AGRICULTURE PRODUCTION OPTIMIZATION ENGINE USING LOGISTIC REGRESSION

A PROJECT REPORT

Submitted by

UDAY KUMAR

Under the guidance of

Mr. Partha Koley

Is the partial fulfilment for the award of the training in

MACHINE LEARNING

At

EUPHORIAGENX KOLKATA

April 2023

ABSTRACT

As we all know that agriculture depends largely on the nature of soil and the climatic conditions and many a times, we face unpredictable changes in climate like, non-seasonal rainfall or heat waves or fluctuations in humidity levels, etc. and all such events cause a great loss to our farmers and farming, because of which they are not able to utilize their agricultural land to its fullest. So to solve all such problems, I have built a Machine Learning Model by the virtue of which we can help farmers, optimize the agricultural production, because this predictive model will help them understand that for a particular soil & given climatic condition, which crop will be best suitable for the harvest.

There are 7 key factors that I've taken into account which will help us in determining, exactly which crop should be grown and at what period of time, viz. Amount of Nitrogen, Phosphorus and Potassium in soil, Temperature in degree Celsius, Humidity, pH and Rainfall in mm.

Tools used: Python & Jupiter Notebook Libraries used: NumPy, Pandas, Seaborn, Matplotlib, Ip widgets .SK Machine Learning Algorithms used: Clustering Analysis and Logistic Regression.

ACKNOWLEDGEMENT

The recent developments in terms of data analytics have been a threshold for Agricultural activities which have help the farming industry to maximise productivity and mitigate risks through predictive data analytics. Big Data is not only helping farmers around the world to gain valuable information on farming but also facilitating them to maximise their production efficiency by the various methods of precision farming.

This next generation Green Revolution has a huge bet on Big Data, and considering India, which as an agricultural nation has over 58% of rural population dependent on farming for livelihood, is in very urgent need of transforming this sector for the growth of the nation.

Agricultural export also nearly one fifth of the total exports in India [1]. As India move along with world it needs to find out ways toward integrated data analytics which will empower Indian agricultural industry towards sustainable growth. This study thus will dwell more into the various methods adopted by developed and some of the developing nations to implement Big Data techniques in Agriculture and will try to analyse where, how and by what means the predictive data analytics can shape the future of Agriculture in India.

To formulate the basic premise of this study, various research papers, books, and online assets have been counselled, which constitute the primary information for the analysis. The information will be subjected to big data analytics framework introduced by various research organisations around the world for precision farming.

SOFTWARE DEVELOPMENT KITS

Python: Python is a popular programming language for data analysis and machine learning, which are essential components of an agriculture production optimization engine. Python has a large community of developers and a vast array of libraries and tools that make it easy to build complex algorithms and models.

Google Collab: Google collab is a free cloud-based service provided by Google that enables users to write, run, and share Python code online. It is a Jupiter notebook environment that allows users to access powerful computing resources, including GPUs and TPUs, without the need for expensive hardware. Google Colab is widely used for data analysis, machine learning, and deep learning projects.

Google Colab provides users with a virtual machine (VM) running on Google's cloud infrastructure that includes all the necessary software packages and libraries, including Python and popular data science libraries such as NumPy, Pandas, and Matplotlib. Users can upload and download data from their Google Drive account, and can also access external data sources using Python libraries or shell commands.

One of the key advantages of Google Colab is its collaborative features. Users can share their notebooks with others, allowing for real-time collaboration and version control. Additionally, Google Colab integrates with GitHub, making it easy to share and collaborate on code repositories.

Overall, Google Colab is a powerful tool for data analysis and machine learning that provides users with the flexibility and scalability of cloud computing, without the need for expensive hardware or infrastructure.

SOFTWARE DEVELOPMENT LIFE CYCLE

There are several software development life cycle (SDLC) models that can be used for developing an agriculture production optimization engine report. Here are a few popular models:

Waterfall Model: The Waterfall Model is a linear and sequential approach to software development. It involves a series of sequential phases that must be completed before moving on to the next phase. These phases include requirements gathering, design, implementation, testing, and maintenance. This model is suitable for agriculture production optimization engine reports that have clear and well defined requirements.

