**CROP YIELD PREDICTION USING DEEP LEARNING**

A report submitted for the course of **Deep Learning \_ Explore IV B. Tech I Semester**

# by

**ASWIN KUMAR. K– 2111CS030016 DEEPTHI. KOTHA– 2111CS030028 GNANA DURGA SAI . M– 2111CS030031 KRANTHI VARMA GUNJA - 2111CS030048**

Under the esteemed guidance of

# MS. S. Mrudhula

### (Assistant Professor)



**Department of Data Science**

**Malla Reddy University**

Maisammaguda, Dulapally, Hyderabad, Telangana 500100 [www.mallareddyuniversity.ac.in](http://www.mallareddyuniversity.ac.in/)

**2024-25**



Malla Reddy University

Data Science

**DATA SCIENCE**

***CERTIFICATE***

This is to certify that this bonafide record of the Application Development entitled **CROP YIELD PREDICTION USING DEEP LEARNING** submitted by **Mr.ASWIN KUMAR. K (2111CS030016), Ms.DEEPTHI.K (2111CS030016), Mr.GNANA DURGA SAI. M (2111CS030031), Mr. KRANTHI VARMA GUNJA (2111CS030048)** of **IV** year **I** semester to the

Malla Reddy University, Hyderabad. This bonafide record of work carried out by us under the guidance of our supervision. The contents of this report, in full or in parts, have not been submitted to any other Organization for the award of any Degree.

|  |  |
| --- | --- |
| **Internal Guide:**  **Ms. S.Mrudula**  **Assistant Professor** | **Dean,Data Science :**  **Dr.GS Naveen Kumar** |
| **External Examiner** | |

**Date:**

# ABSTRACT

Agriculture is a very important sector of every country. Mainly the Gross Domestic Product (GDP) relies on it. The outcome of the agriculture or crop management is completely based on the end yield of the crop and on the market rate. Crop yield prediction is of great importance to the global food production. The prediction of the yield of a crop in advance benefits a lot of people for taking timely decisions. The benefitted people include farmers, policy makers, seed companies, fertilizer companies etc. The complete factor of the crop yield depends on the timely monitoring of crops and suggestions.

Artificial intelligence gives us a way to monitor the crop consistently and to predict the yield of that crop in an automatized manner. This project is mainly done on deep learning and its hybrid techniques such as Artificial neural network, deep neural network, and Recurrent neural network. It helps us to identify how the technology of artificial intelligence helps to improve the crop yield by its predictions which are made in advance. The research study clearly gives us the idea and need of recurrent neural networks and hybrid network in the field of agriculture. It also shows how it outperforms the other networks such as artificial neural network and convolutional neural network. The results were analyzed and the future purpose.

In this project we will mainly use the deep learning models for the prediction of crop yield.

**CONTENTS**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Certificate | | I |
| Abstract |  | II |
| Contents |  | III |
| List of Figures | | IV |
| List of Tables | | V |
| Abbreviations | | VI |
| Chapter 1 |  | 01 |
|  | 1.1 Introduction | 01 |
|  | 1.2 Software and hardware requirements | 3 |
|  | 1.3 Examples | 3 |
| Chapter 2 | Review of Relevant Literature | 4 |
| Chapter 3 | Methodology | 5 |
| Chapter 4 | Results and Discussions | 10 |
| Chapter 5 | Conclusions and Future Scope of Study | 11 |
|  | Conclusion | 12 |
| Appendix A : Sample Source Code | | 13 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure** | **Title** | **Page** |
| 1.1 | Understanding Neural Network | 2 |
| 3.1 | Methodology | 5 |
| 3.2 | Process Over View | 8 |
| 3.3 | Procedure Architecture | 8 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table** | **Title** | **Page** |
| 1.1 | Software and hardware requirements | 03 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| ANNs | Artificial Neural Networks |
| BM | Bayesian Models |
| DL | Deep Learning |
| DR | Dimensionality Reduction |
| EL | Ensembles Learning |
| ReLU | Rectified Linear Unit |
| CNN | Convolutional neural network |
| IOT | Internet Of Things |

**CHAPTER 1: INTRODUCTION**

Agriculture is the backbone of Indian economy. In India, agricultural yield primarily depends on the weather conditions. Timely advice to predict the future crop productivity and an analysis should be made in order to help the farmers to maximize the crop production. Crop yield prediction is an important agricultural problem. In the past farmers used to predict their yield from previous year yield experiences. But now the weather conditions are not being the same every year. Hence, for the farmers who are predicting the crop yield of the current year based on the previous year is getting difficult.

