

CROP YIELD PREDICTION USING RANDOM FOREST ALGORITHM

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

By
Donthu Sai Krishna(38110139)



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

SATHYABAMA

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

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119
MARCH - 2020**



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **DONTHU SAI KRISHNA (38110139)** who carried out the project entitled “**CROP YIELD PREDICTION USING RANDOM FOREST ALGORITHM**” under my supervision from **June 2019** to **March 2020**.

Internal Guide
Dr. L. K JOSHILA GRACE

Head of the Department
Dr. L. Lakshmanan M.E., Ph.D.,

Submitted for Viva-voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **DONTHU SAI KRISHNA (389110139)** hereby declare that the Project Report entitled “**CROP YIELD PREDICTION USING RANDOM FOREST ALGORITHM**” done by me under the guidance of Mrs. Divya S (Internal) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in **Computer Science and Engineering**.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

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

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. L. K Joshila Grace.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

Most agricultural crops have been badly affected by the effect of global climate change in India. In terms of their output over the past 20 years. It will allow policy makers and farmers to take effective marketing and storage steps to predict crop yields earlier in their harvest. This project will allow farmers to capture the yield of their crops before cultivation in the field of agriculture and thus help them make the necessary decisions. Implementation of such a method with a web-based graphic software that is simple to use and the machine learning algorithm can then be distributed. The results obtained are granted access to the farmer. And yet there are various methods or protocols for such very data analytics in crop yield prediction, and we are able to predict agricultural productivity with guidance of all those algorithms. It utilizes a Random Forest Algorithm. By researching such problems and issues such as weather, temperature, humidity, rainfall, humidity, there are no adequate solutions and inventions to resolve the situation we face. In countries like India, even in the agricultural sector, as there are many types of increasing economic growth. In addition, the processing is useful for forecasting the production of crop yields.

TABLE OF CONTENTS

CHAPTER No	TITLE	PAGE
	LIST OF FIGURES	6
	LIST OF ABBREVIATIONS	7
1	INTRODUCTION	8
1.1	About the Project	8
2	SYSTEM ANALYSIS	9
2.1	Existing system	9
2.2	Proposed system	9
3	REQUIREMENTS SPECIFICATION	10
3.1	Introduction	10
3.2	Hardware and Software specification	11
3.3	Technologies Used	11
4.1	Design and Implementation Constraints	29
4.2	Other Nonfunctional Requirements	30
5	Architecture Diagram	31
6	Modules	37

6.1	Modules	37
6.2	Module explanation	37
7	CODING AND TESTING	38
7.1	Coding	38
7.2	Coding standards	38
7.3	Test procedure	40
7.4	Test data and output	41
	SOURCE CODE	48
8	SCREENSHOTS	60
	REFERENCES	69

LIST OF FIGURES

5.1 Sequence Diagram

5.2 Use Case Diagram

5.3 Activity Diagram

5.4 Collaboration Diagram:

LIST OF SCREENSHOTS

- Filling missing values data set using Pandas & NumPy
- Graph for predicting future crop
- Open Firebase Console
- Created android app for Crop prediction
- Crop Prediction
- Enter the values of attributes
- Input given by the user goes to the trained dataset using firebase
- Predicted the crop with price
- Successfully predicted crop and price

CHAPTER 1

INTRODUCTION

Aim:

Predict the crop type and price of the crop using machine learning methodology with accurate results.

Synopsis:

India is the land of agriculture and it is the major source of economy. 70% of Indian population directly relies on agriculture. The common problem existing among the young Indian farmers is to choose the right crop based on the location, humidity, temperature, rainfall. Due to this, they face a serious setback in productivity. Our work proposes to help farmers determine the predict crop type and price by doing analysis on its various parameters and to suggest crops based on the results obtained. The system uses the Classification algorithm of Random Forest to improve the efficiency of Crop Recommendation System. The system maps the location, temperature, humidity, pH value and rainfall to predict the list of suitable crops for the soil and it also provides cost of the crop. Hence it leaves upon the user to decide on the crop to be sown. Thus, the system helps to provide knowledge to the dilettante farmers.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Now a day's, dilettante farmers are hard to understand the cultivation process, crop type, climate change, etc. Farming is that the spine for every nation's economy. Future agriculture depends on dilettante formers. But, the new farmers have low level knowledge in this field. So, Machine learning help to solve this type of problems. In Existing system they provide soil type and crop using Random Forest algorithm. But everyone can able to find a soil type easily. So, we need to predict the crop type and predict the crop price based on Machine learning technology.

2.2 PROPOSED SYSTEM

The system prepared predict major crops yield in a particular district in Tamil Nadu. The client on their first login has to register themselves on the application on android phone .Once the user logs into the system he gets all the access for predicting crop yield and using the input such as location, temperature, pH value, rainfall and humidity depends on their forming land environment. After submitting the inputs, it's redirect into Firebase. The firebase is an intermediate between user input and trained data set. The input goes to the trained data, where it processes random forest algorithm to predict crop and price. After the prediction, the predicted value passes to the fire base. That firebase gives the predict value to the user on android application.

CHAPTER 3

REQUIREMENT SPECIFICATIONS

3.1 INTRODUCTION

Agriculture is the backbone of India. As we known, food stands first in the basic need of survival; agriculture sector needs to be given the highest preference in development. Indian agriculture sector accounts for 18 percent of Indian agriculture gross domestic product (GDP) and provides employment to 50% of the country's workforce. The main reason for considered Agriculture sector is because it plays an important role in developing the country's economy. The proposed System uses the Crop Selection as the area of research, since it the first and most important step in the process of agricultural development and the success of this step guarantees the result of production. Agriculture development provides assistance to the crop producers with the help of various agricultural resources. As a result, it provides high productivity with low consumption of resources.

Crop production may also be an advanced development that is affected by the input parameters of soil and condition. Process parameters of agriculture vary from area to area and producer to producer. It may also be a discouraging challenge to collect such information in an even larger room. The Indian Meteoric Department, however, tabulates the environmental condition information collected in the Republic of India at each 1sq.m space in various components of the district. The huge sets of such information are often used to predict their effect on that district or place's main crops. Within the agriculture or related sciences field, there are entirely different foretelling methodologies developed and evaluated by researchers around the globe. Some studies of that type are: In alternative countries, agricultural researchers have shown that attempts to increase crop yield by pro-pesticide state maximization have been carried out. Driven strategies for dangerously high chemical use have been introduced. The association between chemical use and crop yield has been stated in these studies[1]. Agriculture is a partner trade sector that has gained significantly in recent years from the growth of detector technology, information science, and machine learning (ML) techniques.

3.2 HARDWARE AND SOFTWARE SPECIFICATION

3.2.1 HARDWARE REQUIREMENTS

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

3.2.2 SOFTWARE REQUIREMENTS

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

3.3 TECHNOLOGIES USED

- Python

3.3.1 Introduction to Python

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

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

It is used for:

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

What can Python do?

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

Why Python?

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

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some

features of Python 3 have been backported to Python 2. But in general, they remain not quite compatible.

- Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End Of Life date of January 1, 2020 has been established for Python 2, after which time it will no longer be maintained.
- Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe Monty Python's Flying Circus, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.
- It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

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

Python is Interpreted

- Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.

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

Support Vector Machine (SVM)

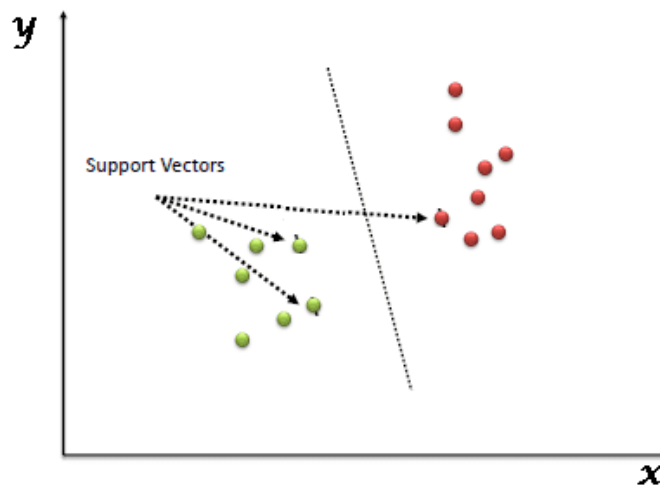
Introduction:

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

Think of machine learning algorithms as an armory packed with axes, sword, blades, bow, dagger etc. You have various tools, but you ought to learn to use them at the right time. As an analogy, think of ‘Regression’ as a sword capable of slicing and dicing data efficiently, but incapable of dealing with highly complex data. On the contrary, ‘Support Vector Machines’ is like a sharp knife – it works on smaller datasets, but on them, it can be much stronger and powerful in building models.

Support Vector Machine

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



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

You can look at definition of support vectors and a few examples of its working here.

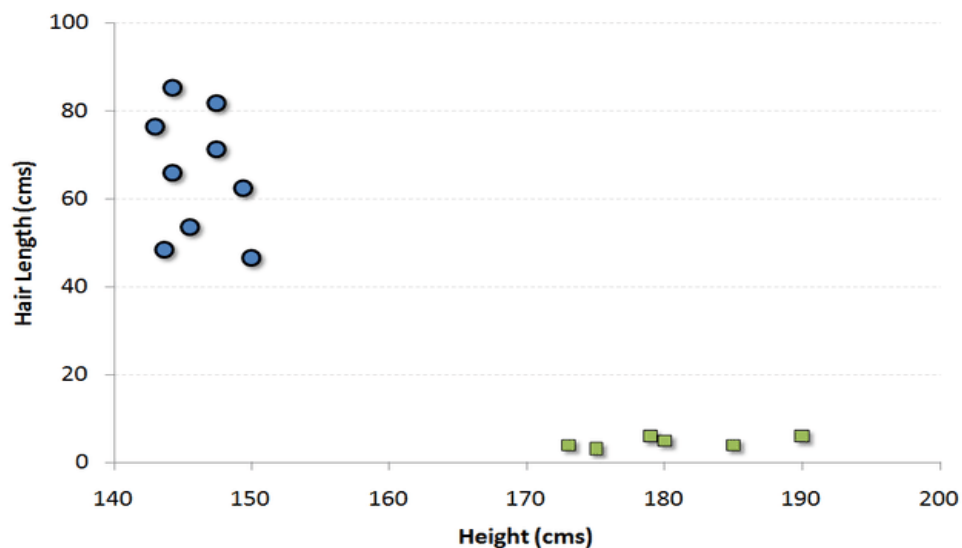
Support Vector Machine – Simplified

Let's talk about how support vector machine works. **This is suitable for readers who do not know much about this algorithm and have a curiosity to learn a new technique.** In

following explore the technique in detail and analyze cases where such techniques are stronger than other techniques.

