**PREDICTION OF CROP RECOMMENDATION ON THE BASIS OF SOIL ANALYSIS**

**ABSTRACT**

In modern agriculture, optimizing crop selection is crucial for achieving higher yields and sustainable farming practices. This paper presents a novel approach to crop recommendation by leveraging machine learning techniques and soil analysis. The proposed system aims to assist farmers in making informed decisions about suitable crops for their specific soil conditions, thereby maximizing productivity and resource efficiency.

Preliminary evaluations of the system demonstrate promising results, indicating its potential to revolutionize traditional farming practices by integrating cutting-edge technology and data-driven decision-making. Future work involves expanding the dataset, refining the machine learning models, and incorporating additional environmental factors to further enhance the accuracy and applicability of the crop recommendation system.

**EXISTING SYSTEM**

There are several existing systems for crop recommendation that utilize soil analysis and machine learning techniques. These systems aim to provide farmers with accurate recommendations on suitable crops based on soil conditions. Here are some examples

1. \*\*Crop recommendation system using classification and regression\*\*:

- This system uses machine learning algorithms to predict suitable crops based on factors such as rainfall, temperature, and geolocation.

- The system employs classification and regression techniques to make accurate crop recommendations.

2. \*\*Model for predicting maize yield using machine learning tools\*\*:

- This system collects field data on maize yield from different countries and applies machine learning methods to predict yield in specific regions.

- The system utilizes machine learning tools to analyze and predict crop yield.

3. \*\*Optimal nutrition recommendation\*\*:

- Another system uses an exploration and exploitation method along with an improved genetic algorithm to recommend optimal nutrition for crops based on soil nutrient data.

- The system employs machine learning techniques to provide precise recommendations for crop nutrition.

4. \*\*Soil analysis and crop recommendation using machine learning\*\*:

- A study proposes a system that analyzes soil parameters to determine soil quality and suggests suitable crops based on the results obtained using a machine learning approach.

- The system utilizes the K-nearest neighbor classification algorithm to improve the efficiency of crop recommendation.

These are just a few examples of existing systems that combine soil analysis and machine learning for crop recommendation. These systems aim to assist farmers in making informed decisions about crop selection based on soil conditions, ultimately improving agricultural productivity.

**DISADVANTAGES OF EXISTING SYSTEM**

While crop recommendation systems based on soil analysis using machine learning offer significant advantages, they also come with certain disadvantages and challenges. Here are some notable drawbacks of existing systems:

1. \*\*Limited Data Availability:\*\*

- Many regions may lack comprehensive and up-to-date soil data. The effectiveness of machine learning models is highly dependent on the quality and quantity of the training data. Inadequate or biased data can lead to inaccurate recommendations.

2. \*\*Sensor and Technology Limitations:\*\*

- The accuracy of soil analysis depends on the quality and precision of the sensors and technologies used. Outdated or low-quality equipment may result in incomplete or inaccurate soil parameter measurements, impacting the reliability of the recommendations.

3. \*\*Complexity of Soil-Plant Interactions:\*\*

- Soil-plant interactions are complex and multifaceted. Machine learning models may struggle to capture all the nuances of these interactions, especially in situations where various factors influence crop growth, such as symbiotic relationships, microbial activity, and crop rotation practices.

4. \*\*Dynamic Environmental Conditions:\*\*

- Environmental conditions, including weather patterns, can change rapidly. Existing systems may face challenges in adapting quickly to these changes, affecting the accuracy of recommendations. Real-time adjustments based on current weather data may not be sufficient in some cases.

5. \*\*Dependency on Historical Data:\*\*

- Machine learning models often rely on historical data to make predictions. However, if there are significant changes in farming practices, land use, or climate, the historical data may become less relevant, leading to suboptimal recommendations.

6. \*\*Sensitivity to Outliers:\*\*

- Machine learning models can be sensitive to outliers or anomalies in the data. Unusual soil conditions or extreme weather events that are not well-represented in the training data may result in inaccurate predictions.

7. \*\*User Resistance and Adoption Challenges:\*\*

- Farmers may be hesitant to adopt new technologies or may lack the necessary skills to interpret machine-generated recommendations. Overcoming resistance and ensuring user-friendly interfaces are critical for successful implementation.

8. \*\*Resource Intensiveness:\*\*

- Implementing and maintaining a soil analysis system with machine learning capabilities can be resource-intensive. This includes the cost of soil testing equipment, data collection, model development, and ongoing maintenance.

9. \*\*Ethical and Privacy Concerns:\*\*

- The collection and use of data for crop recommendations may raise privacy concerns among farmers. Ensuring transparent data practices, obtaining informed consent, and addressing data security issues are essential for maintaining trust in the system.

Addressing these challenges requires ongoing research, technological advancements, and collaboration between stakeholders, including farmers, researchers, and technology developers, to improve the robustness and reliability of crop recommendation systems based on soil analysis using machine learning.

**PROPOSED SYSTEM**

The proposed system aims to enhance crop selection and optimize agricultural productivity by integrating advanced soil analysis techniques with state-of-the-art machine learning algorithms. The primary goal is to provide farmers with accurate and personalized crop recommendations based on their specific soil conditions, thereby promoting sustainable and efficient farming practices.

