CROP YIELD PREDICTION USING MACHINELEARNING

INDEX

ABSTRACT

Agriculture is one of the major and the least paid occupation in India. Machine learning can bring a boom in the agriculture field by changing the income scenario through growing the optimum crop. This paper focuses on predicting the yield of the crop by applying various machine learning techniques. The outcome of these techniques is compared on the basis of mean absolute error. The prediction made by machine learning algorithms will help the farmers to decide which crop to grow to get the maximum yield by considering factors like temperature, rainfall, area, etc. India presence an agriculture country, its economy predominantly depends on agriculture yield growth and agro-industry products. Data Mining is an emerging research field in crop yield analysis. Yield prediction is a very important issue in agricultural. Any farmer is interested in knowing how much yield he is about to expect. Analyze the various related attributes like location, pH value from which alkalinity of the soil is determined. Along with it, percentage of nutrients like Nitrogen Phosphorous, and Potassium. Location is used along with the use of applications like RNN accuracy (Recurrent neural network) LSTM accuracy(Long short-term memory) and Feed forward accuracy for weather and temperature, type of soil, nutrient value of the soil in that region, amount of rainfall in the region, soil composition can be determined. All these attributes of data will be analyzed, train the data with various suitable machine learning algorithms for creating a model. The system comes with a model to be precise and accurate in predicting crop yield and deliver the end user with proper recommendation like crop which can be cultivated based on the atmosphere conditions.

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

SYSTEM ANALYSIS

1 SYSTEM ANALYSIS

1.1 Existing System

Due to the revolution in industrialization, the economic contribution of agriculture to India's GDP is steadily declining with the country's broad-based economic growth. The problem that the Indian Agriculture sector is facing is the integration of technology to bring the desired outputs. With the advent of new technologies and overuse of non-renewable energy resources patterns of rainfall and temperature are disturbed. The inconsistent trends developed from the side effects of global warming make it cumbersome for the farmers to clearly predict the temperature and rainfall patterns thus affecting their crop yield productivity. In order to perform accurate prediction and handle inconsistent trends in temperature and rainfall various machine learning algorithms like RNN, LSTM, etc. can be applied to get a pattern. It will complement the agricultural growth in India and all together augment the ease of living for farmers. In past, many researchers have applied machine learning techniques to enhance agricultural growth of the country

1.1.1 Disadvantages of Existing System

- Efficiency is low.
- More number of repeated works

1.2 Proposed System

This paper focuses on the practical application of machine learning algorithms and its quantification. The work presented here also takes into account the inconsistent data from rainfall and temperature datasets to get a consistent trend. Crop yield prediction is determined by considering all the features in contrast with the usual trend of determining the prediction considering one feature at a time

1.2.1 Advantages of Proposed System.

- Secure and efficient system.
- The advantage of this system in the field of agriculture is that we can select proper crops according to climate, temperature etc.
- As Machine Learning helps us to make predictions using the given data it avoids assumptions and difficulties of using larger sample spaces and complex problems.

1.3 Introduction

1.3.1 MOTIVATION

The history of agriculture in India [1] dates back to the Indus Valley Civilization Era. India ranks second in this sector. Agriculture and allied sectors like forestry and fisheries account for 15.4 percent of the GDP (gross domestic product) with about 31 percent of the workforce. India ranks first globally with the highest net cropped area followed by US and China. Agriculture is demographically the broadest economic sector and plays a significant role in the overall socio-economic fabric of India. Due to the revolution in industrialization, the economic contribution of agriculture to India's GDP is steadily declining with the country's broad-based economic growth.

1.3.2 PROBLEM DEFINITION

The problem that the Indian Agriculture sector is facing is the integration of technology to bring the desired outputs. With the advent of new technologies and overuse of non-renewable energy resources patterns of rainfall and temperature are disturbed. The inconsistent trends developed from the side effects of global warming make it cumbersome for the farmers to clearly predict the temperature and rainfall patterns thus affecting their crop yield productivity. In order to perform accurate prediction and handle inconsistent trends in temperature and rainfall various machine learning algorithms like RNN, LSTM, etc. can be applied to get a pattern. It will complement the agricultural growth in India and all together augment the ease of living for farmers. In past, many researchers have applied machine learning techniques to enhance agricultural growth of the country

1.3.3 OBJECTIVE OF PROJECT

This paper focuses on predicting the yield of the crop by applying various machine learning techniques. The outcome of these techniques is compared on the basis of mean absolute error. The prediction made by machine learning algorithms will help the farmers to decide which crop to grow to get the maximum yield by considering factors like temperature, rainfall, area, etc.

CHAPTER-2

LITERATURE SURVEY

2 LITERATURE SURVEY

[1] P. Priya, U. Muthaiah & M. Balamurugan of "PREDICTING YIELD OF THE CROP USING MACHINE LEARNING ALGORITHM"

The agriculture plays a dominant role in the growth of the country's economy. Climate and other environmental changes have become a major threat in the agriculture field. Machine learning (ML) is an essential approach for achieving practical and effective solutions for this problem. Crop Yield Prediction involves predicting yield of the crop from available historical available data like weather parameter, soil parameter and historic crop yield. This paper focus on predicting the yield of the crop based on the existing data by using Random Forest algorithm. Real data of Tamil Nadu were used for building the models and the models were tested with samples. The prediction will help to the farmer to predict the yield of the crop before cultivating onto the agriculture field. To predict the crop yield in future accurately Random Forest, a most powerful and popular supervised machine learning algorithm is used.

[2] Mishra. s, Mishra. D and Santra. H of "Applications of machine learning techniques in agricultural crop production"

This paper has been prepared as an effort to reassess the research studies on the relevance of machine learning techniques in the domain of agricultural crop production. Methods/Statistical Analysis: This method is a new approach for production of agricultural crop management. Accurate and timely forecasts of crop production are necessary for important policy decisions like import-export, pricing marketing distribution etc. which are issued by the directorate of economics and statistics. However, one has understood that these prior estimates are not the objective estimates as this estimate requires lots of descriptive assessment based on many different qualitative factors. Hence there is a requirement to develop statistically sound objective prediction of crop production. That development in computing and information storage has provided large amount of data. Findings: The problem has been to intricate knowledge from this raw data, this has led to the development of new approach and techniques such as machine learning that can be used to unite the knowledge of the data with crop yield evaluation. This research has been intended to evaluate these innovative techniques such that significant relationship can be found by their applications to the various variables present in the data base. Application/Improvement: The few techniques like artificial neural networks, Information Fuzzy Network, Decision Tree, Regression Analysis, Bayesian belief network. Time series analysis, Markov chain model, k-means clustering, k nearest neighbor, and support vector machine are applied in the domain of agriculture were presented.

[3] Manjula. E of A Model for "Prediction of Crop Yield"

Data Mining is emerging research field in crop yield analysis. Yield prediction is a very important issue in agricultural. Any farmer is interested in knowing how much yield he is about to expect. In the past, yield prediction was performed by considering farmer's experience on particular field and crop. The yield prediction is a major issue that remains to be solved based on available data. Data mining techniques are the better choice for this purpose. Different Data Mining techniques are used and evaluated in agriculture for estimating the future year's crop

production. This research proposes and implements a system to predict crop yield from previous data. This is achieved by applying association rule mining on agriculture data. This research focuses on creation of a prediction model which may be used to future prediction of crop yield. This paper presents a brief analysis of crop yield prediction using data mining technique based on association rules for the selected region i.e., district of Tamil Nadu in India. The experimental results shows that the proposed work efficiently predict the crop yield production.

[4] Dahikar, S. S, Rode and S. V. of "Agricultural crop yield prediction using artificial neural network approach"

By considering various situations of climatologically phenomena affecting local weather conditions in various parts of the world. These weather conditions have a direct effect on crop yield. Various researches have been done exploring the connections between large-scale climatologically phenomena and crop yield. Artificial neural networks have been demonstrated to be powerful tools for modeling and prediction, to increase their effectiveness. Crop prediction methodology is used to predict the suitable crop by sensing various parameter of soil and also parameter related to atmosphere. Parameters like type of soil, PH, nitrogen, phosphate, potassium, organic carbon, calcium, magnesium, Sulphur, manganese, copper, iron, depth, temperature, rainfall, humidity. For that purpose, we are used artificial neural network (ANN).

[5] Gonzlez Snchez. A, Frausto Sols. J and Ojeda Bustamante. W of "Predictive ability of machine learning methods for massive crop yield prediction"

An important issue for agricultural planning purposes is the accurate yield estimation for the numerous crops involved in the planning. Machine learning (ML) is an essential approach for achieving practical and effective solutions for this problem. Many comparisons of ML methods for yield prediction have been made, seeking for the most accurate technique. Generally, the number of evaluated crops and techniques is too low and does not provide enough information for agricultural planning purposes. This paper compares the predictive accuracy of ML and linear regression techniques for crop yield prediction in ten crop datasets. Multiple linear regression, M5-Prime regression trees, perceptron multilayer neural networks, support vector regression and k-nearest neighbor methods were ranked. Four accuracy metrics were used to validate the models: the root mean square error (RMS), root relative square error (RRSE), normalized mean absolute error (MAE), and correlation factor (R). Real data of an irrigation zone of Mexico were used for building the models. Models were tested with samples of two consecutive years. The results show that M5- Prime and k-nearest neighbor techniques obtain the lowest average RMSE errors (5.14 and 4.91), the lowest RRSE errors (79.46% and 79.78%), the lowest average MAE errors (18.12% and 19.42%), and the highest average correlation factors (0.41 and 0.42). Since M5-Prime achieves the largest number of crop yield models with the lowest errors, it is a very suitable tool for massive crop yield prediction in agricultural planning.

[6] Malathy, S. Vanitha, Mohankkanth, E. Rainfall Prediction for Enhancing Crop-Yield based on Machine Learning Techniques.

The authors proposed a novel technique based on Support Vector Machines for auxiliary information on real applications of the agriculture sector. The authors claimed that they obtained an accuracy of 91% compared to the existing applications. The farmers can use this proposed methodology to gain a better yield of the crops, and different governmental sectors can use it to improve crop productivity. However, the authors did not suggest any recommendations for fertilizer systems to improve crop management

[7] Apeksha, R.G.; Swati, S.S of A brief study on the prediction of crop disease using machine learning approaches.

The authors developed a web-based application for crop yield prediction to be used by farmers. This tool provides farmers with a list of various crops planted previously to predict and learn about the best crop to cultivate in the future. Furthermore, the tool can provide farmers with climate data and information to help them make the best decision regarding market demand and prices.

[8] Kumar, R.; Shukla, N. Prince of Plant Disease Detection and Crop Recommendation Using CNN and Machine Learning.

The author's survey's main objective was to evaluate the performance of different publications from 2016 to 2020 that aimed to predict fungal illnesses on the crops. The authors evaluated different machine learning algorithms utilized in the literature. As per the provided comparative results, the authors concluded that the best performance among all machine learning models, SVM, variations of choice trees, and Naïve Bayes has been widely utilized and gained the best results regarding the yearly prediction of crop diseases.

