

## CHAPTER-1

### INTRODUCTION

Nowadays, precision agriculture aims at increasing productivity and maximizing the yields of a crop. The entire crop cycle can benefit from an application of the correct amount of spray (such as water, fertilizers, pesticides or fungicides) at the proper time and place. In parallel, research on multispectral image analysis of agricultural fields is starting to detect diseases in plants. Farmers generate maps

of spatial variabilities based on geo-located sensors. These sensors collect many variables and provide historical and real time information. Recently, Unmanned Aerial Vehicles (UAV) have enabled precision agriculture. These drones are equipped with multispectral or RGB cameras to collect aerial images and create maps of the selected region. High precision agriculture applies mainly to yield monitoring or remote sensing. It is only started to be applied to vineyards on complex landscapes and difficult topographies. To provide better management of vineyards and other cultures, new services are under development around the world. For example, a service for the automatic collection of data related to culture for automatic spotting, designation of zones, and prediction of the evolution of a zone is in preparation. Currently, projects are based on satellites providing high quality images to automatically detect specific zones and generate advanced geographical data. Today, image recognition field massively uses Machine Learning (ML) algorithms, for online images matching or satellite image mapping for example. The current research is based on the needs above and focuses on the detection of the vine and potential disease on images taken by Unmanned Aerial Vehicles (UAV). UAV are used for multiple reasons such as the price and the accessibility of this type of product, compared to satellites. With an auto generated flight plan, the drone can fly over a specifically defined zone of the field, carrying products and spraying it out to the designated area. The present paper will describe how the application of Machine Learning to images taken by a UAV improves the overall performance of automated drone plan generation.

## LITERATURE SURVEY

### 1) **Agricultural Drones Relatively cheap drones with advanced sensors and imaging capabilities are giving farmers new ways to increase yields and reduce crop damage**

AUTHORS:- C. Anderson

Relatively cheap drones with advanced sensors and imaging capabilities are giving farmers new ways to increase yields and reduce crop damage. These aircraft are equipped with an autopilot using GPS and a standard point-and-shoot camera controlled by the autopilot; software on the ground can stitch aerial shots into a high-resolution mosaic map. It is part of a trend toward increasingly data-driven agriculture. Farms today are bursting with engineering marvels, the result of years of automation and other innovations designed to grow more food with less labor. Tractors autonomously plant seeds within a few centimeters of their target locations, and GPS-guided harvesters reap the crops with equal accuracy. Extensive wireless networks backhaul data on soil hydration and environmental factors to faraway servers for analysis.

### 2) **Self-Guided Segmentation and Classification of Multi- Temporal Landsat 8 Images for Crop Type Mapping in Southeastern Brazil**

AUTHORS:- B. Schultz, M. Formaggio, A. R. Del' Arco Sanchez, L. Barreto Luiz and C.N Atzberger,

Only well-chosen segmentation parameters ensure optimum results of object-based image analysis (OBIA). Manually defining suitable parameter sets can be a time-consuming approach, not necessarily leading to optimum results; the subjectivity of the manual approach is also obvious. For this reason, in supervised segmentation as proposed by Stefanski et al. (2013) one integrates the segmentation and classification tasks. The segmentation is optimized directly with respect to the subsequent classification. In this contribution, we build on this work and developed a fully autonomous workflow for supervised object-based classification, combining image segmentation and random forest (RF) classification. Starting from a fixed set of randomly selected and manually interpreted training samples, suitable segmentation parameters are automatically identified. A sub-tropical study site located in Sao Paulo State (Brazil) was used to evaluate the proposed approach. Two multi-temporal Landsat 8 image mosaics were used as input (from August 2013 and January 2014) together with training samples from field visits and VHR (RapidEye) photo-interpretation. Using four test sites of  $15 \times 15$  km<sup>2</sup> with manually interpreted crops as independent validation samples, we demonstrate that the approach leads to robust classification results. On these samples (pixel wise,  $n \approx 1$  million) an overall accuracy (OA) of 80% could be reached while classifying five

classes: sugarcane, soybean, cassava, peanut and others. We found that the overall accuracy obtained from the four test sites was only marginally lower compared to the out-of-bag OA obtained from the training samples. Amongst the five classes, sugarcane and soybean were classified best, while cassava and peanut were often misclassified due to similarity in the spatio-temporal feature space and high within-class variabilities. Interestingly, misclassified pixels were in most cases correctly identified through the RF classification margin, which is produced as a by-product to the classification map.

### **3) A Comprehensive Review on Pixel Oriented and Object Oriented Methods for Information Extraction from Remotely Sensed Images with a Special Emphasis on Cryospheric Applications**

**AUTHORS:-** S. D. Jawak, P. Devliyal and A. J. Luis

Image classification is one of the most basic operations of digital image processing. The present review focuses on the strengths and weaknesses of traditional pixel-based classification (PBC) and the advances of object-oriented classification (OOC) algorithms employed for the extraction of information from remotely sensed satellite imageries. The state-of-the-art classifiers are reviewed for their potential usage in urban remote sensing (RS), with a special focus on cryospheric applications. Generally, classifiers for information extraction can be divided into three catalogues: 1) based on the type of learning (supervised and unsupervised), 2) based on assumptions on data distribution (parametric and non-parametric) and, 3) based on the number of outputs for each spatial unit (hard and soft). The classification methods are broadly based on the PBC or the OOC approaches. Both methods have their own advantages and disadvantages depending upon their area of application and most importantly the RS datasets that are used for information extraction. Classification algorithms are variedly explored in the cryosphere for extracting geospatial information for various logistic and scientific applications, such as to understand temporal changes in geographical phenomena. Information extraction in cryospheric regions is challenging, accounting to the very similar and conflicting spectral responses of the features present in the region. The spectral responses of snow and ice, water, and blue ice, rock and shadow are a big challenge for the pixel-based classifiers. Thus, in such cases, OOC approach is superior for extracting information from the cryospheric regions. Also, ensemble classifiers and customized spectral index ratios (CSIR) proved extremely good approaches for information extraction from cryospheric regions. The present review would be beneficial for developing new classifiers in the cryospheric environment for better understanding of spatial-temporal changes over long time scales.

#### 4) **Segmentation Performance Evaluation for Object Based Remotely Sensed Image Analysis**

**AUTHORS:-** P. Corcoran, A. Winstanley and P. Mooney

Remote sensing imagery needs to be converted into tangible information which can be utilised in conjunction with other data sets, often within widely used Geographic Information Systems (GIS). As long as pixel sizes remained typically coarser than, or at the best, similar in size to the objects of interest, emphasis was placed on per-pixel analysis, or even sub-pixel analysis for this conversion, but with increasing spatial resolutions alternative paths have been followed, aimed at deriving objects that are made up of several pixels. This paper gives an overview of the development of object based methods, which aim to delineate readily usable objects from imagery while at the same time combining image processing and GIS functionalities in order to utilize spectral and contextual information in an integrative way. The most common approach used for building objects is image segmentation, which dates back to the 1970s. Around the year 2000 GIS and image processing started to grow together rapidly through object based image analysis (OBIA - or GEOBIA for geospatial object based image analysis). In contrast to typical Landsat resolutions, high resolution images support several scales within their images. Through a comprehensive literature review several thousand abstracts have been screened, and more than 820 OBIA-related articles comprising 145 journal papers, 84 book chapters and nearly 600 conference papers, are analysed in detail. It becomes evident that the first years of the OBIA/GEOBIA developments were characterised by the dominance of ‘grey’ literature, but that the number of peer-reviewed journal articles has increased sharply over the last four to five years. The pixel paradigm is beginning to show cracks and the OBIA methods are making considerable progress towards a spatially explicit information extraction workflow, such as is required for spatial planning as well as for many monitoring programmes.

## **1.1 ORGANIZATION PROFILE**

LETSLEARN.TECH located at NLR, has a rich background in developing academic student projects, especially in solving latest IEEE Papers, Software Development and continues its entire attention on achieving transcending excellence in the Development and Maintenance of Software Projects and Products in Many Areas.

In Today's Modern Technological Competitive Environment, Students in Computer Science Stream Want To Ensure That They Are Getting Guidance In An Organization That Can Meet Their Professional Needs. With Our Well-Equipped Team of Solid Information Systems Professionals, Who Study, Design, Develop, Enhance, Customize, Implement, Maintain and Support Various Aspects of Information Technology, Students Can Be Sure.

We Understand the Students' Needs, And Develop Their Quality Of Professional Life By Simply Making The Technology Readily Usable For Them. We Practice Exclusively in Software Development, Network Simulation, Search Engine Optimization, Customization And System Integration. Our Project Methodology Includes Techniques For Initiating A Project, Developing The Requirements, Making Clear Assignments To The Project Team, Developing A Dynamic Schedule, Reporting Status To Executives And Problem Solving.

The indispensable factors, which give the competitive advantages over others in the market, may be slated as:

- Performance
- Pioneering efforts
- Client satisfaction
- Innovative concepts
- Constant Evaluations
- Improvisation
- Cost Effectiveness

## **1.2 ABOUT THE PEOPLE:**

As a team we have the clear vision and realize it too. As a statistical evaluation, the team has more than 15,000 hours of expertise in providing real-time solutions in the fields of Android Mobile Apps Development, Networking, Web Designing, Secure Computing, Mobile Computing, Cloud Computing, Image Processing And Implementation, Networking With OMNET++ Simulator, client

Server Technologies in Java,(J2EE\J2ME\EJB), ANDROID, DOTNET (ASP.NET, VB.NET, C#.NET), MATLAB, NS2, SIMULINK, EMBEDDED, POWER ELECTRONICS, VB & VC++, Oracle and operating system concepts with LINUX.

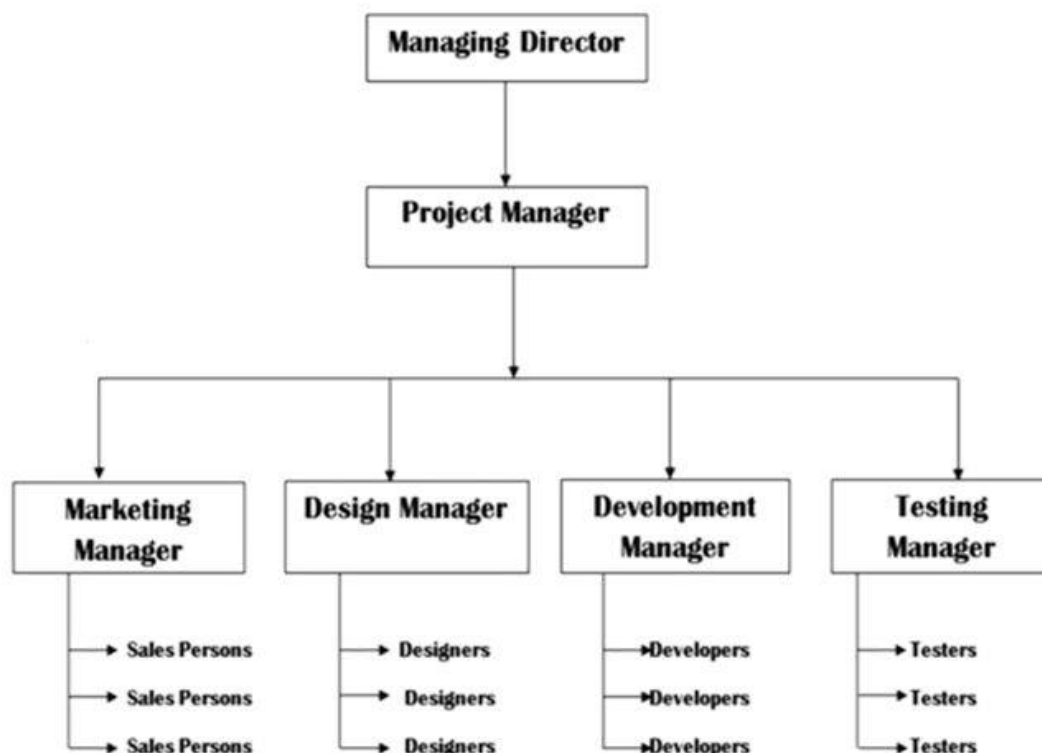
## OUR VISION:

**“WE CAN DEVELOP YOUR OWN IDEAS”** this is our vision; we work according to our vision.

## Customer Focus

We view our customer relationships as partnerships and are committed to being accountable to ensure that the development, integration and implementation of solutions are performed in a professional and timely manner. We unite this accountability with our dedication to applying the most appropriate methodologies & technologies, and we dispatch our 'best-of-breed' technology professionals to make it all happen. The result is satisfied customers who consistently give us high marks for our expansive offerings of precise engineering.