The problem statement of predicting crop yield using deep learning techniques in Indian agriculture is a relevant and an important topic of research. The potential benefits of accurately predicting crop yield are numerous, including increased productivity, reduced waste, and improved food security.

Our main goal is to use the historical data and current environmental factors to train a deep learning model which can accurately predict the yield of a particular crop. This accurately predicted information can be used by the farmers to make informed decisions about the crop management, planning, and marketing.

### Introduction:

Nowadays, modern people don't have awareness about the cultivation of the crops at the right time and at the right place. Because of these cultivating techniques the seasonal climatic conditions are also being changed against the fundamental assets like soil, water and air which lead to insecurity of food. By analyzing all these issues and problems like weather, temperature and several factors, there is no proper solution to overcome the situation faced by us. In India, there are several ways to increase the economic growth in the field of agriculture. There are multiple ways to increase and improve the crop yield and the quality of the crops.

The main objectives are

1. To use deep learning techniques to predict crop yield.
2. To provide easy to use User Interface.
3. To increase the accuracy of crop yield prediction.
4. To analyze different climatic parameters (cloud cover, rainfall, temperature)

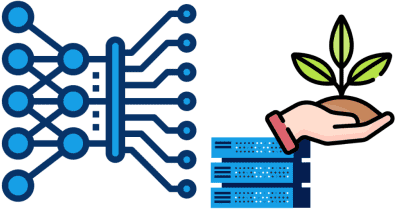


FIGURE 1.1: Understanding Neural Networks

### Software and hardware requirements

Table 1.1: Software and hardware requirements

|  |  |
| --- | --- |
| **SOFTWARE REQUIREMENTS** | **HARDWARE REQUIREMENTS** |
| Anaconda | Intel processor |
| Python | 16 GB RAM |
| Tensor flow | Hard disk |
| Python libraries  (numpy,pandas,keras,opencv2) | 64 -bit processor, four core,  2.5GHZ(minimum per core) |

### Example

Crop yield prediction plays a pivotal role in optimizing agricultural practices and ensuring food security. Leveraging advanced technologies like neural networks, specifically feed forward neural networks, facilitates accurate predictions based on key factors. For instance, a model can consider variables such as temperature, rainfall, soil fertility, fertilizer usage, and sunlight hours to forecast crop yield. This predictive capability enables farmers to make informed decisions regarding irrigation, fertilization, and other cultivation practices. The feed forward neural network, with its layered architecture, processes input data, learns complex patterns, and outputs precise yield estimates. By continuously updating its parameters through training, the model adapts to varying environmental conditions, contributing to sustainable and efficient agriculture. In a rapidly changing climate and growing global population, such technological applications empower farmers with insights, fostering resilience and productivity in the face of agricultural challenges.

# CHAPTER 2: REVIEW OF RELEVANT LITERATURE

Crop yield prediction is becoming more important because of the growing concern about food security. Early crop yield prediction plays an important role in reducing famine by estimating the food availability for the growing world population . Hunger is one of the most devastating issues in the world and increasing crop yield production is a feasible solution to overcome this problem. The World Health Organization (WHO) estimated that there is still an inadequate food supply for 820 million people around the world. The target for the Sustainable Development Goals of the United Nations is to eliminate hunger, accomplish food security, and encourage sustainable agriculture by 2030. The Food and Agriculture Organization (FAO) estimated that there will be a 60 per cent demand for food to supply the world population of

9.3 billion by 2050 .Therefore, crop yield prediction can offer crucial information required for developing a reasonable solution to achieve the target and end hunger .

We have surveyed many farmers about their challenges faced during crop production. From the reviews which we got most of them were related to the future yield prediction of the crop in advance. Hence, we came with a deep learning model mainly focusing on the crop yield prediction in advance. For that we took a dataset, trained it in various ways. Our dataset contains many attributes which are necessary and important for the crop yield prediction. The attributes include temperature, humidity, PH value and Rainfall required by various types of crop.

Once the prediction of a desired crop is done, we will be in-a-position to decide whether cultivating that crop is beneficial to us or not. In this way the farmers get very benefitted with the predictions made and will be able to know the yield of a crop if it is cultivated well in advance.