[9] Saeed Khaki and Lizhi Wang of Crop Yield Prediction Using Deep Neural Networks.

Crop yield is a highly complex trait determined by multiple factors such as genotype, environment, and their interactions. Accurate yield prediction requires fundamental understanding of the functional relationship between yield and these interactive factors, and to reveal such relationship requires both comprehensive datasets and powerful algorithms. In the 2018 Syngenta Crop Challenge, Syngenta released several large datasets that recorded the genotype and yield performances of 2,267 maize hybrids planted in 2,247 locations between 2008 and 2016 and asked participants to predict the yield performance in 2017. A deep neural network approach that took advantage of state-of-the-art modelling and solution techniques. Our model was found to have a superior prediction accuracy, with a root-mean-square-error being 12% of the average yield and 50% of the standard deviation for the validation dataset using predicted weather data. With perfect weather data, the RMSE would be reduced to 11% of the average yield and 46% of the standard deviation. Also performed feature selection based on the trained DNN model, which successfully decreased the dimension of the input space without significant drop in the prediction accuracy. Our computational results suggested that this model significantly outperformed other popular methods such as Lasso, shallow neural networks, and regression tree.

[10] S.Pudumalar, E.Ramanujam, R.Harine Rajashreen, C.Kavyan, T.Kiruthikan, J.Nishan. of Crop Recommendation System for Precesion agriculture

Agriculture segment gives different yields, for example, sustenance, crude material for industry, affordable lift and business. The Agriculture part contains huge information regarding factors influencing its info and yield. With advances in innovation different information mining systems are presented. These information mining methods can be utilized to dissect the multidimensional, time explicit information of horticulture area to create powerful learning from it which can be utilized to support the economy. Moreover, machine learning explored the most performing calculation having amazing expectation exactness to suggest the best harvest for better yield. The proposed framework will coordinate the information got from archive, climate office and by applying machine learning calculation: Multiple Linear Regression, an expectation of most reasonable yields as indicated by current natural conditions is made. This furnishes an agriculturist with assortment of alternatives of harvests that can be developed. This exploration goes for examination of soil dataset utilizing information mining procedures. It centers around characterization of soil utilizing different calculations accessible.

CHAPTER-3

SYSTEM DESIGN

3 SYSTEM DESIGN

3.1 System Architecture

System design is the second step in the system life cycle, in which overall design of the system is achieved. The functionalities of the system is designed and studied in this phase. The first step is designing of program specification. This determines the various data inputs to the system, data flow and the format in which output is to be obtained.

Design phase is a transmission phase because it is a transition from user oriented document to computer data. The activity in the design phase is the allocation of functions to manual operations, equipment and computer programs. Flow charts are prepared in the study time and is decomposed until all functions in the system perform evidently.

Design is a multi-step process that focuses on data structures, software architecture, procedural details(algorithms etc) and links between the modules. The design process goes through logical and physical stages. In logical design reviews are made linking existing system and specification gathered. The physical plan specifies any hardware and software requirement, which satisfies the local design.

Modularization of task is made in this phase. The success of any integrated system depends on the planning of each and every fundamental module. Usually a project is revised in step by step sequence. Inter-phase management of such module is also important. Software design methodology changes continually as new methods, better analysis and broader understanding evolve.

Various techniques for software design do exit with the availability of criteria for design quality. Software design leads three technical activities-design, code and test.

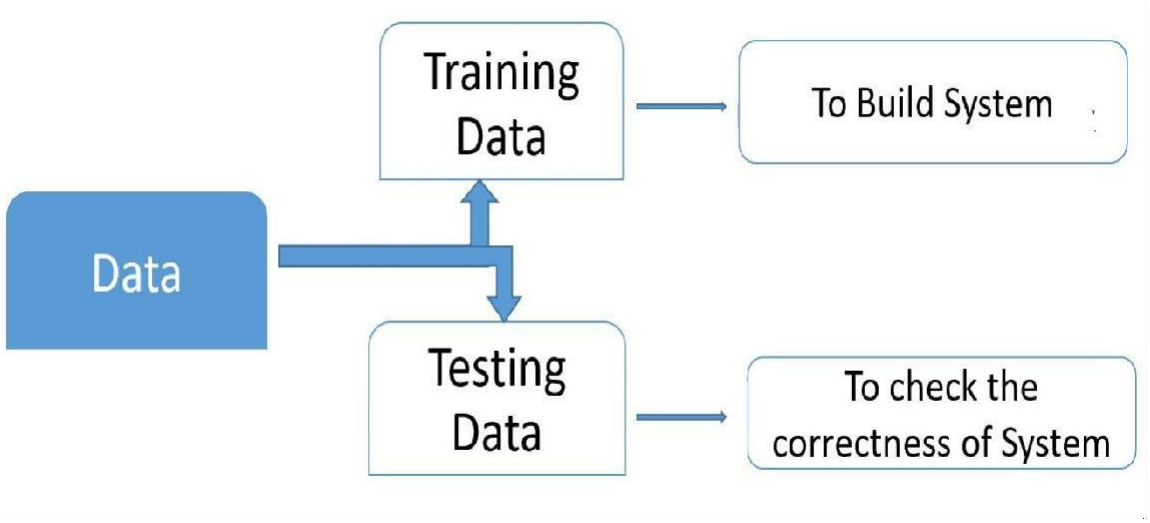

Fig 3.1 System Architecture

3.1.1 MODULES:

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and I Python shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font

properties, axes properties, etc., via an object-oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

Upload Crop Dataset

The crop production dataset that is used to predict the name and yield of the crop is fed into classification and regression algorithms.

Preprocess Dataset

Experiments were conducted on Indian government dataset and it has been established that Random Forest Regressor gives the highest yield prediction accuracy. Sequential model that is Simple Recurrent Neural Network performs better on rainfall prediction while LSTM is good for temperature prediction. By combining rainfall, temperature along with other parameters like season and area, yield prediction for a certain district can be made.

Train Machine Learning

This focuses on district wise yield prediction according to the crop sown in the district. Yield is being predicted for given crops district wise and crops with best yield.

Upload Test Data &Predict Yield

Results reveals that Random Forest is the best classifier when all parameters are combined. This will not only help farmers in choosing the right crop to grow in the next season but also bridge the gap between technology and the agriculture sector. R

3.1.2 BLOCK DIAGRAM

Block 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.

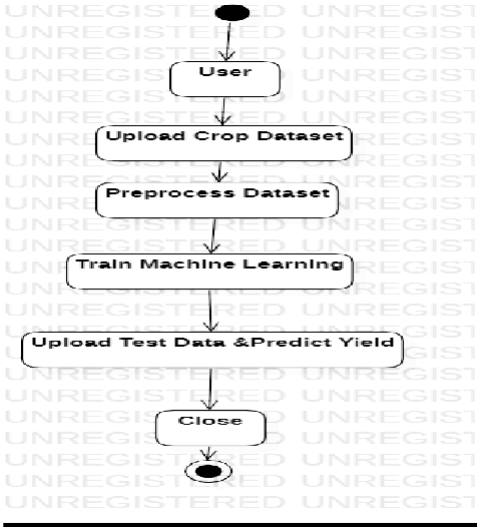

Fig 3.1.2.1 Block Diagram

3.2 System Requirements

3.2.1 Hardware Requirements

• System	11thGenIntel(R)Core(TM)i5-1135G7 @2.40GHz
• Hard Disk	1TB
• Ram	8 GB
• Monitor	15.6 HD WVA Display

3.2.2 Software Requirements

• Operating system	Windows 11
• Coding Language	Python3.7.1

CHAPTER-4

INPUT AND OUTPUT DESIGN

4 INPUT AND OUTPUT DESIGN

4.1 Input Design

Upload Crop Dataset

The crop production dataset that is used to predict the name and yield of the crop is fed into classification and regression algorithms.

Preprocess Dataset

Experiments were conducted on Indian government dataset and it has been established that Random Forest Regressor gives the highest yield prediction accuracy. Sequential model that is Simple Recurrent Neural Network performs better on rainfall prediction while LSTM is good for temperature prediction. By combining rainfall, temperature along with other parameters like season and area, yield prediction for a certain district can be made.

Train Machine Learning

This focuses on district wise yield prediction according to the crop sown in the district. Yield is being predicted for given crops district wise and crops with best yield.

Upload Test Data &Predict Yield

Results reveals that Random Forest is the best classifier when all parameters are combined. This will not only help farmers in choosing the right crop to grow in the next season but also bridge the gap between technology and the agriculture sector.