Classification analysis

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



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

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

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

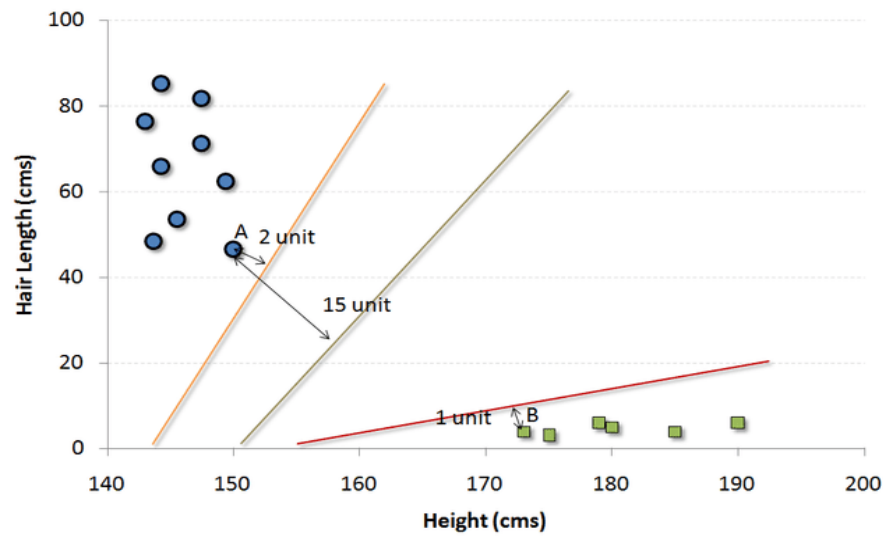
A Support Vector and SVM

Support Vectors are simply the co-ordinates of individual observation. For instance, (45,150) is a support vector which corresponds to a female. Support Vector Machine is a frontier

which best segregates the Male from the Females. In this case, the two classes are well separated from each other; hence it is easier to find a SVM.

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

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

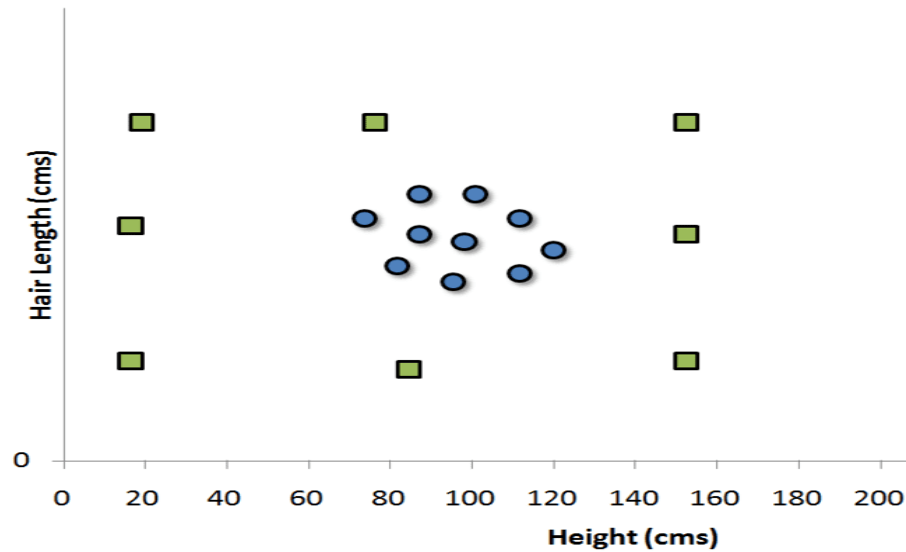


How do we decide which is the best frontier for this particular problem statement?

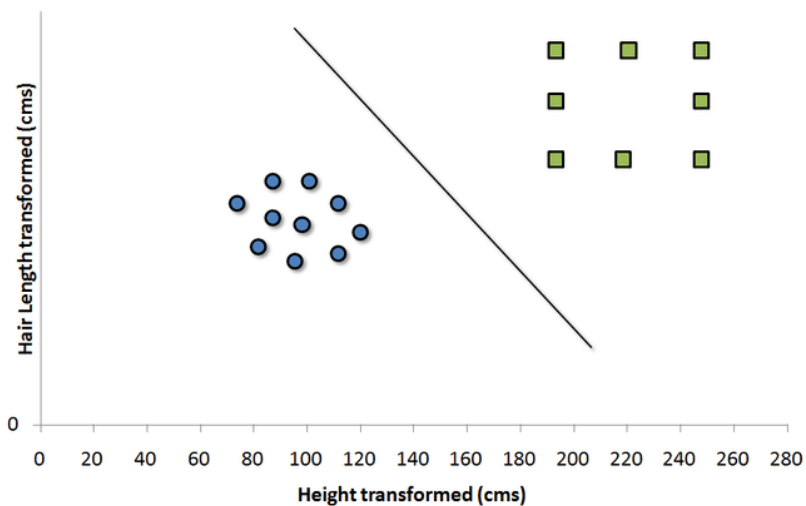
The easiest way to interpret the objective function in a SVM is to find the minimum distance of the frontier from closest support vector (this can belong to any class). For instance, orange frontier is closest to blue circles. And the closest blue circle is 2 units away from the frontier. Once we have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

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

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



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



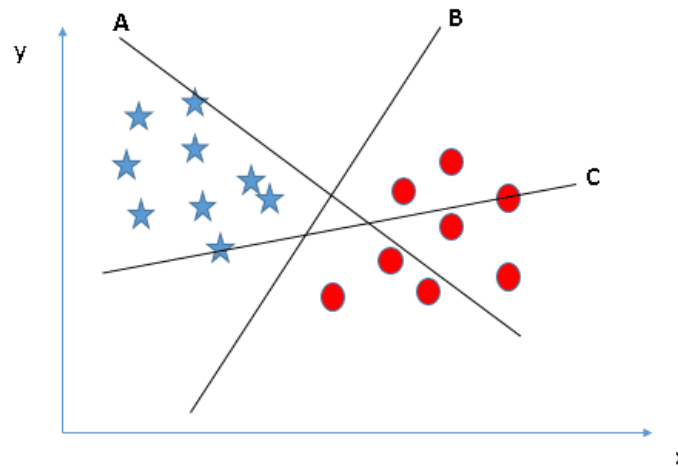
Each of the green squares in original distribution is mapped on a transformed scale. And transformed scale has clearly segregated classes. Many algorithms have been proposed to make these transformations.

How does it work?

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

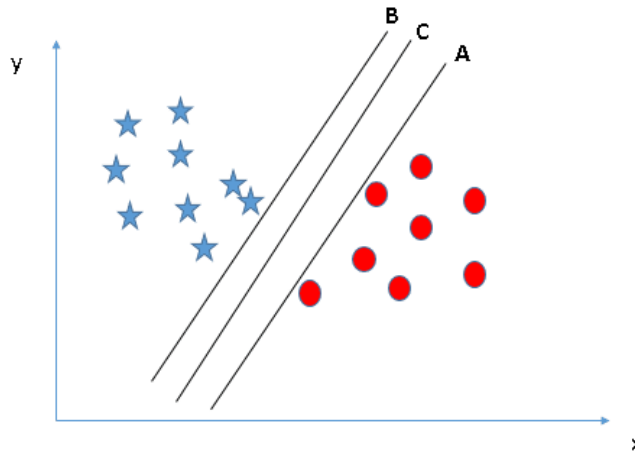
Let’s understand:

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

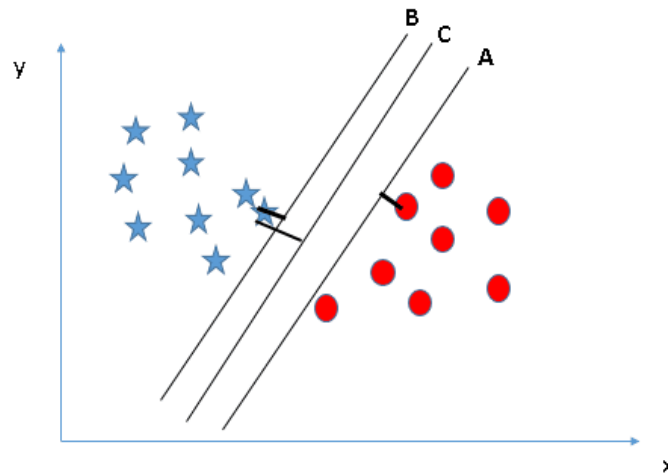


You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.

