TRAINING REPORT

OF

SUMMER TRAINING,

UNDERTAKEN AT

NATIONAL INSTITUTE OF ELECTRONICS & INFORMATION TECHNOLOGY(NIELIT)

ON

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

PROJECT: CROP PREDICTION

SUBMITTED BY:

Sukhpreet Singh & Yashpreet Singh

Abstract

Crop prediction plays a pivotal role in modern agriculture by providing valuable insights into crop yields, growth patterns, and potential challenges. This abstract summarizes a comprehensive study on crop prediction, which combines advanced data analysis techniques, machine learning models, and multi-source data integration to accurately forecast crop production.

The study employs historical agricultural data, weather information, soil characteristics, and satellite imagery to build robust predictive models. These models offer accurate yield forecasts, empowering farmers and policymakers to make informed decisions on crop planning, resource allocation, and risk management.

By promoting sustainable agricultural practices, the project emphasizes efficient resource utilization and environmental conservation. Addressing challenges related to climate change, unpredictable weather patterns, and market demands, the study contributes to long-term food security.

The abstract highlights the significance of continuous data collection and updates for model improvement, ensuring adaptability to dynamic agricultural conditions. Moreover, the study explores collaboration with agricultural experts to align findings with real-world practices and challenges.

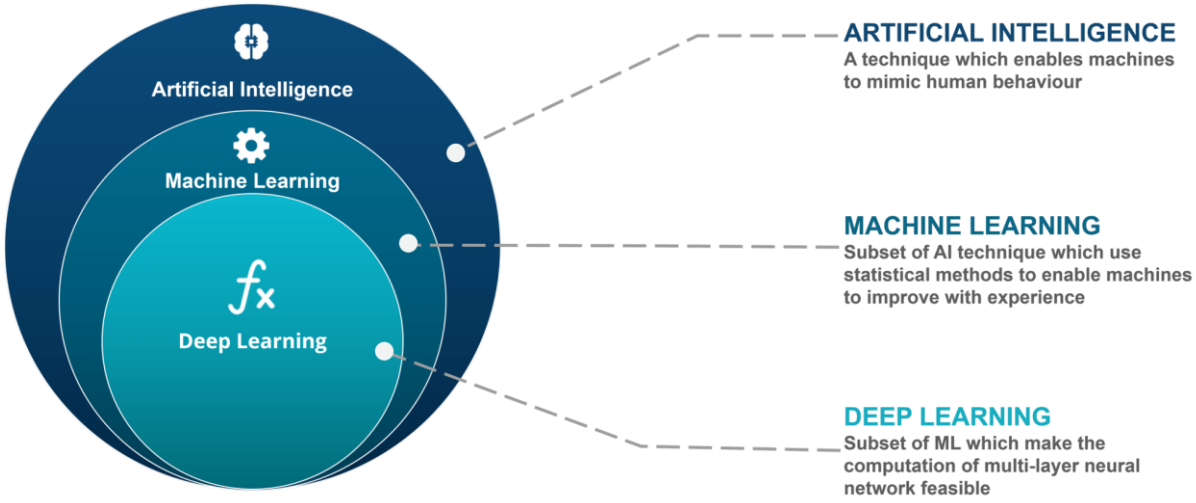
The future scope of the project involves enhancing model accuracy, expanding to new crop types and regions, integrating IoT and sensor data, and adopting advanced deep learning techniques. Additionally, it delves into forecasting climate change impacts, precision agriculture integration, and market dynamics analysis

Table of content

CHAPTER 1 – INTRODUCTION

1.1 Artificial Intelligence and Machine learning

Artificial Intelligence and Machine Learning are two rapidly evolving fields at the forefront of technological advancements. They both revolve around the idea of creating intelligent systems capable of performing tasks that traditionally require human intelligence. While AI is a broader concept, encompassing a wide range of techniques and applications, Machine Learning is a subset of AI that focuses on creating algorithms and models that enable machines to learn and improve from experience.



1. **Artificial Intelligence (AI):**

Artificial Intelligence is the theory and development of computer systems capable of performing tasks that typically require human intelligence. These tasks can include reasoning, problem-solving, perception, learning, language understanding, and decision making. The ultimate goal of AI is to create machines that can mimic, simulate, or even surpass human intelligence in various domains. AI systems can be broadly categorized into two types:

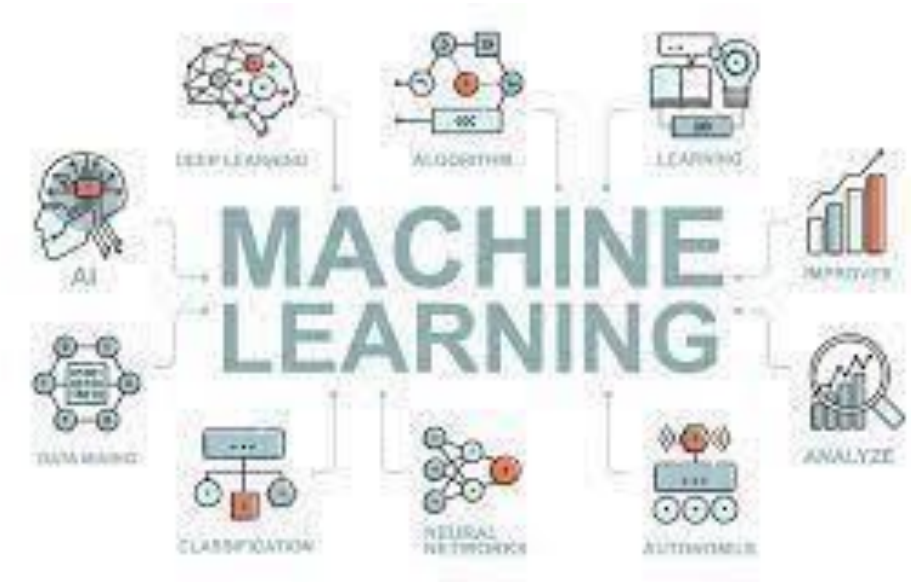
a. **Narrow AI (Weak AI):** This type of AI is designed to perform specific tasks or solve particular problems. Examples include virtual personal assistants like Siri, speech recognition systems, and recommendation algorithms used by online platforms.

