VISVESVARAYA TECHNOLOGICAL UNIVERSITY
"Jnana Sangama", Belgaum - 590018

A project report on

"Intelligent Crop Recommendation System using ML"

submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
in
INFORMATION SCIENCE & ENGINEERING
by

1CR16IS002	Aayush Kumar
1CR16IS061	Omen Rajendra Pooniwala
1CR16IS125	Swapneel Chakraborty

Under the Guidance of

Ms Priyadharshini A
Asst. Professor,Department of ISE, CMRIT, Bengaluru
Department of ISE, CMRIT, Bengaluru

CMR INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
#132, AECS Layout, IT Park Road, Bengaluru - 560037

2019-20

CMR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

#132, AECS Layout, IT Park Road, Bengaluru - 560037

Certificate

This is to Certified that the project work entitled **"Intelligent Crop Recommendation System using ML"** carried out by **Aayush Kumar(1CR16IS002), Omen Rajendra Pooniwala(1CR16IS061) and Swapneel Chakraborty(1CR16IS125)** in partial fulfillment for the award of Bachelor of Engineering in **Information Science & Engineering** of the Visveswaraiah Technological University, Belgaum during the year **2019-20**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

------------------------	------------------------	------------------------
Signature of Guide	Signature of HoD	Signature of Principal
Ms. Priyadharshini A	**Dr. Farida Begam**	**Dr. Sanjay Jain**
Assistant Professor	Professor & Head	Principal
Department of ISE	Department of ISE	CMRIT,
CMRIT	CMRIT	Bengaluru - 37

External Viva

	Name of the Examiners	Institution	Signature with Date
1.	------------------------	------------------------	------------------------
2.	------------------------	------------------------	------------------------

CMR INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
#132, AECS Layout, IT Park Road, Bengaluru - 560037

Declaration

We, **Aayush Kumar(1CR16IS002),Omen Rajendra Pooniwala(1CR16IS061), Swapneel Chakraborty (1CR16IS125)** bonafide students of **CMR Institute of Technology**, Bangalore, hereby declare that the dissertation entitled, **"Intelligent Crop Recommendation System using ML"** has been carried out by us under the guidance of Ms. Priyadharshini A, Asst. Professor, CMRIT, Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science & Engineering, of the Visvesvaraya Technological University, Belgaum during the academic year 2018-2019. The work done in this dissertation report is original and it has not been submitted for any other degree in any university.

Aayush Kumar
Omen Rajendra Pooniwala
Swapneel Chakraborty

Acknowledgement

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of people who made it possible. Success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So, it is with gratitude that we acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

We would like to thank Dr. Sanjay Jain, Principal, CMRIT, Bangalore, for providing an excellent academic environment in the college and his never-ending support for the B.E program.

We would like to express our gratitude towards Dr. Farida Begam, Professor and HOD, Department of Information Science & Engineering CMRIT, Bangalore, who provided guidance and gave valuable suggestions regarding the project.

We consider it a privilege and honour to express our sincere gratitude to our internal guide Ms Priyadharshini A, Asst. Professor, Department of Information Science & Engineering, CMRIT, Bangalore, for her valuable guidance throughout the tenure of this project work.

We would like to thank all the faculty members who have always been very cooperative and generous. Conclusively, we also thank all the non-teaching staff and all others who have done immense help directly or indirectly during our project.

Aayush Kumar
Omen Rajendra Pooniwala
Swapneel Chakraborty

Abstract

Agriculture is a major contributor to the Indian economy. The common problem existing among the Indian farmers are they don't choose the right crop based on their soil requirements. Due to this they face a serious setback in productivity. This problem of the farmers has been addressed through precision agriculture. Precision agriculture is a modern farming technique that uses research data of soil characteristics, soil types, crop yield data collection and suggests the farmers the right crop based on their site-specific parameters. This reduces the wrong choice on a crop and increases the productivity. In this project, we are building an intelligent system, which intends to assist the Indian farmers in making an informed decision about which crop to grow depending on the sowing season, his farm's geographical location and soil characteristics. Further the system will also provide the farmer, the yield prediction if he plants the recommended crop.

Keywords: *Precision Agriculture, yield pridiction.*

Table of Contents

List of Figures

Chapter 1

Preamble

1.1 Introduction

A farmer's decision about which crop to grow is generally clouded by his intuition and other irrelevant factors like making instant profits, lack of awareness about market demand, overestimating a soil's potential to support a particular crop, and so on. A very misguided decision on the part of the farmer could place a significant strain on his family's financial condition. Perhaps this could be one of the many reasons contributing to the countless suicide cases of farmers that we hear from media on a daily basis. In a country like India, where agriculture and related sector contributes to approximately 20.4 per cent of its Gross Value Added (GVA) [2], such an erroneous judgment would have negative implications on not just the farmer's family, but the entire economy of a region. For this reason, we have identified a farmer's dilemma about which crop to grow during a particular season, as a very grave one. The need of the hour is to design a system that could provide predictive insights to the Indian farmers, thereby helping them make an informed decision about which crop to grow. With this in mind, we propose a system, an intelligent system that would consider environmental parameters (temperature, rainfall, geographical location in terms of state) and soil characteristics (pH value, soil type and nutrients concentration) before recommending the most suitable crop to the user.

1.2 Existing System

More and more researchers have begun to identify this problem in Indian agriculture and are increasingly dedicating their time and efforts to help alleviate the issue. Different works include the use of Regularized Greedy Forest to determine an appropriate crop sequence at a given time stamp. Another approach proposes a model that makes use of historical records of meteorological data as training set. Model is trained to identify weather conditions that are deterrent for the production of apples. It then efficiently predicts the yield of apples on the basis of monthly weather patterns. The use of several algorithms like Artificial Neural Network, K Nearest Neighbors, and Regularized Greedy Forest is demonstrated in [5] to select a crop based on the prediction yield rate, which, in turn, is influenced by multiple parameters. Additional features included in the system are pesticide prediction and online trading based on agricultural commodities.

1.2.1 Drawbacks

One shortcoming that we identified in all these notable published works was that the authors of each paper concentrated on a single parameter (either weather or soil) for predicting the suitability of crop growth. However, in our opinion, both these factors should be taken together into consideration concomitantly for the best and most accurate prediction. This is because, a particular soil type may be fit for supporting one type of crop, but if the weather conditions of the region are not suitable for that crop type, then the yield will suffer.

1.3 Proposed System

We to eliminate the aforementioned drawbacks, we propose an Intelligent Crop Recommendation system- which takes into consideration all the appropriate parameters, including temperature, rainfall, location and soil condition, to predict crop suitability. This system is fundamentally concerned with performing the primary function of AgroConsultant, which is, providing crop recommendations to farmers algorithms.We also provide the profit analysis on crops grown in different states which gives the user an easy and reliable insight to decide and plan the crops.

1.4 Plan of Implementation

The steps involved in this system implementation are:-

a) Acquisition of Training Dataset:The accuracy of any machine learning algorithm depends on the number of parameters and the correctness of the training dataset.For the system, we are using various datasets all downloaded for government website and kaggle.
Datasets include:-
Cost of cultivation per ha dataset for major crops in each state
Yield dataset

Modal price of crops
Standard price of crops
Soil nutrient content dataset
Rainfall Temperature dataset

b) Data Preprocessing: This step includes replacing the null and 0 values for yield by -1 so that it does not effect the overall prediction.Further we had to encode the data-set so that it could be fed into the neural network.

c)Training model and crop recommendation:After the preprocessing step we used the data-set to train different machine learning models like neural network and linear regression to attain accuracy as high as possible.