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

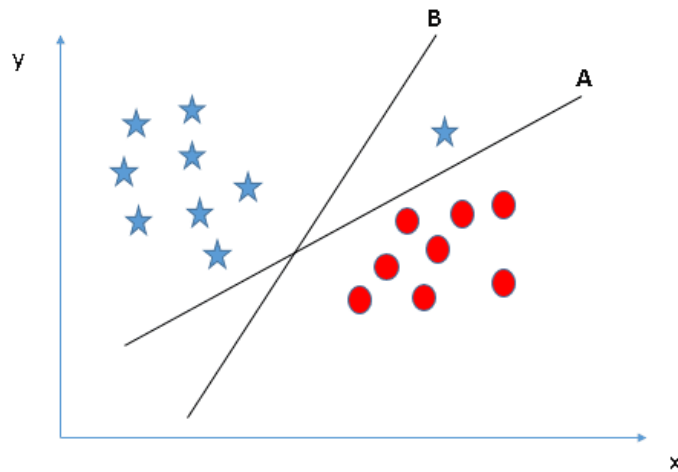


Here, maximizing the distances between nearest data points (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**. Let's look at the below snapshot:



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

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

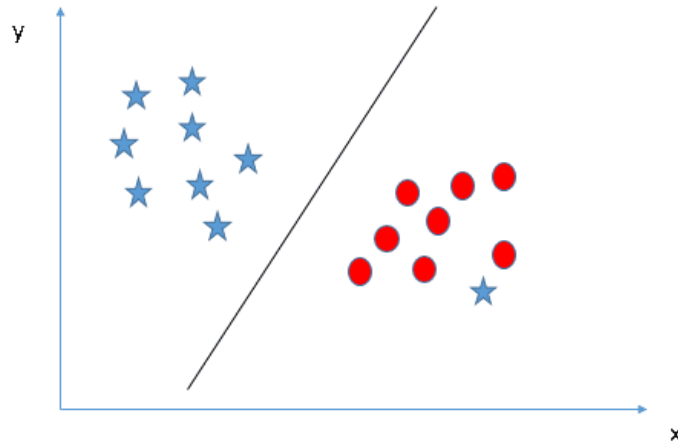


Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A**. But, here is the catch; SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore,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

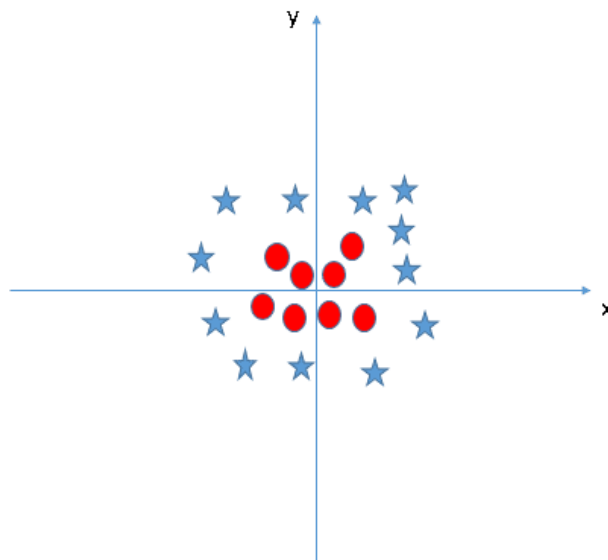
- **Can we classify two classes (Scenario-4)?:** Below, I am unable to segregate the two classes using a straight line, as one of star lies in the territory of other(circle) class as an outlier.



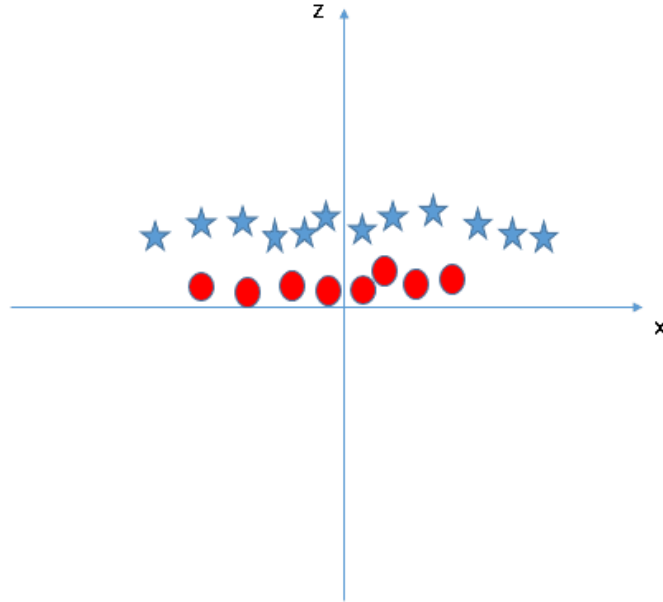
As I have already mentioned, one star at other end is like an outlier for star class. SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin. Hence, we can say, SVM is robust to outliers.



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



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

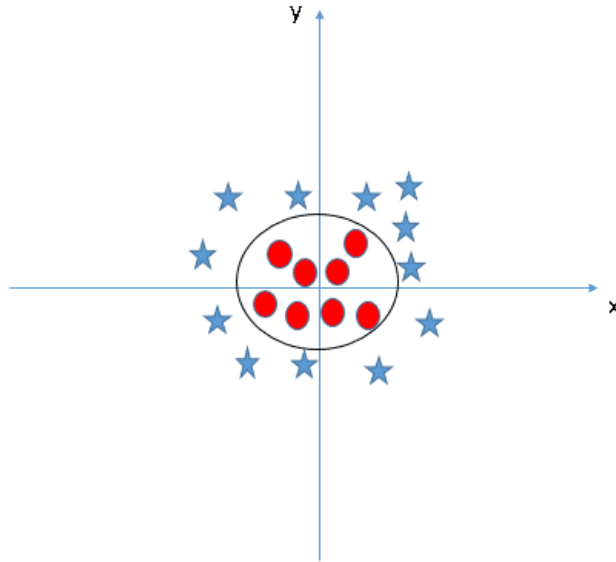


In above plot, points to consider are:

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

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

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



Pros and Cons associated with SVM

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

Random Forest

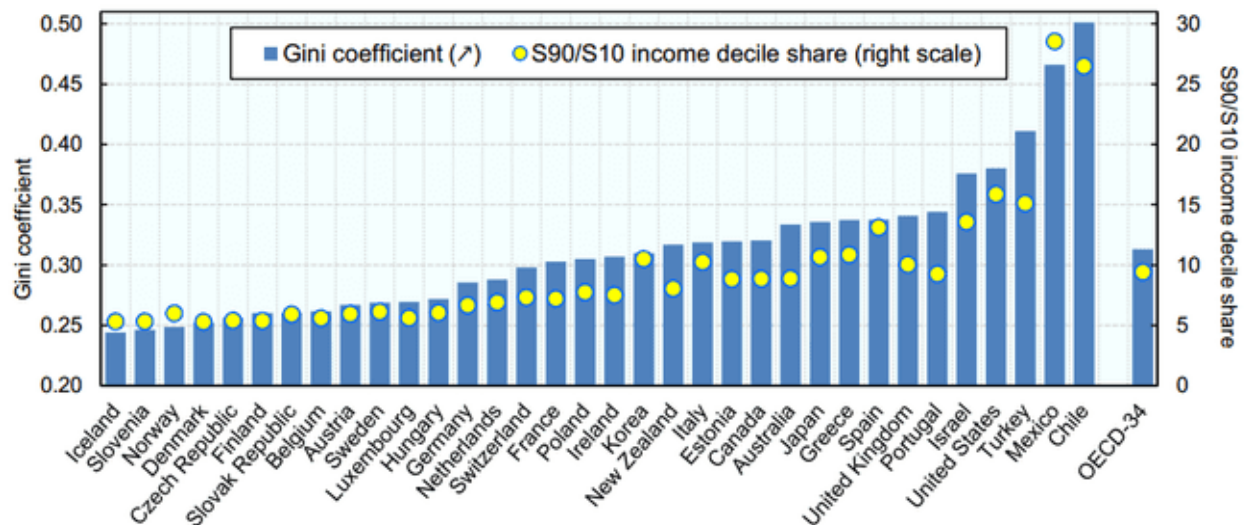
Introduction

With increase in computational power, we can now choose algorithms which perform very intensive calculations. One such algorithm is “Random Forest”, which we will discuss in this article. While the algorithm is very popular in various competitions (e.g. like the ones running on Kaggle), the end output of the model is like a black box and hence should be used judiciously.

Before going any further, here is an example on the importance of choosing the best algorithm.

Random Forest Case Study

Following is a distribution of Annual income Gini Coefficients across different countries:



Mexico has the second highest Gini coefficient and hence has a very high segregation in annual income of rich and poor. Our task is to come up with an accurate predictive algorithm to estimate annual income bracket of each individual in Mexico. The brackets of income are as follows:

1. below \$40,000
2. \$40,000 – 150,000
3. More than \$150,000

Following are the information available for each individual:

1. Age, 2. Gender, 3. Highest educational qualification, 4. Working in Industry, 5. Residence in Metro/Non-metro

We need to come up with an algorithm to give an accurate prediction for an individual who has following traits:

1. Age: 35 years, 2. Gender: Male, 3. Highest Educational Qualification: Diploma holder, 4. Industry: Manufacturing, 5. Residence: Metro

We will only talk about random forest to make this prediction in this article.

The algorithm of Random Forest

Random forest is like bootstrapping algorithm with Decision tree (CART) model. Say, we have 1000 observation in the complete population with 10 variables. Random forest tries to build multiple CART model with different sample and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a CART model. It will repeat the process (say) 10 times and then make a final prediction on each observation. Final prediction is a function of each prediction. This final prediction can simply be the mean of each prediction.

Back to Case study

Disclaimer: The numbers in this article are illustrative

Mexico has a population of 118 MM. Say, the algorithm Random forest picks up 10k observation with only one variable (for simplicity) to build each CART model. In total, we are looking at 5 CART models being built with different variables. In a real life problem, you will have more number of population sample and different combinations of input variables.

Salary bands:

Band 1: Below \$40,000

Band 2: \$40,000 – 150,000

Band 3: More than \$150,000

Following are the outputs of the 5 different CART model.