b. **General AI (Strong AI):** General AI aims to possess human-like intelligence, enabling machines to understand, learn, and perform any intellectual task that a human being can do.

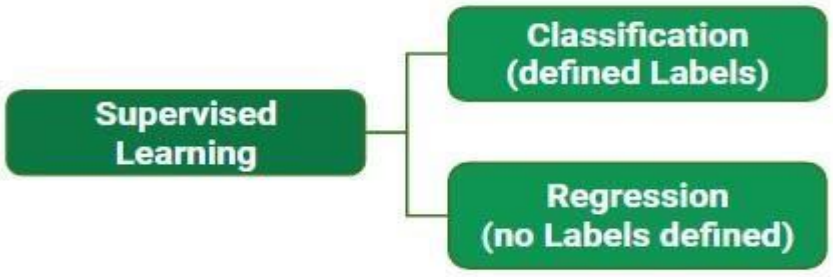
Achieving this level of AI remains a significant challenge and is still largely theoretical.

2. **Machine Learning (ML):**

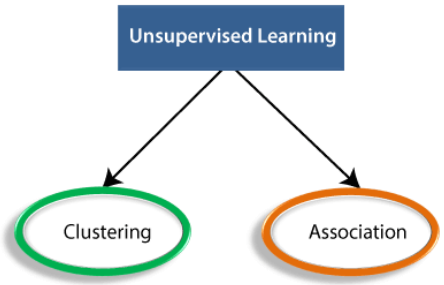
Machine Learning is a subset of AI that focuses on developing algorithms and statistical models to enable computers to learn and improve from experience automatically. Instead of being explicitly programmed, these algorithms learn patterns from data and make predictions or decisions based on that learning. There are three main types:



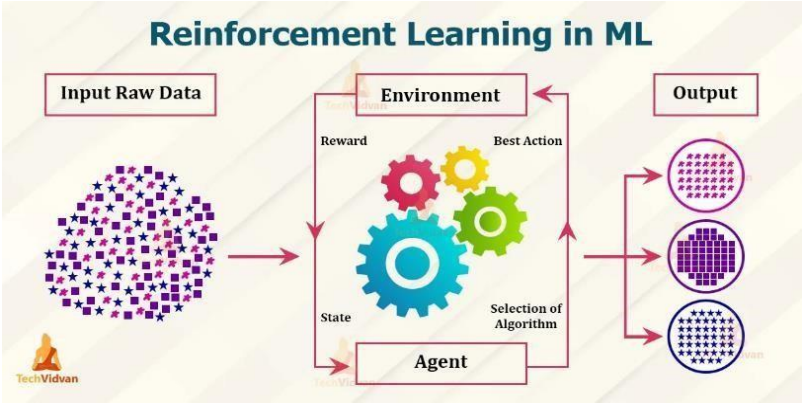
a. **Supervised Learning**: In this approach, the algorithm is trained on labelled data, where the correct output is provided. The model learns to map inputs to outputs, making it capable of making predictions on new, unseen data.



b. **Unsupervised Learning**: Unsupervised learning deals with unlabelled data, where the algorithm tries to find patterns and relationships within the data without specific guidance. Clustering and dimensionality reduction are common tasks in unsupervised learning.



a. **Reinforcement Learning**: This learning paradigm involves an agent interacting with an environment and learning from feedback (rewards or penalties) received based on its actions. The agent aims to maximize its cumulative reward over time.



1.2 Applications of Machine Learning

Machine Learning has a wide range of applications across various industries and domains. Its ability to analyse data, recognize patterns, and make predictions has led to numerous practical use cases. Here are some popular applications of Machine Learning:

1. **Image and Video Analysis:**

Machine Learning is used in image recognition and object detection tasks. Applications include facial recognition, automated tagging of images, detecting objects in photos, and medical image analysis for diagnosis.

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

NLP enables machines to understand, interpret, and generate human language. Applications include sentiment analysis, chatbots, language translation, speech recognition, and text summarization.

3. **Recommender Systems:**

Recommender systems use Machine Learning to suggest products, movies, music, or content based on a user's preferences and behaviour. Examples include personalized recommendations on streaming platforms and e-commerce product recommendations.

4. **Autonomous Vehicles**:

Machine Learning plays a crucial role in developing self-driving cars. It helps in object detection, lane detection, real-time decision-making, and mapping for navigation.