1.5 Problem Statement

Failure of farmers to decide on the best suited crop for his land using traditional and non-scientific methods is a serious issue for a country where approximately 50 percent of the population is involved in farming.Both availability and accessibility of correct and up to date information hinders potential researchers from working on developing country case studies.With resources within our reach we have proposed a system which can address this problem by providing predictive insights on crop sustainability and recommendations based on machine learning models trained considering essential environmental and economic parameters.

1.6 Objective of the Project

- To build a robust model to give correct and accurate prediction of crop sustainability in a given state for the particular soil type and climatic conditions.
- Provide recommendation of the best suitable crops in the area so that the farmer does not incur any losses
- Provide profit analysis of various crops based on previous years data.

.

Chapter 2

Literature Survey

Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique Authors: Rakesh Kumar, M.P. Singh, Prabhat Kumar and J.P. Singh

This paper proposed a method named Crop Selection Method (CSM) to solve crop selection problem, and maximize net yield rate of crop over season and subsequently achieves maximum economic growth of the country. The proposed method may improve net yield rate of crops.

AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms Authors: Zeel Doshi, Subhash Nadkarni, Rashi Agrawal, Prof. Neepa Shah

This paper, proposed and implemented an intelligent crop recommendation system, which can be easily used by farmers all over India. This system would assist the farmers in making an informed decision about which crop to grow depending on a variety of environmental and geographical factors. We have also implemented a secondary system, called Rainfall Predictor, which predicts the rainfall of the next 12 months.

Development of Yield Prediction System Based on Real-time Agricultural meteorological Information Haedong Lee *, Aekyung Moon* * ETRI, 218 Gajeong-ro, Yuseong-gu, 305-700, Korea

This paper contains about the research and the building of an effective agricultural yield forecasting system based on real-time monthly weather. It is difficult to predict the agricultural crop production because of the abnormal weather that happens every year and rapid regional climate change due to global warming. The development of agricultural yield forecasting system that leverages real-time weather information is urgently required. In this research, we cover how to process the number of weather data(monthly, daily) and how to configure the prediction system. We establish a non-parametric statistical model on the basis of 33 years of agricultural weather information. According to the implemented model, we predict final production using the monthly weather information. This paper contains the results of the simulation.

Analysis of Soil Behaviour and Prediction of Crop Yield using Data Mining Approach Monali Paul, Santosh K. Vishwakarma, Ashok Verma Computer science and Engineering GGITS, Jabalpur

This work presents a system, which uses data mining techniques in order to predict the category of the analyzed soil datasets. The category, thus predicted will indicate the yielding of crops. The problem of predicting the crop yield is formalized as a classification rule, where Naive Bayes and K-Nearest Neighbor methods are used.

Crop Recommendation System for Precision Agriculture S.Pudumalar*, E.Ramanujam*, R.Harine Rajashree, C.Kavya, T.Kiruthika, J.Nisha.

This paper, proposes a recommendation system through an ensemble model with majority voting technique using Random tree, CHAID, K-Nearest Neighbor and Naive Bayes as learners to recommend a crop for the site specific parameters with high accuracy and efficiency.

Chapter 3

Theoretical Background

3.1 Overview on Machine Learning

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

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

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

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

3.1.1 Supervised and Unsupervised Learning

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

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

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

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

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

3.2 Machine Learning Tools

There are many different software tools available to build machine learning models and to apply these models to new, unseen data. There are also a large number of well defined machine learning algorithms available. These tools typically contain libraries implementing some of the most popular machine learning algorithms. They can be categorised as follows :

- Pre-built application-based solutions.
- Programming languages which have specialised libraries for machine learning

Using programming languages to develop and implement models is more flexible and gave us better control of the parameters to the algorithms. It also allows us to have a better understanding of the output models produced. Some of the popular programming languages used in the field of machine learning are:

- Python: Python is an extremely popular choice in the field of machine learning and AI development. Its short and simple syntax make it extremely easy to learn

- R: R is one of the most effective and efficient languages for analyzing and manipulating data in statistics. Using R, we can easily produce well-designed publication-quality plot, including mathematical symbols and formulae where needed. Apart from being a general purpose language, R has numerous of packages like RODBC, Gmodels, Class and Tm which are used in the field of machine learning. These packages make the implementation of machine learning algo- rithms easy, for cracking the business associated problems

- Tensorflow:TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications. TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence Research organization to conduct machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains, as well. TensorFlow provides stable Python and C++ APIs, as well as non-guaranteed backward compatible API for other languages.

3.3 SciKit-learn

SciKit learn is an open source machine learning library built for python. Since its release in 2007, Scikit-learn has become one of the most popular open source machine learning libraries. Scikit-learn (also called sklearn) provides algorithms for many machine learning tasks including classification, regression, dimensionality reduction and clustering.

The documentation for scikit-learn is comprehensive, popular and well maintained. Sklearn is built on mature Python Libraries such as NumPy, SciPy, and matplotlib. While languages such as R and MATLAB are extremely popular and useful for machine learning, we decided to choose Python along with its SciKit-learn libraries as our programming language of choice. The reasons for this are:

- We already have some familiarity and exposure to Python, and thus have a smaller learning curve.

- Both Python and Scikit-learn have excellent documentation and tutorials available online

- The number of classic machine learning algorithms that come with Scikit-learn, and the consistent patterns for using the different models i.e., each model can be used with the same basic commands for setting up the data, training the model and using the model for prediction. This makes it easier to try a range of machine learning algorithms on the same data.

- The machine learning algorithms included with sklearn have modifiable parameters known as hyperparameters that effect the performance of the model. These usually have sensible default values, so that we can run them without needing a detailed knowledge or understanding of their semantics.

- The IPython notebook, which is an interactive computational environment for Python, in which a user can combine code execution, rich text, mathematics and plots in a web page. This functionality allows us to provide the notebooks we used to run our experiments almost as an audit and in a presentable.

3.4 Dataset

For the system, we are using various datasets all downloaded for government website and kaggle.

Datasets include:-
Cost of cultivation per ha dataset for major crops in each state
Yield dataset
Modal price of crops
Standard price of crops
Soil nutrient content dataset
Rainfall Temperature dataset

A brief description of the datasets:

- Yield Dataset: This dataset contains yield for 16 major crops grown across all the states in kg per hectare.Yield of 0 indicates that the crop is not cultivated in the respective state.
- Cost of Cultivation dataset:This dataset provides the cost of cultivation for each crop in Rs. per hectare.
- Modal price of crops: This dataset gives the average market prices for those crops over a period of two months
- Standard price of crops:This dataset gives the current market price of the crops in Rs per hectare .
- Soil nutrient content dataset:This dataset has five columns with the attributes in the order-State,Nitrogen content,Phosophorous content,Potassium content and average ph.The nutrient content is represented with encoded alphabets-VL,L,M,H,VH with the meaning:

  VL -Very Low
  L-Low
  M-Medium
  H-High
  VH-Very high

- Rainfall Temperature dataset:This dataset contains crops,max and min rainfall,max and min temprature,max and min rainfall and ph values.

3.5 Data Preprocessing

This step includes replacing the null and 0 values for yield by -1 so that it does not effect the overall prediction.Further we had to encode the data-set so that it could be fed into the neural network.