\*\*Components of the Proposed System:\*\*

1. \*\*Comprehensive Soil Analysis:\*\*

- Utilize advanced soil testing techniques to collect detailed information on key soil parameters, including nutrient levels, pH, moisture content, texture, and organic matter. Incorporate cutting-edge sensors and technologies for precise and real-time data collection.

2. \*\*Machine Learning Model Selection:\*\*

- Evaluate and select the most suitable machine learning algorithms, considering factors such as decision trees, support vector machines, and neural networks. The chosen models should effectively capture the complex relationships between soil properties and crop performance.

3. \*\*Integration of Environmental Factors:\*\*

- Expand the scope of recommendations by integrating additional environmental factors such as climate data, weather patterns, and historical performance of crops in the region. This ensures a holistic approach that considers external influences on crop growth.

4. \*\*Real-Time Data Updates:\*\*

- Implement mechanisms for real-time data updates, especially for weather conditions. This feature enables the system to adapt to changing environmental dynamics, ensuring that recommendations remain accurate and relevant.

5. \*\*User-Friendly Interface:\*\*

- Design an intuitive and user-friendly interface accessible via web or mobile applications. Farmers should be able to easily input their soil analysis results and receive personalized crop recommendations. The interface should also provide insights into the reasoning behind each recommendation.

6. \*\*Feedback Mechanism:\*\*

- Incorporate a feedback loop where farmers can provide information on the success and challenges of the recommended crops. This feedback will contribute to continuous improvement in the machine learning models and enhance the system's adaptability over time.

7. \*\*Customization Options:\*\*

- Allow farmers to input their specific preferences, constraints, and farming objectives. Customization options should consider factors such as crop rotation practices, water availability, and market demand, enabling the system to tailor recommendations to individual farming scenarios.

8. \*\*Data Security and Privacy Measures:\*\*

- Implement robust data security measures to protect sensitive information collected from farmers. Adhere to ethical standards and privacy regulations to build trust and ensure the responsible use of data.

9. \*\*Education and Support:\*\*

- Provide educational resources and support to help farmers understand the benefits of the system and build their confidence in adopting new technologies. Training programs and user assistance can contribute to successful implementation and adoption.

10. \*\*Scalability and Compatibility:\*\*

- Design the system to be scalable and compatible with diverse agricultural settings. Consider the varying scales of farming operations and the need for compatibility with different types of soil analysis equipment.

The proposed system seeks to revolutionize traditional farming practices by combining advanced soil analysis with machine learning, offering a tailored and intelligent approach to crop recommendations. Ongoing research, collaboration with agricultural experts, and continuous system refinement are crucial for the success of this innovative approach to precision agriculture.

**ADVANTAGES OF PROPOSED SYSTEM**

The proposed system of crop recommendation using soil analysis and machine learning offers several advantages that can significantly improve agricultural practices. Here are some key advantages:

1. \*\*Precision Agriculture:\*\*

- The system enables precision agriculture by providing accurate and personalized crop recommendations based on the specific soil conditions of each farm. This targeted approach helps optimize resource utilization and improve overall crop yield.

2. \*\*Optimized Resource Efficiency:\*\*

- By tailoring crop recommendations to the soil characteristics of each field, the proposed system helps farmers optimize the use of resources such as fertilizers, water, and pesticides. This not only reduces input costs but also minimizes environmental impact.

3. \*\*Increased Crop Yield and Quality:\*\*

- Improved crop selection based on advanced soil analysis and machine learning contributes to higher crop yields and better quality produce. Farmers can make informed decisions that align with the specific requirements of their soil, leading to healthier and more productive crops.

4. \*\*Adaptability to Changing Environmental Conditions:\*\*

- The integration of real-time environmental data allows the system to adapt to changing weather patterns and other external factors. This adaptability ensures that crop recommendations remain relevant and effective, even in dynamic agricultural environments.

5. \*\*Data-Driven Decision Making:\*\*

- Farmers can make informed decisions backed by data-driven insights. The machine learning models consider a wide range of soil parameters and environmental factors, providing a comprehensive analysis that goes beyond human intuition.

6. \*\*Sustainable Farming Practices:\*\*

- The proposed system supports sustainable farming practices by promoting efficient resource use, reducing the reliance on chemical inputs, and minimizing the environmental impact of agriculture. This aligns with global efforts to achieve more sustainable and eco-friendly farming methods.

7. \*\*Time and Cost Savings:\*\*

- Farmers can save time and reduce costs associated with trial-and-error approaches to crop selection. The system streamlines decision-making processes by automating the analysis of complex data, allowing farmers to focus on other critical aspects of their operations.

8. \*\*User-Friendly Interface:\*\*

- The user-friendly interface of the system makes it accessible to a wide range of farmers, including those with limited technical expertise. Easy input of soil analysis results and clear presentation of recommendations enhance user adoption and satisfaction.

9. \*\*Continuous Improvement through Feedback:\*\*

- The feedback loop incorporated into the system enables continuous improvement. Farmers' feedback on the success and challenges of recommended crops contributes to refining the machine learning models and enhancing the system's accuracy over time.

10. \*\*Customization for Diverse Farming Scenarios:\*\*

- The system allows farmers to customize recommendations based on their preferences, constraints, and farming objectives. This flexibility ensures that the system can adapt to diverse farming scenarios and cater to the unique needs of each farmer.