## Organization Chart:



## **CHAPTER-2**

### **SYSTEM STUDY AND ANALYSIS**

#### **2.1 EXISTING SYSTEM:**

The methodology utilizes Principal Components Analysis (PCA) not withstanding Support Vector Machines (SVMs) and Linear Discriminate Analysis (LDA) algorithms for feature extraction and arrangement. We demonstrated that machine

Learning algorithms dependent on morpho-colorimetric parameters and NIR examination independently were capable to classify leaves of 16 grapevine cultivars. Automated picture analysis for morphological and color highlights extraction of checked leaves combined with ANN displaying rendered fast, exact and modest techniques to be utilized for ampelo graphy/ cultivar characterization.

#### **Disadvantages of Existing System:**

- 1) PCA not supporting to calculate accuracy for SVM and LDA algorithm the main problem is PCA doesnot supporting more than 10 components from the given dataset.
- 2) Existing model does not support to classify leaves of 16 and also model is not trained to predict and detect disease in leaves and plants so that ANN algorithm failed to resolve problem in Agriculture Field.

#### **2.2 PROPOSED SYSTEM:**

The farmers rely on the traditional ways of farming which is based on the reliability of the suggestions from the elderly and their experience. This method leaves farmers at the mercy of random climatic conditions which are already getting random due to global warming and uneven rainfall patterns. The manual spraying method for pesticides led to improper usage of resources and harms the environment.

#### **Advantages of proposed System:**

- 1) To resolve the problems in agriculture such as lack of water, excess rain, soil pollution due to plastics, synthetic fertilizer we applied selected dataset with decision tree, Random forest , Support vector Machine Learning,
- 2) It is imperative to monitor the pH value, temperature and concentration changes in nutrient solution composition as the performance of soilless cultivation is highly dependent on these parameters. The significant variables in a nutrient solution cannot be measured directly hence these are determined with the help of auxiliary variables.

## **IMPLEMENTATION:**

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

### **Rule-based features:**

Human experts with years of experience created many rules to detect whether a user submitted YouTube comment is spam or not. An example of such rules is “blacklist”, i.e. whether the user has been detected or complained as spam before. Each rule can be regarded as a binary feature that indicates the fraud likeliness.

### **Selective labeling:**

If the spam score is above a certain threshold, the case will enter a queue for further investigation by human experts. Once it is reviewed, the final result will be labeled as Boolean, i.e. spam or clean. Cases with higher scores have higher priorities in the queue to be reviewed. The cases whose spam score are below the threshold are determined as clean by the system without any human judgment.

### **Spam churn:**

Once one case is labeled as spam by human experts, it is very likely that the reviewer is not trustable and may be also commenting other spam; hence all the spam submitted by the same reviewer are labeled as spam too. The fraudulent spammer along with his/her cases will be removed from the website immediately once detected.

## **2.3 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY



### **2.3.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **2.3.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### **2.3.2 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 level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

## **2.4 FUNCTIONAL REQUIREMENTS**

### **Admin**

It is used by the admin for activating and deactivating user request.

### **User**

It is used to add document into database or used to download document from database.

### **Modular Manager**

It is used to accept the document that is to be upload

## 2.5 NON-FUNCTIONAL REQUIREMENTS:

Non-Functional Requirements (quality attributes) ensure the delivery of an operable and manageable system which provides the required functionality reliable, uninterrupted or with minimal time of interruption even under unusual situations.

- **Security**

Login requirements - access levels, CRUD levels

Password requirements - length, special characters, expiry, recycling policies Inactivity timeouts – durations, actions

- **Audit**

Audited elements – what business elements will be audited?

Audited fields – which data fields will be audited?

Audit file characteristics - before image, after image, user and time stamp, etc

- **Performance**

Response times - application loading, screen open and refresh times, etc Processing times – functions, calculations, imports, exports

Query and Reporting times – initial loads and subsequent loads Capacity.

- **Availability**

Hours of operation – when is it available? Consider weekends, holidays, maintenance times, etc

Locations of operation – where should it be available from, what are the connection requirements?

- **Reliability**

Mean Time Between Failures – What is the acceptable threshold for down-time? e.g. One a year, 4,000 hours

Mean Time To Recovery – if broken, how much time is available to get the system back up again?

- **Integrity**

Fault trapping (I/O) – how to handle electronic interface failures, etc

Bad data trapping - data imports, flag-and-continue or stop the import policies, etc

Data integrity – referential integrity in database tables and interfaces. Image compression and decompression standards

- **Recovery**

Recovery process – how do recoveries work, what is the process?

Recovery time scales – how quickly should a recovery take to perform?

Backup frequencies – how often is the transaction data, set-up data, and system (code) backed-up?

- **Compatibility**

Compatibility on different operating systems – What does it have to be able to run on?

Compatibility on different platforms – What are the hardware platforms it needs to work on?

- **Maintainability**

Conformance to architecture standards – What are the standards it needs to conform to or have exclusions from?

Conformance to design standards – What design standards must be adhered to or exclusions created?

Conformance to coding standards – What coding standards must be adhered to or exclusions created?

- **Usability**

Look and feel standards - screen element density, layout and flow, colours, UI metaphors, keyboard shortcuts

Internationalization / localization requirements – languages, spellings, keyboards, paper sizes, etc

- **Documentation**

Required documentation items and audiences for each item

## **CHAPTER-3**

### **DEVELOPMENT ENVIRONMENT**

#### **3.1 REQUIREMENT ANALYSIS:**

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

##### **Software Requirements:**

- OS : Windows 10
- Python IDE : python 3.7.x and above  
: Pycharm IDE, Jupyter notebook
- setup tools and pip to be installed for 3.7.x and above

##### **Hardware Requirements:**

- RAM : 4GB and Higher
- Processor : Intel i3 and above
- Hard Disk : 500GB: Minimum

#### **3.2 REQUIREMENT SPECIFICATION**

##### **Functional Requirements**

- Graphical User interface with the User.

##### **Software Requirements**

For developing the application the following are the Software Requirements:

1. Python
2. Django

## **Operating Systems supported**

1. Windows 7
2. Windows XP
3. Windows 8
4. Windows 10

## **Technologies and Languages used to Develop**

1. Python
2. Jupyter Notebook
3. Pycharm

## **Debugger and Emulator**

- Any Browser (Particularly Chrome)

## **Hardware Requirements**

For developing the application the following are the Hardware Requirements:

- Processor: Core I5 or higher
- RAM: 8 GB
- Space on Hard Disk: minimum 500 GB

## **3.3 ABOUT PYTHON TECHNOLOGY:**

Python is an open source, high-level programming language developed by Guido van Rossum in the late 1980s and presently administered by Python Software Foundation. It came from the ABC language that he helped create early on in his career. Python is a powerful language that you can use to create games, write GUIs, and develop web applications.

It is a high-level language. Reading and writing codes in Python is much like reading and writing regular English statements. Because they are not written in machine-readable language, Python programs need to be processed before machines can run them. Python is an interpreted language. This means that every time a program is run, its interpreter runs through the code and translates it into machine readable byte code.

Python is an object-oriented language that allows users to manage and control data structures or objects to create and run programs. Everything in Python is, in fact, first class. All objects, data types, functions, methods, and classes take equal position in Python. Programming languages are created to satisfy the needs of programmers and users for an effective tool to develop applications that impact lives, lifestyles, economy, and society. They help make lives better by increasing productivity, enhancing communication, and improving efficiency. Languages die and become obsolete when they fail to live up to expectations and are replaced and superseded by languages that are more powerful. Python is a programming language that has stood the test of time and has remained relevant across industries and businesses and among programmers, and individual users. It is a living, thriving, and highly useful language that is highly recommended as a first programming language for those who want to dive into and experience programming. Advantages of Using Python Here are reasons why you would prefer to learn and use Python over other high-level languages.

### **Readability**

Python programs use clear, simple, and concise instructions that are easy to read even by those who have no substantial programming background. Programs written in Python are, therefore, easier to maintain, debug, or enhance.

### **Higher productivity**

Codes used in Python are considerably shorter, simpler, and less verbose than other high-level programming languages such as Java and C++. In addition, it has well-designed built-in features and standard library as well as access to third party modules and source libraries. These features make programming in Python more efficient.

### **Less learning time**

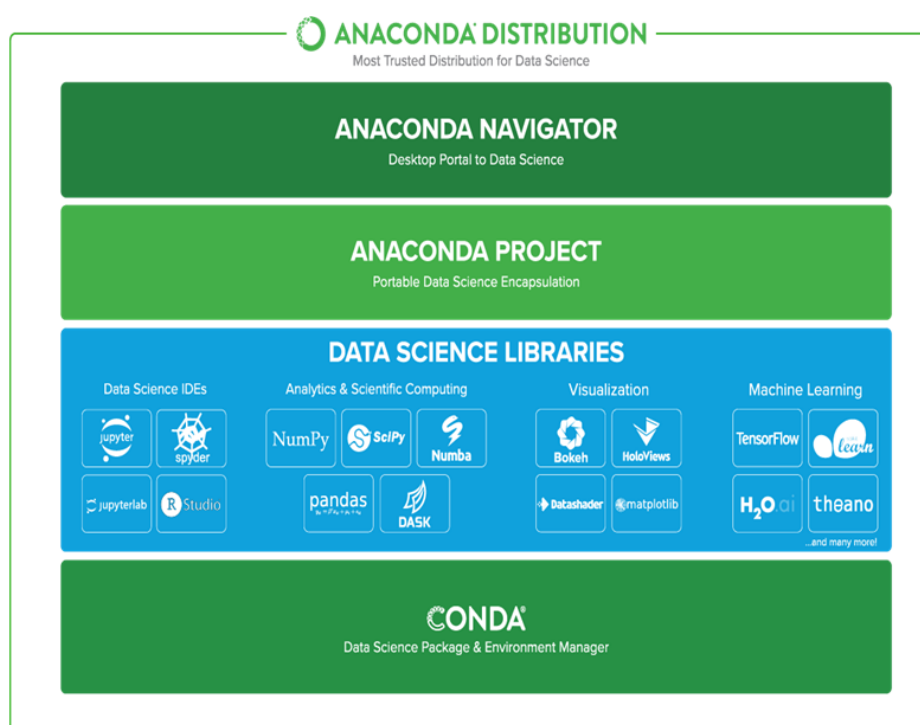
Python is relatively easy to learn. Many find Python a good first language for learning programming because it uses simple syntax and shorter codes. Python works on Windows, Linux/UNIX, Mac OS X, other operating systems and small form devices. It also runs on microcontrollers used in appliances, toys, remote controls, embedded devices, and other similar devices.