5. **Healthcare**:

Machine Learning is used for medical image analysis, disease diagnosis, drug discovery, personalized treatment plans, and predicting patient outcomes.

6. **Financial Services**:

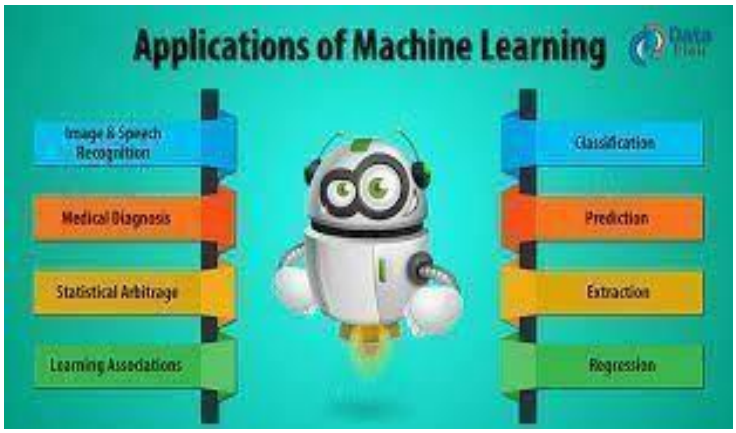
In finance, Machine Learning is applied for credit scoring, fraud detection, algorithmic trading, risk assessment, and customer service chatbots.

7. **Predictive Maintenance**:

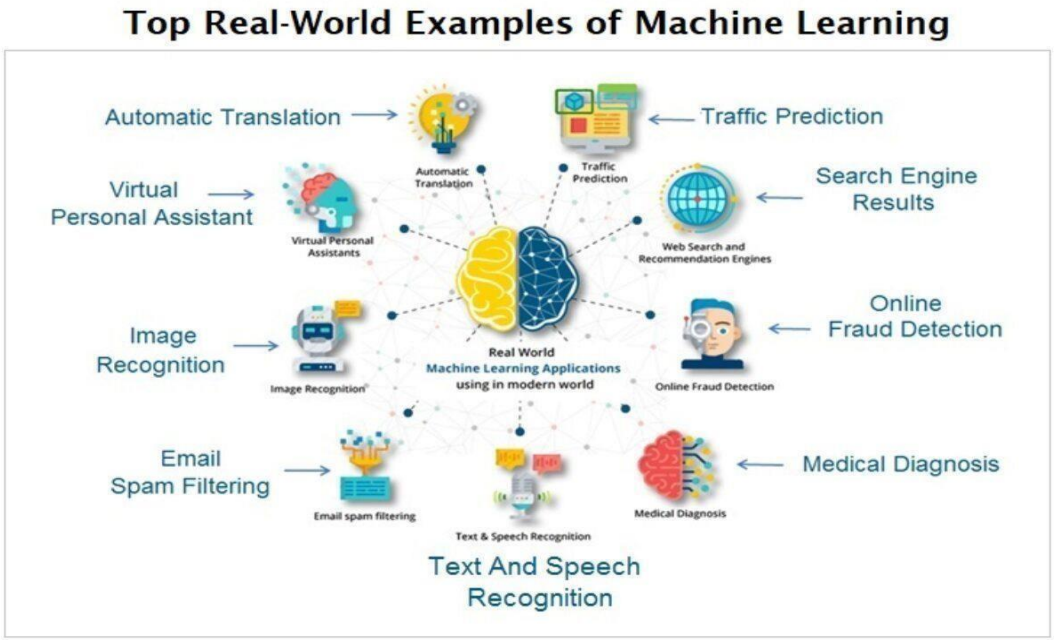
In manufacturing and industrial sectors, Machine Learning is used for predictive maintenance, optimizing equipment maintenance schedules, and reducing downtime.

8. **Fraud Detection**:

Machine Learning models can detect fraudulent transactions, account takeovers, and identify unusual patterns in financial activities.



1.3 Real World Examples of Machine Learning



1. **Speech & Image Recognition**

Computer Speech Recognition *or* Automatic Speech Recognition helps to convert speech into text. Many applications convert the live speech into an audio file format and later convert it into a text file.

Similar to speech recognition, Image recognition is also the most widely used example of Machine Learning technology that helps identify any object in the form of a digital image.

2. **Traffic alerts using Google Map**

Google map uses different technologies, including machine learning which collects information from different users, analyse that information, update the information, and make predictions. With the help of predictions, it can also tell us the traffic before we start our journey.
Machine Learning also helps identify the best and fastest route while we are in traffic using Google Maps

3. **Chatbot (Online Customer Support)**

A chatbot is the most widely used software in every industry like banking, Medical, education, health, etc. You can see chatbots in any banking application for quick online support to customers. These chatbots also work on the concepts of Machine Learning. The programmers feed some basic questions and answers based on the frequently asked queries. So, whenever a customer asks a query, the chatbot recognizes the question's keywords from a database and then provides appropriate resolution to the customer. This helps to make quick and fast customer service facilities to customers.

4. **Google Translation**

Suppose you work on an international banking project like French, German, etc., but you only know English. In that case, this will be a very panic moment for you because you can't proceed further without reviewing documents. Google Translator software helps to translate any language into the desired language. So, in this way, you can convert French, German, etc., into English, Hindi, or any other language. This makes the job of different sectors very easy as a user can work on any country's project hassle-free.