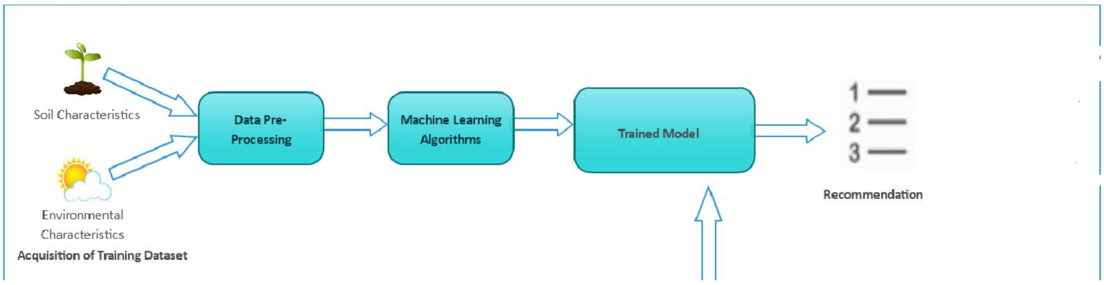

Figure 3.1: Machine Learning process

3.6 Machine Learning Algorithms

Machine Learning algorithms used in the recommendation system are:

- Linear Regression:Linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). Linear regression is used for finding linear relationship between target and one or more predictors. It fits a linear model with coefficients w = (w1, . . . , wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. Linear regression is used for finding linear relationship between target and one or more predictors.

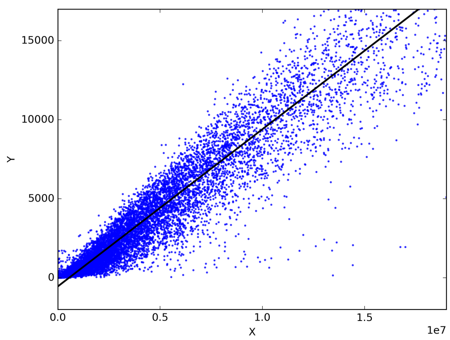

Figure 3.2:Linear regression example

- Logistic Regression:Logistic Regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). The logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1 and the sum adding to one.

- Neural Network:Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated. Neural networks help us cluster and classify.Neural Networks are themselves general function approximations, which is why they can be applied to almost any machine learning problem about learning a complex mapping from the input to the output space.

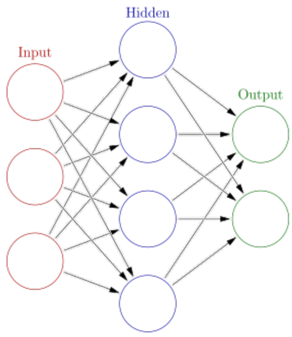

Figure 3.3:Neural network example

Chapter 4

System Requirements Specification

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.
In order to fully understand one's project, it is very important that they come up with a SRS listing out their requirements, how are they going to meet it and how will they complete the project. It helps the team to save upon their time as they are able to comprehend how are going to go about the project. Doing this also enables the team to find out about the limitations and risks early on.
Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources.

4.1 Functional Requirement

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. Following are the functional requirements on the system:

1. All the data must be in the same format as a structured data.
2. The data collected will be vectorized and sent across to the classifier

4.2 Non Functional Requirements

Non functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system properties such as reliability, response time and store occupancy. Non functional requirements arise through the user needs, because of budget constraints, organizational policies and the need for interoperability with other software and hardware systems.

4.2.1 Product Requirements

Correctness: It followed a well-defined set of procedures and rules to engage a conversation with the user and a pre-trained classification model to compute also rigorous testing is performed to confirm the correctness of the data. Modularity: The complete product is broken up into many modules and well-defined interfaces are developed to explore the benefit of flexibility of the product. Robustness: This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness. Non functional requirements are also called the qualities of a system. These qualities can be divided into execution quality and evolution quality. Execu- tion qualities are security and usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

4.2.2 Organizational Requirements

Process Standards: The standards defined by w3 are used to develop the application which is the standard used by the developers. Design Methods: Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs.

4.2.3 Basic Operational Requirements

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points:

- **Mission profile or scenario**: It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.
- **Performance and related parameters:** It points out the critical system parameters to accomplish the mission.
- **Utilization environments:** It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.
- **Operational life cycle**: It defines the system lifetime.

4.2.4 System Configuration

Hardware System Configuration:

- Processor: 2 gigahertz (GHz) or faster processor or SoC.
- RAM: 6 gigabyte (GB) for 32-bit or 8 GB for 64-bit.
- Hard disk space: =16GB.

Software Configuration:

- Operating System: Windows XP/7/8/8.1/10, Linux and Mac
- Coding Language: Python
- Tools:
  1. Pandas
  2. Numpy
  3. Tensorflow
  4. Keras
  5. Sickitlearn

Chapter 5

System Analysis

5.1 Feasibility Study

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

5.1.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. Since the project is Machine learning based, the cost spent in exectuating this project would not demand cost for softwares and related products, as most of the products are open source and free to use. Hence the project would consumed minimal cost and is economically feasible.

5.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical re- quirements of the system. Since machine learning algorithms is based on pure math there is very less requirement for any professional software. And also most of the tools are open source. The best part is that we can run this software in any system without any software requirements which makes them highly portable. Also most of the documentation and tutorials make easy to learn the technology

5.1.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The main purpose of this project which is based on crop prediction is to prevent the farmer from incurring losses and improve productivity.This also ensures that there is no scarcity of food as lack of production may lead to severe consequences. Thus, this is a noble cause for the sake of the society, a small step taken to achieve a secure future.

5.2 Analysis

5.2.1 Performance Analysis

Most of the software we use is open source and free. The models which we use in this software,learn only once ,i.e once they are trained they need not be again fed in for the training phase. One can directly predict for values, hence time-complexity is very less. Therefore this model is temporally sound.

5.2.2 Technical Analysis

As mentioned earlier,the tools used in building this software is open source. Each tool contains simple methods and the required methods are overridden to tackle the problem.

5.2.3 Economical Analysis

The completion of this project can be considered free of cost in its entirety. As the software used in building the model is free of cost and all the data sets used are being downloaded from kaggle and Govt. of India website.

Chapter 6

System Design

6.1 System Development Methodology

System Development methodology is the the development of a system or method for a unique situation. Having a proper methodology helps us in bridging the gap between the problem statement and turning it into a feasible solution. It is usually marked by converting the System Requirements Specifications (SRS) into a real world solution. System design takes the following inputs:

- Statement of work.
- Requirement determination plan.
- Current situation analysis.
- Proposed system requirements including a conceptual data model and metadata (data about data).