**Installing Python in Windows:** To install Python, you must first download the installation package of your preferred version from this link: <https://www.python.org/downloads/> On this page, you will be asked to choose between the two latest versions for Python 2 and 3: Python 3.5.1 and Python 2.7.11. Alternatively, if you are looking for a specific release, you can scroll down the page to find

download links for earlier versions. You would normally opt to download the latest version, which is Python 3.5.1. This was released on December 7, 2015. However, you may opt for the latest version of Python 2, 2.7.11. Your preferences will usually depend on which version will be most usable for your project. While Python 3 is the present and future of the language, issues such as third-party utility or compatibility may require you to download Python 2.

## ANACONDA:

Anaconda is a complete, open-source data science package with a community of over 6 million users. It is easy to download and install, and it is supported on Linux, MacOS, and Windows. The distribution comes with more than 1,000 data packages as well as the Anaconda package and virtual environment manager, so it eliminates the need to learn to install each library independently. As Anaconda's website says, "The Python and Rconda packages in the Anaconda Repository are curated and compiled in our secure environment so you get optimized binaries that 'just work' on your system".



**Fig: 3.1** Anaconda Distribution

## What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage anaconda packages, environments, and channels without using command-line commands. Navigator can search for



packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

### **Why use Navigator?**

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.

The command-line program `anaconda` is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly. Navigator is an easy, point-and-click way to work with packages and environments without needing to type `anaconda` commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

### **What applications can we access using Navigator?**

The following applications are available by default in Navigator:

- Jupyter Notebook
  - Spyder
  - PyCharm
  - VSCode
  - Glueviz
  - Orange3 App
  - RStudio
  - Anaconda Prompt (Windows only)
  - Anaconda Power Shell (Windows only)
  - Jupyter Lab
- 
- **Jupyter Lab:** This is an extensible working environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

- **Qt Console:** It Is the PyQtGUI that supports in line figures, proper multi line editing with syntax highlighting, graphical call tips and more.
- **Spyder:** Spyder is a scientific Python Development Environment. It is a powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.
- **VSCode:** It is a stream lined coded it or with support for development operations like debugging, task running and version control.
- **Glueviz:** This is used for multi-dimensional data visualization acrossfiles. It explores relationships within and among related datasets.
- **Orange3:** It is a component-based datamining framework. This can be used for data visualization and data analysis. The workflows in Orange 3 are very interactive and provide a large tool box.
- **Rstudio:**It is a set of integrated tools designed to help youtube more productive with R. It includes R essentials and notebooks
- **JupyterNotebook:** This is a web-based, interactive computing notebook environment. We can edit and run human-readable docs while describing the data analysis.
- The Jupyter Notebook is an open-source web application that you can use to create and shared documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are as pin-off project from the I Python project, which used to have an I Python Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the I Python kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use. The Jupyter Notebook is not included with Python, so if you an to try it out, you will need to install Jupyter. There are many distributions of the Python language. This article will focus on just two of them for the purposes of installing Jupyter Notebook. The most popular is CPython, which is the reference version of Python that you can change from their website. It is also assumed that you are using Python.

**PyCharm:** It is the most popular IDE for Python, and includes great features such as excellent code completion and inspection with advanced debugger and support for web programming and various frameworks. PyCharm is created by Czech company, Jet brains which focusses on creating integrated development environment for various web development languages like JavaScript and PHP. PyCharm offers some of the best features to its users and developers in the following aspects

- Code completion and inspection.

- Advanced debugging.
- Support for web programming and frame works such as Django and Flask.

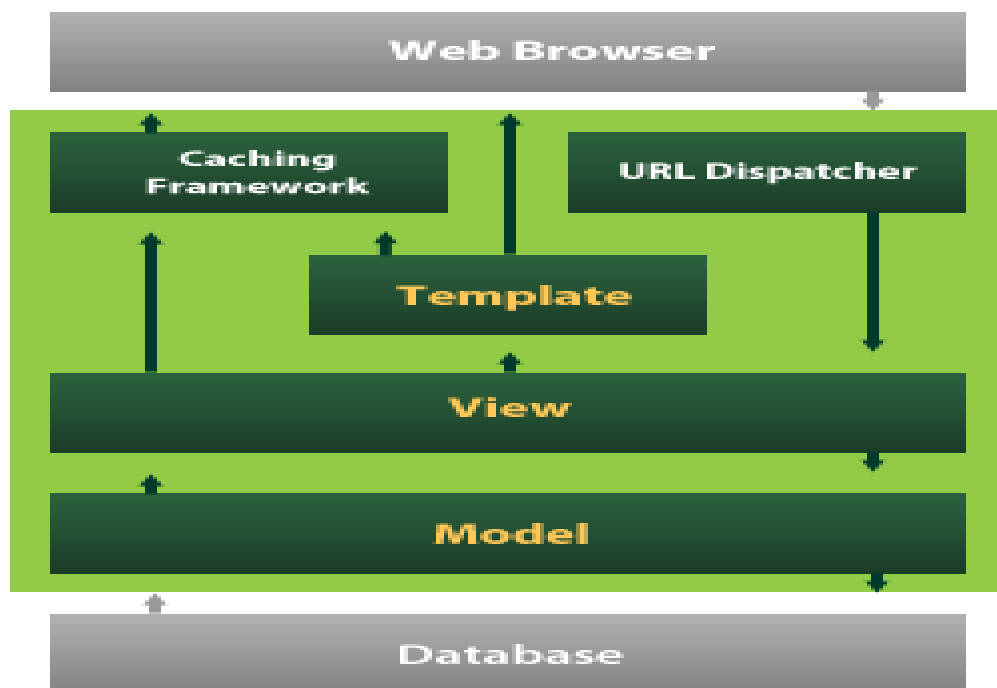
## Features of PyCharm

Besides, a developer will find PyCharm comfortable to work with because of the features mentioned below—

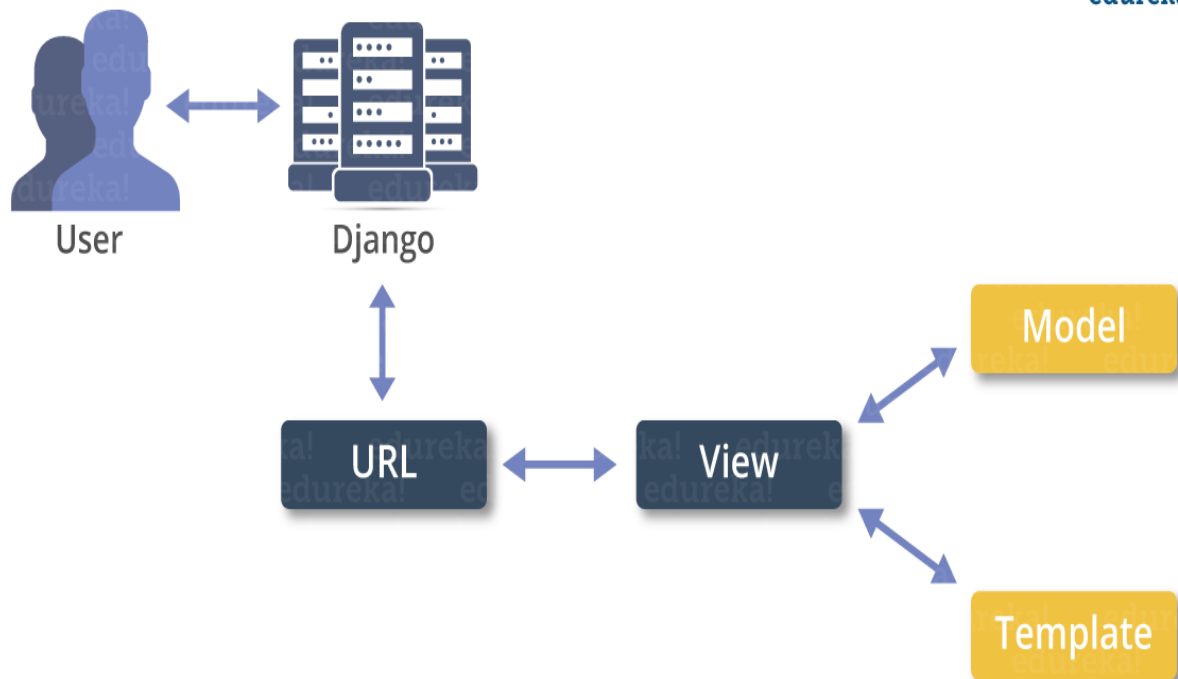
- **Code Completion:** PyCharm enables smoother code completion when it is for built in or for an external package.
- **SQL Alchemy as Debugger:** You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.
- **Git Visualization in Editor:** When coding in Python, queries are normal for a developer. You can check the last commit easily in PyCharm as it has the blue sections that can define the difference between the last commit and the current one.
- **Code Coverage in Editor:** You can run .py files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.
- **Package Management:** All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.
- **LocalHistory** is always keeping track of the changes in a way that complements like Git. Local history in PyCharm gives complete details of what is needed to rollback and what is to be added.
- **Refactoring** is the process of renaming one or more files at a time and PyCharm includes various shortcuts for a smooth refactoring process.
- **WampServer:** WAMPs are packages of independently-created programs installed on computers that use a Microsoft Windows operating system. Apache is a web server. MySQL is an open-source database. PHP is a scripting language that can manipulate information held in a database and generate web pages dynamically each time content is requested by a browser. Other programs may also be included in a package, such as php My Admin which provides a graphical user interface for the MySQL database manager, or the alternative scripting languages Python or Perl.