5. **Prediction**

Prediction system also uses Machine learning algorithms for making predictions. There are various sectors where predictions are used. For example, in bank loan systems, error probability can be determined using predictions with machine learning. For this, the available data are classified into different groups with the set of rules provided by analysts, and once the classification is done, the error probability is predicted.

6**. Extraction**

One of the best examples of machine learning is the extraction of information. In this process, structured data is extracted from unstructured data, and which is used in predictive analytics tools. The data is usually found in a raw or unstructured form that is not useful, and to make it useful, the extraction process is used.

7. **Self-driving cars**

The future of the automobile industry is self-driving cars. These are driverless cars, which are based on concepts of deep learning and machine learning. Some commonly used machine learning algorithms in self-driving cars are Scale-invariant feature transform (SIFT), AdaBoost, TextonBoost, YOLO (You only look once).

8. **Ads Recommendation**

Nowadays, most people spend multiple hours on google or the internet surfing. And while working on any webpage or website, they get multiples ads on each page. But these ads are different for each user even when two users are using the same internet and on the same location. These ads recommendations are done with the help of machine learning algorithms. These ads recommendations are based on the search history of each user. For example, if one user searches for the Shirt on Amazon or any other e-commerce website, he will get start ads recommendation of shirts after some time.

1.4 Python Programming Language

Python Language Introduction Python is a widely used general-purpose, high level programming language. It was initially designed by **Guido Van Rossum in 1991** and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

1.5 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

1.6 Features of Python

Python's features include −

Easy-to-learn – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read − Python code is more clearly defined and visible to the eyes.

Easy-to-maintain − Python's source code is fairly easy-to-maintain.

A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode − 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 − You can 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.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

It supports functional and structured programming methods as well as OOP.

It can be used as a scripting language or can be compiled to byte-code for building large applications.

It provides very high-level dynamic data types and supports dynamic type checking.

IT supports automatic garbage collection.

1.7 Role of Python in Machine Learning:

Python plays a significant role in the field of machine learning and is widely regarded as one of the most popular and versatile programming languages for this domain. Its role in machine learning can be attributed to several key factors:

1. **Rich Ecosystem of Libraries:** Python offers an extensive collection of libraries and frameworks specifically designed for machine learning and data science. Some of the most prominent ones include:

- NumPy: A fundamental library for numerical computation, which forms the backbone of many other machine learning libraries.
- pandas: A powerful library for data manipulation and analysis, often used for data preprocessing.
- scikit-learn: A comprehensive library for various machine learning algorithms, including classification, regression, clustering, and more.
- TensorFlow and PyTorch: Leading deep learning frameworks used for building and training neural networks.
- Keras: An easy-to-use high-level neural networks API built on top of TensorFlow and Theano (now integrated with TensorFlow as tf.keras).

2. **Ease of Use**: Python's simple and clean syntax makes it easy to read and write, reducing the learning curve for newcomers to machine learning. Its readability also facilitates better collaboration among team members.

3. **Community and Documentation:** Python has a vast and active community of data scientists, machine learning researchers, and developers. This community actively contributes to the development of machine learning tools, libraries, and frameworks. Additionally, there is extensive documentation available for most popular Python libraries, making it easier to get started and find solutions to common problems.

4. **Versatility and Integration:** Python is a versatile language that can be used for various tasks beyond machine learning. Its ability to integrate well with other languages and

technologies makes it a preferred choice for incorporating machine learning models into larger applications.

5. **Data Processing and Visualization:** Python's ecosystem includes libraries like pandas and matplotlib, which enable efficient data preprocessing, cleaning, and visualization. These steps are essential for preparing the data before feeding it into machine learning models.

6. **Support for Deep Learning**: With the rise of deep learning, Python has become the goto language for building, training, and deploying deep neural networks. TensorFlow and PyTorch, both primarily Python-based, are among the most popular frameworks for deep learning.

7. **Open-Source and Free:** Python is an open-source language, which means it is freely available for anyone to use and modify. This accessibility has contributed to its widespread adoption in academia and industry.

Due to these advantages and its strong community support, Python has become the de facto language for machine learning tasks. It has democratized access to machine learning tools and made it easier for individuals and organizations to leverage the power of machine learning in their projects and research.

1.8 Jupyter Notebook

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It is a popular tool among data scientists, researchers, and developers for interactive computing and data analysis.

Here's a brief overview of Jupyter Notebook's key features and how to use it:

1. **Notebook Interface:** When you start Jupyter Notebook, it opens in your web browser and provides a file browser to navigate your system's directories. You can create new notebooks or open existing ones from this interface.

2. **Cell-based Structure**: Notebooks are organized into cells, which can be of two types:

- Code Cells: Used for writing and executing code (Python, R, Julia, etc.).

- Markdown Cells: Used for writing formatted text, including headings, lists, images, and more, using the Markdown syntax.