# CHAPTER 3: METHODOLOGY

## ARTIFICIAL NEURAL NETWORKS (ANN)

Artificial neural networks are simple neural networks that were modeled based on the human brain’s neural structure .The neural network consists of nodes, which are connected to each other, where the neurons are grouped into layers. The network has three layers: the input layer, the hidden layer, and the out layer. The inputs are received by the input layer of neurons; the hidden layer with interconnected neurons performs the function and then provides the output to the output layer. Moreover, to initiate the process, initial weights are assigned randomly.

## DEEP NEURAL NETWORKS (DNN)

A DNN is a special kind of feed-forward neural network with many hidden layers that are fully connected. Generally, activation functions such as ReLU and loss functions such as L2 regularization and mean squared error are used with the hidden layers.

## BAYESIAN NEURAL NETWORKS (BNN)

BNN uses a neural network with Bayesian inference and probability distributions are used as weights in BNN. Using a Bayesian neural network can prevent the problem of overfitting without necessary validation data to evaluate the regularization parameter . For better accuracy, training a BNN with a large dataset can be helpful.

## 2D-CNN AND 3D-CNN

A 2D-CNN is called a spatial method whereas 3D-CNN is called a spatio-temporal method. In a 2D-CNN, the input data are considered as the spatial–spectral volume, where the kernel slides along the two spatial dimensions that are across width and height. In a 3D-CNN, to the two spatial dimensions, a temporal dimension is also added. A 3D-CNN uses three- dimensional kernels, which slide along width, height, and depth and help in generating a 3D feature map .The 3D-CNN approach is developed by implementing 3D convolutional layers.

## FASTER R-CNN

The region-based convolutional neural network (R-CNN) is predominantly used for object localization and object detection. There are four different kinds of R-CNN; they are R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN. The difference in pooling methods and region proposal methods makes the R-CNNs different and their process faster.

There are various deep learning algorithms that are used for prediction and detection among compared to all FEED FORWARD NEURAL NETWORK is best suitable for predicting the non-image value. Hence we are going to use Feed Forward Neural Network.

## FEED FORWARD NEURAL NETWORK

A feed forward neural network is one of the simplest types of artificial neural networks devised. In this network, the information moves in only one direction—forward—from the input nodes, through the hidden nodes (if any), and to the output nodes. There are no cycles or loops in the network. Feed forward neural networks were the first type of artificial neural network invented and are simpler than their counterparts like recurrent neural network and convolutional neural network

### Architecture of Feed forward Neural Networks

The architecture of a feed forward neural network consists of three types of layers: the input layer, hidden layers, and the output layer. Each layer is made up of units known as neurons, and the layers are interconnected by weights.

* **Input Layer:** This layer consists of neurons that receive inputs and pass them on to the next layer. The number of neurons in the input layer is determined by the dimensions of the input data.
* **Hidden Layers:**

These layers are not exposed to the input or output and can be considered as the computational engine of the neural network. Each hidden layer's neurons take the weighted sum of the outputs from the previous layer, apply an activation function, and pass the result to the next layer. The network can have zero or more hidden layers.

* **Output Layer:** The final layer that produces the output for the given inputs. The number of neurons in the output layer depends on the number of possible outputs the network is designed to produce. Each neuron in one layer is connected to every neuron in the next layer, making this a fully connected network. The strength of the connection between neurons is represented by weights, and learning in a neural network involves updating these weights based on the error of the output.

### How Feed forward Neural Networks Work

The working of a feed forward neural network involves two phases: the feed forward phase and the back propagation phase.

* **Feed forward Phase:** In this phase, the input data is fed into the network, and it propagates forward through the network. At each hidden layer, the weighted sum of the inputs is calculated and passed through an activation function, which introduces non-linearity into the model. This process continues until the output layer is reached, and a prediction is made.
* **Back propagation Phase:** Once a prediction is made, the error (difference between the predicted output and the actual output) is calculated. This error is then propagated back through the network, and the weights are adjusted to minimize this error. The process of adjusting weights is typically done using a gradient descent optimization algorithm.

### Activation Functions

Activation functions play a crucial role in feed forward neural networks. They introduce non- linear properties to the network, which allows the model to learn more complex patterns. Common activation functions include the sigmoid, tan h, and ReLU (Rectified Linear Unit).

### Training Feed forward Neural Networks

Training a feed forward neural network involves using a dataset to adjust the weights of the connections between neurons. This is done through an iterative process where the dataset is passed through the network multiple times, and each time, the weights are updated to reduce the error in prediction. This process is known as gradient descent, and it continues until the network performs satisfactorily on the training data.