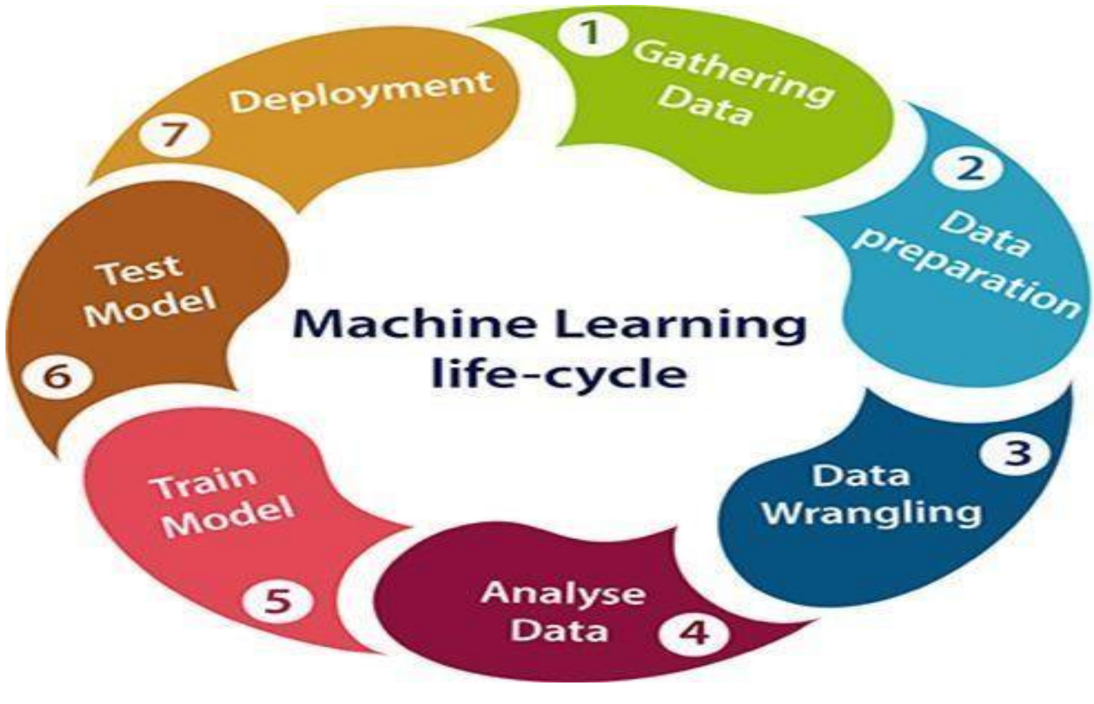
Agile Model: The Agile Model is an iterative and incremental approach to software development. It involves a series of short development cycles, or sprints, where the requirements, design, implementation, testing, and feedback are continuously evaluated and refined. This model is suitable for agriculture production optimization engine reports that have evolving requirements or need to adapt to changing circumstances.

V-Model: The V-Model is a variant of the Waterfall Model that places a strong emphasis on testing. It involves a series of phases that are mirrored on each side of a V-shaped diagram, with the requirements and design phases on the left side and the testing and deployment phases on the right side. This model is suitable for agriculture production optimization engine reports that require a high degree of testing and validation.

Spiral Model: The Spiral Model is a risk-driven model that combines elements of the Waterfall Model and the Agile Model. It involves a series of iterations that build upon the previous ones, with each iteration focusing on identifying and mitigating risks. This model is suitable for agriculture production optimization engine reports that involve high levels of risk or uncertainty. Here we have used Spiral model.

MACHINE LEARNING LIFE CYCLE

The machine learning life cycle refers to the process of developing, implementing and maintaining a machine learning model. The ML life cycle is similar to the software development life cycle (SLDC) and involves the following stages:



Data Gathering: Data gathering refers to the process of collecting data that will be used to train and test machine learning models. The quality and quantity of the data gathered is crucial for the accuracy and effectiveness of the model.
It is a critical step in the machine learning process as the quality and quantity of the data will directly impact the accuracy and effectiveness of the model. A thorough and well-executed data gathering process is essential for developing high-performing machine learning models.