6.2 Model Phases

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

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

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

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

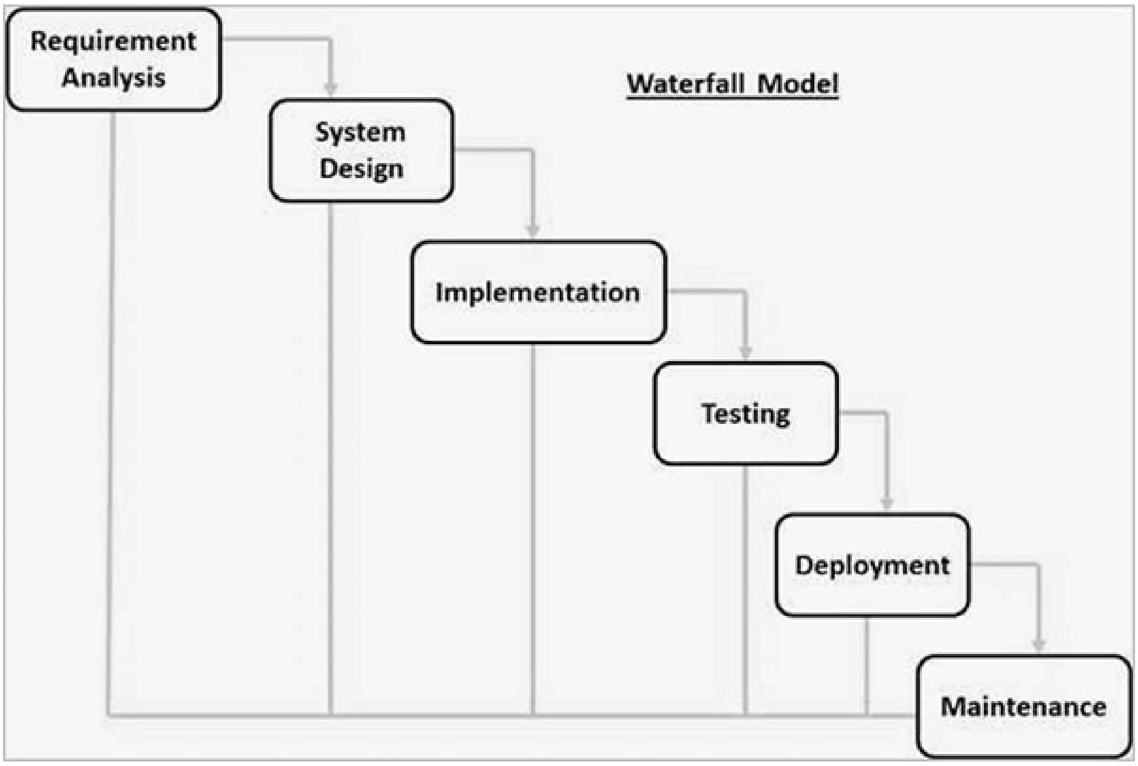

Figure 6.1 Waterfall Model

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

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

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

6.2.1 Advantages of Waterfall model

- Clear project objective
- Stable project requirements
- Progress of system is measurable.
- Logic of software development is clearly understood.
- Better resource allocation.

6.3 System Architecture

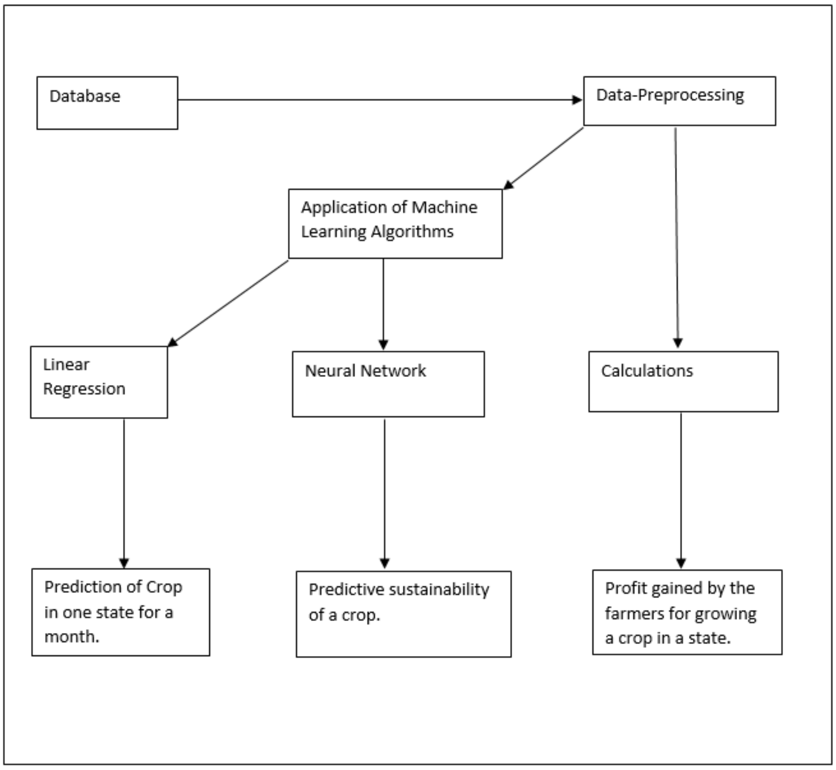

Figure 6.2 System Architecture

A system architecture is a conceptual model using which we can define the structure and behaviour of that system. It is a formal representation of a system. Depending on the context, system architecture can be used to refer to either a model to describe the system or a method used to build the system. Building a proper system architecture helps in analysis of the project, especially in the early stages. Figure 6.2 depicts the system architecture and is explained in the following section.

6.4 Sequence diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. Sequence diagram are an easy and intuitive way of describing the behavior of a sys- tem by viewing the interaction between the system and the environment. A sequence diagram shows an interaction arranged in a time sequence. A sequence diagram has two dimensions: vertical dimension represents time, the horizontal dimension repre- sents the objects existence during the interaction.

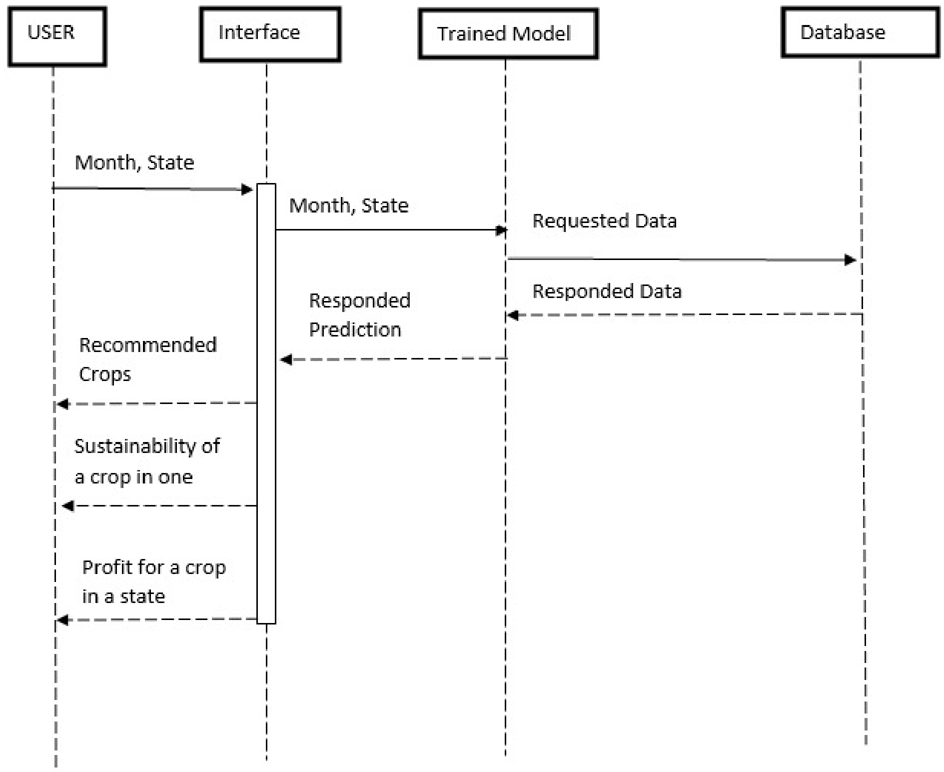

Figure 6.3 Sequence Diagram

Chapter 7

Implementation

7.1 Data Analysis

One of the first steps we perform during implementation is an analysis of the data. This was done by us in an attempt to find the presence of any relationships between the various attributes present in the dataset.