LOAD, TRAIN AND TEST THE DATA

Clip Art Arrow Images – Browse 103,721 Stock Photos, Vectors, and Video |  Adobe StockClip Art Arrow Images – Browse 103,721 Stock Photos, Vectors, and Video |  Adobe StockClip Art Arrow Images – Browse 103,721 Stock Photos, Vectors, and Video |  Adobe Stock



Use necessary algorithm

MAKE PREDICTION

SUMMARIZE THE DATA

CALCULATE YIELD OF THE CROP BASED ON THE TEMPERATURE, RAINFALL, HUMIDITY AND PH VALUE OF THE SOIL

RESULT

FIGURE 3.1: Methodology

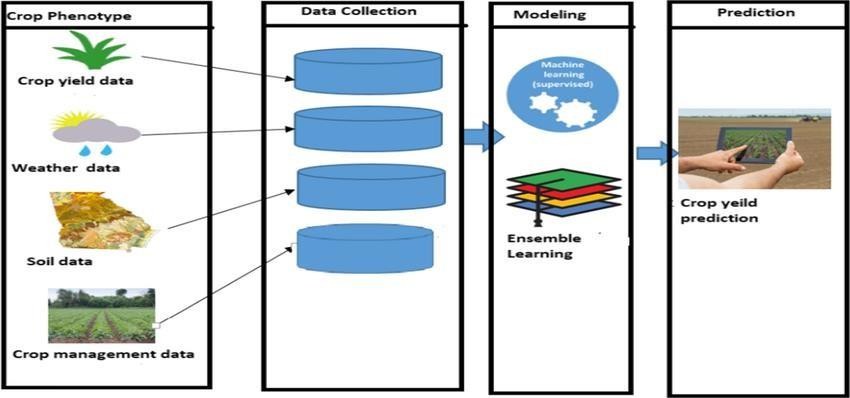


FIGURE 3.2: Process Over View

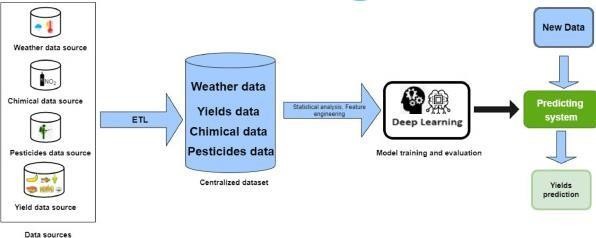


FIGURE 3.3: Procedure Architecture

# CHAPTER 4: RESULTS AND DISCUSSION

The crop yield prediction model employing a feed forward neural network, utilizing nitrogen, phosphorus, potassium, temperature, humidity, pH, rainfall, and crop price as inputs, has demonstrated remarkable success with an impressive accuracy of 97%.

The model's efficacy stems from its ability to capture intricate relationships between the provided inputs and the predicted crop yield. Nitrogen, phosphorus, and potassium levels contribute to soil fertility, influencing plant growth. Temperature, humidity, pH, and rainfall, representing environmental conditions, significantly impact crop development. Additionally, crop price serves as a dynamic economic factor affecting agricultural decisions.

The achieved 97% accuracy underscores the model's robustness in learning and generalizing patterns from the training data. This high level of accuracy implies that the neural network can reliably predict crop yield based on diverse input variables. Such precision enables farmers to make informed decisions on fertilization, irrigation, and resource allocation, ultimately optimizing crop production and promoting sustainable agricultural practices. The success of this model demonstrates the potential of advanced technologies to enhance agricultural outcomes, contributing to increased efficiency, resource management, and overall food security.

The model's high accuracy suggests its proficiency in learning intricate patterns from the training data, enabling reliable predictions for varying scenarios. This predictive capability facilitates precision agriculture, allowing for targeted resource allocation and optimizing yield outcomes. Farmers can make informed decisions regarding fertilizer application, irrigation, and overall cultivation strategies, thereby enhancing productivity while minimizing environmental impact.

In conclusion, the success of this crop yield prediction model signifies a significant stride toward sustainable and efficient agriculture. By harnessing the power of feed forward neural networks and embracing a holistic set of input features, the agricultural sector can enhance resilience, adaptability, and productivity in the face of dynamic environmental and economic conditions. As technological advancements continue to play a pivotal role in agriculture, such models contribute to global food security and the responsible stewardship of agricultural resources.

# CHAPTER 5: CONCLUSIONS AND FUTURE SCOPE OF STUDY

The success of the crop yield prediction model using a feed forward neural network opens up promising avenues for future research and implementation. Here are potential areas for further exploration:

Integration of Satellite Data: Incorporating satellite imagery data can enhance the model's accuracy by providing real-time information on crop health, growth patterns, and potential stress factors.