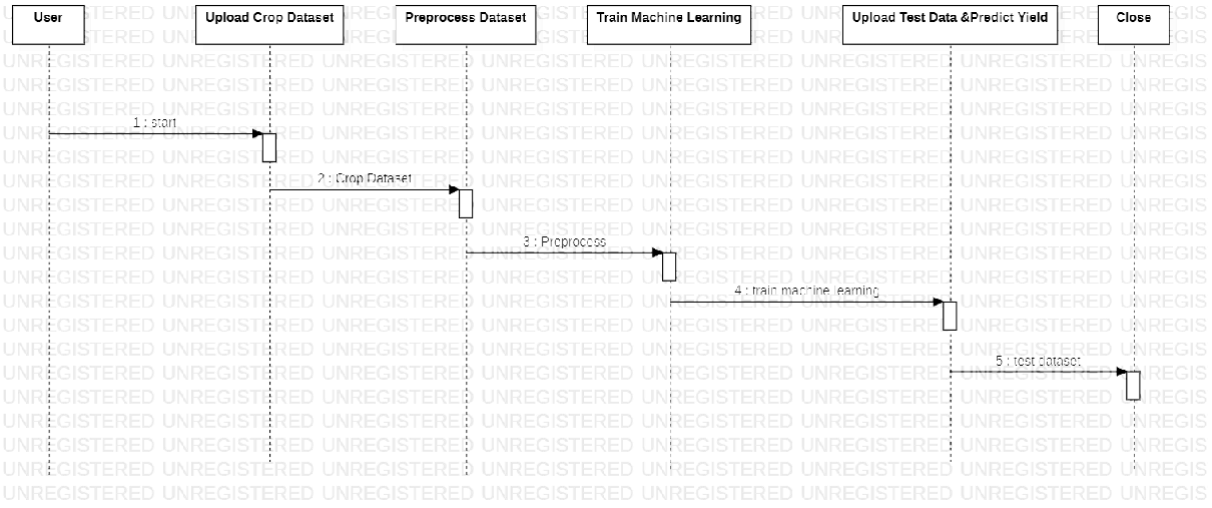

Fig 4.1.1 Input Design

4.2 Output Design

To run project double click on ‘run.bat’ file to get below screen

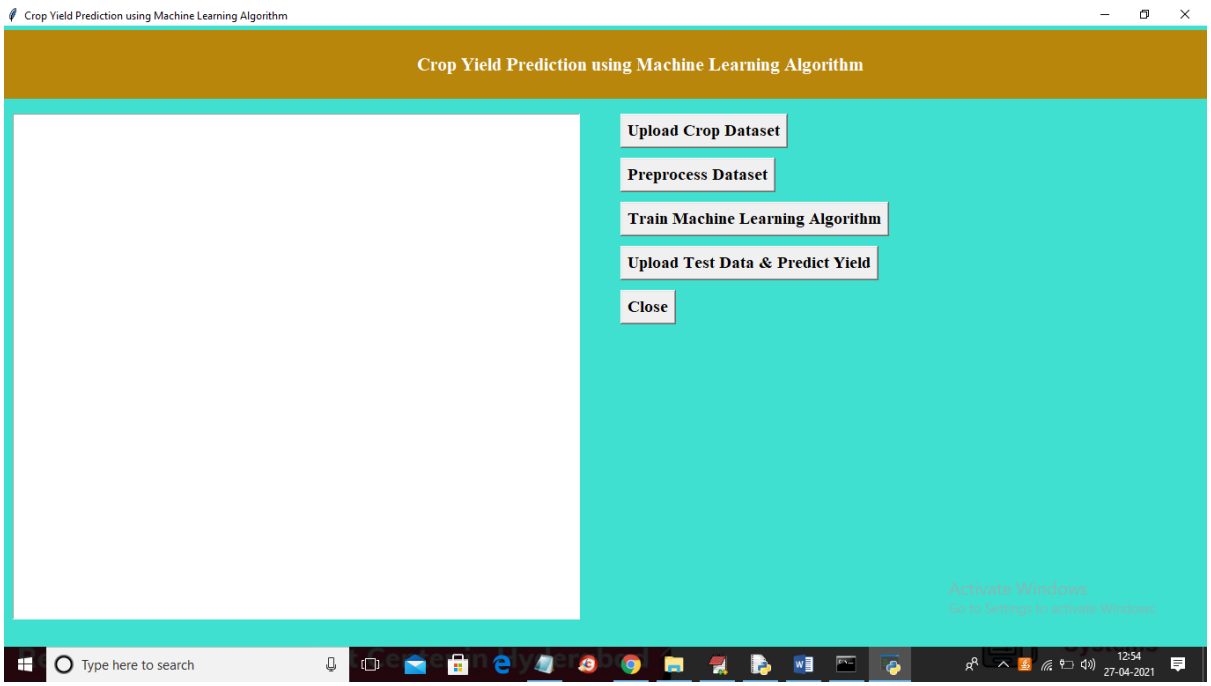

Fig 4.2.1 Output Design

In fig 4.2.1 screen click on ‘Upload Crop Dataset’ button to upload dataset

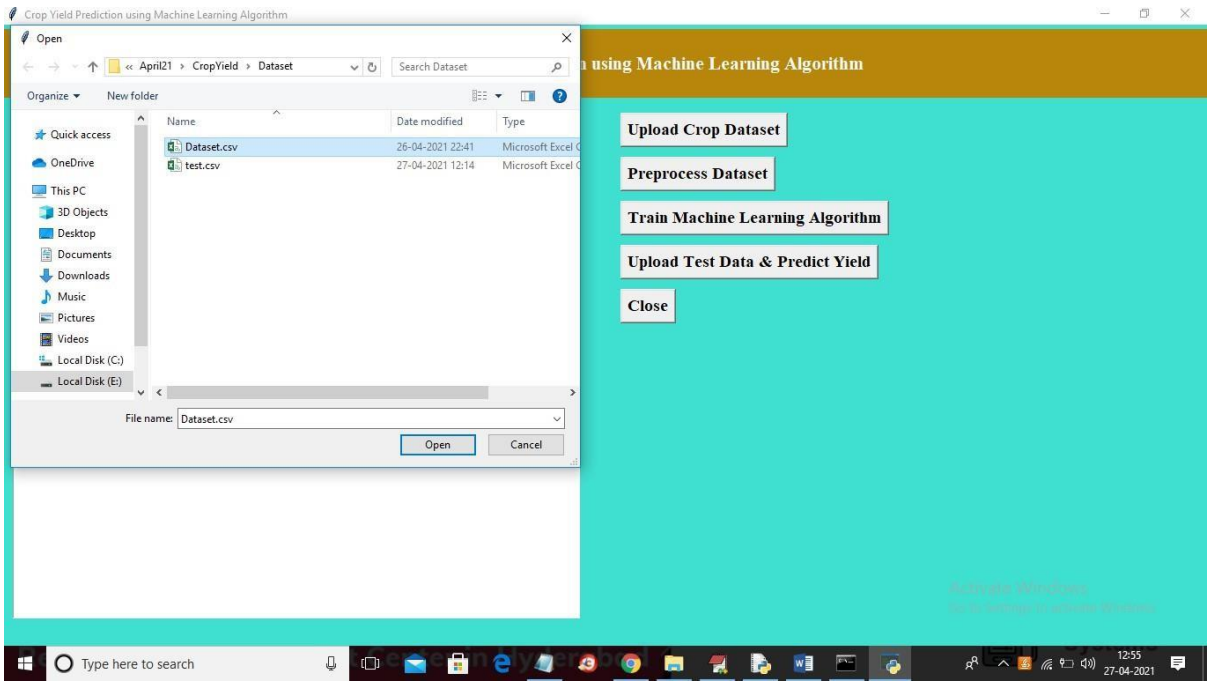

Fig 4.2.2 Output Design

In fig 4.2.2 screen selecting and uploading 'Dataset.csv' file and then click on 'Open' button to load dataset and to get below screen

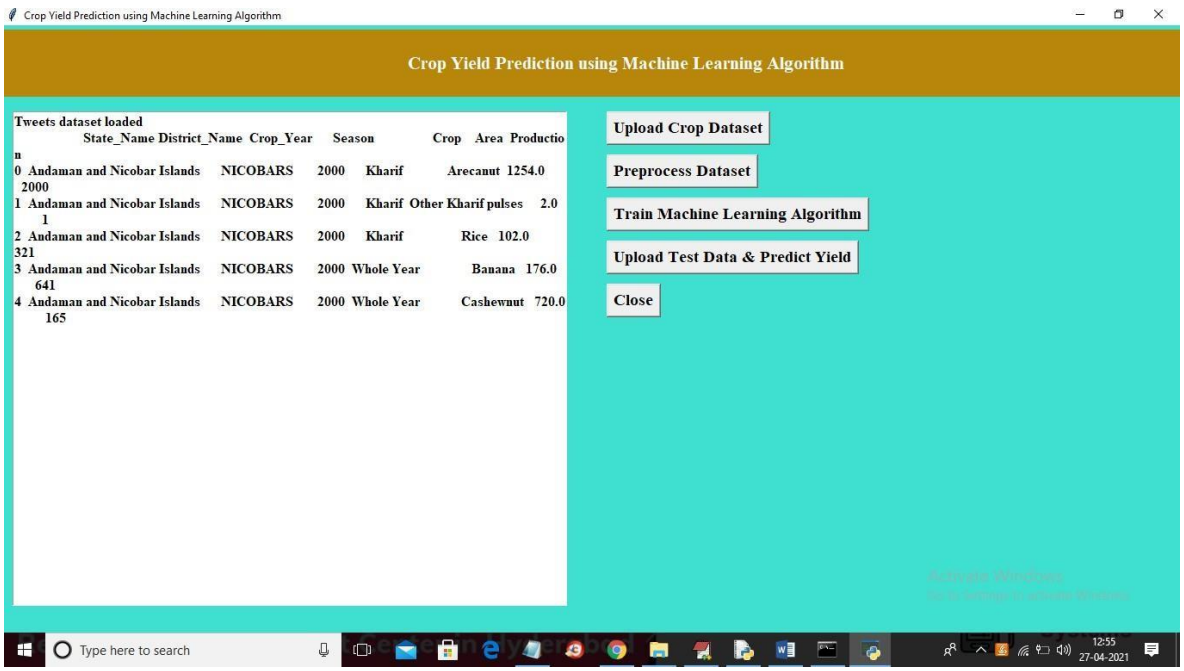

Fig 4.2.3 Output Design

In fig 4.2.3 screen dataset loaded and we can see dataset contains some non-numeric values and ML will not take non-numeric values so we need to preprocess dataset to convert non-numeric values to numeric values by assigning ID to each non-numeric value. So, click on 'Preprocess Dataset' button to process dataset

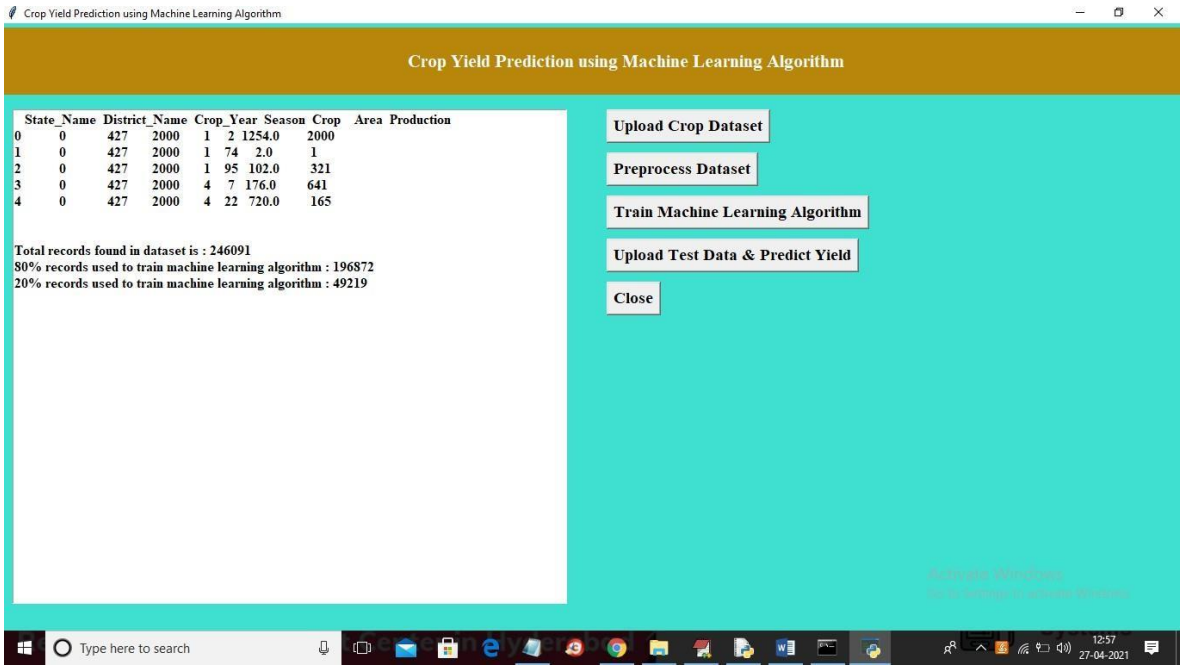

Fig 4.2.4 Output Design

In above screen all non-numeric values converted to numeric format and in below lines we can see dataset contains total 246091 records and application using (80%) 196872 records

to train ML and using (20%) 49219 records to test ML prediction error rate (RMSE (root mean square error)). Now click on 'Train Machine Learning Algorithm' button to train Decision Tree Machine learning algorithm on above dataset and then calculate prediction error rate