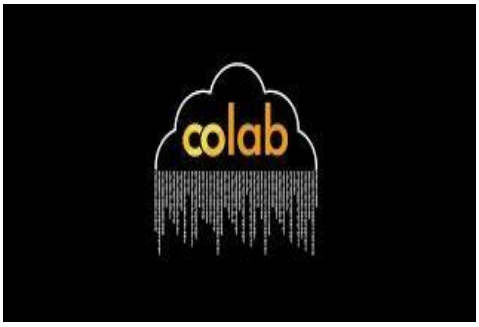
3. **Code Execution**: Code cells allow you to write and execute code in a step-by-step manner. To run a code cell, you can either click the "Run" button in the toolbar or use the keyboard shortcut "Shift + Enter." The output of the code execution is displayed directly below the cell.

4. **Kernel**: A notebook runs a kernel, which is essentially the computational engine that executes the code within the notebook. For example, if you write Python code, the notebook uses a Python kernel to execute it. You can switch kernels to work with different programming languages.

5. **Saving and Exporting**: Notebooks are automatically saved, but you can also manually save them by clicking the "Save" button or using the keyboard shortcut "Ctrl + S" (or "Cmd + S" on Mac). You can export notebooks to various formats, such as HTML, PDF, or plain Python scripts.

6. **Rich Media Support:** Notebooks allow you to embed images, videos, audio, and other media directly in the cells. This is particularly useful when creating interactive data visualizations.

1.9 Google Colab:



Google Colab (short for Collaboratory) is a cloud-based platform provided by Google that offers a Jupyter Notebooklike environment with access to free GPU and TPU (Tensor Processing Unit) resources. Colab is especially popular among data scientists, machine learning researchers, and students due to its ease of use, free access to computational resources, and the ability to collaborate and share notebooks easily.

Features of google colab

1. **Jupyter Notebook Environment**: Colab provides a Jupyter Notebook environment similar to what you would find in Jupyter Notebook. It supports both code cells (for writing and executing code) and markdown cells (for formatted text).

2. **Free GPU and TPU Support**: One of the major advantages of Colab is the provision of free access to GPUs and TPUs. This is particularly useful for training deep learning models that require substantial computational power.

3. **Cloud-based:** Colab runs entirely in the cloud, which means you don't need to install anything on your local machine. You can access your notebooks from any device with an internet connection and a web browser.

4. **Collaboration**: Colab allows multiple users to collaborate on the same notebook simultaneously. You can share the notebook with others and work on it together in real-time, making it a valuable tool for team projects and remote collaborations.

5. **Pre-installed Libraries**: Colab comes with many popular Python libraries pre-installed, such as TensorFlow, PyTorch, NumPy, pandas, matplotlib, and more. This saves time on setup and configuration, enabling you to focus on your analysis or machine learning tasks.

6. **File Storage:** Colab provides limited file storage in Google Drive. You can save and load files directly from your Google Drive, making it easy to keep your data and notebooks organized.

7. **Code Snippets and Examples:** Colab provides a range of code snippets and examples for different use cases, which can help users get started quickly and learn new techniques.

CHAPTER 2 – TRAINING WORK UNDERTAKEN

2.1 Project introduction – Crop Prediction (Agriculture production optimization)

Crop prediction , also known as agricultural yield forecasting ,is crucial aspect of modern agriculture that involves estimating the potential output of various crops based on specific soil content requirements . It plays a vital role in agricultural planning , food security , and economic decision – making for farmers, governments ,and stakeholders within the agricultural sector.

The primary goal of crop prediction is to recommend the most suitable crops for a given region based on the soil's nutrient content and other relevant factors. By accurately predicting the best crops to plant, farmers can make informed decisions regarding planting, irrigation ,fertilization, and pest control ,which can optimize production and reduce potential losses.

Crop prediction is a complex task that relies on the integration of various data sources, including historical agricultural data, weather patterns, soil characteristics, and satellite imagery. Machine learning algorithms and statistical models are often employed to analyze these datasets and extract patterns and relationships that can be used to make predictions.

Benefits of accurate crop prediction include:

- Improved Resource Management: Farmers can optimize the use of resources such as water, fertilizers, and pesticides based on recommended crops, leading to cost savings and reduced environmental impact.
- Mitigating Food Shortages: Governments and organizations can use crop prediction to identify regions at risk of food shortages and take proactive measures to address potential food security challenges.
- Risk Management for Farmers: Knowing the best crops to plant in advance helps farmers manage financial risks and plan for market fluctuations.
- Sustainable Agriculture: By recommending crops that are well-suited to the soil's nutrient content, agricultural practices can be more sustainable, minimizing waste and environmental damage.

Overall, crop prediction based on soil content data is a powerful tool that empowers farmers and decision-makers with valuable insights to ensure the efficient and sustainable production of crops, contributing to food security and economic stability. This technique can be very well adapted to predict suitable crops, ensuring the efficient use of resources and enhancing agricultural productivity.

2.2 Libraries used

2.2.1 Pandas - Pandas is a popular Python library for data manipulation and analysis. It provides data structures for handling tabular and timeseries data, including the Data Frame and Series objects. Pandas offers a wide range of functions for data cleaning, filtering, grouping, merging, and reshaping. It also supports data visualization and integration with other Python libraries, such as NumPy and Matplotlib. Pandas' intuitive and powerful API has made it a go-to library for data analysts and scientists.

2.2.2 Numpy - NumPy is the fundamental package for scientific computing in Python which provides a multidimensional array object. Other mathematical operations can be performed using this but simply speaking we just need it to convert our images into some form of an array so that we can store the model that has been trained.