Acquisition of Training Dataset: The accuracy of any machine learning algorithm depends on the number of parameters and the correctness of the training dataset. We In this project analysed mutiple datasets collected from Government wesite -https://data.gov.in/ and Kaggle and carefully selected the parameters that would give the best results.Many work done in this field have considered environmental parameters to predict crop sutainability some have used yield as major factor where as in some works only economic factors are taken into consideration. We have tried to combine both environmental parameters like rainfall,temperature,ph,nutrients in soil,soil type,location and economic parameters like production, and yield to provide accurate and reliable recommendation to the farmer on which crop will be most suitable for his land.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	state	Rice	Jowar	Bajra	Maize	Ragi	Wheat	Barley	Gram	Tur	Groundnu	Mustard	Soyabean	Sunflowe	Cotton	Jute	Mesta	Sugarcan
2	Andhra Pr	2921.186	1054.115	906.734	3328.871	1202.798	792.5525	0	1139.872	441.4852	852.6383	356.25	1450.464	768.493	337.1242	0	1507.329	76472.3
3	Arunachal	1190.479	0	0	1365.529	0	1641.025	0	0	1000	0	963.015	1258.428	0	0	0	0	19073.6
4	Assam	1449	0	0	710	0	1150	0	519	707	0	508	0	0	106	1736	909	3794
5	Bihar	1300.151	929.1806	933.166	2281.541	794.861	1876.421	1124.002	938.5558	1183.194	652.7778	853.5547	0	1365.018	0	1474.782	1344.15	43134.9
6	Chattisgar	1177	843	0	1567	263	996	888	728	471	1122	369	857	459	136	0	361	255
7	Goa	2652	0	0	0	1000	0	0	0	0	1813	0	0	0	0	0	0	5216
8	Gujarat	1610	901	1152	1300	868	2451	0	739	796	1161	1341	716	0	373	0	0	7405
9	Haryana	2894	296	1313	2228	0	3979	2735	725	988	809	1304	0	1598	452	0	0	5998
10	Himachal	1447	0	0	2251	1104	1482	1207	901	0	0	495	1342	0	0	0	0	1801
11	Jammu &	1960	589	571	1535	0	1543	631	0	0	0	635	0	0	0	0	0	0
12	Jharkhand	1413	988	1253	1465	632	1682	922	886	860	698	558	0	0	0	0	1051	3460
13	Karnataka	2561.393	845.2645	626.1794	2654.594	1492.443	736.6376	0	506.7264	489.3979	696.5809	278.5281	683.0272	456.3873	219.2412	0	265.7269	83235.9
14	Kerala	2197	490	0	0	1070	0	0	0	0	763	0	0	0	250	0	0	9136
15	Madhya P	862	985	1244	1525	351	1630	1228	855	754	992	925	928	453	164	0	382	3893
16	Maharash	1594	812	695	1835	992	1320	637	614	683	1066	317	1175	534	189	0	273	7520
17	Manipur	2315.246	0	0	2495.178	0	0	0	0	0	0	461.1111	0	0	0	0	0	32206.8
18	Meghalay	1692	0	0	1452	0	1699	0	0	769	0	648	945	0	172	1430	835	0
19	Mizoram	1501	0	0	1814	0	0	0	0	0	0	742	1113	0	368	0	0	407
20	Nagaland	1556	1246	1269	1609	0	1716	1756	1027	992	1308	842	1282	1197	570	587	0	4475
21	Orissa	1366.031	608.9923	559.4005	1412.742	641.3716	1445.9	0	627.3451	704.1701	1111.208	205.4544	769.2308	802.4356	327.1654	1797.091	796.46	60066.4
22	Punjab	3686	0	984	2702	0	4259	3309	892	876	868	1105	0	1602	563	0	0	6027

Figure 7.1 Yield Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Bajra	3	18	30	3	8	350	750	L	L	M		
2	Banana	4	15	35	6.5	8.5	450	750	M	VL	VL		
3	Barley	4	12	32	3	8	800	1100	VL	VL	M		
4	Bean	2	14	32	5.5	6.5	300	500	L	VL	M		
5	Black pep	6	23	33	5.5	6.5	1200	2500	H	VL	M		
6	Blackgram	2	23	35	5	7	500	700	L	H	VL		
7	Bottle Gou	2	24	27	6.5	7.5	400	650	VL	VL	VL		
8	Brinjal	3	15	32	5.5	6.5	600	1000	VL	L	M		
9	Cabbage	4	12	30	5.5	6.5	300	600	M	VL	H		
10	Cardamon	8	18	35	4.5	7	1200	4000	H	M	M		
11	Carrot	4	7	23	5.5	7	750	1000	M	H	M		
12	Castor see	6	20	30	5	8.5	500	800	VL	H	VL		
13	Cauliflow	4	12	30	6	7	100	300	M	M	M		
14	Chillies	3	18	40	5.5	7	625	1500	VL	VL	L		
15	Coriander	3	15	30	6	10	750	1000	L	L	M		
16	Cotton	4	15	35	6	8	500	1100	M	VL	VL		
17	Cowpea	5	22	35	5	7	700	1100	VL	VL	VL		
18	Drum Sticl	4	20	30	6	7	750	2000	M	L	H		
19	Garlic	4	10	30	6	7	500	800	VL	M	H		
20	Ginger	8	15	35	5	7	1200	1800	VL	M	VL		
21	Gram	4	20	30	5	7	600	900	VL	VL	H		
22	Grapes	4	15	35	6.5	8.5	650	850	VL	H	L		
23	Groundnu	3	20	35	5	7	500	750	VL	VL	VL		

Figure 7.2 Temperature Rainfall and Nutrients dataset

A1 Rainfall

	A	B	C	D	E	F
1	Rainfall	Temperat	Ph	Crop	Production	
2	400.1508	20	3	Bajra	0.0069	
3	400.1633	20	3.2	Bajra	0.00747	
4	400.1639	20	3.2	Bajra	0.00749	
5	400.1797	20	3.2	Bajra	0.00822	
6	400.1958	20	3.2	Bajra	0.00895	
7	400.2905	20	3.2	Bajra	0.01328	
8	400.3887	20	3.2	Bajra	0.01777	
9	400.3989	20	3.2	Bajra	0.01824	
10	400.4159	20	3.2	Bajra	0.01902	
11	400.4561	20	3.2	Bajra	0.02085	
12	400.4604	20	3.2	Bajra	0.02105	
13	400.4672	20	3.2	Bajra	0.02136	
14	400.5277	20	3.2	Bajra	0.02413	
15	400.542	20	3.2	Bajra	0.02478	
16	400.5644	20	3.2	Bajra	0.02581	
17	400.5686	20	3.2	Bajra	0.026	

Figure 7.3 Regression model training dataset

A1 Andaman and Nicobar

	A	B	C	D	E	F	G	H	I	J
1	Andaman	L	VL	L	6.5					
2	Andhra Pr	H	VH	M	6.3					
3	Arunachal	H	M	M	5.6					
4	Assam	M	L	VL	6.5					
5	Bihar	VL	VH	H	6					
6	Goa	M	VL	M	5.4					
7	Gujarat	L	VH	H	6					
8	Haryana	VL	M	H	6					
9	Himachal	M	M	H	6.5					
10	Jammu an	H	M	M	6					
11	Jharkhand	L	M	M	6.2					
12	Kerala	H	VH	M	6.2					
13	Madhya P	L	M	H	6.2					
14	Maharash	M	H	M	6					
15	Manipur	L	VH	L	5.7					
16	Meghalya	L	M	M	5.9					
17	Mizoram	M	M	M	5					
18	Nagaland	M	VL	H	6.1					

Figure 7.4 Soil Nutrients distribution as per state(Nitrogen,Phosphorous,Potassium).