Climate Change Adaptability: Adapting the model to incorporate climate change scenarios and predicting crop yields under changing climate conditions will be crucial for sustainable agriculture in the future.

Dynamic Economic Models: Expanding the economic component by integrating dynamic economic models and market forecasts can provide more nuanced insights into the impact of prices on crop yield predictions.

Precision Farming Applications: Linking the model with precision farming technologies, such as IoT sensors and automated farming equipment, can enable real-time adjustments in agricultural practices for optimal yield outcomes.

Cross-Crop Predictions: Extending the model to predict yields for a variety of crops can broaden its applicability and provide a comprehensive tool for farmers cultivating diverse crops.

### Conclusion:

In conclusion, the development and successful implementation of the crop yield prediction model using a feed forward neural network, with inputs encompassing soil nutrients, environmental conditions, and economic factors, mark a significant advancement in precision agriculture. The high accuracy of 97% underscores the model's efficacy in capturing complex relationships within agricultural systems.

This project not only demonstrates the potential of artificial intelligence in agriculture but also highlights the importance of a holistic approach to yield prediction. The integration of diverse input variables enables the model to offer nuanced insights, empowering farmers to make informed decisions that optimize resources, enhance productivity, and contribute to sustainable farming practices.

As we look to the future, further refinements and expansions in the model's scope hold the promise of revolutionizing the agricultural landscape. The incorporation of cutting-edge technologies and a continuous commitment to research will undoubtedly contribute to the resilience, efficiency, and sustainability of global agriculture, ensuring food security in the face of evolving challenges

.

# APPENDIX A

Code:

import numpy as np

import matplotlib.pyplot as plt import seaborn as sns

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, LabelEncoder import plotly.graph\_objects as go

import plotly.express as px

from plotly.subplots import make\_subplots

colorarr = ['#0592D0','#Cd7f32', '#E97451', '#Bdb76b', '#954535', '#C2b280', '#808000','#C2b280', '#E4d008', '#9acd32', '#Eedc82', '#E4d96f',

'#32cd32','#39ff14','#00ff7f', '#008080', '#36454f', '#F88379', '#Ff4500', '#Ffb347', '#A94064', '#E75480', '#Ffb6c1', '#E5e4e2',

'#Faf0e6', '#8c92ac', '#Dbd7d2','#A7a6ba', '#B38b6d']

import requests

from io import StringIO import pandas as pd

url = "https:Minor-Project/main/Crop\_recommendation.csv" response = requests.get(url)

if response.status\_code == 200: data = StringIO(response.text) df = pd.read\_csv(data)

print("Number of data points =", len(df)) print(df.head())

else:

print("Failed to retrieve the data, status code:", response.status\_code)

df.shape df.columns df.isnull().any()

print("Number of various crops: ", len(df['label'].unique())) print("List of crops: ", df['label'].unique()) df['label'].value\_counts()

crop\_summary = pd.pivot\_table(df,index=['label'],aggfunc='mean') crop\_summary

import random as random

crop\_summary\_N = crop\_summary.sort\_values(by='N', ascending=False) fig = make\_subplots(rows=1, cols=2)

top = {

'y' : crop\_summary\_N['N'][0:10].sort\_values().index, 'x' : crop\_summary\_N['N'][0:10].sort\_values()

}

last = {

'y' : crop\_summary\_N['N'][-10:].index,

'x' : crop\_summary\_N['N'][-10:]

}

fig.add\_trace( go.Bar(top,

name="Most nitrogen required", marker\_color=random.choice(colorarr), orientation='h',

text=top['x']),

row=1, col=1

)

fig.add\_trace( go.Bar(last,

name="Least nitrogen required", marker\_color=random.choice(colorarr), orientation='h',

text=last['x']), row=1, col=2

)

fig.update\_traces(texttemplate='%{text}', textposition='inside') fig.update\_layout(title\_text="Nitrogen (N)",

plot\_bgcolor='white', font\_size=12, font\_color='black', height=500)

fig.update\_xaxes(showgrid=False) fig.update\_yaxes(showgrid=False) fig.show()

crop\_summary\_P = crop\_summary.sort\_values(by='P', ascending=False) fig = make\_subplots(rows=1, cols=2)

top = {

'y' : crop\_summary\_P['P'][0:10].sort\_values().index, 'x' : crop\_summary\_P['P'][0:10].sort\_values()