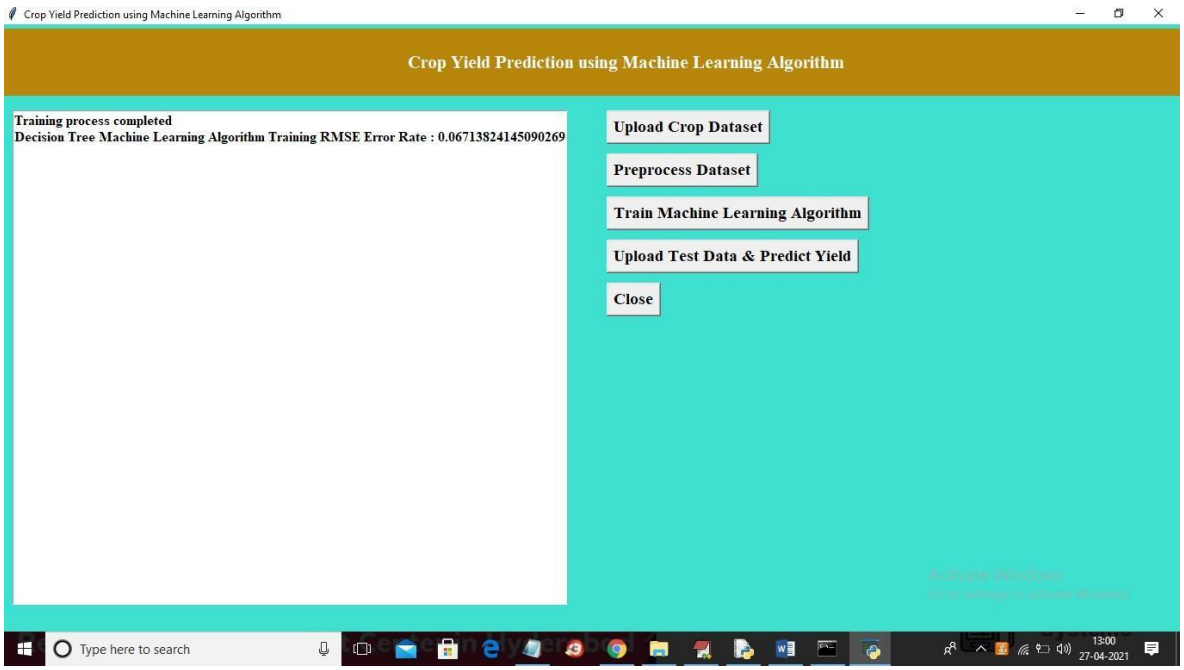

Fig 4.2.5 Output Design

In fig 4.2.5 screen ML is trained and we got prediction error rate as 0.067% and now Decision Tree model is ready and now click on 'Upload Test Data & Predict Yield' button to upload test data and then application will predict production

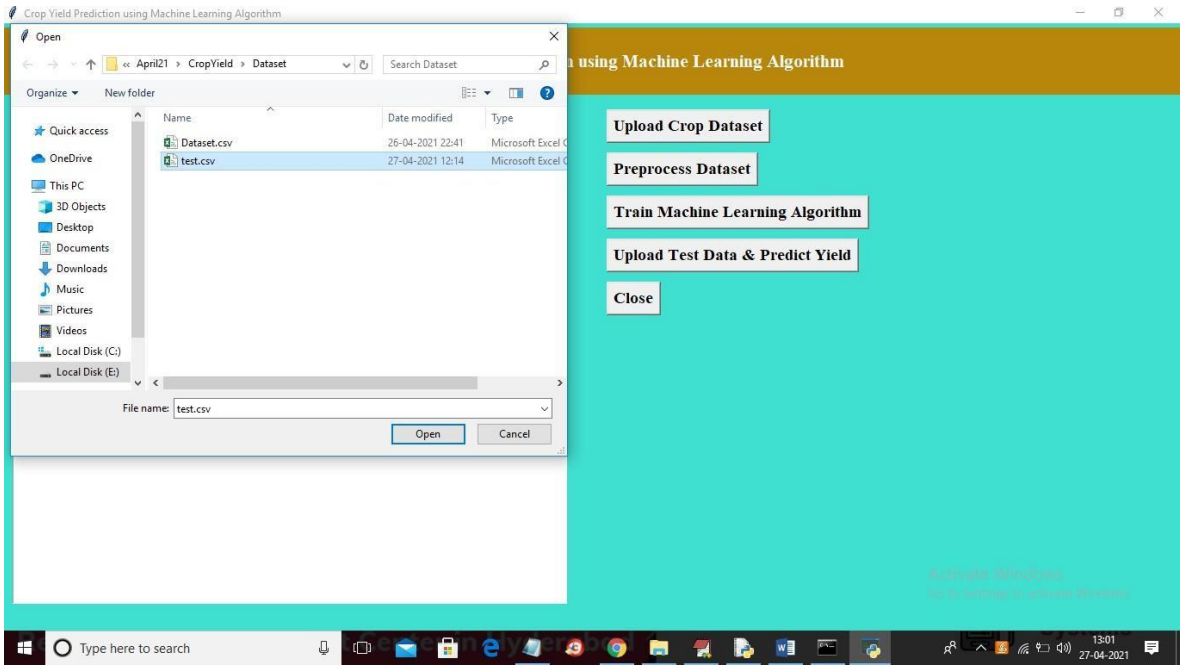

Fig 4.2.6 Output Design

In fig 4.2.6 screen selecting and uploading 'test.csv' file and then click on 'Open' button to load test data and then application will give below prediction result

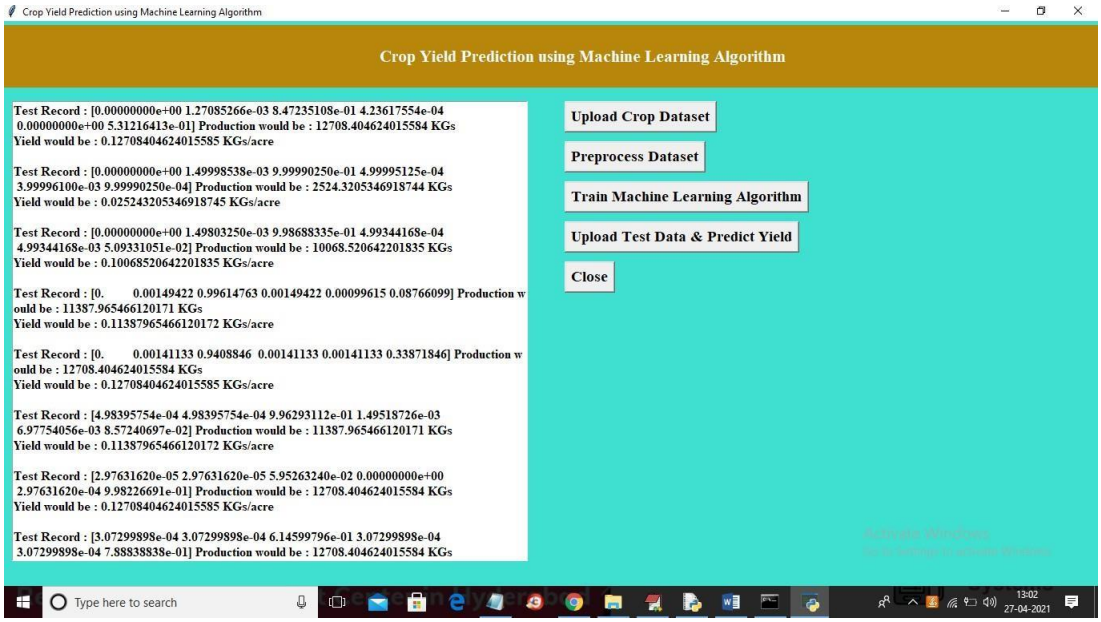

Fig 4.2.7 Output Design

In fig 4.2.7 screen each test record is separated with newline and in above screen in square bracket we can see test data values and after square bracket we can see predicted production and after that we can see predicted YIELD per acre. So, each test record and its prediction are separated with newline.

CHAPTER-5

SYSTEM ENVIRONMENT

5 SYSTEM ENVIRONMENT

5.1 Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

5.1.1 History of Python: -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskundeen Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So, I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

5.1.2 What can Python do: -

Below are some facts about Python. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, Beautiful Soup, Selenium)
- Test frameworks
- Multimedia

5.1.3 Why python

1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. **Extensible**

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

8. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

5.1.4 Python syntax compared to other programming languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Git hub annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

5.1.5 Uses of Python

Python is a versatile programming language with a wide range of applications across various domains. Here are some of the common uses of Python:

1. **Web Development:**

- Python is used for building dynamic websites and web applications using web frameworks such as Django, Flask, and Pyramid.
- It can be used for server-side scripting, handling HTTP requests, and interacting with databases.

2. **Data Science and Data Analysis:**

- Python is a popular choice for data analysis, visualization, and manipulation with libraries like NumPy, Pandas, Matplotlib, and Seaborn.
- Data scientists use Python for machine learning and artificial intelligence tasks using libraries such as scikit-learn, TensorFlow, and PyTorch.

3. **Scientific Computing:**

- Python is widely used in scientific research and engineering for tasks like simulations, computational modeling, and data analysis.

- Libraries like SciPy and SymPy provide tools for scientific computing and symbolic mathematics.

4. **Automation and Scripting**:

- Python is an excellent language for automating repetitive tasks, system administration, and scripting due to its simple syntax.
- It's used to create scripts for tasks like file manipulation, data extraction, and process automation.

5. **Game Development**:

- Python can be used for game development with libraries like Pygame and Panda3D.
- It's often used for developing 2D games and prototypes.

6. **Internet of Things (IoT)**:

- Python is used in IoT projects to control and monitor devices using platforms like Raspberry Pi and microcontrollers.
- It provides libraries and tools for hardware interaction.

7. **Natural Language Processing (NLP):**

- Python is a preferred language for NLP tasks such as text analysis, language translation, and sentiment analysis.
- Libraries like NLTK and spaCy are commonly used for NLP applications.

5.1.6 Python features

Python is a high-level, interpreted programming language known for its simplicity and readability. It offers a wide range of features that make it a popular choice for various applications. Here are some of the key features of Python:

1. **multi-paradigm**

- Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming.
- This flexibility allows developers to choose the best approach for their specific needs.

2. **High-Level Language**:

- Python is a high-level language, which means it abstracts many low-level details, making it more user-friendly and reducing the need for managing memory and other low-level operations.