## DJANGO

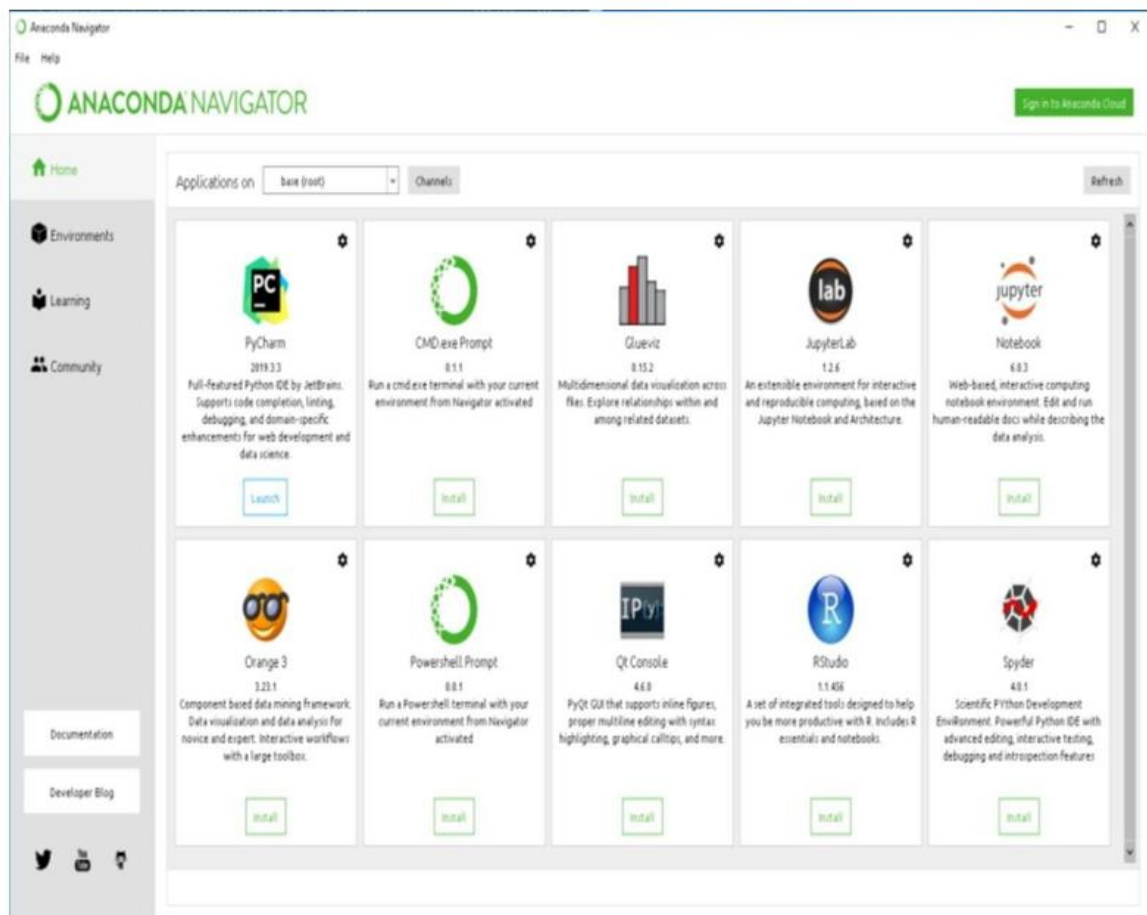
- Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.
- Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.



Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models



**Installation on Windows:** Visit the link <https://www.python.org/downloads/to> to download the latest release of Python. In this process, we will install Python 3.6.7 on our Windows operating system. Double-click the executable file which is downloaded; the following window will open. Select Customize installation and proceed.



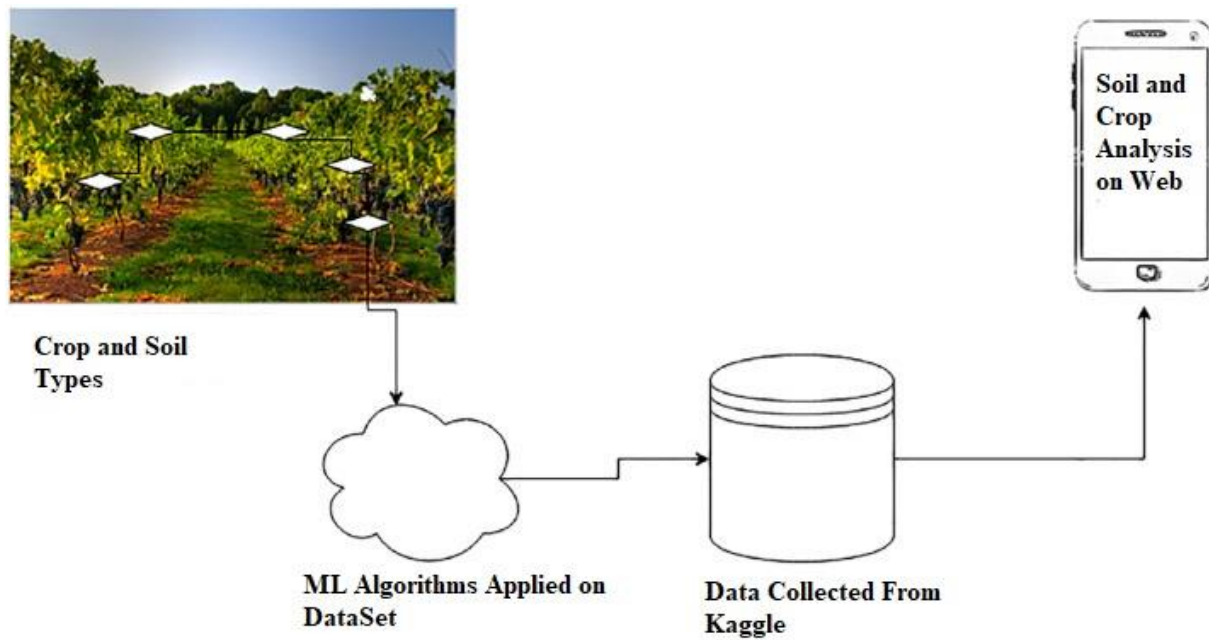
## How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code. You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images, and interactive interfaces into a single notebook file that is edited, viewed, and used in a web browser

## CHAPTER-4

### SYSTEM DESIGN

#### SYSTEM ARCHITECTURE



*Fig: System Architecture*

## **MODULES**

### **Data Collection**

Data collection is a very basic module and the initial step towards the project. It generally deals with the collection of the right dataset. The dataset that is to be used in the market prediction has to be used to be filtered based on various aspects. Data collection also complements to enhance the dataset by adding more data that are external. Our data mainly consists of the previous year stock prices. Initially, we will be analyzing the Kaggle dataset and according to the accuracy, we will be using the model with the data to analyze the predictions accurately.

### **Pre-Processing Data**

pre-processing is a part of data mining, which involves transforming raw data into a more coherent format. Raw data is usually, inconsistent or incomplete and usually contains many errors. The data pre-processing involves checking out for missing values, looking for categorical values, splitting the data-set into training and test set and finally do a feature scaling to limit the range of variables so that they can be compared on common environs.

### **Training the Machine**

The machine is similar to feeding the data to the algorithm to touch up the test data. The training sets are used to tune and fit the models. The test sets are untouched, as a model should not be judged based on unseen data. The training of the model includes cross-validation where we get a well-grounded approximate performance of the model using the training data. Tuning models are meant to specifically tune the hyperparameters like the number of trees in a random forest. We perform the entire cross-validation loop on each set of hyperparameter values. Finally, we will calculate a cross-validated score, for individual sets of hyperparameters. Then, we select the best hyperparameters. The idea behind the training of the model is that we some initial values with the dataset and then optimize the parameters which we want to in the model. This is kept on repetition until we get the optimal values. Thus, we take the predictions from the trained model on the inputs from the test dataset. Hence, it is divided in the ratio of 80:20 where 80% is for the training set and the rest 20% for a testing set of the data.

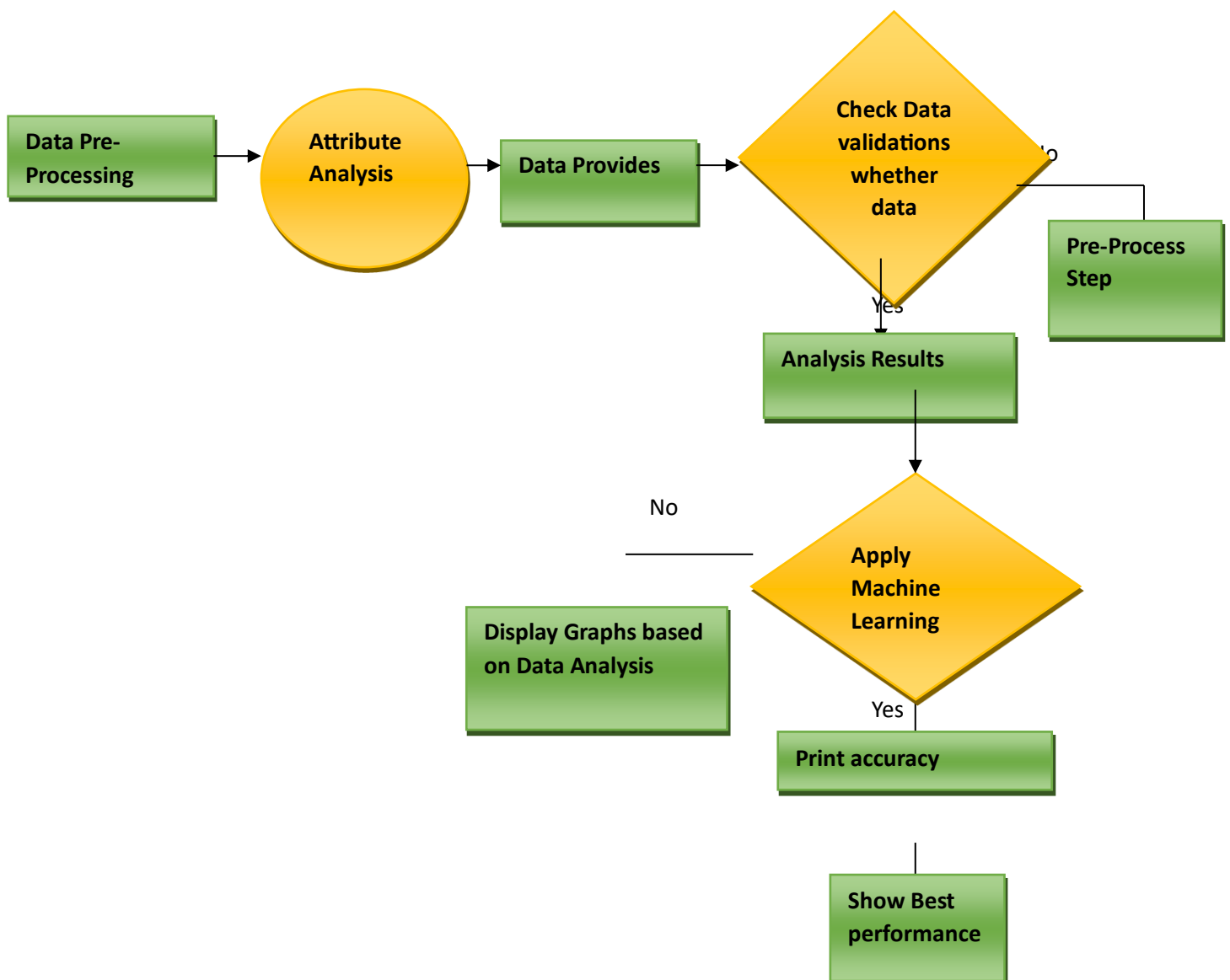


## **Data Scoring**

The process of applying a predictive model to a set of data is referred to as scoring the data. The technique used to process the dataset is the Random Forest Algorithm. Random forest involves an ensemble method, which is usually used, for classification and as well as regression. Based on the learning models, we achieve interesting results. The last module thus describes how the result of the model can help to predict the probability of a stock to rise and sink based on certain parameters. It also shows the vulnerabilities of a particular stock or entity. The user authentication system control is implemented to make sure that only the authorized entities are accessing the results.