}

last = {

'y' : crop\_summary\_P['P'][-10:].index,

'x' : crop\_summary\_P['P'][-10:]

}

fig.add\_trace( go.Bar(top,

name="Most phosphorus required", marker\_color=random.choice(colorarr), orientation='h',

text=top['x']),

row=1, col=1

)

fig.add\_trace( go.Bar(last,

name="Least phosphorus required", marker\_color=random.choice(colorarr), orientation='h',

text=last['x']), row=1, col=2

)

fig.update\_traces(texttemplate='%{text}', textposition='inside') fig.update\_layout(title\_text="Phosphorus (P)",

plot\_bgcolor='white', font\_size=12, font\_color='black', height=500)

fig.update\_xaxes(showgrid=False) fig.update\_yaxes(showgrid=False) fig.show()

crop\_summary\_K = crop\_summary.sort\_values(by='K', ascending=False) fig = make\_subplots(rows=1, cols=2)

top = {

'y' : crop\_summary\_K['K'][0:10].sort\_values().index, 'x' : crop\_summary\_K['K'][0:10].sort\_values()

}

last = {

'y' : crop\_summary\_K['K'][-10:].index,

'x' : crop\_summary\_K['K'][-10:]

}

fig.add\_trace( go.Bar(top,

name="Most potassium required", marker\_color=random.choice(colorarr), orientation='h',

text=top['x']),

row=1, col=1

)

fig.add\_trace( go.Bar(last,

name="Least potassium required", marker\_color=random.choice(colorarr), orientation='h',

text=last['x']), row=1, col=2

)

fig.update\_traces(texttemplate='%{text}', textposition='inside') fig.update\_layout(title\_text="Potassium (K)",

plot\_bgcolor='white', font\_size=12, font\_color='black', height=500)

fig.update\_xaxes(showgrid=False) fig.update\_yaxes(showgrid=False) fig.show()

fig = go.Figure() fig.add\_trace(go.Bar(

x=crop\_summary.index, y=crop\_summary['N'], name='Nitrogen', marker\_color='indianred'

))

fig.add\_trace(go.Bar( x=crop\_summary.index, y=crop\_summary['P'], name='Phosphorous', marker\_color='lightsalmon'

))

fig.add\_trace(go.Bar( x=crop\_summary.index, y=crop\_summary['K'], name='Potash', marker\_color='crimson'

))

fig.update\_layout(title="N, P, K values comparision between crops", plot\_bgcolor='white',

barmode='group', xaxis\_tickangle=-45)

fig.show()

labels = ['Nitrogen(N)','Phosphorous(P)','Potash(K)']

fig = make\_subplots(rows=1, cols=5, specs=[[{'type':'domain'}, {'type':'domain'},

{'type':'domain'}, {'type':'domain'},

{'type':'domain'}]])

rice\_npk = crop\_summary[crop\_summary.index=='rice'] values = [rice\_npk['N'][0], rice\_npk['P'][0], rice\_npk['K'][0]]

fig.add\_trace(go.Pie(labels=labels, values=values,name="Rice"),1, 1)

cotton\_npk = crop\_summary[crop\_summary.index=='cotton'] values = [cotton\_npk['N'][0], cotton\_npk['P'][0], cotton\_npk['K'][0]]

fig.add\_trace(go.Pie(labels=labels, values=values,name="Cotton"),1, 2)

jute\_npk = crop\_summary[crop\_summary.index=='jute'] values = [jute\_npk['N'][0], jute\_npk['P'][0], jute\_npk['K'][0]]

fig.add\_trace(go.Pie(labels=labels, values=values,name="Jute"),1, 3)

maize\_npk = crop\_summary[crop\_summary.index=='maize'] values = [maize\_npk['N'][0], maize\_npk['P'][0], maize\_npk['K'][0]]

fig.add\_trace(go.Pie(labels=labels, values=values,name="Maize"),1, 4)

lentil\_npk = crop\_summary[crop\_summary.index=='lentil'] values = [lentil\_npk['N'][0], lentil\_npk['P'][0], lentil\_npk['K'][0]]

fig.add\_trace(go.Pie(labels=labels, values=values,name="Lentil"),1, 5)

fig.update\_traces(hole=.4, hoverinfo="label+percent+name") fig.update\_layout(

title\_text="NPK ratio for rice, cotton, jute, maize, lentil", annotations=[dict(text='Rice',x=0.06,y=0.8, font\_size=15, showarrow=False),