2.2.3 sklearn - scikit-learn (often abbreviated as sklearn) is a popular and widely-used machine learning library for Python. It provides a rich set of tools for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and preprocessing of data. scikit-learn is built on top of other foundational Python libraries like NumPy and SciPy, making it a fundamental part of the Python data science ecosystem.

2.3 Dataset

As of my last update in September 2021, I don't
have access to real-time or future datasets. However, I can provide some general information about crop prediction datasets and what they typically contain.

Crop prediction datasets are collections of historical agricultural data, weather data, soil information, and other relevant features that are used to build models for forecasting crop yields or crop outcomes. These datasets are valuable resources for researchers, data scientists, and policymakers in the agriculture sector to analyze and predict crop production patterns and make informed decisions**.**

The availability of crop prediction datasets varies depending on the region, crop type, and the specific factors considered in the dataset. Some datasets are publicly available through agricultural organizations, government agencies, research institutions, and data repositories.

When working with crop prediction datasets, it's crucial to consider the quality of the data, data preprocessing, and the choice of machine learning models to ensure accurate and meaningful predictions.

For up-to-date and specific crop prediction datasets, I recommend conducting a web search or exploring reputable agricultural data sources to find the most relevant and recent data for your research or analysis.

Link of dataset used –

https://github.com/sukhchannu/crop-prediction/blob/main/Crop_recommendation.csv

It contains data with attributes which is necessary for the predicting the crop yield like

- Temperature

- Humidity
- Potassium
- Nitrogen
- Phosphorus
- Ph
- Rainfall
- Label

ALL INFORMATION ABOUT DATASET

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   N            2200 non-null   int64
 1   P            2200 non-null   int64
 2   K            2200 non-null   int64
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   object
dtypes: float64(4), int64(3), object(1)
memory usage: 137.6+ KB
```

2.4 Training process

- First step is to upload required libraries in jupyter notebook that will be used for crop prediction.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

- Then we will describe our data and check the minimum and maximum values of different columns.

```
df.describe()
```

	N	P	K	temperature	humidity	ph	rainfall
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.211267
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.551686
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.867624
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.267508
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.560117

- Then we will use groupby function to check the total values in a particular column. For example, label function for ‘label’ is shown in following figure.

```
df.groupby('label')['label'].count()
```

```
label
apple          100
banana         100
blackgram      100
chickpea       100
coconut        100
coffee         100
cotton         100
grapes         100
jute           100
kidneybeans    100
lentil         100
maize          100
mango          100
mothbeans      100
mungbean       100
muskmelon      100
orange         100
papaya         100
pigeonpeas     100
pomegranate    100
rice           100
watermelon     100
Name: label, dtype: int64
```

- As the same above we well check for other columns.
- Now we will replace the strings into integer so that our whole dataset will be converted into numeric form. For example, for ‘label’ column this is shown in following figure:

```
# replace type into numeric
df.label.replace(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',
       'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',
       'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',
       'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21],inpla
df['label'].unique()
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21], dtype=int64)
```

- As the same above we will do for other columns.
- Now we will check information about our dataset. We can see from the following figure that there is no string in our dataset. Thus, we data will be converted into numeric form.

```
df.info()
```

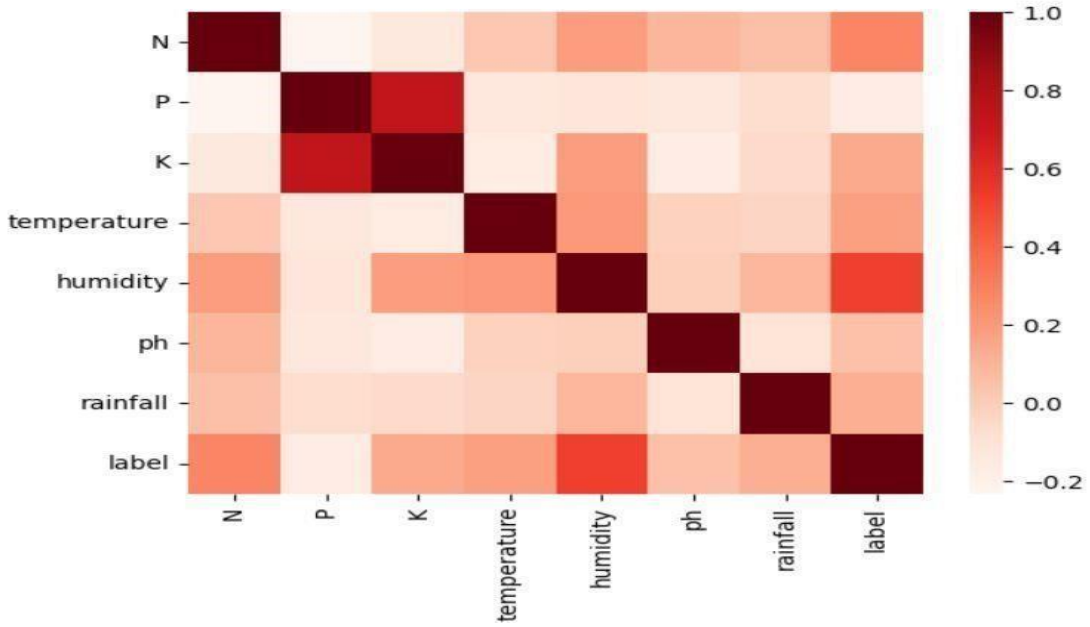
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   N            2200 non-null   int64  
 1   P            2200 non-null   int64  
 2   K            2200 non-null   int64  
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   int64  
dtypes: float64(4), int64(4)
memory usage: 137.6 KB
```

Now we will import one more library this is seaborn.