Data Preparation: Data preparation is a crucial step in machine learning that involves transforming raw data into a format that can be used for training a machine learning model. The process of data preparation is also known as data pre-processing.
Data preparation is a time-consuming and often iterative process, as each step may need to be repeated several times to achieve the desired results. The goal of data preparation is to create a high-quality dataset that accurately represents the underlying patterns in the data and is suitable for use in training a machine learning model.

Data Wrangling: Data wrangling in machine learning refers to the process of cleaning, transforming, and preparing raw data for analysis or use in machine learning models. It involves a series of operations such as filtering, sorting, reshaping, aggregating, and joining data to create a clean and structured dataset. The primary goal of data wrangling is to prepare the data in a way that is suitable for analysis or use in machine learning algorithms.
Data wrangling is a time-consuming and iterative process that requires a combination of domain knowledge, technical skills, and creativity. It is a critical step in the machine learning process as it directly impacts the quality and reliability of the machine learning model.

Data Analysis: Data analysis in machine learning refers to the process of applying statistical and machine learning techniques to extract insights and knowledge from data. The goal of data analysis is to understand patterns and relationships in the data and to use this information to develop predictive models or make informed decisions.
Data analysis is a complex and iterative process that requires a combination of technical and domain-specific knowledge. The goal of data analysis is to transform data into insights and knowledge that can be used to improve business processes, make informed decisions, or develop predictive models that can be used to automate tasks and optimize outcomes.

Model Training: Model training in machine learning refers to the process of using a dataset to teach a machine learning algorithm to make accurate predictions or classifications. The goal of model training is to develop a model that can accurately generalize from the training data to make predictions or classifications on new, unseen data.
Model training is a critical step in the machine learning process as it directly impacts the accuracy and reliability of the model.

Model Testing: Model testing in machine learning refers to the process of evaluating the performance of a trained machine learning model on a new, unseen dataset. The goal of model testing is to assess the ability of the model to generalize to new data and to identify potential issues such as overfitting or underfitting.
Model testing is a critical step in the machine learning process as it provides an unbiased estimate of the performance of the model on new, unseen data. It helps to ensure that the model is not overfitting to the training data and that it can generalize well to new data.

Deployment: Deployment in machine learning refers to the process of integrating a trained machine learning model into a production environment, where it can be used to make predictions or classifications on new, real-world data. The goal of deployment is to create a system that can automate a specific task or provide valuable insights based on data.
Deployment is a critical step in the machine learning process, as it is the point at which the model is put into action and begins providing value to stakeholders.

SUPERVISED AND UNSUPERVISED MACHINE LEARNING

Supervised learning and unsupervised learning are two fundamental types of machine learning techniques used to train artificial intelligence (ML) models.
Supervised learning involves training a machine learning model using labelled data, where the input data and the desired output are already known. The labelled data is fed into the model, and the model is trained to recognize patterns and relationships between the input data and the desired output. Once the model is trained, it can be used to predict the output for new, unseen data.
Unsupervised learning, on the other hand, involves training an ML model using unlabelled data, where the desired output is unknown. The model is left to identify patterns and relationships in the data on its own, without any guidance or supervision.

Supervised and unsupervised machine learning can both be used in agriculture production optimization engines to improve crop yields and efficiency.

Supervised learning involves training a model using labelled data, where the input data and the desired output are already known. This can be useful for predicting outcomes such as crop yield or soil moisture content based on input variables such as weather patterns, soil composition, and irrigation frequency. These predictions can be used to optimize farming practices and improve overall crop production.

Unsupervised learning, on the other hand, involves training a model without labelled data, instead identifying patterns and relationships in the data on its own. This can be useful for tasks such as clustering crops based on similar characteristics or identifying anomalies in plant growth or soil conditions that could indicate the presence of pests or disease.

Both supervised and unsupervised learning can be used in combination in agriculture production optimization engines to provide a comprehensive view of crop performance and improve overall efficiency and productivity. Additionally, machine learning techniques can be used to analyse large amounts of data quickly, making it possible for farmers to make data-driven decisions in real-time, improving their chances of success.