#### 4.1 DATA FLOW DIAGRAM:

The data flow diagram (DFD) is one of the most important tools used by system analysis. Data flow diagrams are made up of number of symbols, which represents system components. Most data flow modeling methods use four kinds of symbols: Processes, Data stores, Data flows and external entities. These symbols are used to represent four kinds of system components. Circles in DFD represent processes. Data Flow represented by a thin line in the DFD and each data store has a unique name and square or rectangle represents external entities name and square or rectangle represents external entities.



## 4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

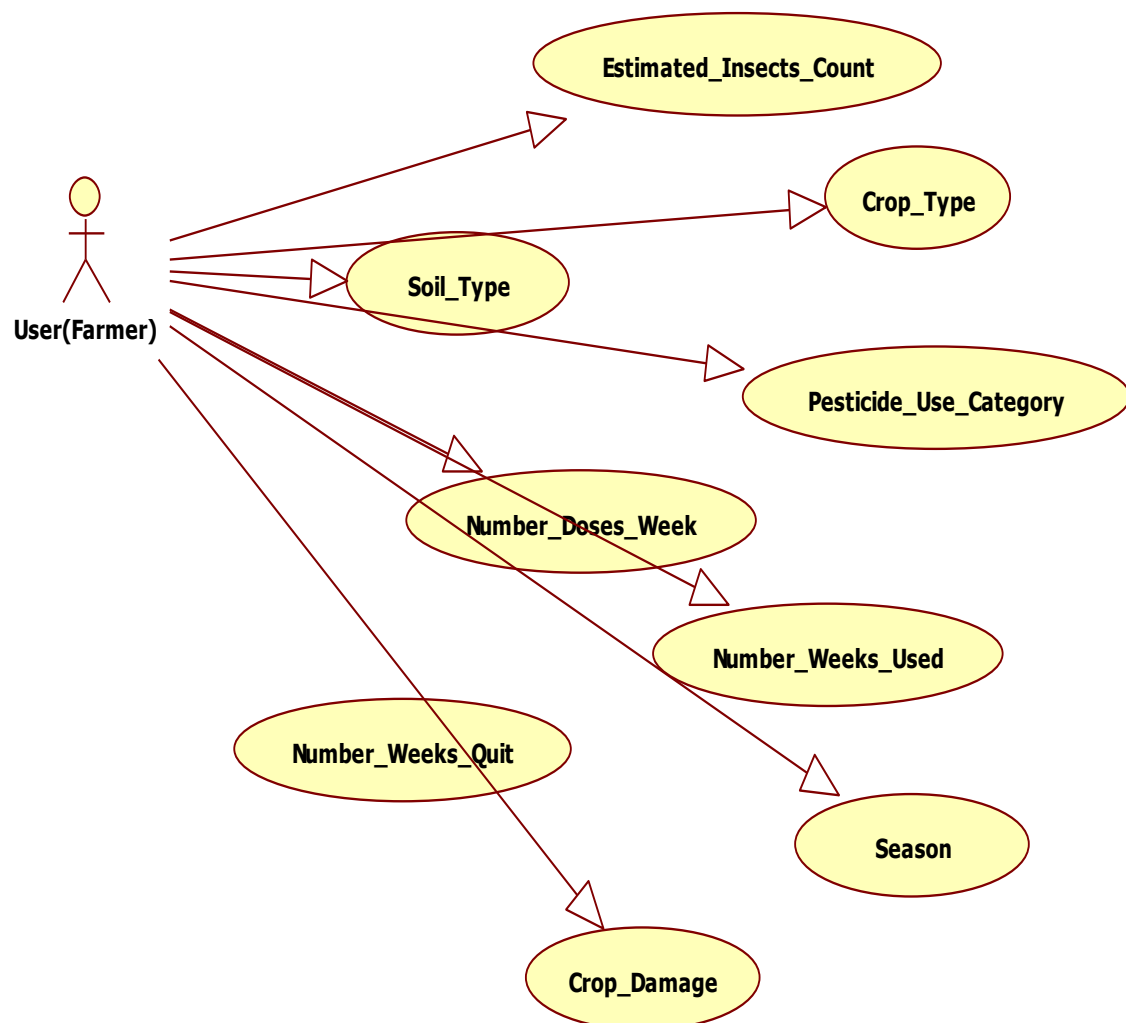
### GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

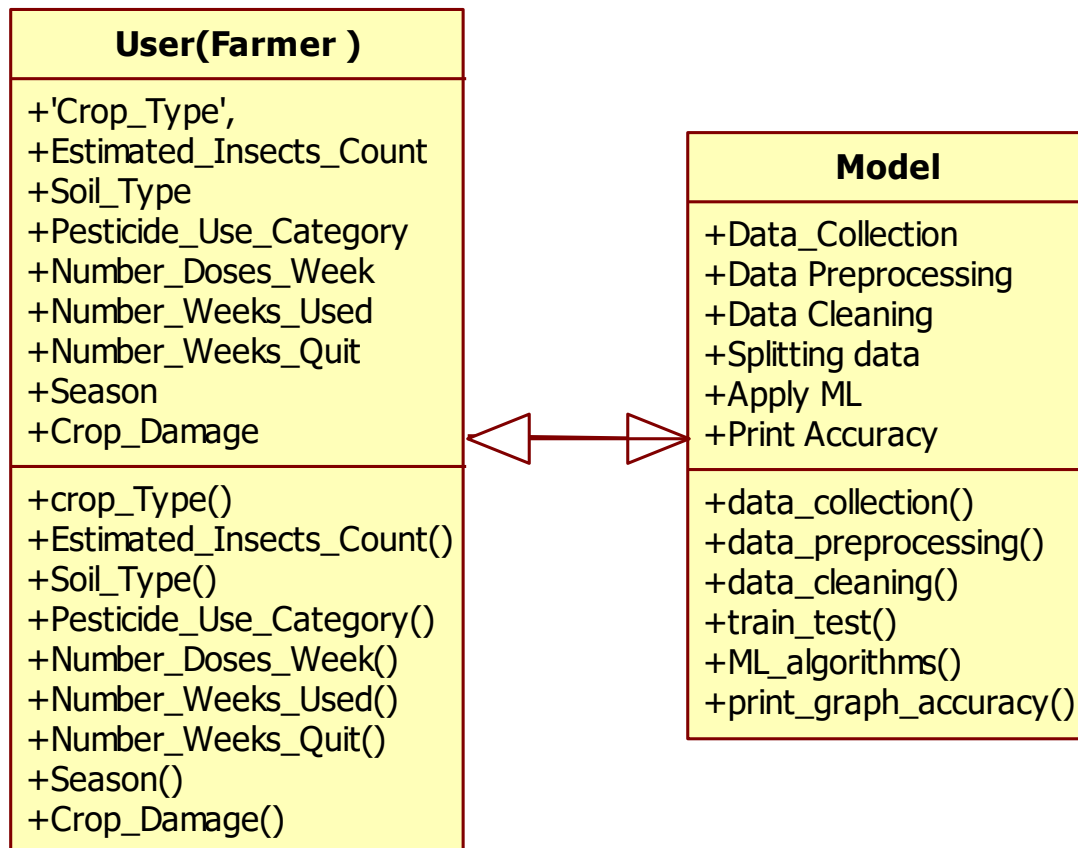
## USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



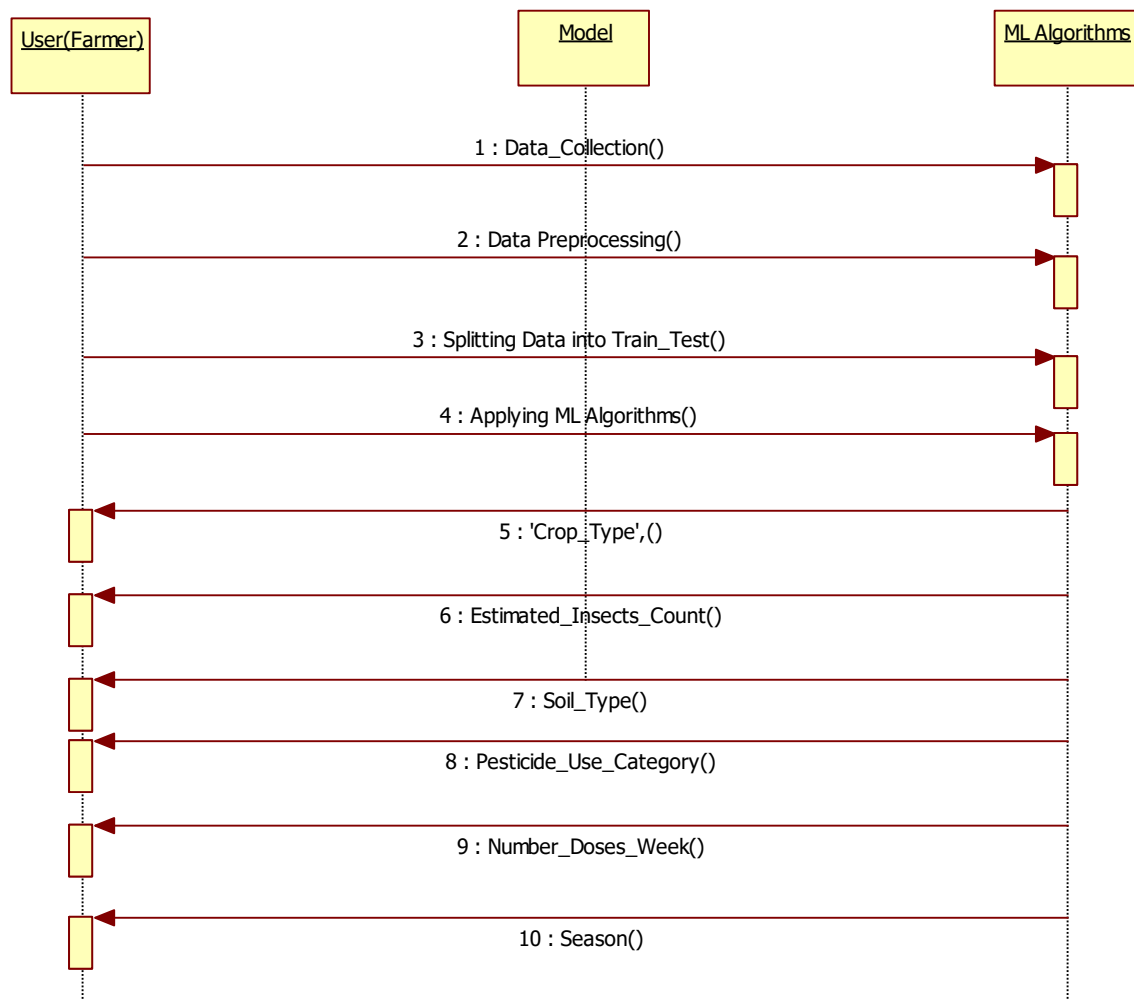
## CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