- Though seaborn library we will check the dependency using corr() function and plot it by using heatmap.

```
import seaborn as sns
corr=df.corr()
sns.heatmap(corr,cmap="Reds")
```

```
<Axes: >
```



- From the above figure we can say that our dataset is now properly cleaned.
- Now we will split it into x(input) and y(output).
- X is our input
- Y is our output
- X contains the n no. of inputs and y contains only 1 that is label

```
x=df.iloc[:,:-1]
y=df.iloc[:,-1]
```

```
x
```

	N	P	K	temperature	humidity	ph	rainfall
0	90	42	43	20.879744	82.002744	6.502985	202.935536
1	85	58	41	21.770462	80.319644	7.038096	226.655537
2	60	55	44	23.004459	82.320763	7.840207	263.964248
3	74	35	40	26.491096	80.158363	6.980401	242.864034
4	78	42	42	20.130175	81.604873	7.628473	262.717340
...	...	...	...	...	...	...	...
2195	107	34	32	26.774637	66.413269	6.780064	177.774507
2196	99	15	27	27.417112	56.636362	6.086922	127.924610
2197	118	33	30	24.131797	67.225123	6.362608	173.322839
2198	117	32	34	26.272418	52.127394	6.758793	127.175293
2199	104	18	30	23.603016	60.396475	6.779833	140.937041

```
y
```

```
0       1
1       1
2       1
3       1
4       1
       ..
2195    22
2196    22
2197    22
2198    22
2199    22
Name: label, Length: 2200, dtype: int64
```

- Now we split our dataset for training and testing.

split

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,shuffle=True,random_state=42)
print("x_train:",x_train.shape)
print("y_train:",y_train.shape)
print("x_test",x_test.shape)
print("y_test",y_test.shape)
```

```
x_train: (1760, 7)
y_train: (1760,)
x_test (440, 7)
y_test (440,)
```

- Now we use the LogisticRegression algorithm

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\linear_model\
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data a
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver op
    https://scikit-learn.org/stable/modules/linear_model.html#lo
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression()
```

- Prediction

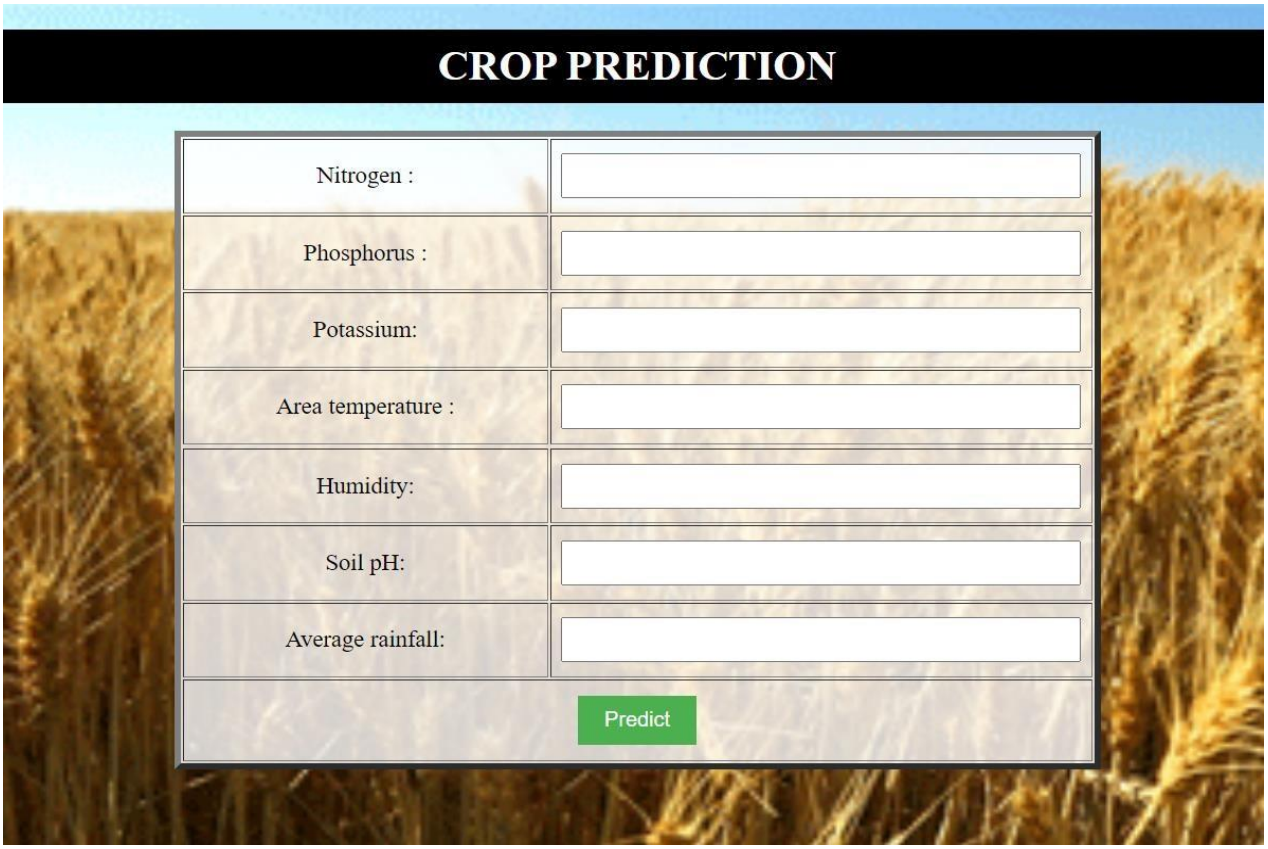
```
lr.predict([[74,35,40,26.491096,80.158363,6.980401,242.864034]])
```

```
array([0.])
```