11. \*\*Enhanced Profitability:\*\*

- By optimizing crop selection and resource use, the proposed system contributes to enhanced profitability for farmers. Improved yields, reduced input costs, and sustainable practices can positively impact the economic viability of farming operations.

**MODULES DESCRIPTION**

The proposed system for crop recommendation using soil analysis and machine learning can be broken down into several modules, each serving a specific purpose in the overall functionality of the system. Here are the key modules with their descriptions:

1. \*\*Data Collection Module:\*\*

- \*Description:\* This module involves the collection of soil samples from agricultural fields. It includes the use of advanced soil testing techniques and sensors to gather data on key soil parameters such as nutrient levels, pH, moisture content, texture, and organic matter.

2. \*\*Data Preprocessing Module:\*\*

- \*Description:\* The collected soil data may need preprocessing to clean and transform it into a suitable format for machine learning analysis. This module involves data cleaning, normalization, and feature engineering to enhance the quality and effectiveness of the input data.

3. \*\*Machine Learning Model Training Module:\*\*

- \*Description:\* In this module, machine learning models are trained using historical data that includes soil characteristics and corresponding crop yields. Various machine learning algorithms, such as decision trees, support vector machines, or neural networks, are evaluated and the best-performing model is selected.

4. \*\*Real-Time Data Integration Module:\*\*

- \*Description:\* This module integrates real-time environmental data, such as weather conditions and climate patterns, into the system. The continuous updates ensure that the machine learning models can adapt to changing environmental conditions, enhancing the accuracy of crop recommendations.

5. \*\*Crop Recommendation Module:\*\*

- \*Description:\* The core of the system, this module utilizes the trained machine learning model to generate personalized crop recommendations based on the input soil analysis and real-time environmental data. The recommendations consider the specific requirements of the soil and external factors influencing crop growth.

6. \*\*User Interface Module:\*\*

- \*Description:\* The user interface module provides a user-friendly platform, accessible through web or mobile applications. Farmers can input their soil analysis results, view personalized crop recommendations, and receive insights into the reasoning behind each recommendation. The interface should be intuitive and easy to navigate.

7. \*\*Feedback and Improvement Module:\*\*

- \*Description:\* This module incorporates a feedback loop where farmers can provide information on the success and challenges of the recommended crops. The feedback is used to continuously improve the machine learning models, ensuring the system's adaptability and reliability over time.

8. \*\*Customization Module:\*\*

- \*Description:\* The customization module allows farmers to input specific preferences, constraints, and farming objectives. This information is considered in the crop recommendation process, enabling the system to tailor recommendations to individual farming scenarios.

9. \*\*Security and Privacy Module:\*\*

- \*Description:\* To address data security and privacy concerns, this module implements robust measures to protect sensitive information collected from farmers. It adheres to ethical standards and privacy regulations, ensuring the responsible use of data.

10. \*\*Educational and Support Module:\*\*

- \*Description:\* This module provides educational resources and user support to help farmers understand the benefits of the system and overcome any resistance to adopting new technologies. Training programs and assistance contribute to successful implementation and user adoption.

11. \*\*Scalability and Compatibility Module:\*\*

- \*Description:\* The scalability and compatibility module ensures that the system can scale to accommodate various agricultural settings. It considers the varying scales of farming operations and is designed to be compatible with different types of soil analysis equipment.

These modules work collaboratively to create an integrated and intelligent crop recommendation system that leverages soil analysis and machine learning to enhance precision, sustainability, and efficiency in agriculture.

**ABOUT SOFTWARE**

**PYTHON**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**Python Features**

Python has few keywords, simple structure, and a clearly defined syntax. Python code is more clearly defined and visible to the eyes. Python's source code is fairly easy-to-maintaining. Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh. Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

**Extendable**

It allows to add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

**Databases**

Python provides interfaces to all major commercial databases.

**GUI Programming**

Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

**Scalable**

Python provides a better structure and support for large programs than shell scripting.

**Object-Oriented Approach**

One of the key aspects of Python is its object-oriented approach. This basically means that Python recognizes the concept of class and object encapsulation thus allowing programs to be efficient in the long run.

**Highly Dynamic**

Python is one of the most dynamic languages available in the industry today. There is no need to specify the type of the variable during coding, thus saving time and increasing efficiency.

**Extensive Array of Libraries**

Python comes inbuilt with many libraries that can be imported at any instance and be used in a specific program.

**Open Source and Free**

Python is an open-source programming language which means that anyone can create and contribute to its development. Python is free to download and use in any operating system, like Windows, Mac or Linux.

**ANACONDA**

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system .The Anaconda distribution includes data-science packages suitable for Windows, Linux, and MacOS.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them.

It is available for Windows, MacOS and Linux.

**JUPYTER NOTEBOOK**

"Jupyter" is a loose acronym meaning Julia, Python, and R. These programming languages were the first target languages of the Jupyter application. As a server-client application, the Jupyter Notebook App allows you to edit and run your notebooks via a web browser. The application can be executed on a PC without Internet access, or it can be installed on a remote server and it can access through the Internet.