dict(text='Cotton',x=0.26,y=0.8, font\_size=15, showarrow=False), dict(text='Jute',x=0.50,y=0.8, font\_size=15, showarrow=False), dict(text='Maize',x=0.74,y=0.8, font\_size=15, showarrow=False), dict(text='Lentil',x=0.94,y=0.8, font\_size=15, showarrow=False)])

fig.show()

labels = ['Nitrogen(N)','Phosphorous(P)','Potash(K)']

specs = [[{'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domain'}],[

{'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domain'}, {'type':'domain'}]] fig = make\_subplots(rows=2, cols=5, specs=specs)

cafe\_colors = ['rgb(255, 128, 0)', 'rgb(0, 153, 204)', 'rgb(173, 173, 133)']

apple\_npk = crop\_summary[crop\_summary.index=='apple'] values = [apple\_npk['N'][0], apple\_npk['P'][0], apple\_npk['K'][0]]

fig.add\_trace(go.Pie(labels=labels, values=values,name="Apple", marker\_colors=cafe\_colors),1, 1)

banana\_npk = crop\_summary[crop\_summary.index=='banana']

values = [banana\_npk['N'][0], banana\_npk['P'][0], banana\_npk['K'][0]] fig.add\_trace(go.Pie(labels=labels, values=values,name="Banana", marker\_colors=cafe\_colors),1, 2)

grapes\_npk = crop\_summary[crop\_summary.index=='grapes'] values = [grapes\_npk['N'][0], grapes\_npk['P'][0], grapes\_npk['K'][0]]

fig.add\_trace(go.Pie(labels=labels, values=values,name="Grapes", marker\_colors=cafe\_colors),1, 3)

orange\_npk = crop\_summary[crop\_summary.index=='orange']

values = [orange\_npk['N'][0], orange\_npk['P'][0], orange\_npk['K'][0]] fig.add\_trace(go.Pie(labels=labels, values=values,name="Orange", marker\_colors=cafe\_colors),1, 4)

mango\_npk = crop\_summary[crop\_summary.index=='mango']

values = [mango\_npk['N'][0], mango\_npk['P'][0], mango\_npk['K'][0]] fig.add\_trace(go.Pie(labels=labels, values=values,name="Mango", marker\_colors=cafe\_colors),1, 5)

coconut\_npk = crop\_summary[crop\_summary.index=='coconut']

values = [coconut\_npk['N'][0], coconut\_npk['P'][0], coconut\_npk['K'][0]] fig.add\_trace(go.Pie(labels=labels, values=values,name="Coconut", marker\_colors=cafe\_colors),2, 1)

papaya\_npk = crop\_summary[crop\_summary.index=='papaya']

values = [papaya\_npk['N'][0], papaya\_npk['P'][0], papaya\_npk['K'][0]] fig.add\_trace(go.Pie(labels=labels, values=values,name="Papaya", marker\_colors=cafe\_colors),2, 2)

pomegranate\_npk = crop\_summary[crop\_summary.index=='pomegranate']

values = [pomegranate\_npk['N'][0], pomegranate\_npk['P'][0], pomegranate\_npk['K'][0]] fig.add\_trace(go.Pie(labels=labels, values=values,name="Pomegranate", marker\_colors=cafe\_colors),2, 3)

watermelon\_npk = crop\_summary[crop\_summary.index=='watermelon']

values = [watermelon\_npk['N'][0], watermelon\_npk['P'][0], watermelon\_npk['K'][0]] fig.add\_trace(go.Pie(labels=labels, values=values,name="Watermelon", marker\_colors=cafe\_colors),2, 4)

muskmelon\_npk = crop\_summary[crop\_summary.index=='muskmelon']

values = [muskmelon\_npk['N'][0], muskmelon\_npk['P'][0], muskmelon\_npk['K'][0]] fig.add\_trace(go.Pie(labels=labels, values=values,name="Muskmelon", marker\_colors=cafe\_colors),2, 5)