CART 1: Variable Age

	Salary Band	1	2	3
Age	Below 18	90%	10%	0%
	19-27	85%	14%	1%
	28-40	70%	23%	7%
	40-55	60%	35%	5%
	More than 55	70%	25%	5%

CART 2: Variable Gender

	Salary Band	1	2	3
Gender	Male	70%	27%	3%
	Female	75%	24%	1%

CART 3: Variable Education

	Salary Band	1	2	3
Education	<=High School	85%	10%	5%
	Diploma	80%	14%	6%
	Bachelors	77%	23%	0%
	Master	62%	35%	3%

CART 4: Variable Residence

	Salary Band	1	2	3
Residence	Metro	70%	20%	10%
	Non-Metro	65%	20%	15%

CART 5: Variable Industry

	Salary Band	1	2	3
Industry	Finance	65%	30%	5%
	Manufacturing	60%	35%	5%
	Others	75%	20%	5%

Using these 5 CART models, we need to come up with single set of probability to belong to each of the salary classes. For simplicity, we will just take a mean of probabilities in this case study. Other than simple mean, we also consider vote method to come up with the final prediction. To come up with the final prediction let's locate the following profile in each CART model:

1. Age: 35 years, 2. Gender: Male, 3. Highest Educational Qualification: Diploma holder, 4. Industry: Manufacturing, 5. Residence: Metro

For each of these CART model, following is the distribution across salary bands:

CART	Band	1	2	3
Age	28-40	70%	23%	7%
Gender	Male	70%	27%	3%
Education	Diploma	80%	14%	6%
Industry	Manufacturing	60%	35%	5%
Residence	Metro	70%	20%	10%
Final probability		70%	24%	6%

The final probability is simply the average of the probability in the same salary bands in different CART models. As you can see from this analysis, that there is 70% chance of this individual falling in class 1 (less than \$40,000) and around 24% chance of the individual falling in class 2.

Conclusion

Random forest gives much more accurate predictions when compared to simple CART/CHAID or regression models in many scenarios. These cases generally have high number of predictive variables and huge sample size. This is because it captures the variance of several input variables at the same time and enables high number of observations to participate in the prediction.

CHAPTER 4

4.1 Design and Implementation Constraints

4.1.1 Constraints in Analysis

- ◆ Constraints as Informal Text
- ◆ Constraints as Operational Restrictions
- ◆ Constraints Integrated in Existing Model Concepts
- ◆ Constraints as a Separate Concept
- ◆ Constraints Implied by the Model Structure

4.1.2 Constraints in Design

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

4.1.3 Constraints in Implementation

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

4.2 Other Nonfunctional Requirements

4.2.1 Performance Requirements

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

- embedded browser in charge of the navigation and accessing to the web service;
- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

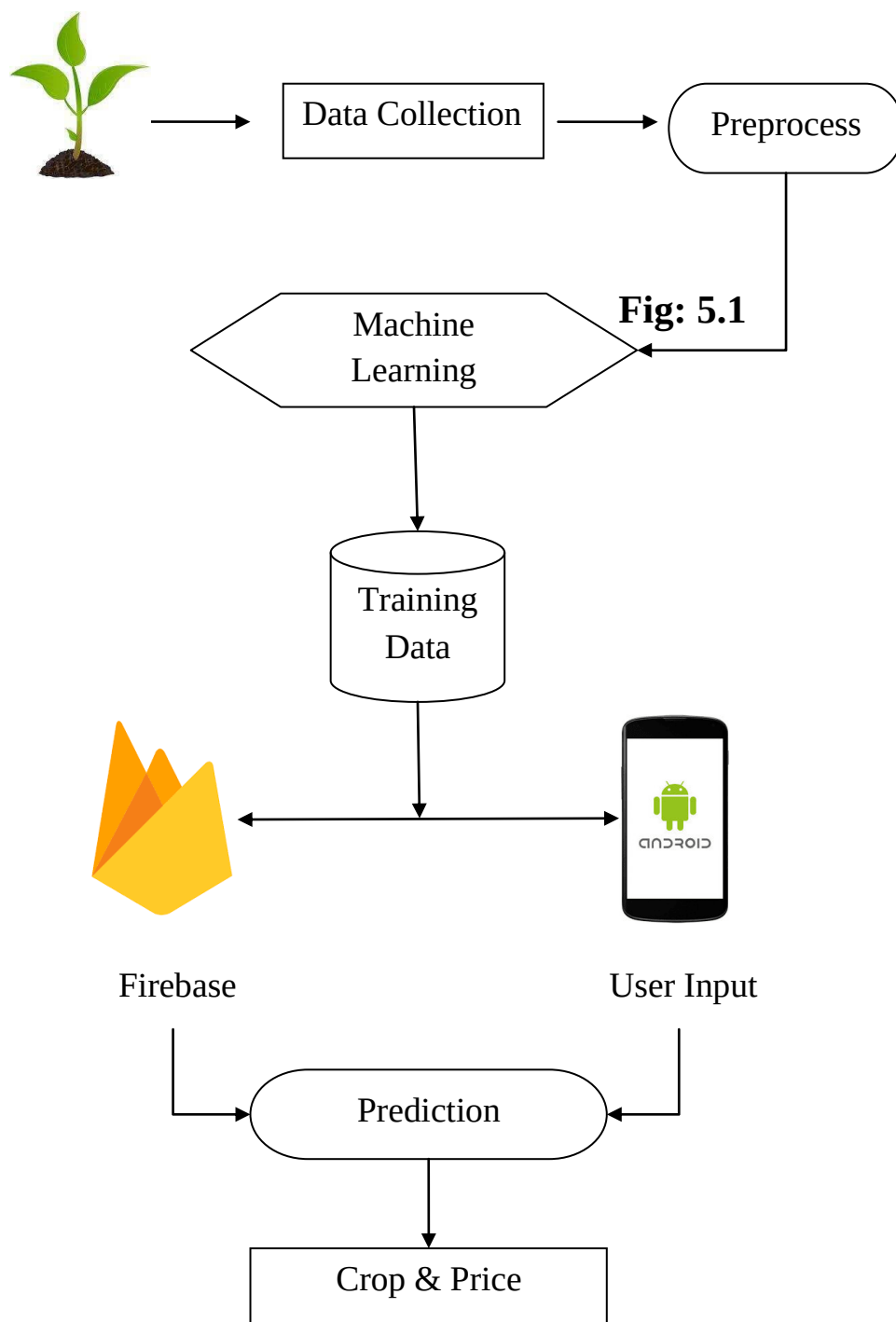
4.2.2 Safety Requirements

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

7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

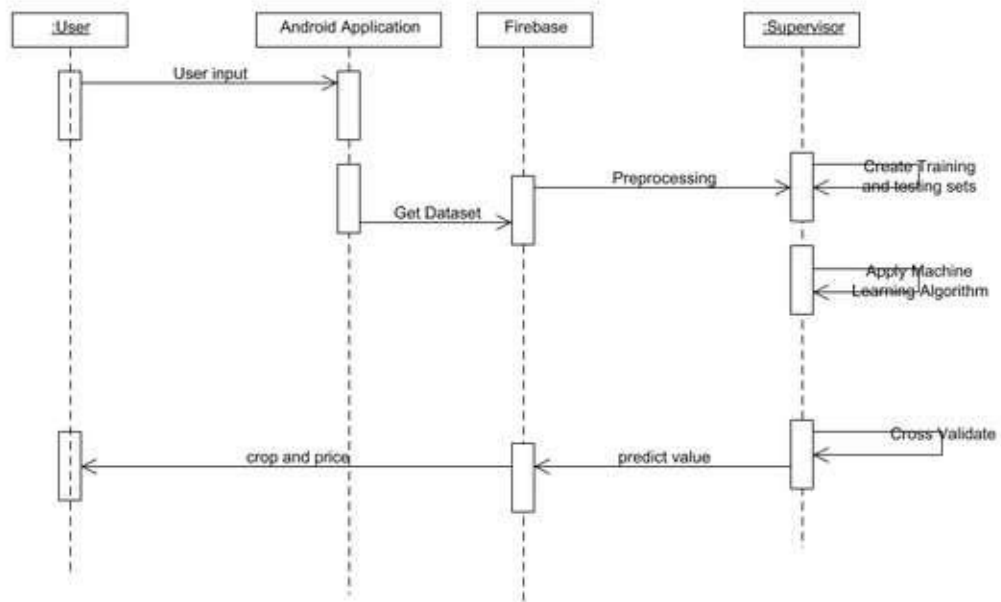
CHAPTER 5

5.1 Architecture Diagram:



5.2 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



5.3 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

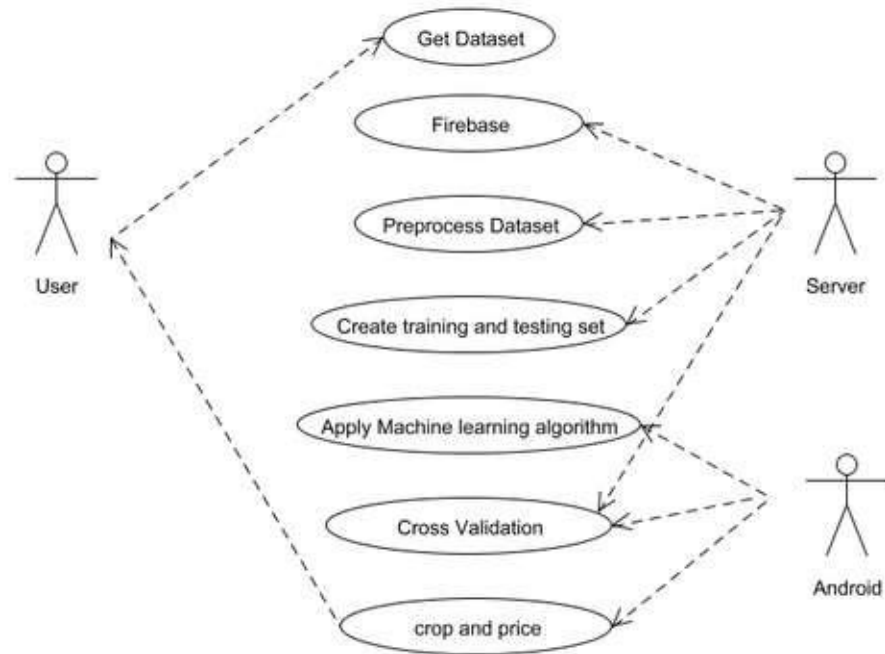
5.3.1. USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