Soil	Month	State	Rice	Wheat	Cotton	Sugarcane	Tea	Coffee	Cashew	Rubber	Coconut	Oilseed	Ragi	Maize	Groundnu	Millet	Barley
Alluvial	January	Andhra Pr	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Assam	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Bihar	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Gujarat	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Haryana	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Himachal	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Karnataka	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Kerala	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Madhya P	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Maharash	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Odisha	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Punjab	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Tamil Nad	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Uttar Prad	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	West Ben	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Chhattisg	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Jharkhand	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	January	Uttrakhan	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alluvial	February	Andhra Pr	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Alluvial	February	Karnataka	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Alluvial	February	Maharash	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Alluvial	February	Tamil Nad	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Figure 7.5 Encoded dataset

Profit analysis was performed using cost of cultivation,market price,standard price and yield dataset.This was performed as a first step to know how much impact does profit as a parameter can have on crop prediction. The function below calculates the profit for each crop grown in the state and assigns a -1 value for the states with 0 or no production of the given crop.

```
for i in range(0,yield_np_mat.shape[0]-1):
    for j in range(1,yield_np_mat.shape[1]):
        if(yield_np_mat[i][j]!=0.0):
    #if yielding in current state then calculate
      profit
            rs_per_kg = grouped_market[(
              grouped_market['state']==states[i]) &
               (grouped_market['commodity'] ==
              crops[j-1])]['modal_pricemean']

            if(rs_per_kg.shape[0]==1):
        #use market price data
```

```
            diff = ((yield_np_mat[i][j]) *
               rs_per_kg.values[0]) -
            coc_dict[str(crops[j-1].lower())]
            val = diff if diff>0.0 else 1.
        else:
#use approximated standard price
            diff = ((yield_np_mat[i][j]) *
               std_prices[str(crops[j-1].lower()
               )]) - coc_dict[str(crops[j-1].
               lower())]
            val = diff if diff>0.0 else 1.0

#append all results to a table
        prediction_table.append([states[i],
          crops[j-1], val])
    else:
#states which cant produce the given crop
        print("**", states[i],crops[j-1])
        prediction_table.append([states[i],
          crops[j-1], -1])
```

7.2 Data Preprocessing

After analysing and visualizing the data, the next step is preprocessing. Data preprocessing is an important step as it helps in cleaning the data and making it suitable for use in machine learning algorithms. Most of the focus in preprocessing is to remove any outliers or erroneous data, as well as handling any missing values.

Missing data can be dealt with in two ways. The first method is to simply remove the entire row which contains the missing or erroneous value. While this an easy to execute method, it is better to use only on large datasets. Using this method on small datasets can reduce the dataset size too much, especially if there are a lot of missing values. This can severely affect the accuracy of the result. Since ours is a relatively small dataset, we will not be using this method.

The datset that we used had values that were in string format so we had to transform and encode the into integer valued so as to pass as an input to the neural network.First we converted the data into pandas categorical data and then generated codes for crops and states respectively we than appended these and created separated datasets.The steps are illustrated below.

```
profit_data['state']=profit_data['state'].
    astype('category')
profit_data['crop'] = profit_data['crop'].
    astype('category')

state_encoding.to_csv("data/result/statecode.
    csv", index=False)
crop_encoding.to_csv("data/result/cropcode.csv"
    , index = False)

dataset[columns].to_csv("data/result/
    encoded_dataset.csv", index = False, header =
      False)
```

Further to reduce the amount of data going into the linear regression model we filtered the crops based on the required nutrients and nutrients present in the soil.If the nutrient content of the soil was below that required by the crops,then that crop was discarded, in this way we were able to reduce the trainig time a lot.

```
for row in reader:
    ncrop=conv(row[8])
    pcrop=conv(row[9])
    kcrop=conv(row[10])
    if(narea>=ncrop and parea>=pcrop and karea
        >=kcrop):
        no_months=int(row[1])
        total=row[0]+","+str(rainfall_final[
            no_months-1])+","+str(temp_final[
            no_months-1])+","+ph+"\n"
        metacrops.writelines(total)
```

7.3 Machine Learning Models

7.3.1 Linear Regression Model

Linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). Linear regression is used for finding linear relationship between target and one or more predictors. It fits a linear model with coefficients w = (w1, . . . , wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. Linear regression is used for finding linear relationship between target and one or more predictors.

After the preprocessing was done we trained our liner regression which would return y-pred value for each crop based on a straight line fit between rainfall,temprature,ph and production.The implementation is given below:

```
for row in reader:
    crop=row[0]
    metadata=dataset.loc[dataset['Crop'] ==
       crop]
    X = metadata.iloc[:, :-2].values#rainfall,
       temp,Ph
    Y = metadata.iloc[:, 4].values#Production

     X_train, X_test, Y_train, Y_test =
        train_test_split(X, Y, test_size = 0.1,
        random_state = 0)
    #print(Y_train)

    regressor = LinearRegression()
    regressor.fit(X_train, Y_train)

    X_locbased = locbased.loc[[n]].values
    X_locbased = X_locbased[:, 1:4]
    Y_pred=regressor.predict(X_locbased)
    print(Y_pred)

        if Y_pred>0:
            crop_Y_pred.append(round(Y_pred
               [0],3))
            crop_name.append(crop)
            print(crop_name)
```

```
    accuracy = regressor.score(X_test ,Y_test)
    print("ACCURACY_SCORE:-" ,accuracy*100, '%')
    sorted_crops=quicksort(crop_name ,crop_Y_pred ,0 ,
        len(crop_Y_pred)-1)
    csvfile.close

    return sorted_crops
```

At the end we have sorted the crops based on the y-ored value returned by the linear regression model using quick sort giving the crop with the best score first in the list.

7.3.2 Neural Network

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated. Neural networks help us cluster and classify.

Our implementation of the neural network was facilitated with the help of keras module.We implemented a sequential model with 3 input layers and 15 out put layers which gave the sustainability of each 15 crops given the input in terms of state,month and soil.

```
    model = Sequential()
    model.add(Dense(15, input_dim=3, init="uniform"
        ,activation="sigmoid"))

    sgd = SGD(lr=0.01)
```

```
model.compile(loss="binary_crossentropy",
   optimizer=sgd, metrics=["accuracy"])
model.fit(trainData.values, trainLabels.values,
    epochs=500, batch_size=10, verbose=1)
```

Chapter 8

Testing

8.1 Testing Methodologies

The program comprises of several algorithms which are tested individually for the accuracy. we check for the correctness of the program as a whole and how it performs.

8.2 Unit Testing

Unit tests focus on ensuring that the correct changes to the world-state take place when a transaction is processed. The business logic in transaction processor functions should have unit tests, ideally with 100 percent code coverage. This will ensure that you do not have typos or logic errors in the business logic. The various modules can be individually run from a command line and tested for correctness. The tester can pass various values, to check the answer returned and verify it with the values given to him/her. The other work around is to write a script, and run all the tests using it and write the output to a log file and using that to verify the results. We tested each of the algorithms individually and made changes in preprocessing accordingly to increase the accuracy.

8.3 System Testing

System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the systems compliance with the specified requirements. System Testing is the testing of a complete and fully integrated software product. and White Box Testing. System test falls under the black box testing category of software testing. Different Types of System Testing:

- Usability Testing - Usability Testing mainly focuses on the users ease to use the application, flexibility in handling controls and ability of the system to meet its objectives.

- Load Testing - Load Testing is necessary to know that a software solution will perform under real-life loads.

- Regression Testing- - Regression Testing involves testing done to make sure none of the changes made over the course of the development process have caused new bugs.

- Recovery Testing - Recovery testing is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.

- Migration Testing - Migration testing is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.

8.4 Quality Assurance

Quality Assurance is popularly known as QA Testing, is defined as an activity to ensure that an organization is providing the best possible product or service to customers. QA focuses on improving the processes to deliver Quality Products to the customer. An organization has to ensure, that processes are efficient and effective as per the quality standards defined for software products.

8.5 Functional Test

Functional Testing is also known as functional completeness testing, Functional Test- ing involves trying to think of any possible missing functions. As chat-bot evolves into new application areas, functional testing of essential chatbot components. Functional testing evaluates use-case scenarios and related business processes, such as the behavior of smart contracts.