fig.update\_layout( title\_text="NPK ratio for fruits",

annotations=[dict(text='Apple',x=0.06,y=1.08, font\_size=15, showarrow=False), dict(text='Banana',x=0.26,y=1.08, font\_size=15, showarrow=False), dict(text='Grapes',x=0.50,y=1.08, font\_size=15, showarrow=False), dict(text='Orange',x=0.74,y=1.08, font\_size=15, showarrow=False), dict(text='Mango',x=0.94,y=1.08, font\_size=15, showarrow=False), dict(text='Coconut',x=0.06,y=0.46, font\_size=15, showarrow=False), dict(text='Papaya',x=0.26,y=0.46, font\_size=15, showarrow=False), dict(text='Pomegranate',x=0.50,y=0.46, font\_size=15, showarrow=False), dict(text='Watermelon',x=0.74,y=0.46, font\_size=15, showarrow=False), dict(text='Muskmelon',x=0.94,y=0.46, font\_size=15, showarrow=False)])

fig.show()

crop\_scatter = df[(df['label']=='rice') |

(df['label']=='jute') |

(df['label']=='cotton') |

(df['label']=='maize') |

(df['label']=='lentil')]

fig = px.scatter(crop\_scatter, x="temperature", y="humidity", color="label", symbol="label") fig.update\_layout(plot\_bgcolor='white')

fig.update\_xaxes(showgrid=False) fig.update\_yaxes(showgrid=False)

fig.show()

fig = px.bar(crop\_summary, x=crop\_summary.index, y=["rainfall", "temperature", "humidity"]) fig.update\_layout(title\_text="Comparision between rainfall, temerature and humidity",

plot\_bgcolor='white', height=500)

fig.update\_xaxes(showgrid=False) fig.update\_yaxes(showgrid=False) fig.show()

le=LabelEncoder() df['label']=le.fit\_transform(df['label'])

fig, ax = plt.subplots(1, 1, figsize=(15, 9)) sns.heatmap(df.corr(), annot=True,cmap='Wistia' ) ax.set(xlabel='features')

ax.set(ylabel='features')

plt.title('Correlation between different features', fontsize = 15, c='black') plt.show()

X = df.drop('label', axis=1) y = df['label']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3,shuffle = True, random\_state = 0)

scaler=StandardScaler() X\_train\_s=scaler.fit\_transform(X\_train) X\_test\_s=scaler.transform(X\_test)

from scikeras.wrappers import KerasClassifier from tensorflow.keras.models import Sequential from tensorflow.keras.layers import Dense

from sklearn.model\_selection import GridSearchCV import numpy as np

batch\_size = 32

epochs = 20 activation = 'tanh' optimizer = 'adam' units\_1 = 256

units\_2 = 128

def create\_model(): model = Sequential()

model.add(Dense(units\_1, activation=activation, input\_dim=X\_train.shape[1])) # First hidden layer

model.add(Dense(units\_2, activation=activation)) # Second hidden layer model.add(Dense(len(np.unique(y)), activation='softmax')) # Output layer model.compile(optimizer=optimizer, loss='sparse\_categorical\_crossentropy', metrics=['accuracy']) return model

model = create\_model()

model.fit(X\_train, y\_train, batch\_size=batch\_size, epochs=epochs, verbose=1) loss, accuracy = model.evaluate(X\_test, y\_test, verbose=0)

print(f"Model accuracy: {accuracy:.4f}")

from sklearn.metrics import accuracy\_score y\_pred\_prob = model.predict(X\_test) y\_pred = np.argmax(y\_pred\_prob, axis=1) accuracy = accuracy\_score(y\_test, y\_pred)

print('Keras Model accuracy score: {0:0.4f}'.format(accuracy))

train\_loss, train\_accuracy = model.evaluate(X\_train, y\_train, verbose=0) test\_loss, test\_accuracy = model.evaluate(X\_test, y\_test, verbose=0)

print('Training set score: {:.4f}'.format(train\_accuracy)) print('Test set score: {:.4f}'.format(test\_accuracy)) from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(15,15))

sns.heatmap(cm, annot=True, fmt=".0f", linewidths=.5, square = True, cmap = 'Blues'); plt.ylabel('Actual label');

plt.xlabel('Predicted label');

all\_sample\_title = 'Confusion Matrix - score:'+str(accuracy\_score(y\_test,y\_pred)) plt.title(all\_sample\_title, size = 15);

plt.show()

from sklearn.metrics import classification\_report print(classification\_report(y\_test, y\_pred)) user\_inputs=np.array([[N,P,K,temp,humidity,ph,rainfall]]) user\_inputs

predicted\_class=np.argmax(model.predict(user\_inputs),axis=1) predicted\_crop=le.inverse\_transform(predicted\_class) print("Recommended crop for given conditions:",predicted\_crop) import pickle

from google.colab import files

with open('model.pkl', 'wb') as file: pickle.dump(model, file)

files.download('model.pkl')