PYTHON

Python is a popular programming language that can be used in agriculture production optimization engines to analyze data, build models, and make predictions. Python offers a variety of tools and libraries that can be used to process and analyze large datasets, such as Pandas, NumPy, and Scikit-Learn.

In agriculture, Python can be used for various tasks, including:

Data collection and preprocessing: Python can be used to collect and pre-process data from various sources, such as sensors, weather stations, and satellite imagery. It can also be used to clean and transform data for use in modeling.

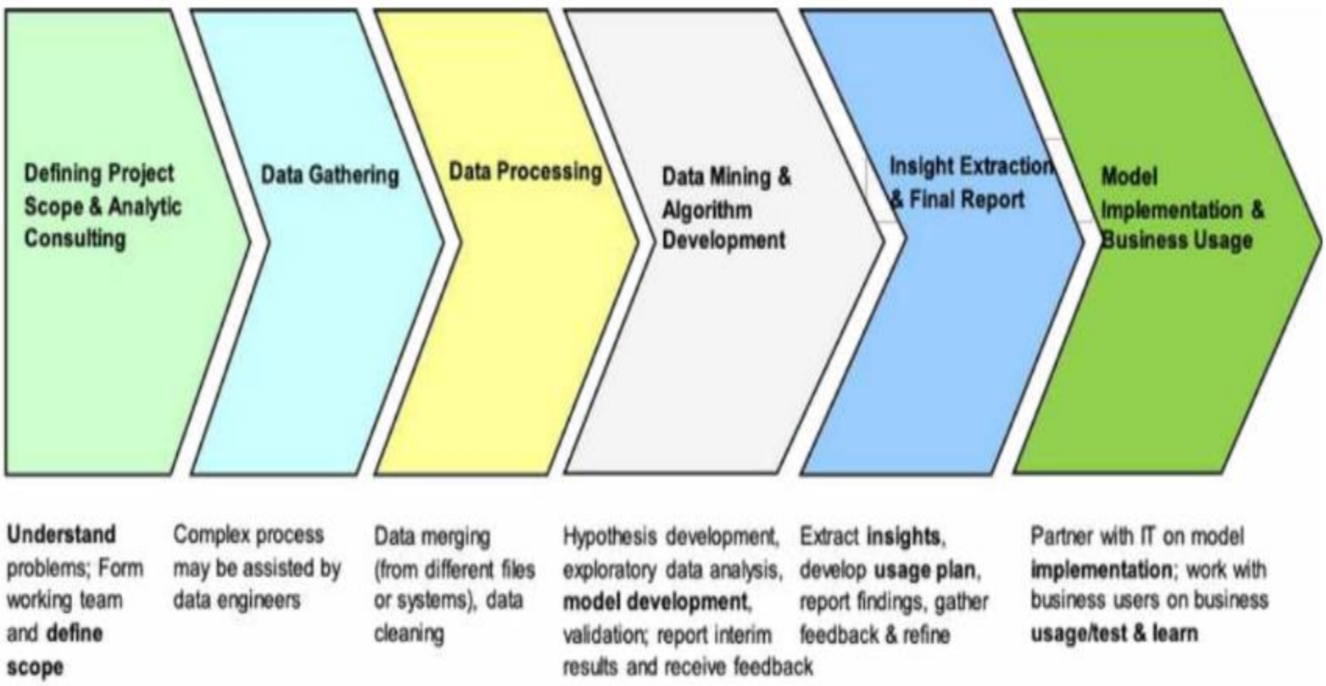
Modeling: Python can be used to build various types of models, such as regression, classification, and clustering models. These models can be used to predict crop yields, soil moisture content, and other variables relevant to agriculture.

Optimization: Python can be used to optimize agriculture production by finding the best values for various input variables, such as fertilizer application rates and irrigation schedules.