A kernel is a program that runs and introspects the user’s code. The Jupyter Notebook App has a kernel for Python code. "Notebook" or "Notebook documents" denote documents that contain both code and rich text elements, such as figures, links, equations. The mix of code and text elements, these documents are the ideal place to bring together an analysis description, and can be executed to perform the data analysis in real time.

Jupyter Notebook contains two components such as web application and notebook documents.

A web application is a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output.

Notebook documents is a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.Structure of a notebook document The notebook consists of a sequence of cells. A cell is a multiline text input field, and its contents can be executed by using Shift-Enter, or by clicking either the “Play” button the toolbar, or Cell , Run in the menu bar. The execution behavior of a cell is determined by the cell’s type. There are three types of cells namely code cells, markdown cells, and raw cells. Every cell starts off being a code cell, but its type can be changed by using a drop-down on the toolbar.

**Code cells**

A code cell allows you to edit and write new code, with full syntax highlighting and tab completion. The programming language you use depends on the kernel, and the default kernel (IPython) runs Python code.

**Markdown cells**

Document the computational process in a literate way, alternating descriptive text with code, using rich text. In IPython this is accomplished by marking up text with the Markdown language. The corresponding cells are called Markdown cells. The Markdown language provides a simple way to perform this text mark-up, to specify which parts of the text should be emphasized (italics), bold, form lists, etc.

**Raw cells**

Raw cells provide a place in which you can write output directly. Raw cells are not evaluated by the notebook. When passed through nbconvert, raw cells arrive in the destination format unmodified.

**MICROSOFT EXCEL**

Microsoft Excel is a spreadsheet developed by Microsoft for Windows, MacOS, Android and iOS. It features calculation, graphing tools, pivot tables and a macro programming language called Visual Basic for applications.

**FEATURES**

**Basic Operation**

Microsoft Excel has the basic features of all spreadsheets, using a grid of cells arranged in numbered rows and letter-named columns to organize data manipulations like arithmetic operations. It has a battery of supplied functions to answer statistical, engineering and financial needs. In addition, it can display data as line graphs, histograms and charts, and with a very limited three-dimensional graphical display. VBA programming The Windows version of Excel supports programming through Microsoft's Visual Basic for Applications (VBA), which is a dialect of Visual Basic. Programmers may write code directly using the Visual Basic Editor (VBE), which includes a window for writing code, debugging code, and code module organization environment. The user can implement numerical methods as well as automating tasks such as formatting or data organization in VBA and guide the calculation using any desired intermediate results reported back to the spreadsheet.

**Charts**

Excel supports charts, graphs, or histograms generated from specified groups of cells. The generated graphic component can either be embedded within the current sheet, or added as a separate object. These displays are dynamically updated if the content of cells change.

For example, suppose that the important design requirements are displayed visually; then, in response to a user's change in trial values for parameters, the curves describing the design change shape, and their points of intersection shift, assisting the selection of the best design.

Data storage and communication

Number of rows and columns

Versions of Excel up to 7.0 had a limitation in the size of their data sets of 16K (2 14 = 16384) rows. Versions 8.0 through 11.0 could handle 64K (2 16 = 65536) rows and 256 columns (2 8 as label 'IV'). Version 12.0 onwards, including the current Version 16.x, can handle over 1M (2 20 = 1048576) rows, and 16384 (2 14 as label 'XFD') columns.

**File formats**

Microsoft Excel up until 2007 version used a proprietary binary file format called Excel Binary File Format (.XLS) as its primary format. Excel 2007 uses Office Open XML as its primary file format, an XML-based format that followed after a previous XML-based format called "XML Spreadsheet" ("XMLSS"), first introduced in Excel 2002.

In addition, most versions of Microsoft Excel can read CSV, DBF, SYLK, DIF, and other legacy formats. Support for some older file formats was removed in Excel 2007. The file formats were mainly from DOS-based programs.

**Binary**

OpenOffice.org has created documentation of the Excel format. Since then Microsoft made the Excel binary format specification available to freely download.Export and migration of spreadsheets Programmers have produced APIs to open Excel spreadsheets in a variety of applications and environments other than Microsoft Excel. These include opening Excel documents on the web using either ActiveX controls, or plugins like the Adobe Flash Player.

**The Apache**

POI open source project provides Java libraries for reading and writing Excel spreadsheet files. Excel Package is another open-source project that provides server-side generation of Microsoft Excel 2007 spreadsheets. PHPExcel is a PHP library that converts Excel5, Excel 2003, and Excel 2007 formats into objects for reading and writing within a web application. Excel Services is a current .NET developer tool that can enhance Excel's capabilities. Excel spreadsheets can be accessed from Python with xlrd and openpyxl.

**CSV File**

A comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. A CSV file typically stores tabular data (numbers and text) in plain text, in which case each line will have the same number of fields. These files serve a few different business purposes. They help companies export a high volume of data to a more concentrated database.

The rules should be followed to format CSV file,

• Each record (row of data) is to be located on a separate line, delimited by a line break.

• The last record in the file may or may not have an ending line break.

• There may be an optional header line appearing as the first line of the file with the same format as normal record lines.

• It should contain the same number of fields as the records in the rest of the file.

• The header contains names corresponding to the fields in the file.