Chapter 9

Results and Performance Analysis

For the purposes of this project we have used three popular algorithms: Linear regression, Logistic regression and Neural network. All the algorithms are based on supervised learning.Our overall system is divided into three modules:

- Profit analysis
- Crop recommender
- Crop Sustainability predictor

Output of profit analysis:

```
In [15]: profit_data
Out[15]:
```

	state	crop	profit
0	Andhra Pradesh	Rice	6.385184e+04
1	Andhra Pradesh	Jowar	1.097407e+04
2	Andhra Pradesh	Bajra	7.414478e+03
3	Andhra Pradesh	Maize	3.136984e+04
4	Andhra Pradesh	Ragi	5.636376e+03
5	Andhra Pradesh	Wheat	1.000000e+00
6	Andhra Pradesh	Barley	-1.000000e+00
7	Andhra Pradesh	Gram	5.058972e+03
8	Andhra Pradesh	Tur	1.000000e+00
9	Andhra Pradesh	Groundnut	1.017747e+04
10	Andhra Pradesh	Mustard	1.000000e+00
11	Andhra Pradesh	Soyabean	7.632153e+03
12	Andhra Pradesh	Sunflower	9.739718e+03
13	Andhra Pradesh	Cotton	1.000000e+00
14	Andhra Pradesh	Jute	-1.000000e+00
15	Andhra Pradesh	Mesta	1.609931e+03
16	Andhra Pradesh	Sugarcane	8.931305e+05
17	Arunachal Pradesh	Rice	9.995866e+02
18	Arunachal Pradesh	Jowar	-1.000000e+00

Figure 9.1 Profit on crops per state

As can be seen above after the analysis we get the profit data for each crop grown in all the states.This provides a clear insight on which crop to be selected.

Output for Crop recommender:

```
['Barley', 'Bottle Gourd']
[-13.77759935]
[-0.81033561]
[-5.14828826]
[-2.21432874]
[-4.81342706]
[-59.60942137]
[-2.84073175]
[-75.81061724]
[0.69908353]
['Barley', 'Bottle Gourd', 'Groundnut']
[0.70479237]
['Barley', 'Bottle Gourd', 'Groundnut', 'Jowar']
[0.61912891]
['Barley', 'Bottle Gourd', 'Groundnut', 'Jowar', 'Khesari']
[-9.91352688]
[-0.28529204]
[-1.7600131]
[4.11882972]
['Barley', 'Bottle Gourd', 'Groundnut', 'Jowar', 'Khesari', 'Orange']
[-262.25858254]
[8.45690409]
['Barley', 'Bottle Gourd', 'Groundnut', 'Jowar', 'Khesari', 'Orange', 'Potato']
[2.20757848]
['Barley', 'Bottle Gourd', 'Groundnut', 'Jowar', 'Khesari', 'Orange', 'Potato', 'Raddish']
[-4.04423303]
[0.71076754]
['Barley', 'Bottle Gourd', 'Groundnut', 'Jowar', 'Khesari', 'Orange', 'Potato', 'Raddish', 'Sannhamp']
[-0.77569453]
[-54.22040544]
[-17.77020347]
[-24.76243297]
ACCURACY SCORE:- 88.26342114086883 %
```

Figure 9.2 Regression model output

```
In [55]: print ('Recommended crop for the month of '+NumtoMonth[month]+' in '+state+' is/are: \n'+final_crop)

         Recommended crop for the month of May in Bihar is/are:
         Potato,Bottle Gourd,Orange,Barley,Raddish,Sannhamp,Jowar,Groundnut,Khesari

In [ ]:
```

Figure 9.3 Crop recommendation

The Crop recommendation model enlists the crops in an order where the first crop will have the highest productivity followed by the remaining in the list.

Output of Crop Sustainability Predictor:

```
173   0.536654  0.242637  0.167341   0.201528  0.163237  0.181734  0.068485
174   0.428104  0.075437  0.008961   0.209194  0.432152  0.157399  0.039976
175   0.523158  0.221789  0.238394   0.327195  0.251305  0.164559  0.104569
176   0.482467  0.185739  0.208894   0.164559  0.030575  0.024127  0.002197

In [62]: Soil=input()
         Month=input()
         State=input()

         Alluvial
         March
         Punjab

In [63]: # df
```

Figure 9.4 Input for the predictor

```
In [68]: pred = model.predict_proba(Choices)
         df2 = pd.DataFrame(pred, columns=["Rice","Wheat","Cotton","Sugarcane","Tea",    "Coffee","Cashew","Rubber","Coconut","O:
         print(df2)
         df2.shape

               Rice   Wheat    Cotton  Sugarcane       Tea    Coffee    Cashew  \
         0  0.495499  0.2027  0.236318   0.184892  0.043014  0.035415  0.004605

              Rubber   Coconut  Oilseed      Ragi    Maize  Groundnut    Millet  \
         0  0.002973  0.005234  0.06702  0.087935  0.13789   0.075435  0.125787

              Barley
         0  0.043136
Out[68]: (1, 15)
```

Figure 9.5 Crop Sustainability prediction values

Here we observe the prediction value of sustainability for each crop given the three inputs.Hence from these values one can get a clear -cut idea on which crop will give better yield.

9.1 Snapshots

```
In [12]: for i in range(0,yield_np_mat.shape[0]-1):
             for j in range(1,yield_np_mat.shape[1]):
                 if(yield_np_mat[i][j]!=0.0):
             #if yielding in current state then calculate profit
                     rs_per_kg = grouped_market[(grouped_market['state']==states[i]) & (grouped_market['commodity'] == crops[j-1]

                     if(rs_per_kg.shape[0]==1):
                 #use market price data
                         diff = ((yield_np_mat[i][j]) * rs_per_kg.values[0]) - coc_dict[str(crops[j-1].lower())]
                         val = diff if diff>0.0 else 1.0
                     else:
                 #use approximated standard price
                         diff = ((yield_np_mat[i][j]) * std_prices[str(crops[j-1].lower())]) - coc_dict[str(crops[j-1].lower())]
                         val = diff if diff>0.0 else 1.0

                 #append all results to a table
                     prediction_table.append([states[i], crops[j-1], val])
                 else:
                 #states which cant produce the given crop
                     print("**", states[i],crops[j-1])
                     prediction_table.append([states[i], crops[j-1], -1])

** Andhra Pradesh Barley
** Andhra Pradesh Jute
** Arunachal Pradesh  Jowar
** Arunachal Pradesh  Bajra
** Arunachal Pradesh  Ragi
```

Figure 9.6 Profit analysis code

```
In [56]: state = input("Enter the name of state: ")

         Enter the name of state: Bihar

In [57]: month = int(input("Enter month: "))

         Enter month: 5

In [58]: import os
         import sys
```

Figure 9.7 Input for regression model

```
    crop_name=[]
    dataset=pd.read_csv('data/regressiondb.csv')
    locbased=pd.read_csv('data/metacrops.csv')

    try:
        with open('data/metacrops11.csv', 'r') as csvfile:
            reader = csv.reader(csvfile)

            for row in reader:
                crop=row[0]
                metadata=dataset.loc[dataset['Crop'] == crop]
                X = metadata.iloc[:, :-2].values#rainfall,temp,Ph
                Y = metadata.iloc[:, 4].values#Production



                X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1, random_state = 0)
                print(Y_train)

                regressor = LinearRegression()
                regressor.fit(X_train, Y_train)


                X_locbased = locbased.loc[[n]].values
                X_locbased = X_locbased[:, 1:4]
                Y_pred=regressor.predict(X_locbased)

                if Y_pred>0:
                    crop_Y_pred.append(round(Y_pred[0],3))
                    crop_name.append(crop)

        sorted_crops=quicksort(crop_name,crop_Y_pred,0,len(crop_Y_pred)-1)
        csvfile.close

        return sorted_crops

    except IOError:
        print ("No file exists named metacrops11.csv")
```