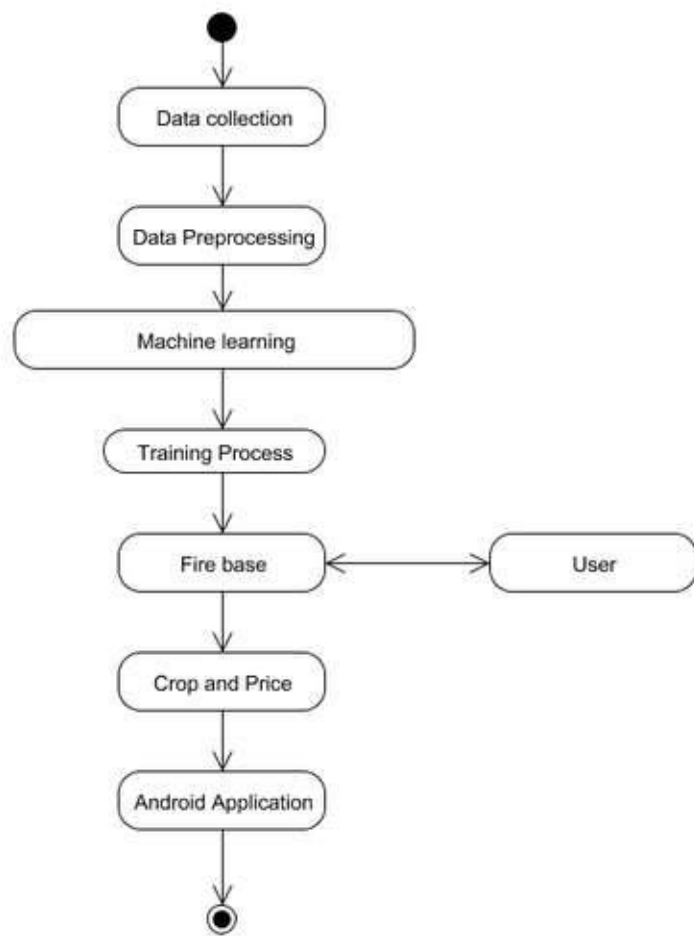


5.4 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

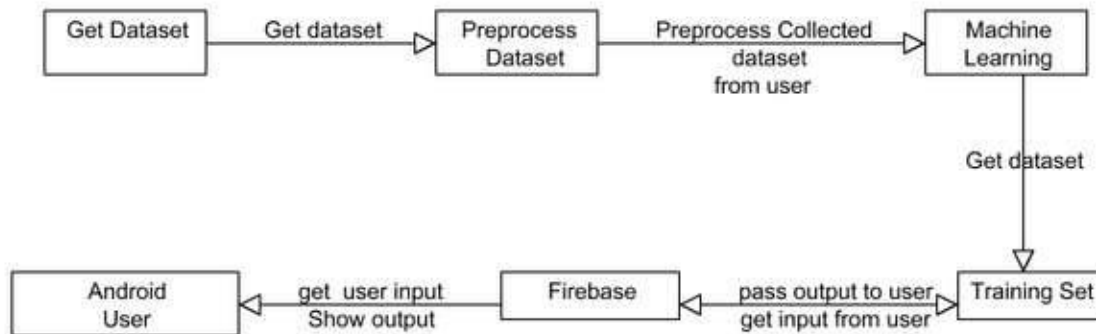
The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.



5.5 Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.



CHAPTER 6

6.1 MODULES

- **Dataset collection**
- **Implementation**
- **Prediction**

6.2 MODULE EXPLANATION:

6.2.1 Dataset Collection:

Our crop prediction project dataset are collected from kaggle.com. Data is pre-processed after collection of various records. The dataset contains a more number of records, where some records are with some missing values. Those missing records have been removed from the dataset and filled the data using python package techniques (Pandas, NumPy).

Implementation:

The classification Algorithms always produces the best results. We are using Random Forest Algorithm to predict the crop and price using ML. On an analysis conducted within various algorithms, the Random Forest was found to provide highest efficiency and precision compared to Decision tree. Because RF contain number of decision tree algorithm, that take the average to improve the predictive accuracy of dataset. Hence the RF algorithm is used in the proposed system to find the suitable crop and crop price.

Prediction:

Preprocessed data are trained and input given by the user goes to the trained dataset using firebase. After prediction the predict value given as a output to android through firebase

CHAPTER 7

CODING AND TESTING

7.1 CODING

Once the design aspect of the system is finalized the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

7.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format

Exception and error handling

7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

7.2.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

7.2.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

7.2.4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.
- ! – Information to the user.

7.3 TEST PROCEDURE

SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

7.4 TEST DATA AND OUTPUT

7.4.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

7.4.2 FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

7.4.3 PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

7.4.4 STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

7.4.5 STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

7.4.6 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all

the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

7.5 TESTING TECHNIQUES / TESTING STRATEGIES

7.5.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for

correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

7.5.1.1 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

7.5.1.2 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to

specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

7.5.2 SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

7.5.2.1 INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly

when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

7.5.2.2 PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

7.5.2.3 SECURITY TESTING:

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

7.5.2.4 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- * The function or performance characteristics confirm to specifications and are accepted.
- * A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

7.5.2.5 USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

Source Code

```
import pandas as pd

import numpy as np

import os


data = pd.read_csv(r"./dataset/Dataset.csv")


data = data.replace('?', np.nan)

print("Crop data. Size={}\nNumber of missing values".format(data.shape))

print(data.isna().sum())

df=data.fillna(data.ph.median())


print("Concatanated dataset. Size={}\nNumber of missing values".format(df.shape))

print(df.isna().sum())

df.to_csv(os.path.join(r"./preprocess_dataset/Dataset.csv'), index=False)

print("Dataset created successfully")


#print(data.isna().sum())
```

```
data.head()

data.tail()

#!/usr/bin/env python

# coding: utf-8


# In[5]:import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

# In[6]:dataset = pd.read_csv(r'./dataset/Dataset.csv')

# In[7]:dataset.head()

# In[4]:type(dataset)

# In[5]:dataset.shape

# In[6]:dataset.info()

# In[7]:dataset.describe().T

# In[8]:dataset.isnull().sum()

# In[11]:from sklearn.preprocessing import LabelEncoder

# In[12]:labelencoder = LabelEncoder()

# In[13]:dataset['class'] = labelencoder.fit_transform(dataset['label'])
```

```

dataset.head(5)

# # feature selection

# In[14]:dataset.corr()

sns.heatmap(dataset.corr(), annot = True)

plt.show()

# In[15]:X = dataset.iloc[:, [1,2,3,4]].values

Y = dataset.iloc[:, [5]].values

# In[16]:from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.20, random_state =
42 )

# In[17]:

# Checking dimensions

print("X_train shape:", X_train.shape)

print("X_test shape:", X_test.shape)

print("Y_train shape:", Y_train.shape)

print("Y_test shape:", Y_test.shape)

print(X_train)

# # Support Vector Classifier Algorithm

from sklearn.svm import SVC

svc = SVC(kernel = 'linear', random_state = 42)

svc.fit(X_train, Y_train)

```

```
# # Decision tree Algorithm
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
dectree = DecisionTreeClassifier(criterion = 'entropy', random_state = 42)
```

```
dectree.fit(X_train, Y_train)
```

```
# # Random forest Algorithm
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
ranfor = RandomForestClassifier(n_estimators = 11, criterion = 'entropy', random_state  
= 42)
```

```
ranfor.fit(X_train, Y_train)
```

```
# # Making predictions on test dataset
```

```
Y_pred_svc = svc.predict(X_test)
```

```
Y_pred_dectree = dectree.predict(X_test)
```

```
Y_pred_ranfor = ranfor.predict(X_test)
```

```
# # Model Evaluation
```

```
from sklearn.metrics import accuracy_score
```

```
accuracy_svc = accuracy_score(Y_test, Y_pred_svc)
```

```
accuracy_dectree = accuracy_score(Y_test, Y_pred_dectree)
```

```
accuracy_ranfor = accuracy_score(Y_test, Y_pred_ranfor)
```

```
print("Support Vector Classifier: " + str(accuracy_svc * 100))
```

```
print("Decision tree: " + str(accuracy_dectree * 100))
```

```
print("Random Forest: " + str(accuracy_ranfor * 100))
```

```

c=accuracy_svc * 100

d=accuracy_dectree * 100

e=accuracy_ranfor * 100

scores = [c,d,e]

algorithms = ["Support Vector Machine","Decision Tree","Random Forest"]

for i in range(len(algorithms)):

    print("The accuracy score achieved using "+algorithms[i]+" is: "+str(scores[i])+" %")

sns.set(rc={'figure.figsize':(8,8)})

plt.xlabel("Algorithms")

plt.ylabel("Accuracy score")

sns.barplot(algorithms,scores)

from sklearn.externals import joblib

joblib.dump(ranfor, r'Models\ranfor_model1.pkl')

ranfor_from_joblib = joblib.load(r'Models\ranfor_model1.pkl')

print("Model successfully created...!")

# # Confusion Matrix

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(Y_test, Y_pred_ranfor)

plt.figure(figsize = (10,10))

sns.heatmap(pd.DataFrame(cm), annot=True)

X1 = dataset.iloc[:, [1,2,3,4]].values

```

```

Y1 = dataset.iloc[:, [6]].values

from sklearn.model_selection import train_test_split

X_train1, X_test1, Y_train1, Y_test1 = train_test_split(X1, Y1, test_size = 0.20,
random_state = 42 )

from sklearn.ensemble import RandomForestClassifier

ranfor = RandomForestClassifier(n_estimators = 11, criterion = 'entropy', random_state
= 42)

ranfor.fit(X_train1, Y_train1)

from sklearn.externals import joblib

joblib.dump(ranfor, r'Models\ranfor_model2.pkl')

ranfor_from_joblib = joblib.load(r'Models\ranfor_model2.pkl')

print("Model successfully created...!")

#!/usr/bin/env python

# coding: utf-8

# # Importing the required Libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