• In the header and each record, there may be one or more fields, separated by commas.

• The last field in the record must not be followed by a comma.

• Each field may or may not be enclosed in double quotes.

• If fields are not enclosed with double quotes, then double quotes may not appear inside the fields.

• Fields containing line breaks (CRLF), double quotes, and commas should be enclosed in double quotes.

• If double quotes are used to enclose fields, then a double quote appearing inside a field must be escaped by preceding it with another double quote.

**SYSTEM DESIGN**

The degree of interest in each concept has varied over the year, each has stood the test of time. Each provides the software designer with a foundation from which more sophisticated design methods can be applied. Fundamental design concepts provide the necessary framework for “getting it right”.

During the design process the software requirements model is transformed into design models that describe the details of the data structures, system architecture, interface, and components. Each design product is reviewed for quality before moving to the next phase of software development.

**INPUT DESIGN**

The design of input focus on controlling the amount of dataset as input required, avoiding delay and keeping the process simple. The input is designed in such a way to provide security. Input design will consider the following steps:

• The dataset should be given as input.

• The dataset should be arranged.

• Methods for preparing input validations.

**OUTPUT DESIGN**

A quality output is one, which meets the requirement of the user and presents the information clearly. In output design, it is determined how the information is to be displayed for immediate need.

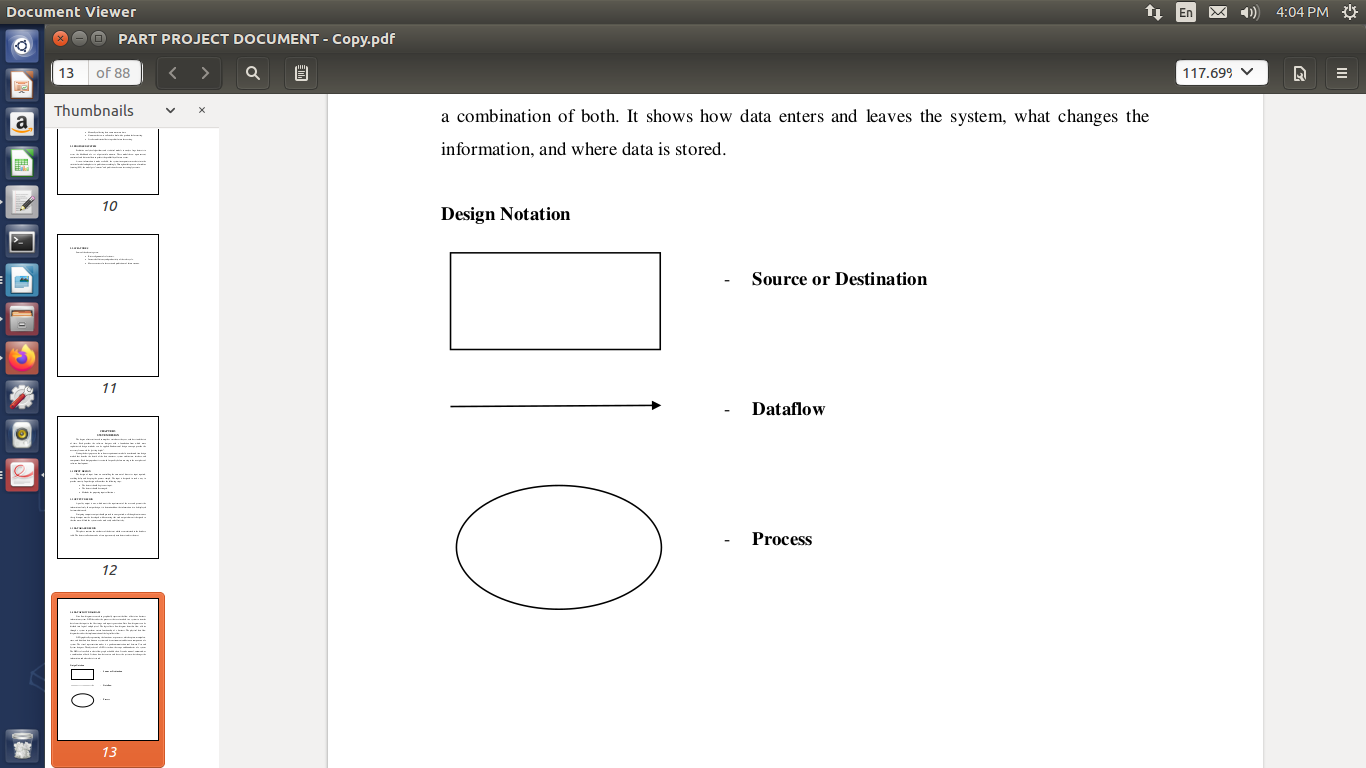
Designing computer output should proceed in an organized, well thought out manner;the right output must be developed while ensuring that each output element is designed so that the user will find the system can be used easily and effectively.