3. **Simple and Readable Syntax**:

- Python's syntax is clear and easy to understand, making it an excellent language for beginners and experienced programmers.
- It uses indentation (whitespace) for code blocks, enhancing readability.

4. **Interpreted Language**:

- Python is an interpreted language, which means that code can be executed directly without the need for compilation.
- This makes development and debugging faster.

5. **Dynamic Typing**:

- Python uses dynamic typing, where variable types are determined at runtime.
- This provides flexibility but also requires careful variable management to avoid type-related errors.

5.2 SDLC

SDLC stands for Software Development Life Cycle. It is a structured and systematic approach to software development that defines the stages, activities, and processes involved in building, maintaining, and updating software systems. SDLC is a fundamental framework used by software development teams to ensure that software projects are completed efficiently, with high quality, and in a way that meets user and business requirements.

The typical stages of the Software Development Life Cycle include:

1. **Planning**: In this initial phase, project requirements are gathered, and the project scope is defined. A feasibility study is conducted to determine whether the project is viable. Project goals, timelines, and budgets are established during this stage.

2. **Analysis**: During this stage, the software's requirements are carefully analyzed and documented. This involves working closely with stakeholders to understand their needs and expectations. Use cases, user stories, and functional specifications are often created.

3. **Design**: In this phase, the architecture and design of the software system are developed. This includes defining the software's structure, components, data models, and user interfaces. Design documents are created to guide the development team.

4. **Implementation (Coding):** This is where the actual code for the software is written. Developers use the design documents as a blueprint to create the software's functionality. This stage involves programming, testing, and debugging.

5. **Testing:** The software undergoes various testing processes to ensure it meets the specified requirements. This includes unit testing, integration testing, system testing, and user acceptance testing. Bugs and issues are identified and fixed during this phase.

6. **Deployment:** Once the software has been thoroughly tested and is considered stable, it is deployed to a production environment for end-users to access and use.

7. **Maintenance and Support:** After deployment, the software enters a maintenance phase. Updates, bug fixes, and enhancements are made as needed to ensure the software continues to function correctly and meet evolving user requirements.

There are different SDLC models, and the choice of model depends on the project's specific needs and goals. Common SDLC models include the Waterfall model, Agile model, Iterative model, and Spiral model, among others. Each model has its own set of principles and characteristics, and it's important to select the one that best fits the project's requirements and constraints.

The SDLC process is essential for managing and controlling the software development process, improving the quality of the final product, and ensuring that it aligns with the business objectives and user needs. It provides a structured framework for project management, collaboration, and risk management throughout the software development journey.

5.2.1 Benefits of SDLC

The Software Development Life Cycle (SDLC) offers several benefits to software development projects and organizations, helping ensure the successful planning, development, and maintenance of software systems. Some of the key advantages of using SDLC are:

1. **Structured Approach:** SDLC provides a structured and organized framework for software development, which helps in systematically managing the project from initiation to completion.

2. **Improved Project Management:** SDLC models help project managers and teams set clear milestones, allocate resources, and establish timelines, leading to better project management and control.

3. **Risk Management:** By identifying and addressing potential risks and issues early in the process, SDLC allows for proactive risk management, reducing the likelihood of costly errors or project failures.

4. **Clear Documentation:** Each phase of the SDLC requires the creation of documentation, which aids in understanding project requirements, design, and implementation, making it easier for team members to work collaboratively.

5. **Quality Assurance:** SDLC emphasizes testing and quality control throughout the development process, ensuring that the software meets defined requirements and quality standards.

6. **User Involvement:** SDLC models often involve end-users and stakeholders in various phases of the development process, leading to better alignment with user needs and expectations.

7. **Efficient Resource Allocation:** SDLC helps in the efficient allocation of resources, ensuring that time and budget constraints are managed effectively.

8. **Cost Control:** By having a well-defined plan and control measures in place, SDLC helps manage costs and avoid budget overruns.

9. **Predictable Results:** SDLC models provide a clear roadmap for the project, making it easier to predict outcomes and deliver a product that meets user expectations.

10. **Scalability:** SDLC can be adapted to projects of various sizes and complexities, making it suitable for both small and large-scale software development.

11. **Standardization:** SDLC promotes the use of standardized development practices and methodologies, leading to greater consistency in the development process and resulting software.

12. **Easier Maintenance:** By emphasizing the importance of documentation and structured coding practices, SDLC models make it easier to maintain and enhance the software in the post-development phase.

13. **Increased Collaboration:** SDLC encourages collaboration among team members and stakeholders throughout the development process, leading to better communication and understanding of project goals.

14. **Clear Exit Criteria:** Each phase of SDLC has well-defined entry and exit criteria, which help ensure that a phase is completed satisfactorily before proceeding to the next one.

15. **Customer Satisfaction:** By involving users and stakeholders in the development process and delivering software that aligns with their needs, SDLC contributes to higher levels of customer satisfaction.

16. **Legal and Regulatory Compliance:** For projects subject to legal or regulatory requirements, SDLC provides a framework for ensuring compliance and mitigating legal risks.

It's important to note that the specific benefits of SDLC may vary depending on the chosen SDLC model and the unique characteristics of the software project. Organizations often select an SDLC model that best fits their needs and constraints to maximize these advantages.

5.3 Machine Learning: -

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories Of Machine Learning: -

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete

categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning: -

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning: -

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

How to Start Learning Machine Learning?

Arthur Samuel coined the term **"Machine Learning"** in 1959 and defined it as a **"Field of study that gives computers the capability to learn without being explicitly programmed".**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is the Best Job of 2019 with a *344%* growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So, this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer.

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on math's as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So, it is no surprise that you need to learn it!!! Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Kera's, TensorFlow, Scikit-learn, etc.

So, if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on Geeks for Geeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. Labelled datasets have both input and output parameters. In Supervised Learning algorithms learn to map points between inputs and correct outputs. It has both training and validation datasets. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning –** Unsupervised learning is a type of machine learning technique in which an algorithm discovers patterns and relationships using unlabeled data. So, this using unlabeled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning –** It is a machine learning algorithm that works between the supervised and unsupervised learning so it uses both labelled and unlabeled data. It's particularly useful when obtaining labeled data is costly, time-consuming, or resource-intensive. This involves using unlabeled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning –** his algorithm is a learning method that interacts with the environment by producing actions and discovering errors. This involves learning optimal actions through trial and error. So, the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

Advantages of Machine learning: -

1. Easily identifies trends and patterns: -Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation): -With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus software's; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement: -As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data, you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data*:* -Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications*:* -It could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning: -

1. Data Acquisition: -Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources: -ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function.

3. Interpretation of Results: -Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

CHAPTER-6

SYSTEM STUDY

6 SYSTEM STUDY

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

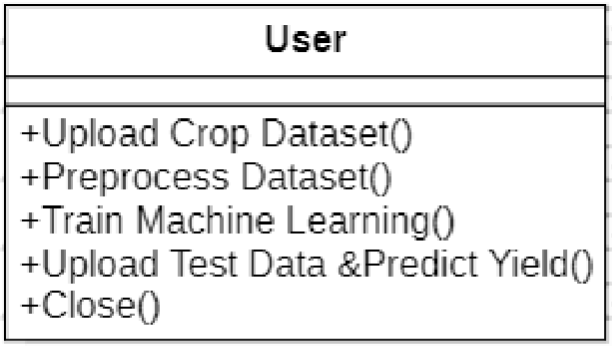

Fig 6.1 Feasibility Study

6.1 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.

6.2 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.

6.3 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 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.

CHAPTER-7:
SYSTEM TESTING

7 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, sub-assemblies, 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.

7.1 TYPES OF TESTS

7.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.1.3 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input: identified classes of valid input must be accepted.
- Invalid Input: identified classes of invalid input must be rejected.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised.
- Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for

testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.1.4 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 in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is 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, 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.

7.1.5 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.

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

7.1.6 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.

CHAPTER-8

RESULTS

8 RESULTS

To run project double click on ‘run.bat’ file to get below screen

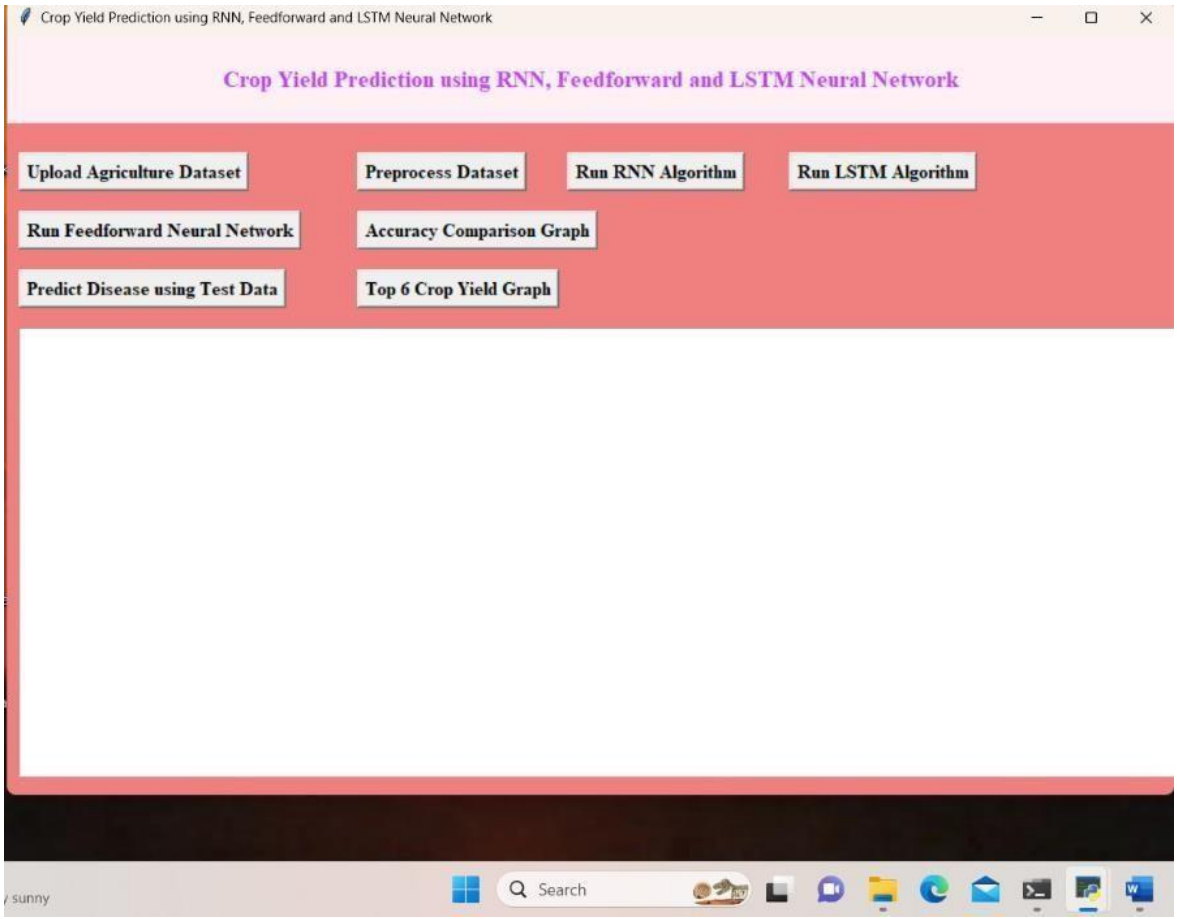

Fig 8.1 Result

In fig 8.1 Result screen click on ‘Upload Crop Dataset’ button to upload dataset and accordingly the output is seen accordingly