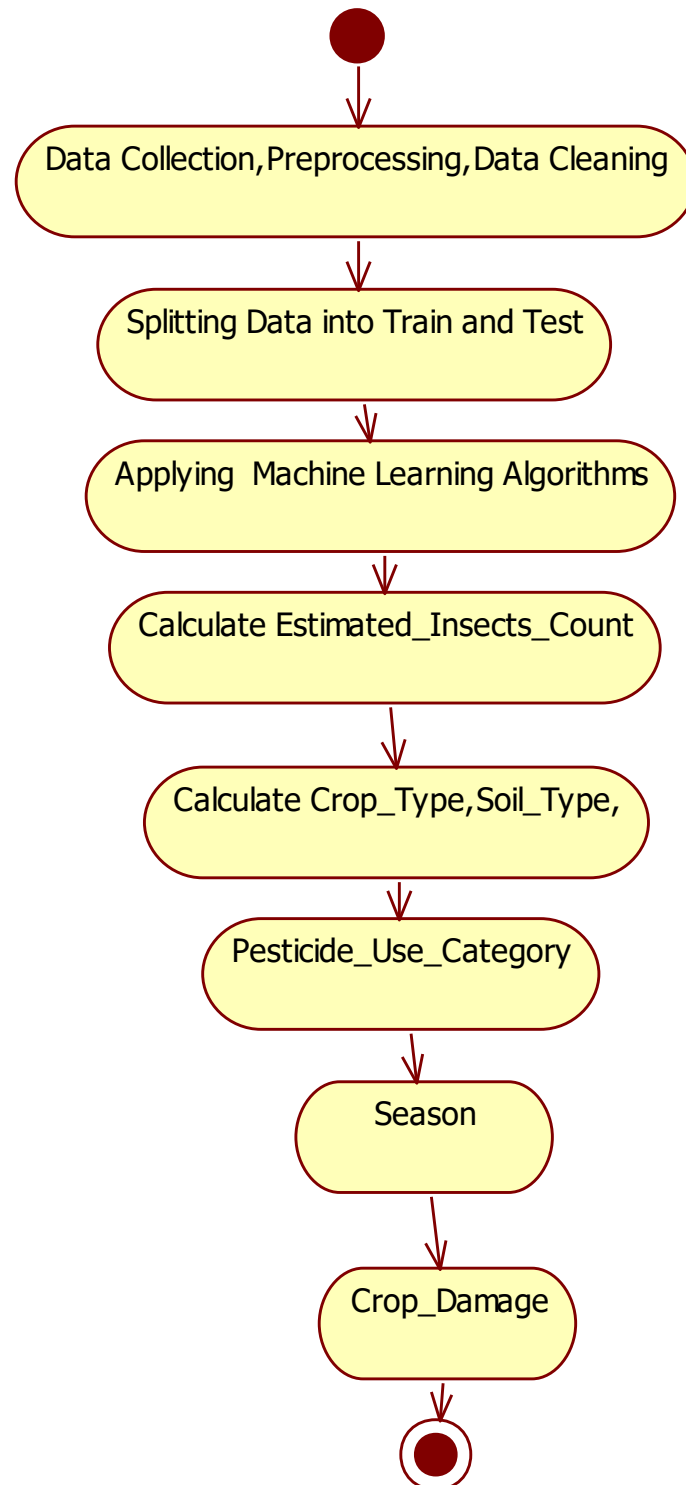
## SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

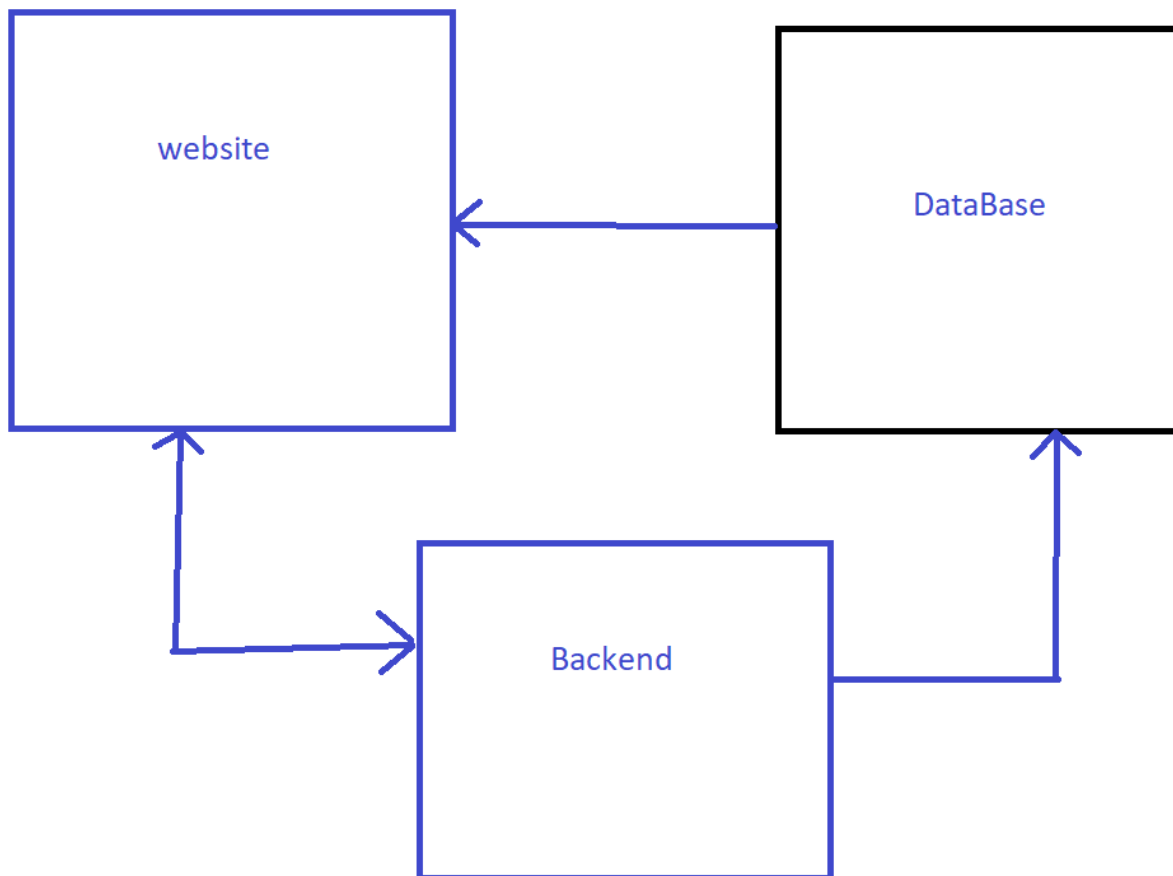


## ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**TOTAL FLOW:**





## **INPUT AND OUTPUT DESIGN:**

### **INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

### **OBJECTIVES**

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

## OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

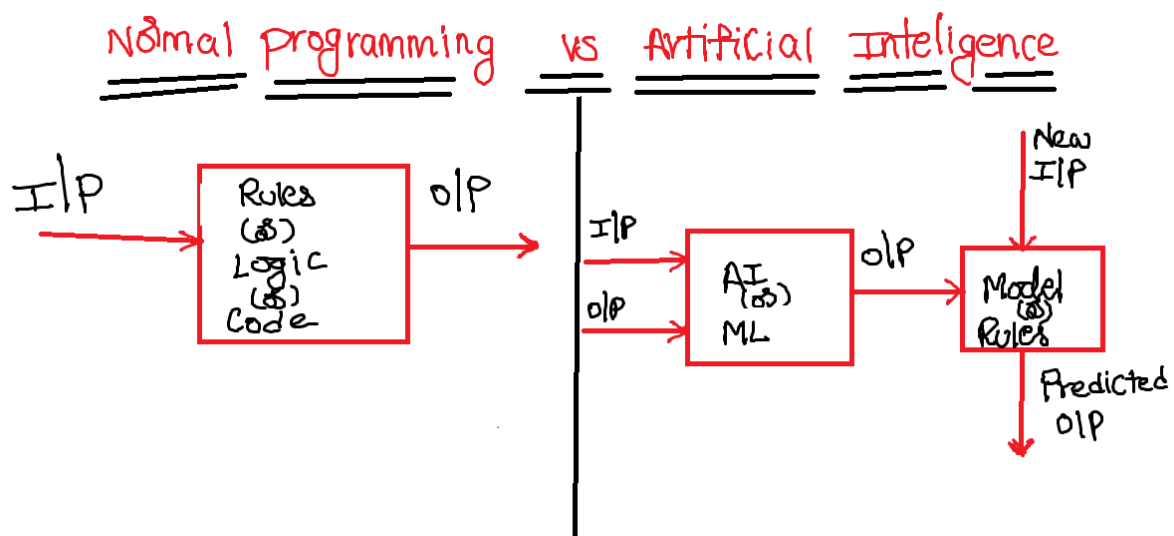
1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.



## **IMPLEMENTATION**

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

### **Pixel-based and Object-oriented classification:**

Image processing most basic operation is image classification. Two classification process are mainly used: (1) the pixel-based classification (PBC) and (2) the object-oriented classification (OOC). The PBC uses conventional statistical techniques. All pixels are categorized into a specific class or thematic. This classification is based on features extracted from the pixel, such as the spectral information and spectral estimated signature. The OOC classifies objects presents on the picture. Spatial spectral features of the high-resolution (HR) satellite data Information is used as the main feature. Last research projects and development for the OOC focuses on the rule-based classifier and the nearest neighbor classifier.

### **Object-based image analysis by application of nonlinear scale-space filtering:**

This research project presents a second powerful and widely used methodology for image recognition. This method analyses images through nonlinear scale-space filtering. Classification of different pictures and various aerial images is provided through a recently developed platform. This framework demonstrates the importance of the qualitative properties and classifier such as multilevel object representation and Support Vector Machine classifier (SVM). Furthermore, it eliminates the need to tune several parameters during the segmentation. These platform algorithms outperform the previous development regarding the final graphical representation and the overall accuracy shows the precision and visualization obtain with the explained framework.