Figure 9.8 Regression model

```
In [15]: print(trainData.shape, testData.shape, trainLabels.shape, testLabels.shape)

(596, 3) (199, 3) (596, 15) (199, 15)

In [16]: sgd = SGD(lr=0.01)
model.compile(loss="binary_crossentropy", optimizer=sgd, metrics=["accuracy"])
model.fit(trainData.values, trainLabels.values, epochs=500, batch_size=10, verbose=1)

Epoch 2/500
596/596 [==============================] - 0s 161us/step - loss: 0.3808 - accuracy: 0.8907
Epoch 3/500
596/596 [==============================] - 0s 153us/step - loss: 0.3534 - accuracy: 0.8922
Epoch 4/500
596/596 [==============================] - 0s 142us/step - loss: 0.3423 - accuracy: 0.8928
Epoch 5/500
596/596 [==============================] - 0s 163us/step - loss: 0.3361 - accuracy: 0.8921
Epoch 6/500
596/596 [==============================] - 0s 171us/step - loss: 0.3321 - accuracy: 0.8917
Epoch 7/500
596/596 [==============================] - 0s 143us/step - loss: 0.3293 - accuracy: 0.8922
Epoch 8/500
596/596 [==============================] - 0s 171us/step - loss: 0.3270 - accuracy: 0.8922
Epoch 9/500
596/596 [==============================] - 0s 161us/step - loss: 0.3254 - accuracy: 0.8917
Epoch 10/500
596/596 [==============================] - 0s 156us/step - loss: 0.3238 - accuracy: 0.8924
Epoch 11/500

In [169]: (loss, accuracy) = model.evaluate(testData.values, testLabels.values, batch_size=40, verbose=1)

print("[INFO] loss={:.4f}, accuracy: {:.4f}%".format(loss,accuracy * 100))

199/199 [==============================] - 0s 121us/step
[INFO] loss=0.3013, accuracy: 89.7487%
```

Figure 9.9 Training the neural network

```
[2.57039 0.5     2.67816 ... 1.18074 1.83628 1.91249]
[0.44726 0.36531 0.42857 0.36364 0.59279 0.26923 0.82258 0.15385 0.43836
 0.78992 0.37079 0.75128 0.27273 0.51613 0.43987 0.81875 0.36842 0.91536
 0.14286 0.79524 0.36667 1.71509 0.26361 0.25703 0.52896 0.44118 0.43519
 0.36706 0.41752 0.33971 1.82295 1.14779 1.0625  0.38889 0.51556 1.49069
 0.3774  0.79592 0.81481 0.42262 1.41485 0.23722 0.3663  0.39048 0.70868
 0.35346 0.42056 0.52615 0.23881 0.35145 0.25    0.4     0.422   0.40683
 0.26087 0.33333 0.125   0.8179  0.33117 2.66667 0.47318 0.33333 0.82456
 0.13107 0.42206 0.13333 1.80906 0.83333 0.31283 0.33333 0.48101 0.35258
 0.52568 0.23446 0.39227 0.36667 0.29882 1.29509 0.24883 0.925   0.6383
 0.79695 1.52349 0.33333 0.31907 0.23776 0.33333 0.32189 0.37175 0.6276
 0.54309 1.45418 0.44444 0.23656 0.2625  0.20513 0.35948 0.11429 0.42857
 0.37143 0.37255 0.11538 0.31818 0.79592 0.31321 0.52773 0.29167 0.19565
 0.25926 0.18334 0.48333 0.16714 1.      0.39683 0.35204 0.25581 0.89474
 0.36749 0.26403 0.48413 0.42177 0.35714 1.46293 0.98039 0.33333 1.45455
 0.6704  0.35825 0.86957 0.36842 0.23656 0.42857 1.1     1.15022 0.33333
 0.275   0.42857 0.25511 0.46154 0.35    0.81939 0.89216 0.41226 0.26427
 0.42593 0.38663 0.26761 0.41667 1.02611 0.424   0.44672 0.26476 0.18072
 0.37727 0.35616 0.32432 0.33333 0.32787 0.37245 1.46167 0.33333 0.26414
 0.36747 1.7525  0.25806 0.4333  0.375   0.31818 0.27273 0.62738 0.09
 0.79669 0.36493 0.1773  0.32495 1.14706 0.4303  0.21224 0.28    0.36842
 0.27419 0.43977 0.42449 0.33495 0.57143 0.28571 0.33333 0.79794 0.28947
 0.38356 0.27778 0.5558  0.40741 0.39413 0.31579 0.35174 0.76415 0.34783
 0.3     0.95293 0.21429 0.93202 0.79778 0.48214 0.37297 0.36667 0.3125
 0.21429 1.4375  0.22761 0.25056 0.35701 0.45124 0.45417 0.36364 0.26222
 0.2164  0.4     0.21212 0.52632 0.45106 0.91371 0.36364 0.20804 0.44361
 0.40909 0.33161 0.44444 0.46341 0.31818 0.36095 0.27485 2.33333 0.46458
 0.48008 0.3629  0.68966 0.3516  0.36232 0.8125  0.35714 0.36667 0.375
 0.25926 0.1961  0.36364 0.36508 0.30451 0.27711 0.33333 0.8     0.69204
 0.22059 0.38858 0.48807 0.26364 0.81799 0.43373 0.15605 0.39374 0.79683
 3.2     0.3662  0.33333 0.2641  0.28577 1.4     0.27642 0.32722 0.33
 0.17647 0.4503  0.27628 0.48011 0.31818 0.36709 0.375   0.5     0.42424
```

Figure 9.10 Training set for Regression model

Chapter 10

Conclusion

This system helps the farmer to choose the right crop by providing insights that ordinary farmers don't keep track of thereby decreasing the chances of crop failure and increasing productivity. It also prevents them form incurring losses. The system can be extended to the web and can be accessed by millions of farmers across the country.We could achieve an accuracy of 89.88 percent from the neural network and an accuracy of 88.26 percent from the linear regression model

Further development is to integrate the crop recommendation system with another subsystem, yield predictor that would also provide the farmer an estimate of production if he plants the recommended crop.

References

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

[2] Haedong Lee and Aekyung Moon, “Development of Yield Prediction System Based on Real-time Agricultural Meteorological Information”, 16th International Conference on Advanced Communication Technology, 2014

[3] T.R. Lekhaa, “Efficient Crop Yield and Pesticide Prediction for Improving Agricultural Economy using Data Mining Techniques”, International Journal of Modern Trends in Engineering and Science (IJMTES), 2016, Volume 03, Issue 10

[4] Jay Gholap, Anurag Ingole, Jayesh Gohil, Shailesh Gargade and Vahida Attar, “Soil Data Analysis Using Classification Techniques and Soil Attribute Prediction”, International Journal of Computer Science Issues, Volume 9, Issue 3

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

[6] AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms. Zeel Doshi , Subhash Nadkarni , Rashi Agrawal, Prof. Neepa Shah.

[7] Crop Recommendation System for Precision Agriculture S.Pudumalar*, E.Ramanujam*, R.Harine Rajashree, C.Kavya, T.Kiruthika, J.Nisha.

[8] Tom M. Mitchell, Machine Learning, India Edition 2013, McGraw Hill Educa- tion.

[9] `https://data.gov.in/g`

[10] Kaggle`https://www.kaggle.com/notebook`