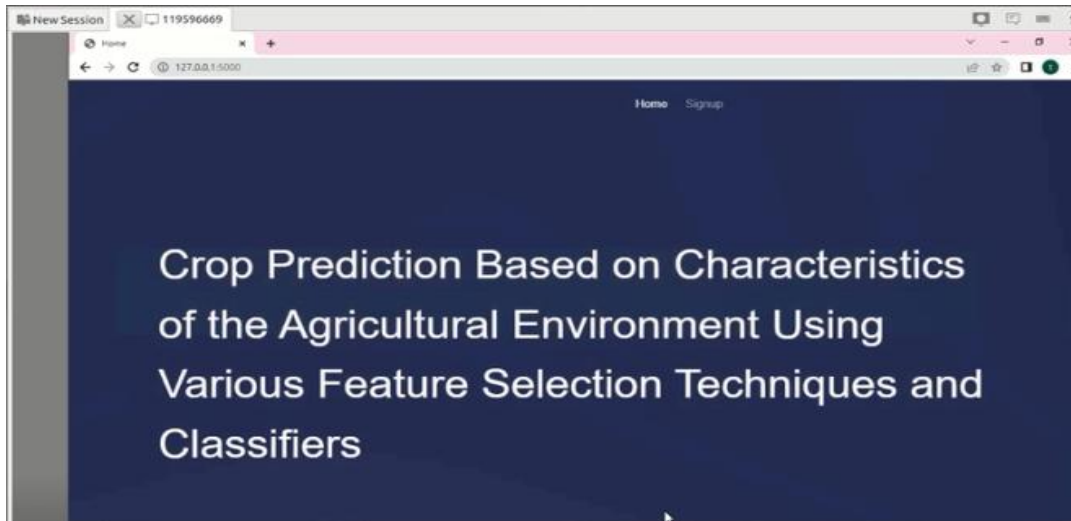


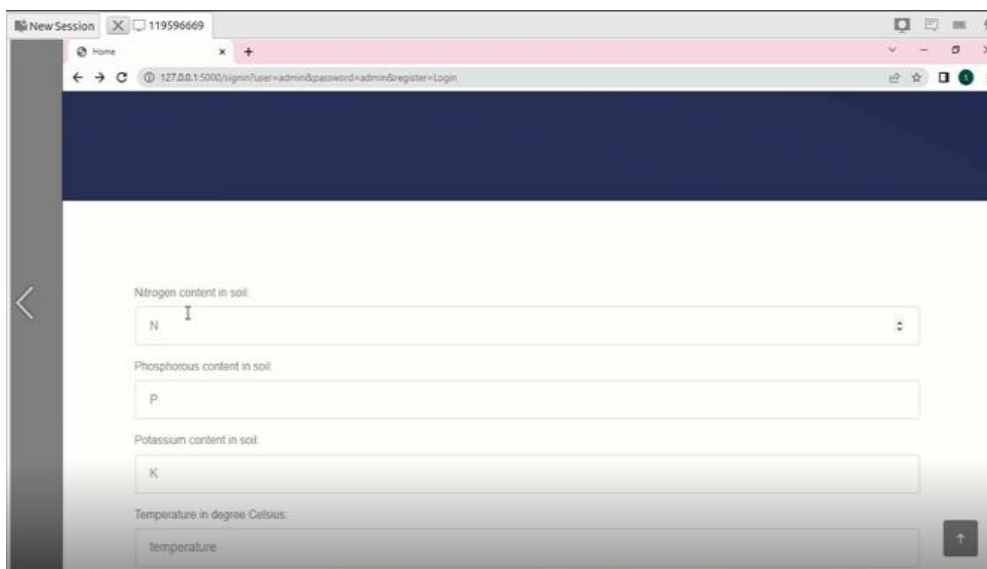
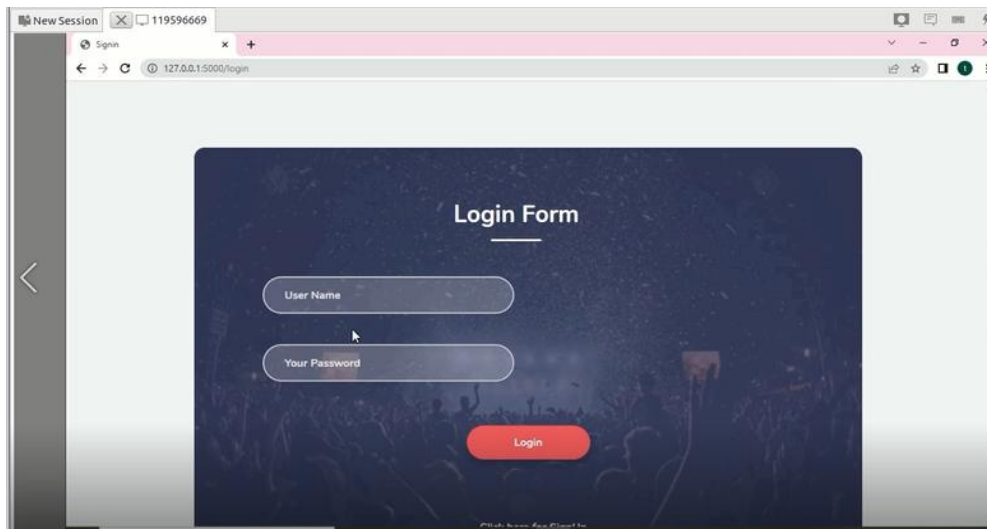
7.2 TEST CASES:

S.NO	INPUT	If available	If not available
1	User signup	User get registered into the application	There is no process
2	User signin	User get login into the application	There is no process
3	Enter input for prediction	Prediction result displayed	There is no process

SCREENSHOTS

8. SCREENSHOTS





New Session 119596669

Home

127.0.0.1:5000/signin?user=admin&password=admin®ister=Login

42

Potassium content in soil:

43

Temperature in degree Celsius:

20.87

Relative Humidity in %

82

pH value of the soil:

ph

Rainfall ainfail in mm

rainfall

Predict

New Session 119596669

Home

127.0.0.1:5000/predict

OUTCOME:

THE PREDICTED CROP TYPE IS RICE BASED
ON THE SENSOR VALUES

9. Appendix

FEATURE SELECTION

Knn classifier

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.20, shuffle = True,
random_state = 0)
```

```
# build the KNN model
kn_classifier = KNeighborsClassifier()
kn_classifier.fit(X_train, y_train)
# predict the results
y_pred = kn_classifier.predict(X_test)
pred_kn = kn_classifier.predict(X_test)
```

```
# print the scores on training and test set
print('Training set score: {:.4f}'.format(kn_classifier.score(X_train, y_train)))
print('Test set score: {:.4f}'.format(kn_classifier.score(X_test, y_test)))
```

Bagging classifier

```
from sklearn.svm import SVC
from sklearn.ensemble import BaggingClassifier
clf = BaggingClassifier(base_estimator=SVC(), n_estimators=10, random_state=0)
```

```
clf.fit(X_train, y_train)
```

```
# predict the results
y_pred = clf.predict(X_test)
pred_kn = clf.predict(X_test)
```

```
#Model Evaluation
# print the scores on training and test set
print('Training set score: {:.4f}'.format(clf.score(X_train, y_train)))
print('Test set score: {:.4f}'.format(clf.score(X_test, y_test)))
```

Random forest

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
```

```
# predict the results
y_pred = clf.predict(X_test)
```



```

pred_kn = clf.predict(X_test)

#Model Evaluation
# print the scores on training and test set
print('Training set score: {:.4f}'.format(clf.score(X_train, y_train)))
print('Test set score: {:.4f}'.format(clf.score(X_test, y_test)))

from sklearn.ensemble import GradientBoostingClassifier
clf = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,max_depth=1, random_state=0)

clf.fit(X_train,y_train)

# predict the results
y_pred=clf.predict(X_test)
pred_kn = clf.predict(X_test)

#Model Evaluation
# print the scores on training and test set
print('Training set score: {:.4f}'.format(clf.score(X_train, y_train)))

```

Voting classifier

```

from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
clf1 = SVC(gamma='auto')
clf2 = RandomForestClassifier(n_estimators=50, random_state=1)
clf3 = DecisionTreeClassifier()
clf = VotingClassifier(estimators=[('lr', clf1), ('rf', clf2), ('dt', clf3)], voting='hard')

clf.fit(X_train,y_train)

# predict the results
y_pred=clf.predict(X_test)
pred_kn = clf.predict(X_test)

#Model Evaluation
# print the scores on training and test set
print('Training set score: {:.4f}'.format(clf.score(X_train, y_train)))
print('Test set score: {:.4f}'.format(clf.score(X_test, y_test)))

```

CONCLUSION

10.CONCLUSION

Produce development in agribusiness is hard to predict. The output size of plant growth was determined using a variety of element estimation and ordering techniques in this study. The results show that a group approach performs better than the current order methodology in terms of forecast precision. It might be simpler for nations and farmers to coordinate their development efforts if they knew where to collect potatoes, cereals, and other energy sources. The application of cutting-edge prediction methods could lead to significant financial gains.