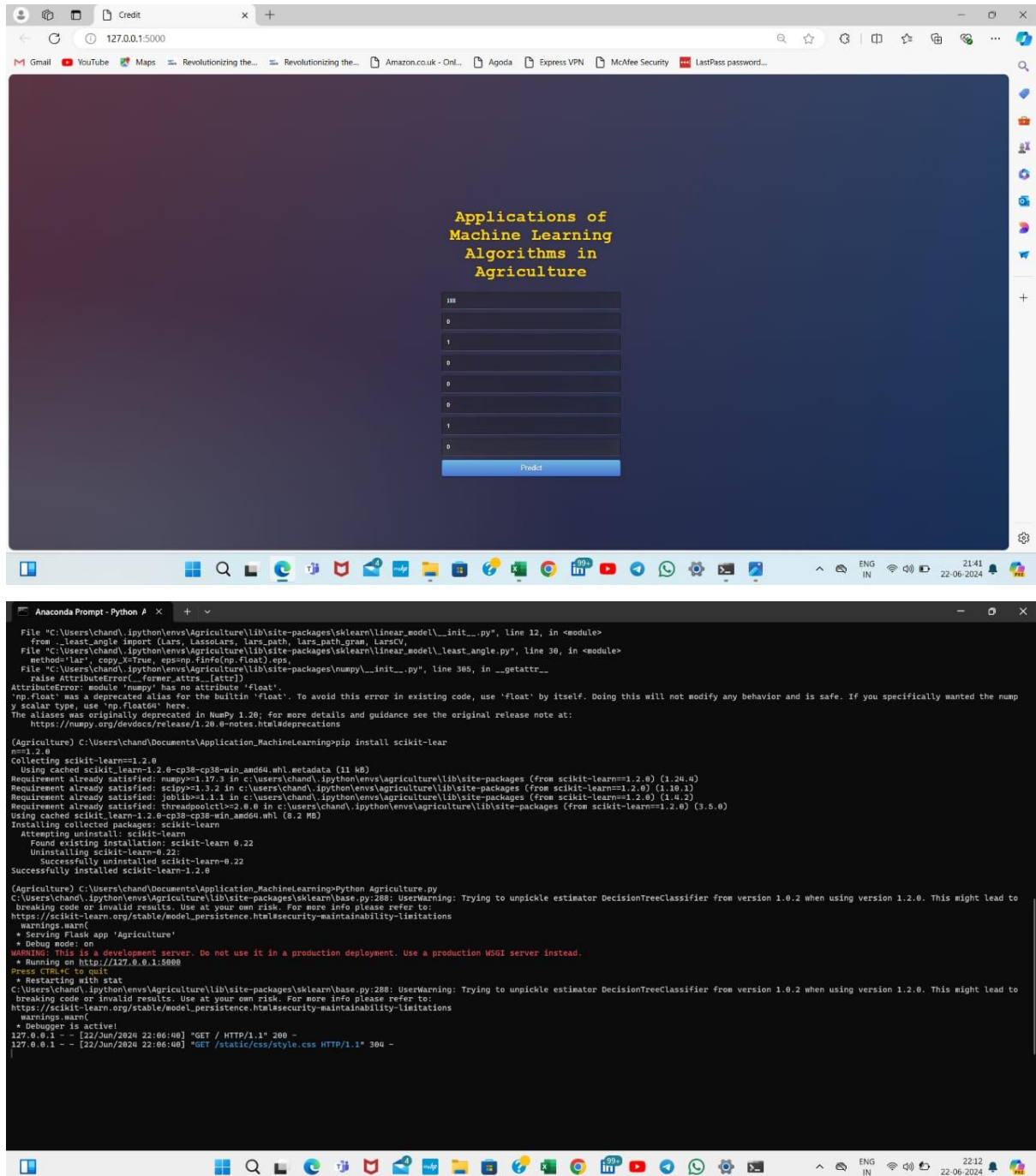
### **Vineyard analysis from very High-Resolution Satellite Data:**

The research focuses on the development of automated and efficient agriculture methods. It is presented as a framework and it is based on multispectral data analysis for object classification. Even if multiple research projects are focusing on aerial images analysis for specific zone

detection because of the complexity of vineyard fields very few of them address this specific use case. The number of analysis regarding precision agriculture is reduced. This complexity depends on:

- The landscape (vines in the mountains for example)
- The paths and roads going through the vineyards
- Leaves and trunks mixed with metal to maintain the vine

## 4.3 USER INTERFACE DESIGN:



Anaconda Prompt - Python 3.11.5

```
(base) C:\Users\chand>conda env list
# conda environments:
#
base                  * C:\Users\chand\.ipython
agriculture            C:\Users\chand\.ipython\envs\agriculture

(base) C:\Users\chand>python -V
Python 3.11.5

(base) C:\Users\chand>Activate Agriculture
(Agriculture) C:\Users\chand>cd Documents
(Agriculture) C:\Users\chand\Documents>cd Application_MachineLearning

(Agriculture) C:\Users\chand\Documents\Application_MachineLearning>python Agriculture.py
C:\Users\chand\.ipython\envs\agriculture\lib\site-packages\sklearn\base.py:288: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.0.2 when using version 1.2.0. This might lead to
breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
 * Serving Flask app 'Agriculture'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
C:\Users\chand\.ipython\envs\agriculture\lib\site-packages\sklearn\base.py:288: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.0.2 when using version 1.2.0. This might lead to
breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
 * Debugger is active!
 * Debugger PIN: 129-385-442

(Agriculture) C:\Users\chand\Documents\Application_MachineLearning>pip install numpy
Requirement already satisfied: numpy in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (1.24.4)

(Agriculture) C:\Users\chand\Documents\Application_MachineLearning>pip install pandas
Requirement already satisfied: pandas in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: numpy>=1.20.3 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from pandas) (1.24.4)
Requirement already satisfied: six>=1.5 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

(Agriculture) C:\Users\chand\Documents\Application_MachineLearning>pip install flask
Requirement already satisfied: flask in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (3.0.3)
Requirement already satisfied: Werkzeug>=3.0.0 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from flask) (3.0.3)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from flask) (3.1.4)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from flask) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from flask) (1.8.2)
Requirement already satisfied: importlib-metadata>=3.6.0 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from flask) (7.1.0)
Requirement already satisfied: colorama in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from click>=8.1.3->flask) (0.4.6)
Requirement already satisfied: zipp>=0.5 in c:\users\chand\.ipython\envs\agriculture\lib\site-packages (from importlib-metadata>=3.6.0->flask) (3.18.2)
```

127.0.0.1:5000/predict

Applications of Machine Learning Algorithms in Agriculture

Estimated_Insects_Count
Soil_Type
Pesticide_Use_Category
Number_Dewey_Week
Number_Weeds_Loud
Number_Weeds_Out
Season
Crop_Damage
Predict

You Can Proceed Food Relevant Crops

```
Agriculture.py
File Edit View

# -*- coding: utf-8 -*-
import numpy as np
import pandas as pd
from flask import Flask, request, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('Agriculture.pk', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    ...
    For rendering results on HTML GUI
    ...
    int_features = [x for x in request.form.values()]
    print(int_features)
    # df1 = pd.DataFrame([int_features])

    final_features = np.array(int_features)

    final_features = final_features.reshape(1,8)
    print(final_features)
    #final_features = pd.DataFrame([final_features])
    print(model.predict(final_features))
    if model.predict(final_features) == [1]:
        predict = "You Can Proceed Food Relevant Crops"
    else:
        predict = "You Can Proceed Horticulture Relevant Crops"

    return render_template('index.html', prediction=predict)

if __name__ == "__main__":
    app.run(debug=True)
# -*- coding: utf-8 -*-

Ln 23, Col 44 1,035 characters 100% Windows (CRLF) UTF-8
```

```
Anaconda Prompt - Python 3.10.12
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\chand\python\envs\agriculture\lib\site-packages (from Jinja2>=3.1.2->Flask) (2.1.5)
(Agriculture) C:\Users\chand\Documents\Application_MachineLearning>pip install scikit-learn==0.22
Collecting scikit-learn==0.22
  Using cached scikit-learn-0.22-cp38-cp38-win_amd64.whl.metadata (7.2 kB)
Requirement already satisfied: numpy>=1.11.0 in c:\users\chand\python\envs\agriculture\lib\site-packages (from scikit-learn==0.22) (1.24.4)
Requirement already satisfied: scipy>=0.17.0 in c:\users\chand\python\envs\agriculture\lib\site-packages (from scikit-learn==0.22) (1.10.1)
Requirement already satisfied: joblib>=0.11 in c:\users\chand\python\envs\agriculture\lib\site-packages (from scikit-learn==0.22) (1.4.2)
Using cached scikit-learn-0.22-cp38-cp38-win_amd64.whl (6.3 MB)
Installing collected packages: scikit-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 1.2.0
    Uninstalling scikit-learn-1.2.0:
      Successfully uninstalled scikit-learn-1.2.0
  Successfully installed scikit-learn-0.22
(Agriculture) C:\Users\chand\Documents\Application_MachineLearning>Python Agriculture.py
Traceback (most recent call last):
  File "Agriculture.py", line 8, in <module>
    model = pickle.load(open('Agriculture.pk', 'rb'))
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\tree\_init_.py", line 6, in <module>
    from .classes import BaseDecisionTree
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\tree\_classes.py", line 40, in <module>
    from ._criterion import Criterion
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\tree\_splitter.py", line 34, in init sklearn.tree._criterion
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\tree\_splitter.py", line 37, in init sklearn.tree._splitter
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\tree\_tree.py", line 56, in init sklearn.tree._tree
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\neighbors\_init_.py", line 17, in <module>
    from ._nca import NeighborhoodComponentAnalysis
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\neighbors\_nca.py", line 22, in <module>
    from ..decomposition import pca
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\decomposition\_init_.py", line 11, in <module>
    from ..sparse_pca import SparsePCA, MiniBatchSparsePCA
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\decomposition\_sparse_pca.py", line 11, in <module>
    from ..linear_model import ridge_regression
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\linear_model\_init_.py", line 12, in <module>
    from ..least_angle import Lars, LarsNet, LarsPath, LarsPathGlm, LarsCV,
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\sklearn\linear_model\_least_angle.py", line 30, in <module>
    methods='lar', copy_X=True, eps=np.finfo(np.float).eps,
  File "C:\Users\chand\python\envs\agriculture\lib\site-packages\numpy\_init_.py", line 365, in __getattr__
    raise AttributeError(f"module 'numpy' has no attribute '{attr}'")
AttributeError: module 'numpy' has no attribute 'float'.
'np.float' was a deprecated alias for the builtin 'float'. To avoid this error in existing code, use 'float' by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use 'np.float64' here.
The aliases was originally deprecated in NumPy 1.20; for more details and guidance see the original release note at:
https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
(Agriculture) C:\Users\chand\Documents\Application_MachineLearning>pip install scikit-learn==1.2.0
Collecting scikit-learn==1.2.0
  Using cached scikit-learn-1.2.0-cp38-cp38-win_amd64.whl.metadata (11 kB)
Requirement already satisfied: numpy>=1.17.3 in c:\users\chand\python\envs\agriculture\lib\site-packages (from scikit-learn==1.2.0) (1.24.4)
Requirement already satisfied: scipy>=1.3.2 in c:\users\chand\python\envs\agriculture\lib\site-packages (from scikit-learn==1.2.0) (1.10.1)
Requirement already satisfied: joblib>=1.1.1 in c:\users\chand\python\envs\agriculture\lib\site-packages (from scikit-learn==1.2.0) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\chand\python\envs\agriculture\lib\site-packages (from scikit-learn==1.2.0) (3.5.0)
```



## **What is SDLC?**

SDLC stands for Software Development Life Cycle. A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

SDLC is the process consisting of a series of planned activities to develop or alter the software products.

## **Benefits of the SDLC Process**

The intent of a SDLC process is to help produce a product that is cost-efficient, effective, and of high quality. Once an application is created, the SDLC maps the proper deployment and decommissioning of the software once it becomes a legacy. The SDLC methodology usually contains the following stages: Analysis (requirements and design), construction, testing, release, and maintenance (response). Veracode makes it possible to integrate automated security testing into the SDLC process through use of its cloud based platform.