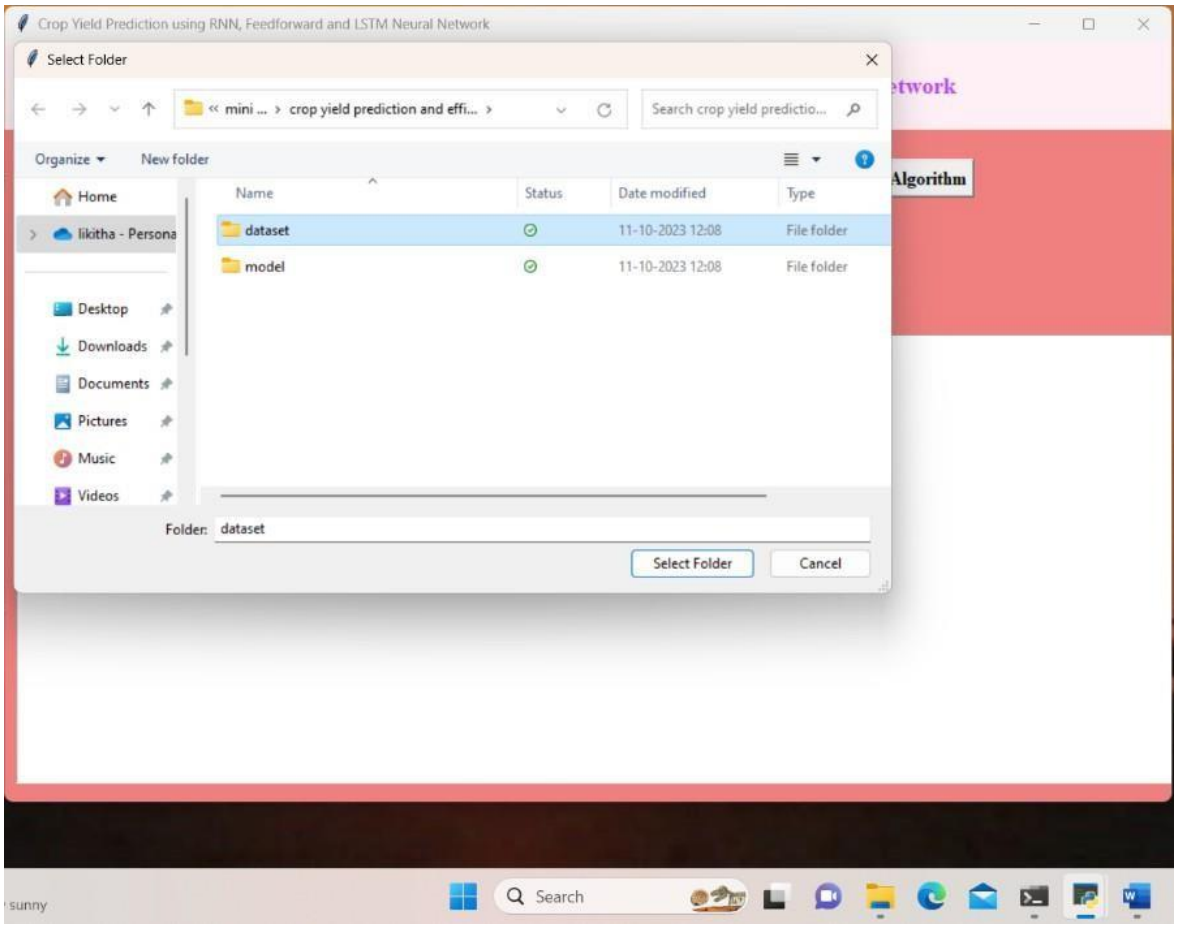

Fig 8.2 Result

In fig 8.2 Result screen selecting and uploading ‘Dataset.csv’ file and then click on ‘Open’ button to load dataset and to get below screen

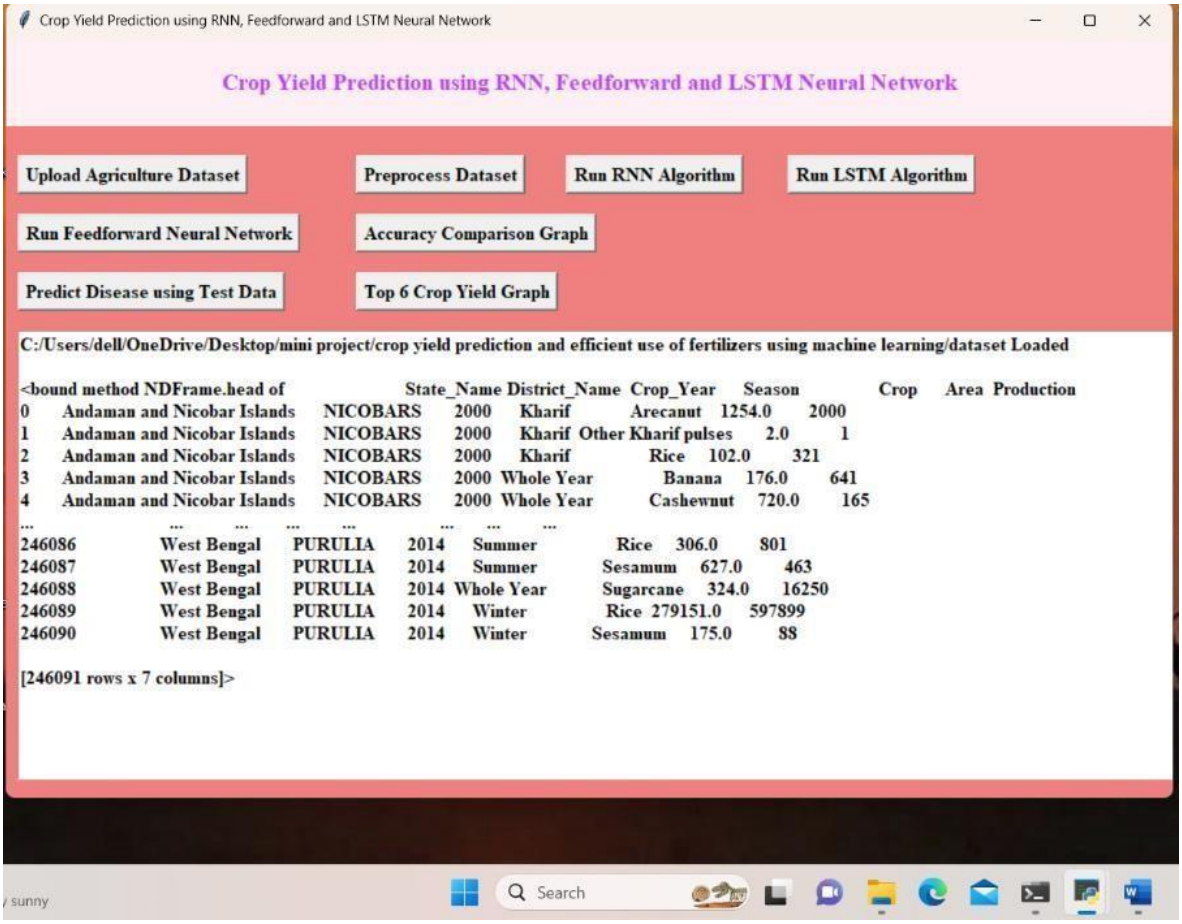

Fig 8.3 Result

In fig 8.3 Result screen dataset loaded and we can see dataset contains some non-numeric values and ML will not take non-numeric values so we need to preprocess dataset to convert non-numeric values to numeric values by assigning ID to each non-numeric value. So, click on 'Preprocess Dataset' button to process dataset.

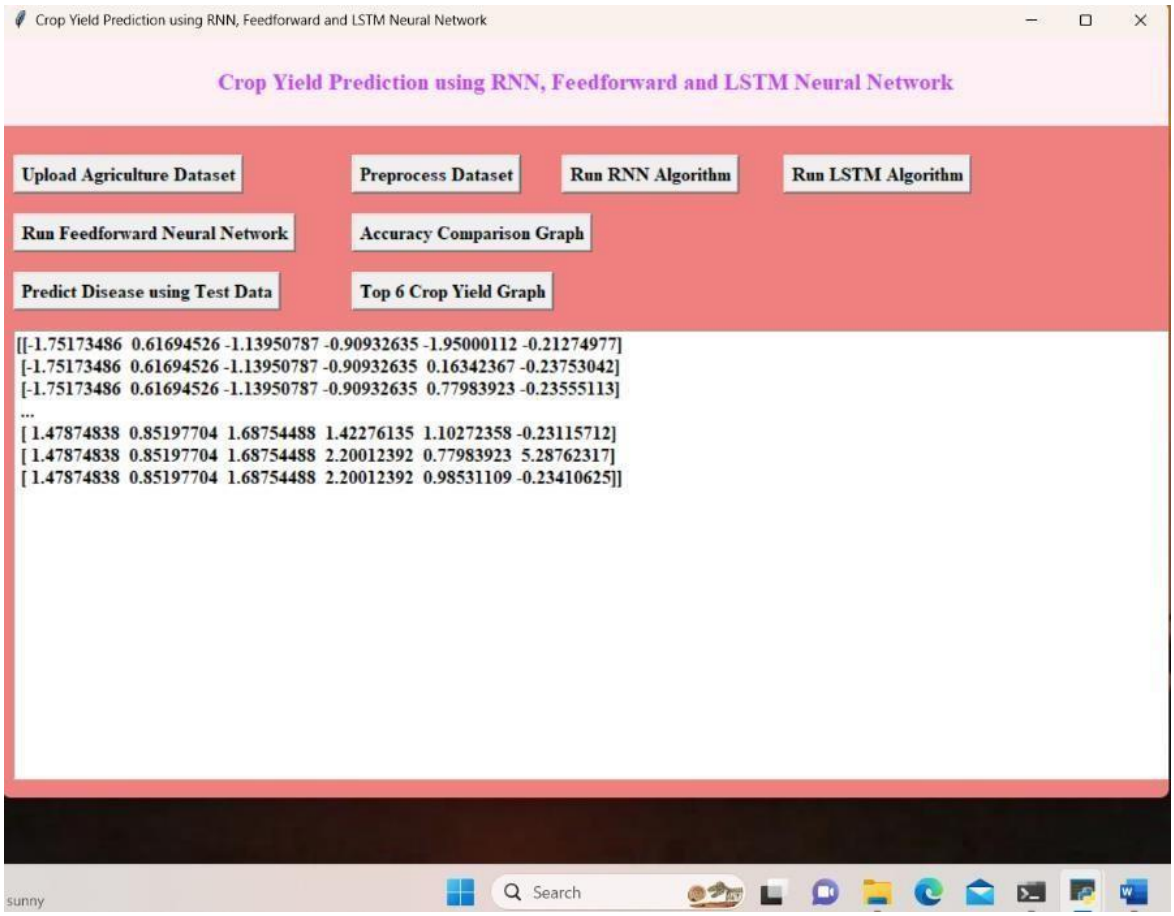

Fig 8.4 Result

In fig 8.4 Result screen all non-numeric values converted to numeric format and in below lines we can see dataset contains total 246091 records and application using (80%) 196872 records to train ML and using (20%) 49219 records to test ML prediction error rate (RMSE (root mean square error)). Now click on 'Train Machine Learning Algorithm' button to train Decision Tree Machine learning algorithm on above dataset and then calculate prediction error rate.