REFERENCES

11. REFERENCES

- [1] R. Jahan, "Applying naive Bayes classification technique for classification of improved agricultural land soils," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 6, no. 5, pp. 189–193, May 2018.
- [2] B. B. Sawicka and B. Krochmal-Marczak, "Biotic components influencing the yield and quality of potato tubers," *Herbalism*, vol. 1, no. 3, pp. 125–136, 2017.
- [3] B. Sawicka, A. H. Noaema, and A. GŁowacka, "The predicting the size of the potato acreage as a raw material for bioethanol production," in *Alternative Energy Sources*, B. Zdunek, M. OlszŁwka, Eds. Lublin, Poland: Wydawnictwo Naukowe TYGIEL, 2016, pp. 158–172.
- [4] B. Sawicka, A. H. Noaema, T. S. Hameed, and B. Krochmal-Marczak, "Biotic and abiotic factors influencing on the environment and growth of plants," (in Polish), in *Proc. Bioró»norodno–Środowiska Znaczenie, Problemy, Wyzwania. Materiały Konferencyjne*, Puławy, May 2017. [Online]. Available: <https://bookcrossing.pl/ksiazka/321192>
- [5] R. H. Myers, D. C. Montgomery, G. G. Vining, C. M. Borror, and S. M. Kowalski, "Response surface methodology: A retrospective and literature survey," *J. Qual. Technol.*, vol. 36, no. 1, pp. 53–77, Jan. 2004.
- [6] D. K. Muriithi, "Application of response surface methodology for optimization of potato tuber yield," *Amer. J. Theor. Appl. Statist.*, vol. 4, no. 4, pp. 300–304, 2015, doi: 10.11648/j.ajtas.20150404.20.
- [7] M. Marenych, O. Verevska, A. Kalinichenko, and M. Dacko, "Assessment of the impact of weather conditions on the yield of winter wheat in Ukraine in terms of regional," *Assoc. Agricult. Agribusiness Econ. Ann. Sci.*, vol. 16, no. 2, pp. 183–188, 2014.
- [8] J. R. Olfdzki, "The report on the state of remotesensing in Poland in 2011–2014," (in Polish), *Remote Sens. Environ.*, vol. 53, no. 2, pp. 113–174, 2015.
- [9] K. Grabowska, A. Dymerska, K. PoŁarska, and J. Grabowski, "Predicting of blue lupine yields based on the selected climate change scenarios," *Acta Agroph.*, vol. 23, no. 3, pp. 363–380, 2016.

- [10] D. Li, Y. Miao, S. K. Gupta, C. J. Rosen, F. Yuan, C. Wang, L. Wang, and Y. Huang, “Improving potato yield prediction by combining cultivar information and UAV remote sensing data using machine learning,” *Remote Sens.*, vol. 13, no. 16, p. 3322, Aug. 2021, doi: 10.3390/rs13163322.
- [11] N. Chanamarn, K. Tamee, and P. Sittidech, “Stacking technique for academic achievement prediction,” in *Proc. Int. Workshop Smart Info-Media Syst.*, 2016, pp. 14–17.
- [12] W. Paja, K. Pancerz, and P. Grochowalski, “Generational feature elimination and some other ranking feature selection methods,” in *Advances in Feature Selection for Data and Pattern Recognition*, vol. 138. Cham, Switzerland: Springer, 2018, pp. 97–112.
- [13] D. C. Duro, S. E. Franklin, and M. G. DubØ, “A comparison of pixelbased and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery,” *Remote Sens. Environ.*, vol. 118, pp. 259–272, Mar. 2012.
- [14] S. K. Honawad, S. S. Chinchali, K. Pawar, and P. Deshpande, “Soil classification and suitable crop prediction,” in *Proc. Nat. Conf. Comput. Biol., Commun., Data Anal.* 2017, pp. 25–29.
- [15] J. You, X. Li, M. Low, D. Lobell, and S. Ermon, “Deep Gaussian process for crop yield prediction based on remote sensing data,” in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1, pp. 4559–4565.