CHAPTER 3 - RESULTS

- Here, 3.1 shows real life model where we give some inputs and submit it and obtain result like ‘banana’,”rice’,’cotton’,’jute’,etc .

  Link for model – https://glitch.com/edit/#!/crop-prediction



CHAPTER 4 – CONCLUSION AND FUTURE SCOPE

4.1 Conclusion

In conclusion, the project on crop production and prediction is a significant endeavor with substantial implications for the agriculture sector and food security. By leveraging advanced data analysis techniques and machine learning models on comprehensive datasets, we have gained valuable insights into crop yields, growth patterns, and factors influencing agricultural productivity.

Through the integration of historical agricultural data, weather information, soil characteristics, and satellite imagery, we have developed predictive models that can forecast crop yields with a high degree of accuracy. These models enable farmers and policymakers to make well-informed decisions about crop planning, resource allocation, and risk management, optimizing production processes and minimizing potential losses.

The project's findings highlight the importance of data-driven approaches in modern agriculture. Utilizing the power of technology, we can address challenges such as climate change, unpredictable weather patterns, and fluctuating market demands. By adopting sustainable agricultural practices and making efficient use of resources, we can promote environmental conservation and ensure long-term food security for a growing global population.

While the project has achieved significant milestones, there are still opportunities for further research and refinement. Continuous data collection and updates will enhance the models' accuracy and robustness, allowing us to adapt to changing agricultural landscapes and evolving conditions.

In summary, the crop production and prediction project contribute valuable insights and tools to the agricultural community, empowering stakeholders with the knowledge needed to make

informed decisions, drive sustainable practices, and secure a prosperous future for agriculture and food supply. With the ongoing advancements in technology and data science, this field will continue to evolve, leading to more efficient and resilient agricultural systems for the benefit of society as a whole.

4.2 Future Scope

1. **Improving Prediction Accuracy:** The project aims to make crop yield predictions more accurate and reliable.

2. **Expansion to Multiple Crops**: It can be extended to predict yields for various crops, not just limited to a few types.

3. **Regional Adaptability**: The project can be adapted for different regions, considering specific climatic and soil conditions.

4. **Early Warning System:** It can be developed as an early warning system to alert farmers about potential crop-related risks.

5. **Sustainable Agriculture**: The project can promote sustainable agricultural practices by optimizing resource utilization.

6. **Mobile Applications:** The predictions can be made accessible to farmers through userfriendly mobile apps.

7. **Precision Farming**: It can support precision farming techniques, guiding precise crop management decisions.

8. **Climate Change Resilience**: The project can help farmers adapt to the impacts of climate change on crops.

9. **Pest and Disease Control:** Extending the project to predict pests and diseases can aid in timely control measures.

10. **Market Trends Analysis**: Integrating market data can assist farmers in making informed decisions about crop choices based on demand and prices.

11. **Real-time Updates:** Continuous data updates can ensure that the predictions reflect current conditions.

12. **Interdisciplinary Collaboration:** Collaboration with agronomists and agricultural experts can ensure practical applications.

13. **Policy Development:** The project's findings can support policymakers in formulating agricultural policies.

14. **Weather Forecasting**: The project can be used to improve local weather forecasting for better crop management.

15. **Cost Optimization:** Farmers can optimize costs by using the predictions to plan resource allocation efficiently.

16. **Crop Rotation Planning:** The predictions can help farmers plan crop rotations for better soil health.

17. **Input Planning**: It can aid in planning the optimal use of fertilizers and irrigation resources.

18. **Sustainable Water Management:** By predicting crop yields, it can contribute to sustainable water management practices.

19. **Risk Management:** The project can help farmers manage risks related to weather, pests, and market fluctuations.

20. **Increased Productivity:** Ultimately, the project aims to increase crop productivity and support food security.

In conclusion, the future scope of the crop prediction project encompasses a wide range of applications, including accuracy improvements, expansion to different crops and regions, early warning systems, and support for sustainable agricultural practices. By leveraging technology, collaboration, and real-time data, the project aims to benefit farmers, policymakers, and the agricultural sector as a whole.

REFERENCES

1. W3SCHOOLS LINK: https://www.w3schools.com/
2. JAVAPOINT LINK: https://www.javatpoint.com/
3. GREEKS FOR GREEK LINK : https://www.geeksforgeeks.org/
4. KAGGLE LINK: https://www.kaggle.com/