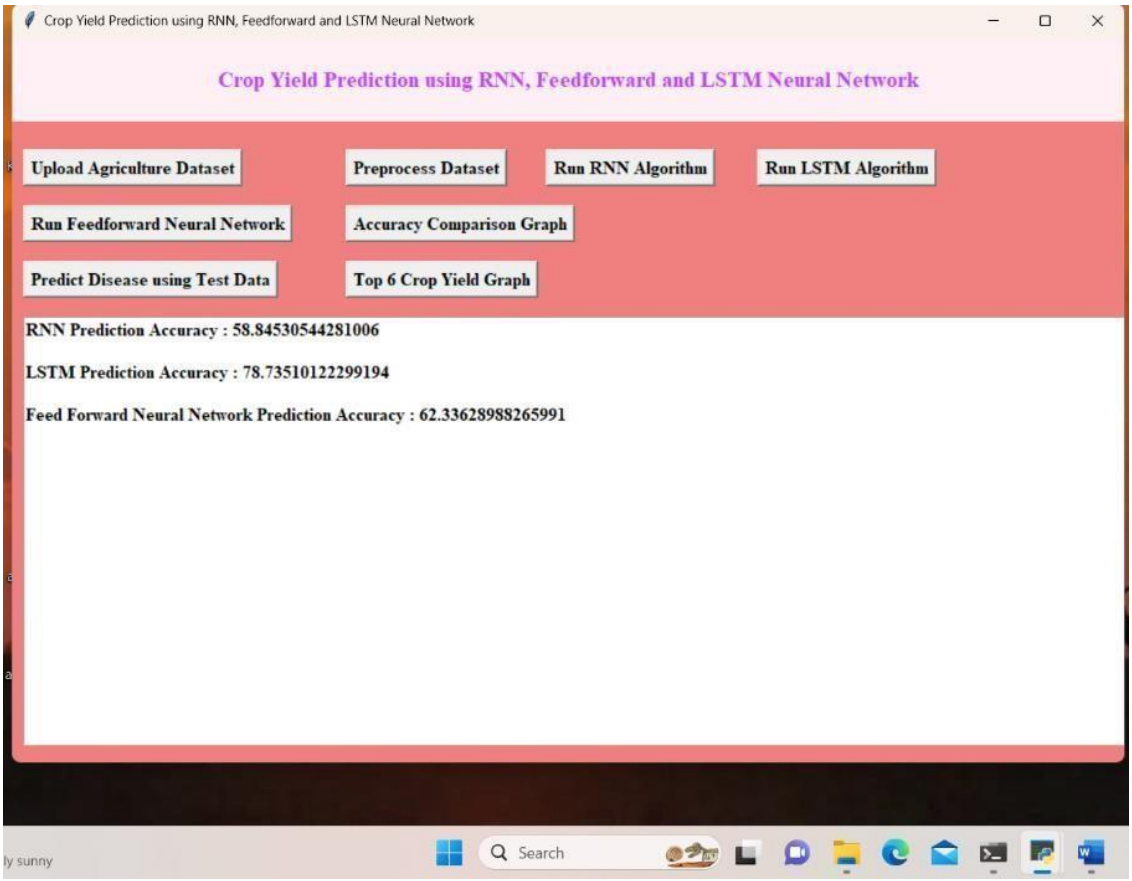

Fig 8.5 Result

After execution of this process the data will be running in the back end and the next screen is visible as the output.

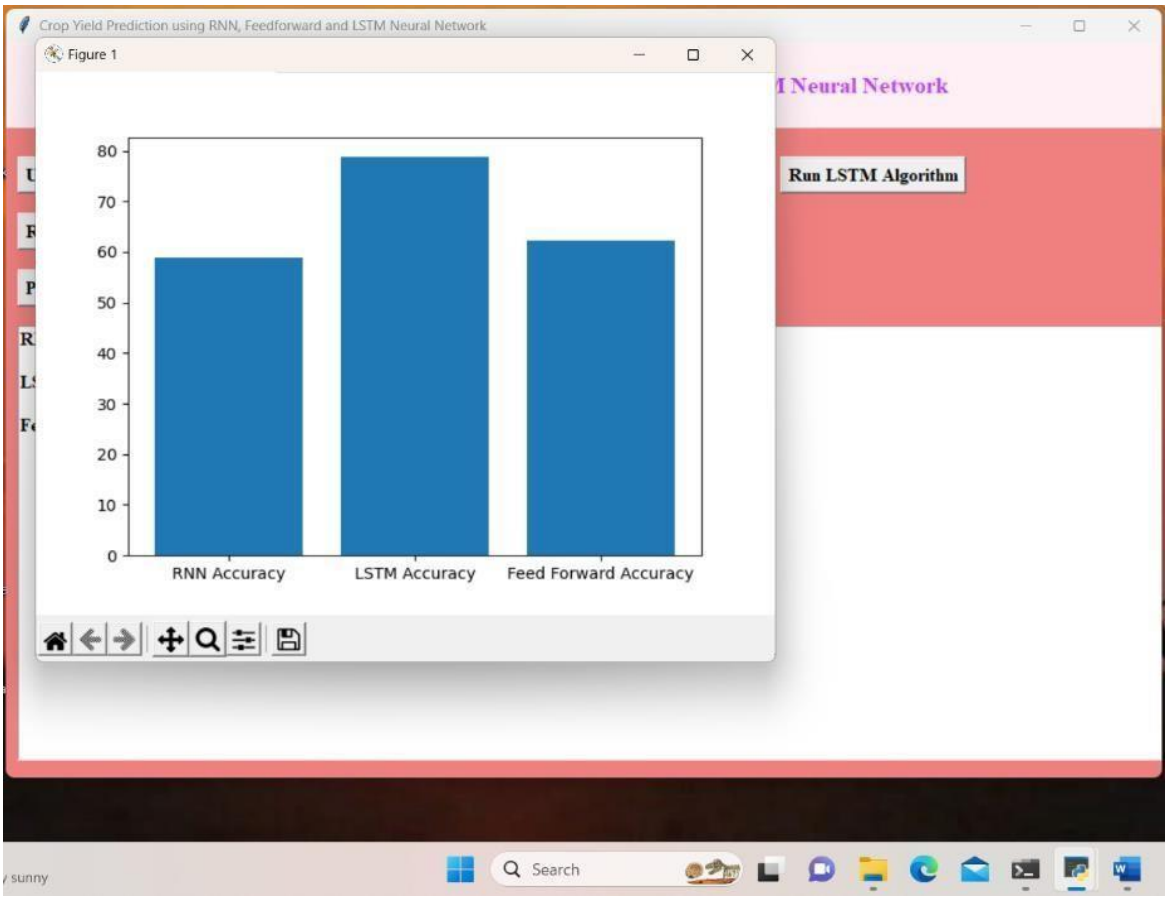

Fig 8.6 Result

In fig 8.6 Result screen ML is trained and we got prediction error rate as 0.067% and now Decision Tree model is ready and now click on 'Upload Test Data & Predict Yield' button to upload test data and then application will predict production

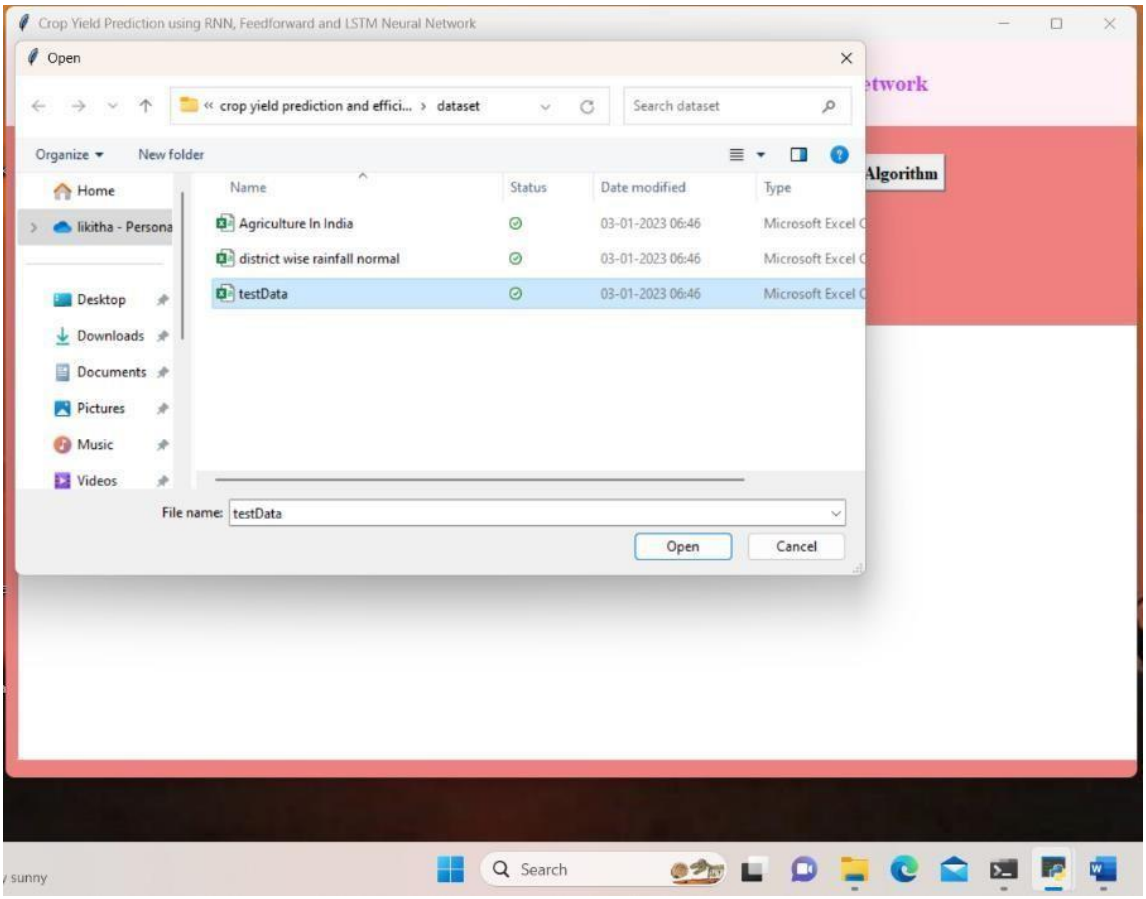

Fig 8.7 Result

In fig 8.7 Result screen selecting and uploading ‘test.csv’ file and then click on ‘Open’ button to load test data and then application will give below prediction result