```

```

import datetime

import time

from firebase import firebase

fixefixed_interval = 3

firebase      =      firebase.FirebaseApplication('https://crop-prediction-ca036-default-
rttdb.firebaseio.com/', None)

count=1

dataset = pd.read_csv(r'./dataset/Dataset.csv')

# Preview data

dataset.head()

#Verifying it as a 'dataframe' object in pandas

type(dataset)

# Dataset dimensions - (rows, columns)

dataset.shape

# Features data-type

dataset.info()

# Statistical summary

dataset.describe().T

# Count of null values

dataset.isnull().sum()

# Outcome column countplot

```

```

from sklearn.preprocessing import LabelEncoder

# creating instance of labelencoder

labelencoder = LabelEncoder()

# Assigning numerical values and storing in another column

dataset['class'] = labelencoder.fit_transform(dataset['label'])

dataset.head(5)

X = dataset.iloc[:, [1,2,3,4]].values

Y = dataset.iloc[:, [5]].values

# Splitting X and Y

#Controls the shuffling applied to the data before applying the split

from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.20, random_state =
42 )

# Checking dimensions

print("X_train shape:", X_train.shape)

print("X_test shape:", X_test.shape)

print("Y_train shape:", Y_train.shape)

print("Y_test shape:", Y_test.shape)

print(X_train)

# Support Vector Classifier Algorithm

from sklearn.svm import SVC

```

```

svc = SVC(kernel = 'linear', random_state = 42)

svc.fit(X_train, Y_train)

# Decision tree Algorithm

from sklearn.tree import DecisionTreeClassifier

dectree = DecisionTreeClassifier(criterion = 'entropy', random_state = 42)

dectree.fit(X_train, Y_train)

# Random forest Algorithm

from sklearn.ensemble import RandomForestClassifier

ranfor = RandomForestClassifier(n_estimators = 11, criterion = 'entropy', random_state
= 42)

ranfor.fit(X_train, Y_train)

# # Making predictions on test dataset

Y_pred_svc = svc.predict(X_test)

Y_pred_dectree = dectree.predict(X_test)

Y_pred_ranfor = ranfor.predict(X_test)

# Evaluating using accuracy_score metric

from sklearn.metrics import accuracy_score

accuracy_svc = accuracy_score(Y_test, Y_pred_svc)

accuracy_dectree = accuracy_score(Y_test, Y_pred_dectree)

accuracy_ranfor = accuracy_score(Y_test, Y_pred_ranfor)

# Accuracy on test set

```

```

print("Support Vector Classifier: " + str(accuracy_svc * 100))

print("Decision tree: " + str(accuracy_dectree * 100))

print("Random Forest: " + str(accuracy_ranfor * 100))

c=accuracy_svc * 100

d=accuracy_dectree * 100

e=accuracy_ranfor * 100

import joblib

joblib.dump(ranfor, r'Models\ranfor_model1.pkl')

ranfor_from_joblib = joblib.load(r'Models\ranfor_model1.pkl')

ranfor_from_joblib.predict(X_test)

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(Y_test, Y_pred_ranfor)

plt.figure(figsize = (10,10))

sns.heatmap(pd.DataFrame(cm), annot=True)

X1 = dataset.iloc[:, [1,2,3,4]].values

Y1 = dataset.iloc[:, [6]].values

from sklearn.model_selection import train_test_split

X_train1, X_test1, Y_train1, Y_test1 = train_test_split(X1, Y1, test_size = 0.20,
random_state = 42 )

from sklearn.ensemble import RandomForestClassifier

```

```

ranfor = RandomForestClassifier(n_estimators = 11, criterion = 'entropy', random_state
= 42)

ranfor.fit(X_train1, Y_train1)

import joblib

joblib.dump(ranfor, r'Models\ranfor_model2.pkl')

ranfor_from_joblib = joblib.load(r'Models\ranfor_model2.pkl')

Location=firebase.get('Crop/Location', None)

date=firebase.get('Crop/date', None)

Temp=float(firebase.get('Crop/temperature',None))

Humidity=float(firebase.get('Crop/humidity', None))

pH=float(firebase.get('Crop/phvalue', None))

Rainfall=float(firebase.get('Crop/rainfall', None))

#print(Location)

test_data=[Temp,Humidity,pH,Rainfall]

print(test_data)

ranfor_from_joblib = joblib.load(r'Models\ranfor_model1.pkl')

prediction_crop=ranfor_from_joblib.predict([test_data])

prediction_crop=str(prediction_crop)

print(type(prediction_crop))

x=prediction_crop

print(x)

```

```

print("yes")

datetime1=datetime.datetime.now()

date=datetime1.strftime("%x")

time=datetime1.strftime("%X")

day=datetime1.strftime("%A")

device = "42"

status="x"

data={"Date":date,"Time":time,"Day":day}

#firebase.get('/Recommandation', 'Data', data)

firebase.put('/', 'Recommendation/Crop/Prediction_crop', x)

count=0

ranfor_from_joblib = joblib.load(r'Models\ranfor_model2.pkl')

# Use the loaded model to make predictions

prediction_price=ranfor_from_joblib.predict([test_data])

prediction_price=str(prediction_price)

print(prediction_price)

firebase.put('/', 'Recommendation/Crop/Price', prediction_price)

```

Screenshots:

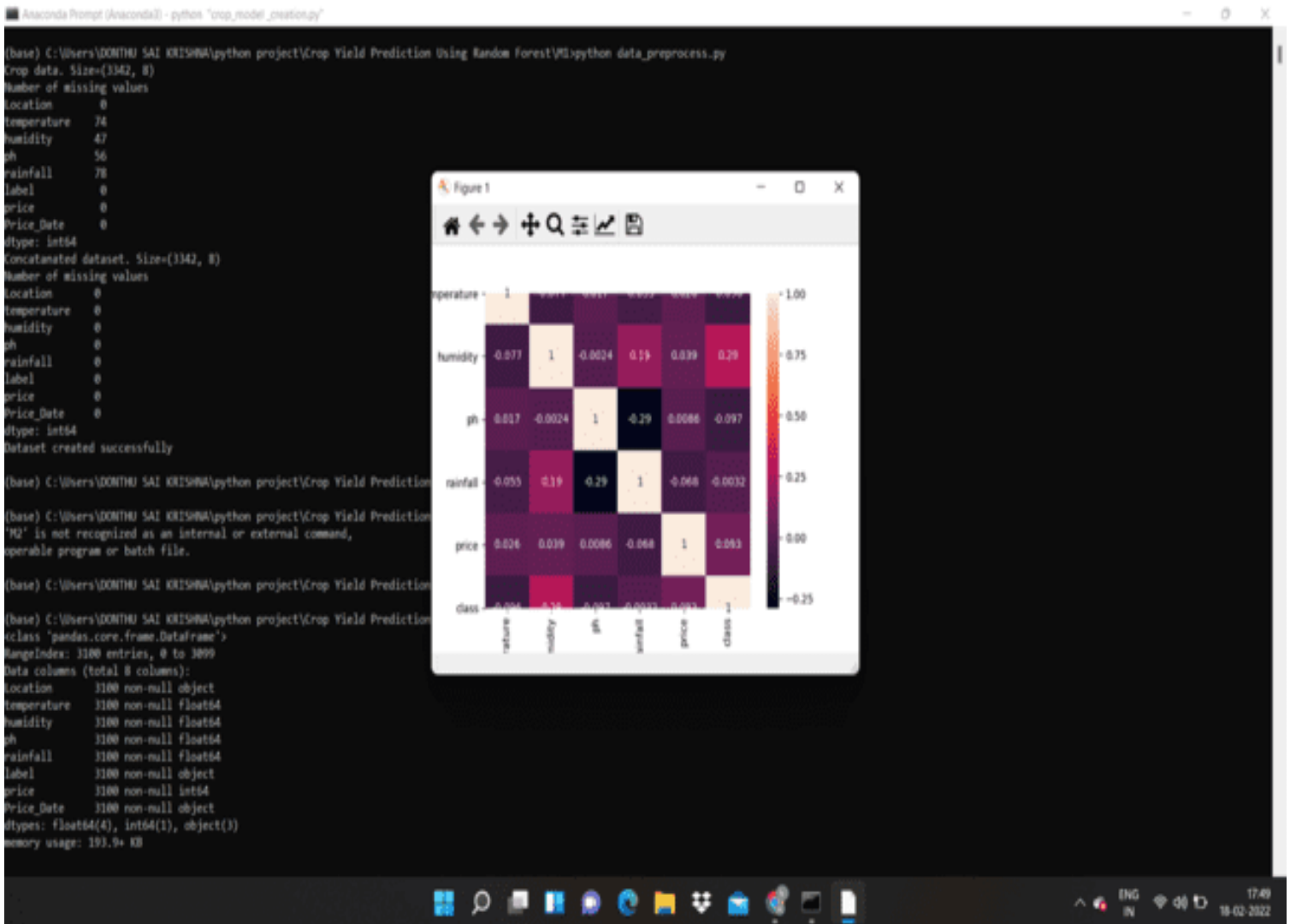
➤ Removing null data in the dataset

```
Anaconda Prompt (Anaconda3)

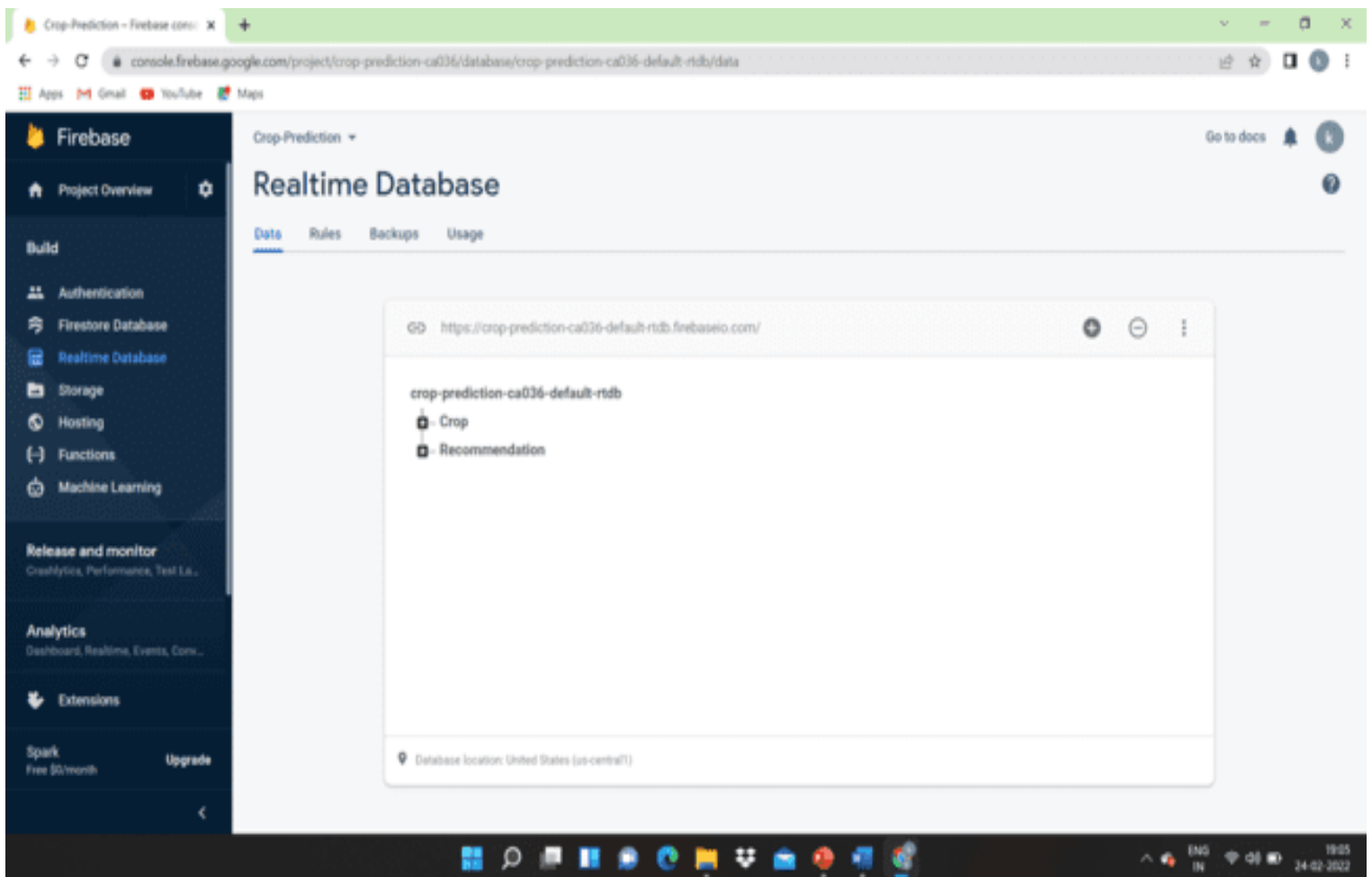
(base) C:\Users\DONITHU SAI KRISHNA\python project\
(base) C:\Users\DONITHU SAI KRISHNA\python project>cd "Crop Yield Prediction Using Random Forest"
(base) C:\Users\DONITHU SAI KRISHNA\python project\Crop Yield Prediction Using Random Forest>cd R0
(base) C:\Users\DONITHU SAI KRISHNA\python project\Crop Yield Prediction Using Random Forest\R0>python data_preprocess.py
Crop data, Size=(3342, 8)
Number of missing values
Location      0
Temperature   74
Humidity       47
pH            56
Rainfall      70
Label         0
Price         0
Price_Date    0
dtype: int64
Concatenated dataset, Size=(3342, 8)
Number of missing values
Location      0
Temperature   0
Humidity      0
pH            0
Rainfall      0
Label         0
Price         0
Price_Date    0
dtype: int64
Dataset created successfully

(base) C:\Users\DONITHU SAI KRISHNA\python project\Crop Yield Prediction Using Random Forest\R0>
```