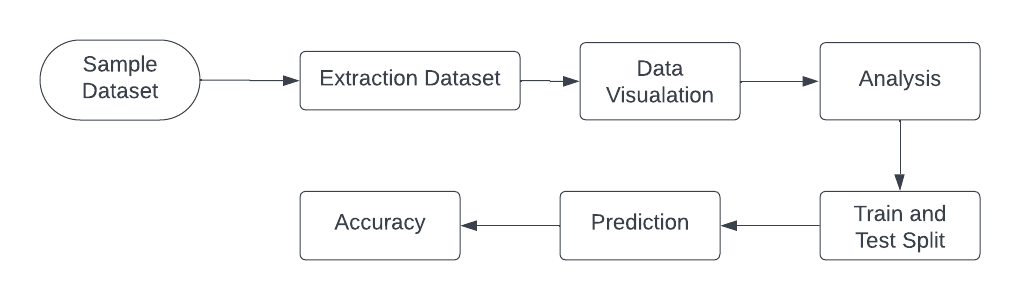
**DATAFLOW DIAGRAM**

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. The objective of a DFD is to show the scope and boundaries of a system.The DFD is also called as a data flow graph or bubble chart. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored.

**Design Notation**





**SYSTEM DEVELOPMENT**

**Libraries in Python**

Python library is vast. There are built in functions in the library which are written in C language. This library provide access to system functionality such as file input output and that is not accessible to Python programmers. This modules and library provide solution to the many problems in programming.

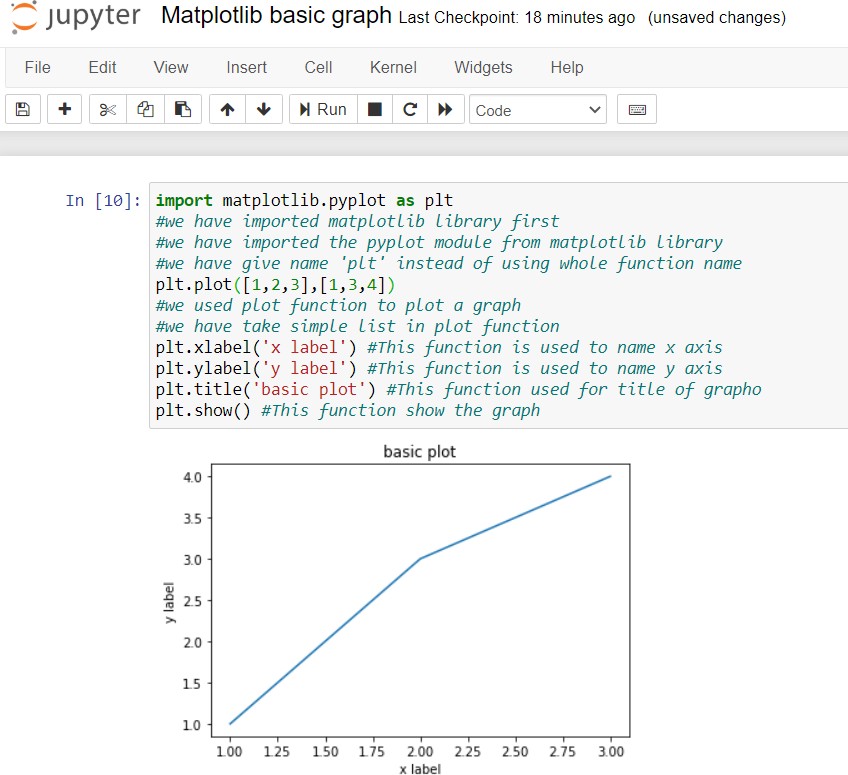
Following are some Python libraries.

* Matplotlib
* Pandas
* Numpy

## Matplotlib

”Matplotlib is a plotting library for the Python programming language and its numerical math- ematics extension [NumPy”[11].](file:///C:\Users\HP\Downloads\Drug%20Discovery-REP.docx#_bookmark98) Matlab provides an application that is used in graphical user interface tool kits. Another such libraby is pylab which is almost same as MATLAB.

It is a library for 2D graphics, it finds its application in web application servers, graphical user interface toolkit and shell.Below is the example of a basic plot in python.