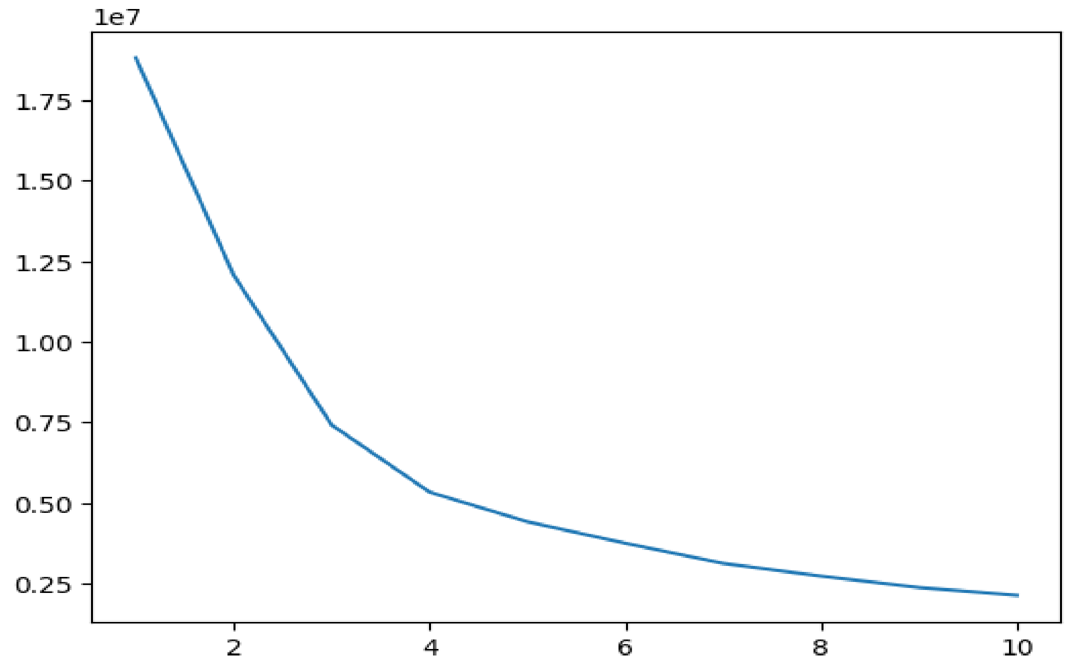
Visualization: Python offers several visualization libraries, such as Matplotlib and Seaborn, which can be used to create visual representations of data, such as crop yield maps and soil moisture heat maps.

Overall, Python can be a powerful tool for agriculture production optimization, offering a flexible and versatile platform for data analysis and modeling. Its popularity and ease of use make it accessible to a wide range of users, from small-scale farmers to large-scale agribusinesses.