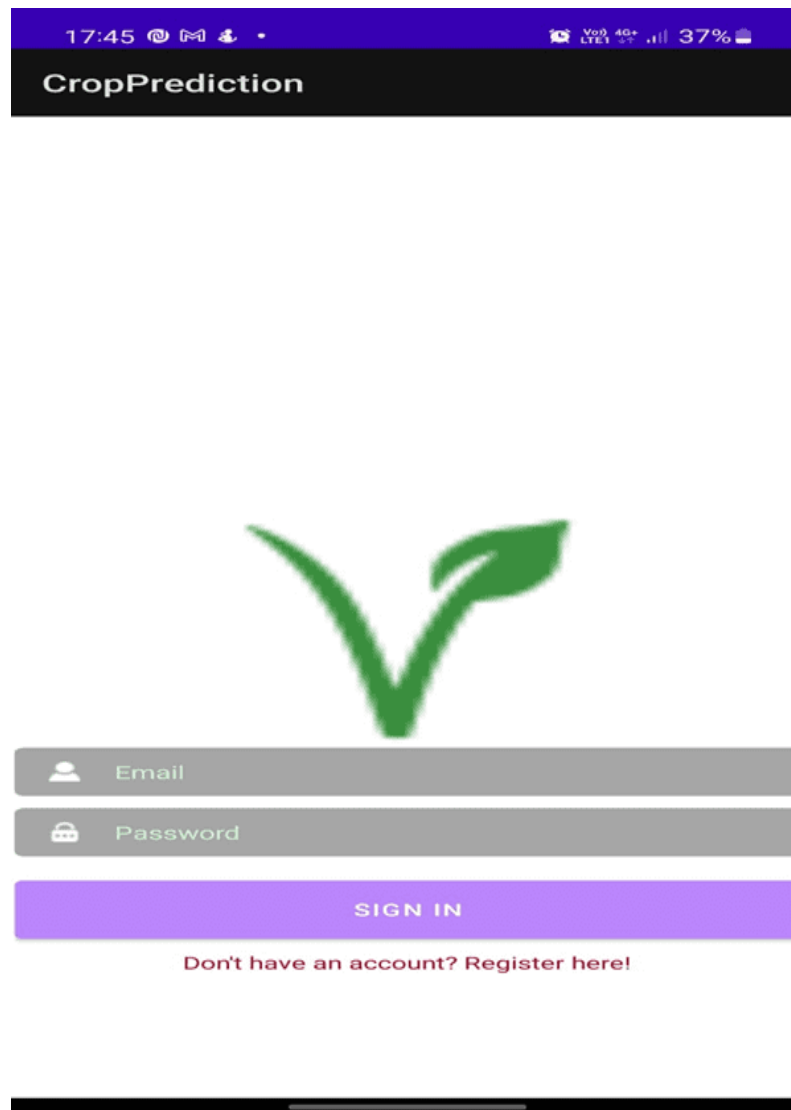
➤ Graph for predicting future crop



➤ Open Firebase Console



➤ Created android app for Crop prediction



Create an account and sign in with email & Password

➤ Crop Prediction

17:45 17:45 36%

CropPrediction

Crop_Recommendation



Enter Location

SELECT_DATE

Enter Temperature

Humidity

Enter PH Value(Maximum 7)