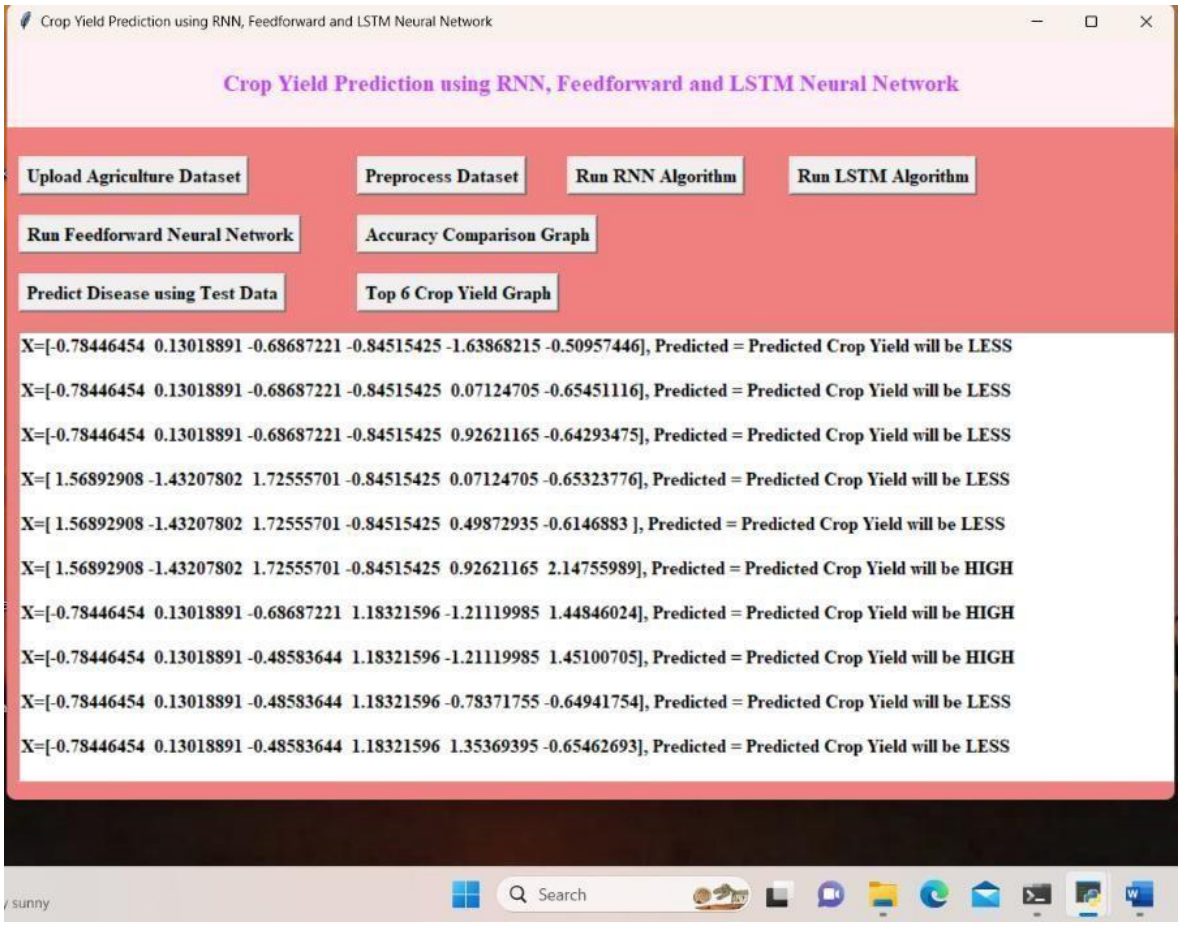

Fig 8.8 Result

In fig 8.8 Result screen each test record is separated with newline and in above screen in square bracket we can see test data values and after square bracket we can see predicted production and after that we can see predicted YIELD per acre. So, each test record and its prediction are separated with newline.

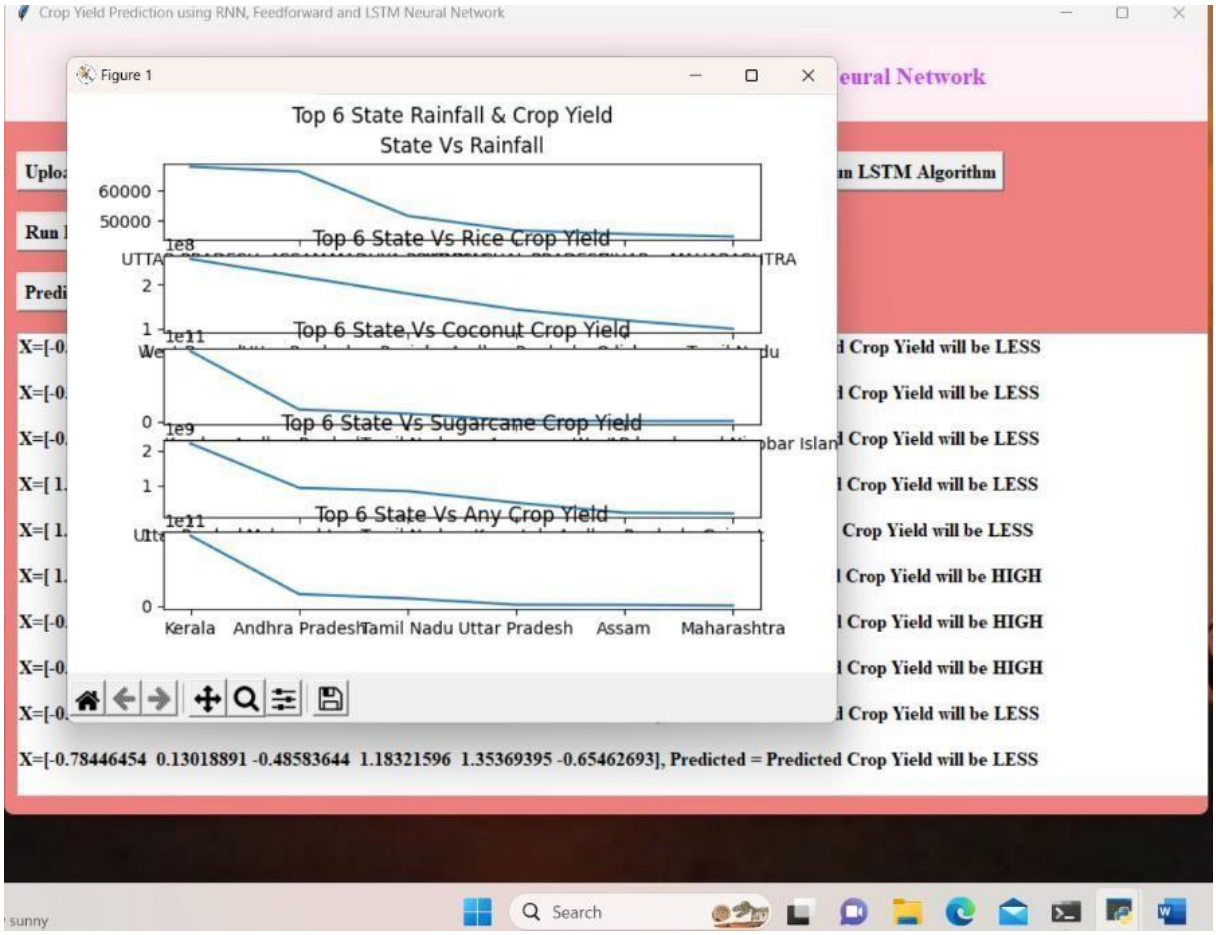

Fig 8.9 Result

From fig 8.9 Result graph the farmer may know what crop should be sown in order to get the good crop yield.

CHAPTER-9

CONCLUSION & FUTURE ENHANCEMENT

9 CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The paper presented the various machine learning algorithms for predicting the yield of the crop on the basis of temperature, rainfall, season and area. Experiments were conducted on Indian government dataset and it has been established that Random Forest Regressor gives the highest yield prediction accuracy. Sequential model that is Simple Recurrent Neural Network performs better on rainfall prediction while LSTM is good for temperature prediction. By combining rainfall, temperature along with other parameters like season and area, yield prediction for a certain district can be made. Results reveals that Random Forest is the best classifier when all parameters are combined. This will not only help farmers in choosing the right crop to grow in the next season but also bridge the gap between technology and the agriculture sector.

FUTURE ENHANCEMENT

Later on, every cultivating gadget can likewise be associated over the web utilizing IOT with web. The sensors in the gadget can be utilized in ranch which will gather the data about the present homestead conditions and what's more these gadgets can build the dampness, acridity, and so forth appropriately. The vehicles utilized in ranch like tractor can be associated with web in future which will help, progressively pass information to rancher about yield gathering and the contaminated harvests might be experiencing in this way helping the rancher sin making proper move. In addition, the best productive harvest can likewise be found considering the financial and expansion proportion.

CHAPTER 10

BIBLIOGRAPHY

10 BIBLOGRAPHY

1. Agriculture Role on Indian Economy Madhusudhan L - https://www.omicsonline.org/open-access/agriculture-role-on-indianeconomy-2151-6219-1000176.php?aid=62176
2. Priya, P., Muthaiah, U., Balamurugan, M. International Journal of Engineering Sciences Research Technology Predicting Yield of the Crop Using Machine Learning Algorithm.
3. Mishra, S., Mishra, D., Santra, G. H. (2016). Applications of machine learning techniques in agricultural crop production: a review paper. Indian J. Sci. Technol,9(38), 1-14.
4. Manjula, E., Djodiltachoumy, S. (2017). A Model for Prediction of Crop Yield. International Journal of Computational Intelligence and Informatics,6(4), 2349-6363.
5. Dahikar, S. S., Rode, S. V. (2014). Agricultural crop yield prediction using artificial neural network approach. International journal of innovative research in electrical, electronics, instrumentation and control engineering,2(1), 683-686.
6. Gonzlez Snchez, A., Frausto Sols, J., Ojeda Bustamante, W. (2014). Predictive ability of machine learning methods for massive crop yield prediction.
7. Mandic, D. P., Chambers, J. (2001). Recurrent neural networks for prediction: learning algorithms, architectures and stability. John Wiley Sons, Inc.
8. Hochreiter, S., Schmid Huber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
9. Sak, H., Senior, A., Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association.