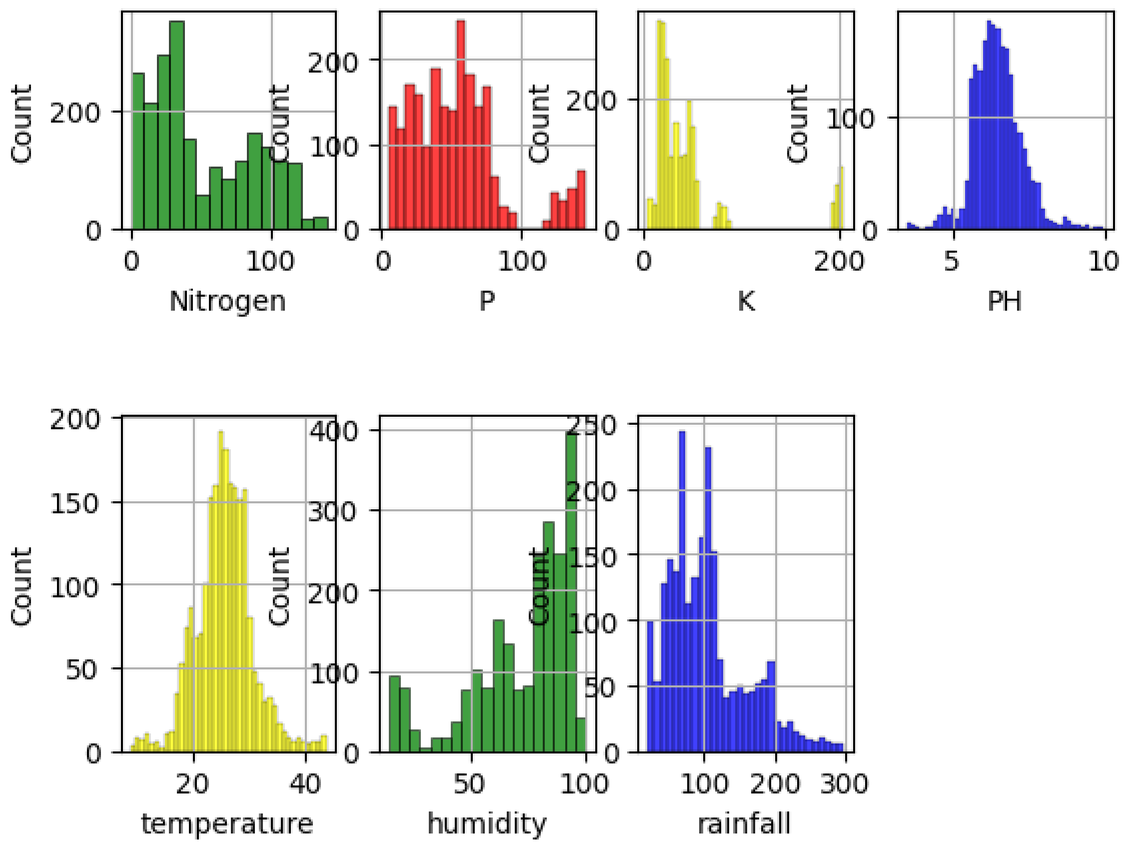
Workflow of Project



The Elbow Method



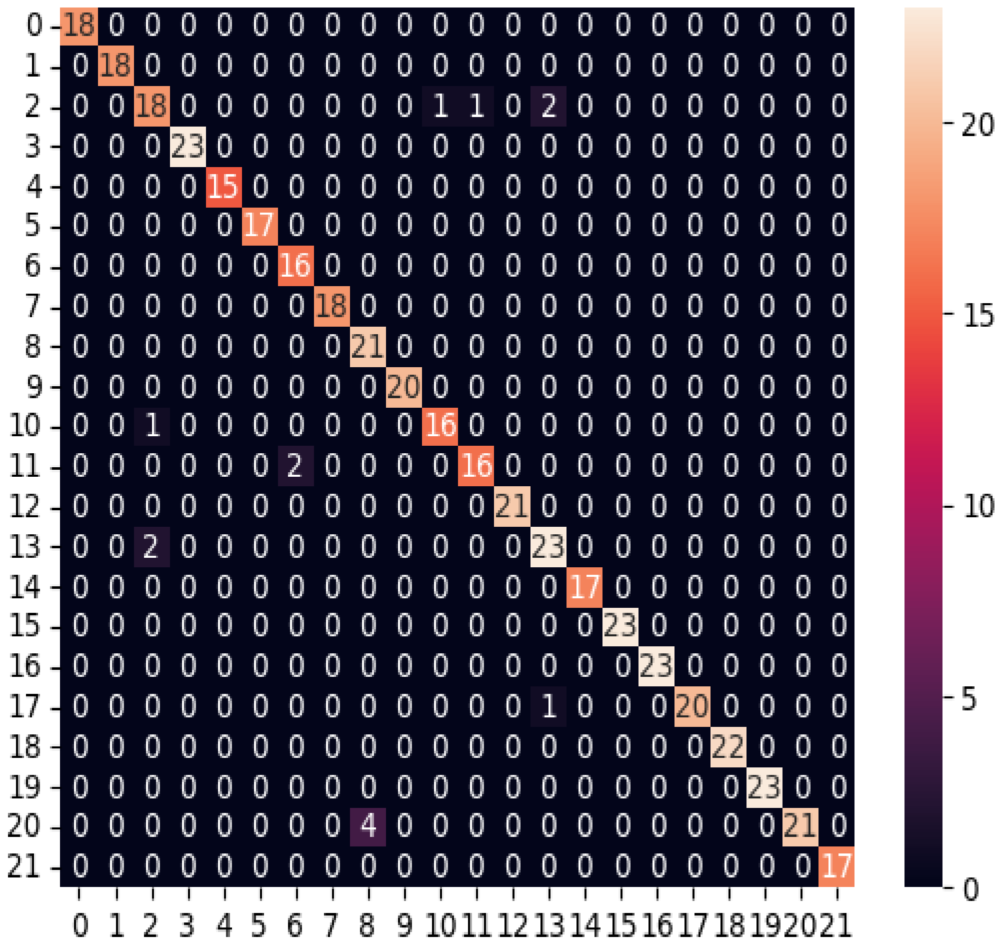
Distribution of Agricultural Conditions



Prediction of Crops

```
[ ]  from sklearn.linear_model import LogisticRegression

     model=LogisticRegression()
     model.fit(x_train,y_train)
     y_pred=model.predict(np.array([[40,40,40,40,100,7,200]]))

     print(y_pred)

     ['coconut']
```

Confusion Matrix using Logistic Regression



Classification report for Logistic Regression

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	18
banana	1.00	1.00	1.00	18
blackgram	0.86	0.82	0.84	22
chickpea	1.00	1.00	1.00	23
coconut	1.00	1.00	1.00	15
coffee	1.00	1.00	1.00	17
cotton	0.89	1.00	0.94	16
grapes	1.00	1.00	1.00	18
jute	0.84	1.00	0.91	21
kidneybeans	1.00	1.00	1.00	20