### **1. Requirements Gathering:**

In this phase we gather all the requirements from the client, i.e. what are the client expected input, output.....

### **2. Analysis:**

In this phase based upon the client requirements we prepare one documentation is called “High Level Design Document”. It contains Abstract, Functional Requirements, Non Functional Requirements, Existing System, Proposed System, SRS,.....

### **3. Design:**

It is difficult to understand the High Level Design Document for all the members, so to understand easily we use “Low Level Design Document”. To design this document we use UML (Unified Modeling Language). In this we have Use case, Sequence, Collaboration.....

### **4. Coding:**

In this phase we develop the coding module by module. After developing all the modules we integrate them.

### **5. Testing:**

After developing we have to check whether client requirements are satisfied or not. If not we are again going to develop.

### **6. Implementation:**



In testing phase if client requirements are satisfied, we go for implementation. i.e. we need to deploy the application in some server.

7. Maintenance:

After deployment, if at all any problems come from the client side; we are providing maintenance for that application.

## **CHAPTER-5**

### **SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

#### **TYPES OF TESTS:**

##### **5.1 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

#### **Test strategy and approach:**

Field testing will be performed manually and functional tests will be written in detail.

#### **Test objectives:**

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

#### **Features to be tested**

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

##### **5.2 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### **5.3 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

#### **System Testing:**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

#### **White Box Testing:**

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

#### **Black Box Testing:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

#### **Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

#### **Test strategy and approach:**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives:**

All field entries must work properly.

Pages must be activated from the identified link.

The entry screen, messages and responses must not be delayed.

**Features to be tested:**

Verify that the entries are of the correct format

No duplicate entries should be allowed

All links should take the user to the correct page.

**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. Components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing:**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**SYSTEM TESTING:**

**TESTING METHODOLOGIES:**

The following are the Testing Methodologies:

**Unit Testing.**

**Integration Testing.**

**User Acceptance Testing.**

**Output Testing.**

**Validation Testing.**

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All-important processing path are tested for the expected results. All error handling paths are also tested.

### **Integration Testing:**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

### **The following are the types of Integration Testing:**

#### **1.Top-Down Integration:**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

#### **2. Bottom-up Integration:**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom-up integration strategy may be implemented with the following steps:

The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.

A driver (i.e.) The control program for testing is written to coordinate test case input and output.

The cluster is tested.

Drivers are removed and clusters are combined moving upward in the program structure

The bottom-up approaches test each module individually and then each module is integrated with a main module and tested for functionality.

## **OTHER TESTING METHODOLOGIES:**

### **User Acceptance Testing:**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

### **Output Testing:**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

### **Validation Checking:**

Validation checks are performed on the following fields.

#### **Text Field:**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

#### **Numeric Field:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error message. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces an output revealing the errors in the system.

### **Preparation of Test Data:**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

### **Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true system test and in fact ignores the cases most likely to cause system failure.

### **Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

### **USER TRAINING:**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose, the normal working of the project was demonstrated

to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

## **MAINTAINENCE**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing are simple and easy to understand which will make maintenance easier.

## **TESTING STRATEGY:**

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.



## **TEST CASES:**

Test Case Id	Test Case Name	Test Case Desc.	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
Aplication -011	Flask application	To create Flask Application in Pycharm Tool	If Application created	Application will open	Application created in Local Drive	High	High
credit.py 02	Create python file with extension.py	To check whether File Created or not	Create Main Method in python File	File created with desired name	We can write main method for flask application	High	High
Parameter Accessing 03	Template creation using HTML	To get parameter for Tuning prediction function	Applying for Model	Check Eye Desease	To get actual prediction based on parameters	High	High
Model 04	Preparing Model	To verify Model generate in .pk or not	Check model working properly or not	Function should accept generated model	Model working successfully	High	High
Flask executin g by using python Libraries 05	Install Libraries	Install flak,matplto lib, sklearn,seab orn...etc	If you have advanced library then It will Shows already installed updated version	We cannot download if proper Internet not found	We can download with proper resources	Medium	Medium

Applying Environment 06	Environment Setting	Go to File select setting in pycharm goto Existed environment	If you are not created environment need to create using anaconda prompt and install required Libraries	He is not Created need to created environment	Environment applied success	Medium	Medium
Executing 07	Executing Project in Terminal	Open Terminal and type python app.py	If the file is not available in pycharm it show file not found	File Executed without Errors	Deployed inside flask server and generating url likes <b>http://localhost:5000/</b>	Medium	Medium

## CHAPTER-6

### CONCLUSION

Decision-making tool has been developed for selecting suitable agricultural crop to be cultivated in the given experimental land. Twenty-six input variables are selected and grouped into six main variables such as soil, water, season, input, support and infrastructure. The evaluation scores of alternatives in the form of main variable sequences are obtained by dominance-based rough set approach and simple additive method. Johnson's reduct classifier algorithm is used in the developed tool to generate classification rules for three agricultural crops such as paddy, groundnut and sugarcane. The validation results showed that the developed tool has sufficient predictive power to help the farmers to select suitable crop for agriculture development. Although illustrations are based on the three agricultural crops, namely paddy, groundnut and sugarcane, the developed decision-making tool can act as a multi-class classification tool to select any agricultural crop for cultivation. The limitations of this tool are as follows: Johnson's classifier produces satisfactory results for smaller training datasets. Discernibility based attribute reduction system has input number restrictions in the dataset. As it supports maximum data set with 12 input variables and 120 alternatives, it needs extreme storage area. Therefore, Johnson's classifier algorithm is suitable for MCDM problems with limited datasets like the one we have considered. Further, the research can be extended by using other classifiers for larger training datasets. The programs in MATLAB were designed in a flexible, modular fashion, and it can be easily adapted to other horticulture crops.

## CHAPTER-7

### REFERENCES

- [1] Agarwal, Apoorv, Fadi Biadisy, and Kathleen R. Mckeown. "Contextual phrase-level polarity analysis using lexical affect scoring and syntactic n-grams." Proceedings of the 12<sup>th</sup> Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2009.
- [2] Barbosa, Luciano, and Junlan Feng. "Robust sentiment detection on twitter from biased and noisy data." Proceedings of the 23rd international conference on computational linguistics: posters. Association for Computational Linguistics, 2010.
- [3] Bermingham, Adam, and Alan F. Smeaton. "Classifying sentiment in microblogs: is brevity an advantage?." Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, 2010.
- [4] Gamon, Michael. "Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis." Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, 2004.
- [5] Kim, Soo-Min, and Eduard Hovy. "Determining the sentiment of opinions." Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, 2004.
- [6] Klein, Dan, and Christopher D. Manning. "Accurate unlexicalized parsing." Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1. Association for Computational Linguistics, 2003..
- [7] Charniak, Eugene, and Mark Johnson. "Coarse-to-fine n-best parsing and MaxEnt discriminative reranking." Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics, 2005.
- [8] Gupta, B., Negi, M., Vishwakarma, K., Rawat, G., & Badhani, P. (2017). Study of Twitter sentiment analysis using machine learning algorithms on Python. International Journal of Computer Applications, 165(9), 0975-8887.
- [9] Sahayak, V., Shete, V., & Pathan, A. (2015). Sentiment analysis on twitter data. International Journal of Innovative Research in Advanced Engineering (IJIRAE), 2(1), 178-183.
- [10] Mamgain, N., Mehta, E., Mittal, A., & Bhatt, G. (2016, March). Sentiment analysis of top colleges in India using Twitter data. In Computational Techniques in Information and Communication Technologies (ICCTICT), 2016 International Conference on (pp. 525-530). IEEE.

## SAMPLE CODE

```
{
  "cells": [
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {},
      "outputs": [],
      "source": []
    },
    {
      "cell_type": "markdown",
      "metadata": {},
      "source": [
        "# Heart Diseases Analysis\n",
        "INTRODUCTION\n",
        "\n",
        "We have a data which classified if patients have heart disease or not according to features in it. We will try to use this data to create a model which tries predict if a patient has this disease or not. We will use logistic regression (classification) algorithm."
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 2,
      "metadata": {},
      "outputs": [
        {
          "name": "stdout",
          "output_type": "stream",
          "text": [
            '['.ipynb_checkpoints',          'heart.csv',          'heartDiseaseAndAges.png',
            'HeartDiseasePrediction.ipynb']\n"
          ]
        }
      ]
    }
  ]
}
```

```

],
"source": [
    "#Import Libraries\n",
    "import numpy as np\n",
    "import pandas as pd\n",
    "import matplotlib.pyplot as plt\n",
    "import seaborn as sns\n",
    "from sklearn.linear_model import LogisticRegression\n",
    "from sklearn.neighbors import KNeighborsClassifier\n",
    "from sklearn.svm import SVC\n",
    "from sklearn.naive_bayes import GaussianNB\n",
    "from sklearn.tree import DecisionTreeClassifier\n",
    "from sklearn.ensemble import RandomForestClassifier\n",
    "from sklearn.model_selection import train_test_split\n",
    "\n",
    "# data files are available in the current directory.\n",
    "\n",
    "import os\n",
    "print(os.listdir(\".\"))"
]
},
{
    "cell_type": "code",
    "execution_count": 4,
    "metadata": {},
}
{
    "cell_type": "markdown",
    "metadata": {},
    "source": [
        "Data Description\n",
        "\n",
        "age: The person's age in years\n",
        "\n",
        "sex: The person's sex (1 = male, 0 = female)\n",
        "\n",

```

```

"cp: The chest pain experienced (Value 1: typical angina, Value 2: atypical angina, Value 3:
non-anginal pain, Value 4: asymptomatic)\n",
"\n",
"trestbps: The person's resting blood pressure (mm Hg on admission to the hospital)\n",
"\n",
"chol: The person's cholesterol measurement in mg/dl\n",
"\n",
"fbs: The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)\n",
"\n",
"restecg: Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave
abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)\n",
"\n",
"thalach: The person's maximum heart rate achieved\n",
"\n",
"exang: Exercise induced angina (1 = yes; 0 = no)\n",
"\n",
"oldpeak: ST depression induced by exercise relative to rest ('ST' relates to positions on the
ECG plot. See more here)\n",
"\n",
"slope: the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value
3: downsloping)\n",
"\n",
"ca: The number of major vessels (0-3)\n",
"\n",
"thal: A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversible
defect)\n",
"\n",
"target: Heart disease (0 = no, 1 = yes)"
]
},
{
"cell_type": "markdown",
"metadata": {},
"source": [
"# Exploratory Data Analysis"

```

```

]
},
{
  "cell_type": "code",
  "execution_count": 5,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/html": [
          "<div>\n",
          "<style scoped>\n",
          "  .dataframe tbody tr th:only-of-type {\n",
          "    vertical-align: middle;\n",
          "  }\n",
          "\n",
          "  .dataframe tbody tr th {\n",
          "    vertical-align: top;\n",
          "  }\n",
          "\n",
          "  .dataframe thead th {\n",
          "    text-align: right;\n",
          "  }\n",
          "</style>\n",
          "<table border='1' class='dataframe'>\n",
          "  <thead>\n",

```