Enter RainFall

PREDICT

- Enter the values of attributes

17:50 @ 100% 35%

CropPrediction

Crop_Recommendation



vellore

SELECT_DATE

18-02-2022

35

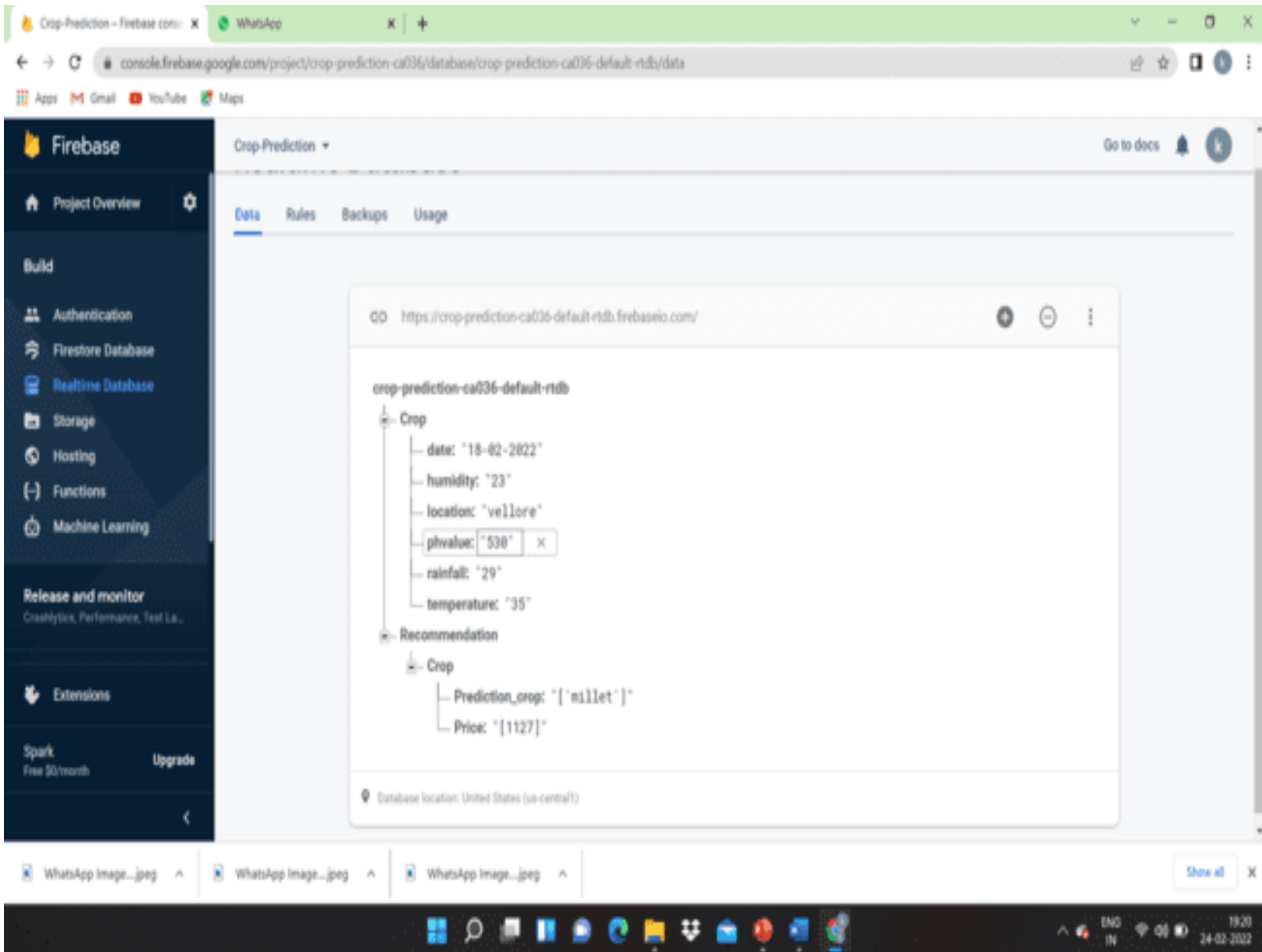
23

530

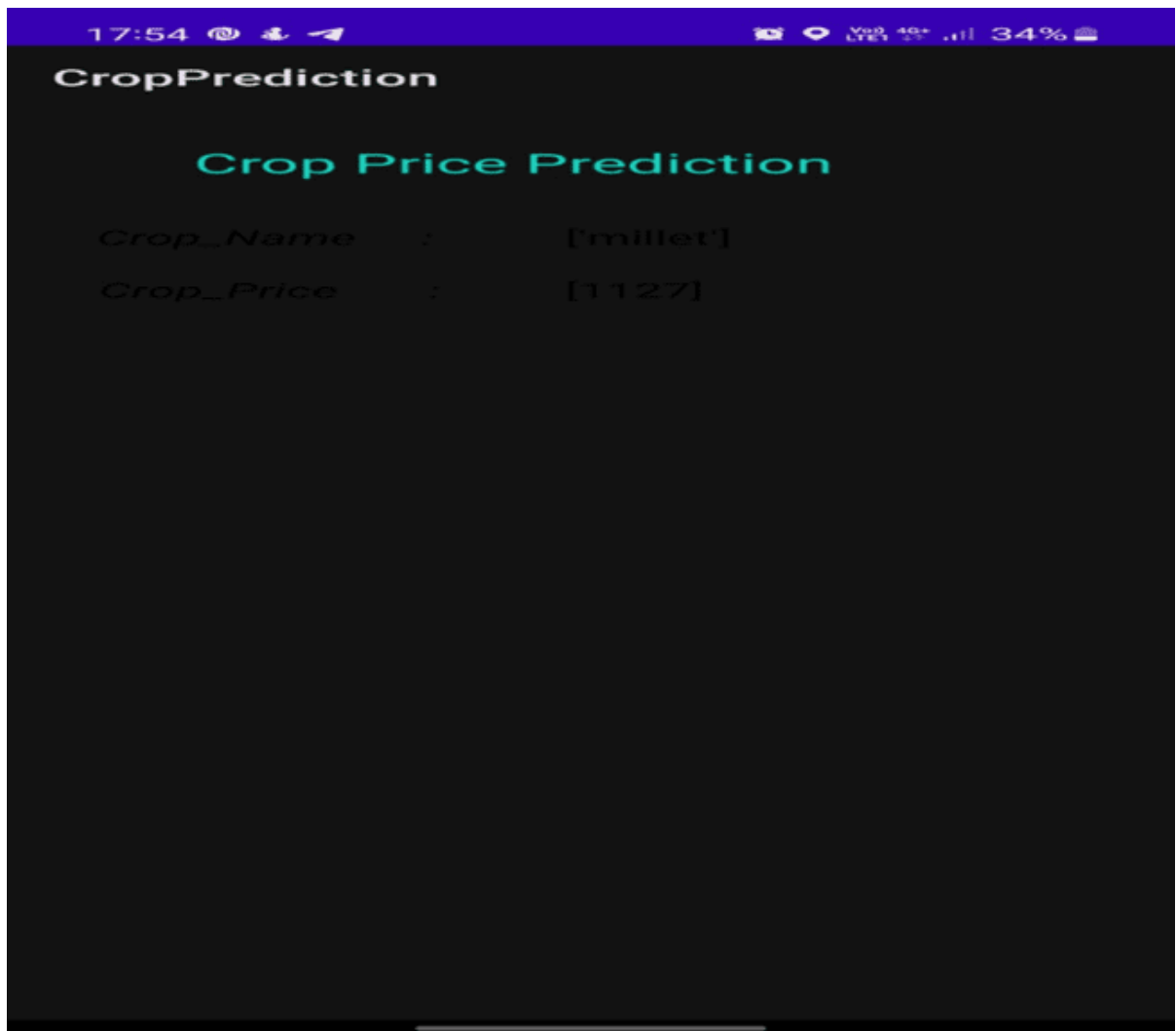
29

PREDICT

- Input given by the user goes to the trained dataset using firebase.



- Predicted the crop with a price.



➤ Successfully predicted crop and price.

```
Anaconda Prompt (Anaconda3)
File "C:\Users\DONTHU SAI KRISHNA\Anaconda3\lib\site-packages\requests\sessions.py", line 546, in get
    return self.request('GET', url, **kwargs)
File "C:\Users\DONTHU SAI KRISHNA\Anaconda3\lib\site-packages\requests\sessions.py", line 533, in request
    resp = self.send(prepare, **kwargs)
File "C:\Users\DONTHU SAI KRISHNA\Anaconda3\lib\site-packages\requests\sessions.py", line 646, in send
    r = adapter.send(request, **kwargs)
File "C:\Users\DONTHU SAI KRISHNA\Anaconda3\lib\site-packages\requests\adapters.py", line 529, in send
    raise ReadTimeout(e, request=request)
requests.exceptions.ReadTimeout: HTTPConnectionPool(host='crop-prediction-ca016-default-rtdb.firebaseio.com', port=443): Read timed out. (read timeout=60)

(base) C:\Users\DONTHU SAI KRISHNA\python project\Crop Yield Prediction Using Random Forest\PO>python crop_prediction.py
'python' is not recognized as an internal or external command,
operable program or batch file.

(base) C:\Users\DONTHU SAI KRISHNA\python project\Crop Yield Prediction Using Random Forest\PO>python crop_prediction.py
cclass 'pandas.core.frame.DataFrame'>
RangeIndex: 3100 entries, 0 to 3099
Data columns (total 8 columns):
location      3100 non-null object
temperature   3100 non-null float64
humidity      3100 non-null float64
ph            3100 non-null float64
rainfall      3100 non-null float64
label         3100 non-null object
price         3100 non-null int64
Price_Date    3100 non-null object
dtypes: float64(4), int64(1), object(3)
memory usage: 193.9+ KB
X_train shape: (2480, 4)
X_test shape: (620, 4)
Y_train shape: (2480, 1)
Y_test shape: (620, 1)
[[ 17.19628279  13.68862614  5.93952042  53.87227743 ]
 [ 31.32728734  54.17967763  6.62367846 144.8018325 ]
 [ 38.01821337  50.0983181  5.67683258  96.08745982]
 ...
 [ 17.92114547  13.04534931  6.43169009  54.56423739]
 [ 38.85454844  70.0216622  5.42888436 219.745386 ]
 [ 26.36629863  52.25738495  7.45646037 177.3176161 ]]
Support Vector Classifier: 86.45361290322581
Decision tree: 92.25806451612984
Random Forest: 93.709677741935485
[35.0, 23.0, 530.0, 29.0]
cclass 'str'>
['millett']
yes
[[1127]]

(base) C:\Users\DONTHU SAI KRISHNA\python project\Crop Yield Prediction Using Random Forest\PO>
```

Conclusion

The present study showed the practical use of data mining techniques in predicting crop yield based on climate input parameters. The built website is user-friendly, and that reliability of prediction in all of the other grains and regions chosen in the analysis should be above 75 percent, indicating greater predictive performance. The net page is developed to predict crop yield by providing data from that area. Any user uses their crop option.

REFERENCES

- [1] . S.Veenadhari, Dr Bharat Misra, Dr CD Singh.2019."Machine learning approach for forecasting crop yield based on climatic parameters.". International Conference on Computer Communication and Informatics (ICCCI), 2014.
- [2] . P.Priya, U.Muthaiah M.Balamurugan.Predicting yield of the crop using machine learning algorithm. International Journal of Engineering Science Research Technology(IJESRT),2018.
- [3] . D Ramesh ,B Vishnu Vardhan. Analysis Of Crop Yield Prediction Using Data Mining Techniques. IJRET: International Journal of Research in Engineering and Technology.(IJERT),2015.
- [4] . F. Baheri, F. Davardoost and V. Ahmadzadeh, "Data mining with learning decision tree and Bayesian network for data replication in Data Grid", Proceedings of the Third International Conference on Contemporary Issues in Computer and Information Sciences, pp. 51- 51, 2012..
- [5] . Abdullah, A., Brobst, S, Pervaiz.I., Umer M.,and A.Nisar.2004. Learning dynamics of pesticide abuse through data mining. Proceedings of Australian Workshop on Data mining and Web Intelligence, New Zealand, January,2004.
- [6] . C.T. Dhanya and D. Nagesh Kumar, "Data mining for evolution of association rules for droughts and floods in India using climate inputs", J. of Geo. Phy. Res.114: 1-15, 2009.

- [7] . VS Boddu, BNK Reddy, MK Kumar, “Low-power and area efficient N-bit parallel processors on a chip,” 2016 IEEE annual India conference (INDICON), pp. 1-4, 2016.
- [8] . M. Kaur, H. Gulati, and H. Kundra, "Data Mining in Agriculture on Crop Price Prediction: Techniques and Applications," International Journal of Computer Applications, vol. 99, no. 12, pp.1-3, August.2014.
- [9] . F. Baheri, F. Davardoost and V. Ahmadzadeh, "Data mining with learning decision tree and Bayesian network for data replication in Data Grid", Proceedings of the Third International Conference on Contemporary Issues in Computer and Information Sciences, pp. 51- 51, 2012