lentil	0.94	0.94	0.94	17
maize	0.94	0.89	0.91	18
mango	1.00	1.00	1.00	21
mothbeans	0.88	0.92	0.90	25
mungbean	1.00	1.00	1.00	17
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	23
papaya	1.00	0.95	0.98	21
pigeonpeas	1.00	1.00	1.00	22
pomegranate	1.00	1.00	1.00	23
rice	1.00	0.84	0.91	25
watermelon	1.00	1.00	1.00	17
accuracy			0.97	440
macro avg	0.97	0.97	0.97	440
weighted avg	0.97	0.97	0.97	440

<u>Code:</u>

Importing the modules

```
#for data manipulations
import numpy as np
import pandas as pd
#for data visualizations
import matplotlib.pyplot as plt
import seaborn as sns
#for interactive analysis
from ipywidgets import interact
import warnings
warnings.filterwarnings("ignore")
```

Reading the data sets

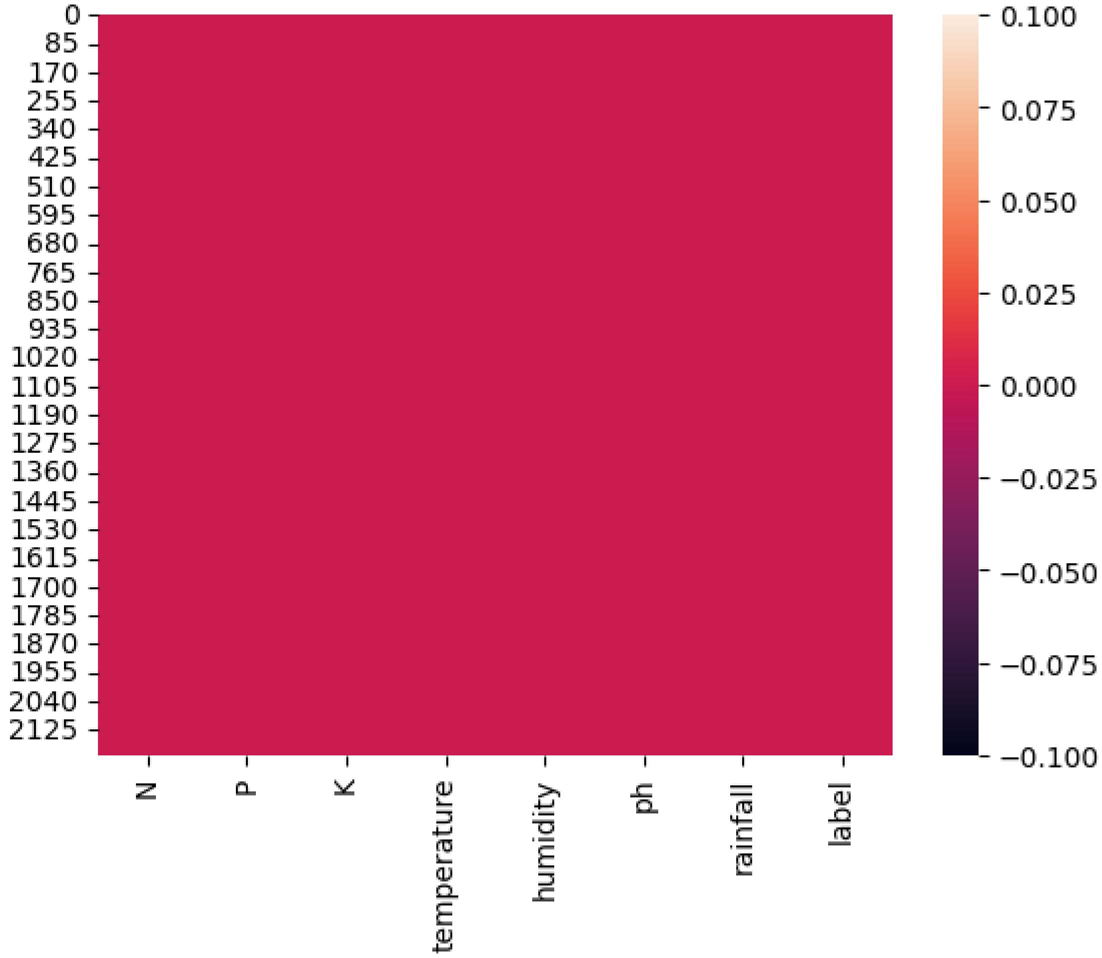
```
data=pd.read_csv("/content/data.csv")
print(data)
```