Matplotlib basic example

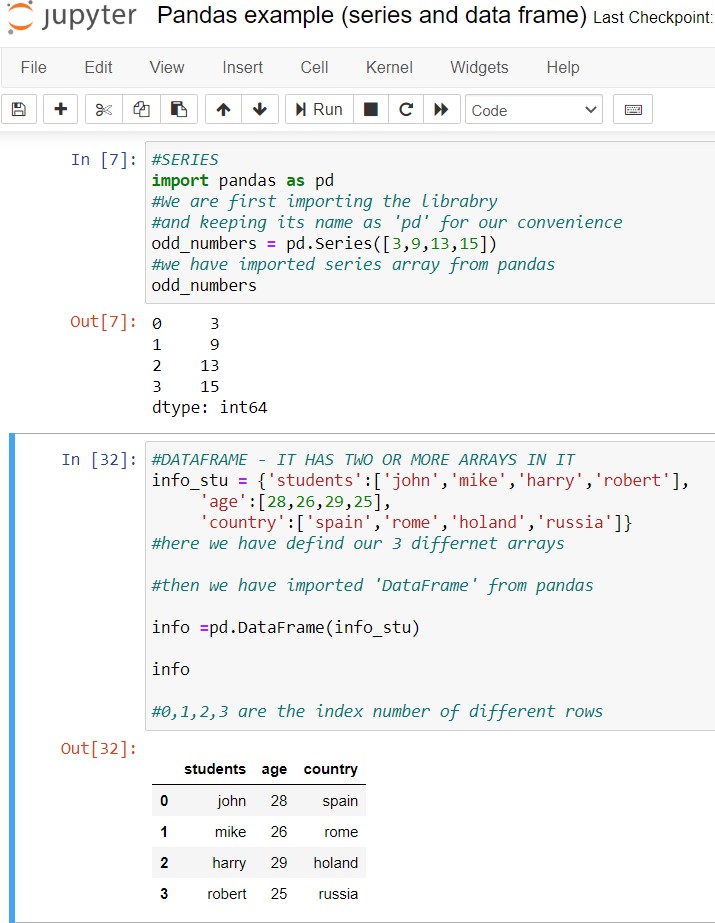
## Pandas

Pandas is also a library or a data analysis tool in python which is written in python program- ming language. It is mostly used for data analysis and data manipulation. It is also used for data structures and time series.

We can see the application of python in many fields such as - Economics, Recommendation Systems - Spotify, Netflix and Amazon, Stock Prediction, Neuro science, Statistics, Advertising, Analytics, Natural Language Processing. Data can be analyzed in pandas in two ways -

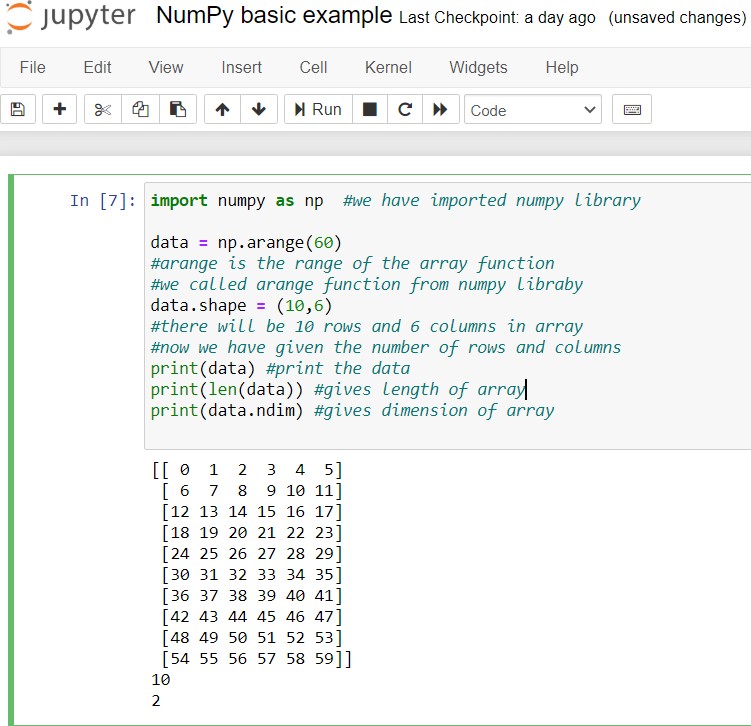
**Data frames -** In this data is two dimensional and consist of multiple series. Data is always represented in rectangular table.

**Series -** In this data is one dimensional and consist of single list with index.



NumPy

”NumPy is a library for the Python programming language, adding support for large, multi- dimensional arrays and matrices, along with a large collection of high-level mathematical func- tions to operate on these arrays”. The previous similar programming of NumPy is Numeric, and this language was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. [[12]](file:///C:\Users\HP\Downloads\Drug%20Discovery-REP.docx#_bookmark99) It is an open source library and free of cost.



NumPy basic example

**DESCRIPTION OF MODULES**

* DATASET COLLECTION
* HYPOTHESIS DEFINITION
* DATA EXPLORATION
* DATA CLEANING
* FEATURE ENGINEERING

**DATASET COLLECTION**

Before analyzing and visualization we need the raw data and this raw data can gathered from different open source data websites available on the internet. This data will be in raw form, it may be the PV solar panel sales, renewable energy consumption or production in any specific area or regions where solar or wind which one is more favorable. As here we are focusing on the renewable energy data sets so we will be considering following websites where this data is available.

<https://www.eia.gov/>

This website contains the energy data mostly of US. EIA is the abbreviated form of Energy Information Administration. Here we have different data of prices, consumption, production, exports and imports of the energy data.

<https://www.energy.gov/>

Energy.gov is the other website for data related to renewable energy, This Energy Department is responsible to make sure USA’s Energy Future and solve the energy related problems

<https://openei.org/>

Open Energy Information is a website for policy makers, researchers, technology investors, venture capitalists, and market professionals with energy data, information, analyzes, tools, images, maps, and other resources.

<https://data.world/>

Here we can find data related to each and every field, it is the most widely used website for data analysis. We can also gather energy related data from this website.

<https://catalog.data.gov/dataset>

Data.gov is powered by two open source applications, CKAN and WordPress, and it is de- veloped publicly on GitHub.Data.gov is managed and hosted by the U.S. General Services Administration, Technology Transformation Service.

<https://www.kaggle.com/>

The data sets available here is not specifically for renewable energy. Kaggle is general data sets website, here you can get generalized data.

**HYPOTHESIS DEFINITION**

This is a very important step to analyse any problem. The first and foremost step is to understand the problem statement. The idea is to find out the factors of a product that creates an impact on the sales of a product. A null hypothesis is a type of hypothesis used in statistics that proposes that no statistical significance exists in a set of given observations.

An alternative hypothesis is one that states there is a statistically significant relationship between two variables.

**DATA EXPLORATION**

Data exploration is an informative search used by data consumers to form true analysis from the information gathered. Data exploration is used to analyse the data and information from the data to form true analysis. After having a look at the dataset, certain information about the data was explored. Here the dataset is not unique while collecting the dataset. In this module, the uniqueness of the dataset can be created.

**DATA CLEANING**

In data cleaning module, is used to detect and correct the inaccurate dataset. It is used to remove the duplication of attributes. Data cleaning is used to correct the dirty data which contains incomplete or outdated data, and the improper parsing of record fields from disparate systems. It plays a significant part in building a model.

**FEATURE ENGINEERING**

In the feature engineering module, the process of using the import data into machine learning algorithms to predict the accurate directions. A feature is an attribute or property shared by all the independent products on which the prediction is to be done. Any attribute could be a feature, it is useful to the model.

**VALIDATION TESTING**

Validation testing begins at the culmination of integration testing, when individual components have been exercised, the software is completely assembled as a package. The testing focuses on user visible actions and user recognizable output from the system. The testing has been conducted on possible condition such as the function characteristic conforms the specification and a deviation or error is uncovered. The alpha test and beta test is conducted at the developer site by end-users.

**Machine Learning Algorithm used:**

**Support Vector Machines (SVM)**

**SUPPORT VECTOR MACHINES**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. which there are two different categories that are classified using a decision boundary or hyperplane:

**Example:** SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat.

SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

## Types of SVM

**SVM can be of two types:**

* **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
* **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

**Hyperplane and Support Vectors in the SVM algorithm:**

**Hyperplane:** There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

**Support Vectors:**

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

## How does SVM works?

**Linear SVM:**

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue.

So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes.

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.

**Non-Linear SVM:**

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:

So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

**z=x2 +y2**

By adding the third dimension,

So now, SVM will divide the datasets into classes in the following way.

Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:

Hence we get a circumference of radius 1 in case of non-linear data.

**CONCLUSION**

In conclusion, the implementation of a crop recommendation system based on soil analysis using machine learning holds immense potential to revolutionize modern agriculture. The synergy of advanced soil analysis techniques and cutting-edge machine learning algorithms offers a promising solution to address the challenges faced by farmers in optimizing crop selection and resource utilization

The integration of soil analysis and machine learning in crop recommendation systems represents a transformative step towards precision agriculture. As technology continues to advance, further research, development, and collaboration between the agricultural and technology sectors will be essential to fully realize the potential of this innovative approach. The proposed system offers a pathway to sustainable, efficient, and data-driven agriculture that addresses the challenges of the modern farming landscape.

**BIBLOGRAPHY**

1. Food and Agriculture Organization of the United Nations. (2018). "Guidelines for soil description." Available online: http://www.fao.org/3/i6653e/i6653e.pdf

2. Jones, R. J. A., Hiederer, R., Rusco, E., & Montanarella, L. (2005). "Estimating organic carbon in the soils of Europe for policy support." European Journal of Soil Science, 56(5), 655-671.

3. Lal, R. (2004). "Soil carbon sequestration impacts on global climate change and food security." Science, 304(5677), 1623-1627.

4. Malhi, S. S., Johnston, A. M., Schoenau, J. J., & Zentner, R. P. (2006). "Soil amendments and crop sequence effects on spring wheat production and soil chemical properties." Canadian Journal of Soil Science, 86(1), 57-68.

5. Martin-Neto, L., Lima, E., Colnago, L. A., & Milori, D. M. B. P. (2007). "Hummification process of soil organic matter as viewed by 13C CPMAS NMR spectroscopy and chemical analysis." Geoderma, 139(1-2), 32-39.

6. Powlson, D. S., Whitmore, A. P., & Goulding, K. W. T. (2011). "Soil carbon sequestration to mitigate climate change: a critical re-examination to identify the true and the false." European Journal of Soil Science, 62(1), 42-55.

7. Reicosky, D. C., & Forcella, F. (1998). "Cover crop and soil quality interactions in agroecosystems." Journal of Soil and Water Conservation, 53(3), 224-229.

8. Stenberg, M., Viscarra Rossel, R. A., Mouazen, A. M., & Wetterlind, J. (2010). "Visible and near infrared spectroscopy in soil science." Advances in Agronomy, 107, 163-215.

9. Su, Y., He, Y., Li, Y., Jiang, X., Liu, F., Liu, J., ... & Huang, Y. (2017). "Estimation of soil organic matter content using hyperspectral remote sensing based on different spectral preprocessing methods." Remote Sensing, 9(4), 356.

10. Teixeira, E. I., Fischer, G., Van Velthuizen, H., Walter, C., & Ewert, F. (2013). "Global hot-spots of heat stress on agricultural crops due to climate change." Agricultural and Forest Meteorology, 170, 206-215.

These resources cover various aspects of soil analysis, crop suitability, and climate change impacts on agriculture, which are all relevant to developing a crop recommendation system based on soil analysis.