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

[2200 rows x 8 columns]
```

Exploratory Data Analysis (EDA):

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

Checking the missing data



Summary of crops (Checking Minimum, Average, Maximum)

```
def summary(crops=list(data['label'].value_counts().index)):
  x=data[data['label']==crops]
  print("min N required ",x['N'].min())
  print("Avg N required ",x['N'].mean())
  print("Max N required ",x['N'].max())

  print("min P required ",x['P'].min())
  print("Avg P required ",x['P'].mean())
  print("Max P required ",x['P'].max())

  print("min K required ",x['K'].min())
  print("Avg K required ",x['K'].mean())
  print("Max K required ",x['K'].max())

  print("min temperature required ",x['temperature'].min())
  print("Avg temperature required ",x['temperature'].mean())
  print("Max temperature required ",x['temperature'].max())

  print("min humidity required ",x['humidity'].min())
  print("Avg humidity required ",x['humidity'].mean())
  print("Max humidity required ",x['humidity'].max())

  print("min ph required ",x['ph'].min())
  print("Avg ph required ",x['ph'].mean())
  print("Max ph required ",x['ph'].max())

  print("min rainfall required ",x['rainfall'].min())
  print("Avg rainfall required ",x['rainfall'].mean())
  print("Max rainfall required ",x['rainfall'].max())
```

crops rice

```
min N required  60
Avg N required  79.89
Max N required  99
min P required  35
Avg P required  47.58
Max P required  60
min K required  35
Avg K required  39.87
Max K required  45
min temperature required  20.0454142
Avg temperature required  23.6893322105
Max temperature required  26.92995077
min humidity required  80.12267476
Avg humidity required  82.27282153889999
Max humidity required  84.96907151
min ph required  5.005306977
Avg ph required  6.425470922139999
Max ph required  7.868474653
min rainfall required  182.5616319
Avg rainfall required  236.18111359399998
Max rainfall required  298.5601175
```

Checking which crops grow in summer, winter and monsoon seasons

Summer's crops

```
[ ]  print(data[(data['temperature']>30) & (data['humidity']>50)]['label'].unique())
```

```
['pigeonpeas' 'mothbeans' 'blackgram' 'mango' 'grapes' 'orange' 'papaya']
```

Monsoon's Crops

```
[ ]  print(data[(data['rainfall']>200) & (data['humidity']>50)]['label'].unique())
```

```
['rice' 'papaya' 'coconut']
```

Winter crops

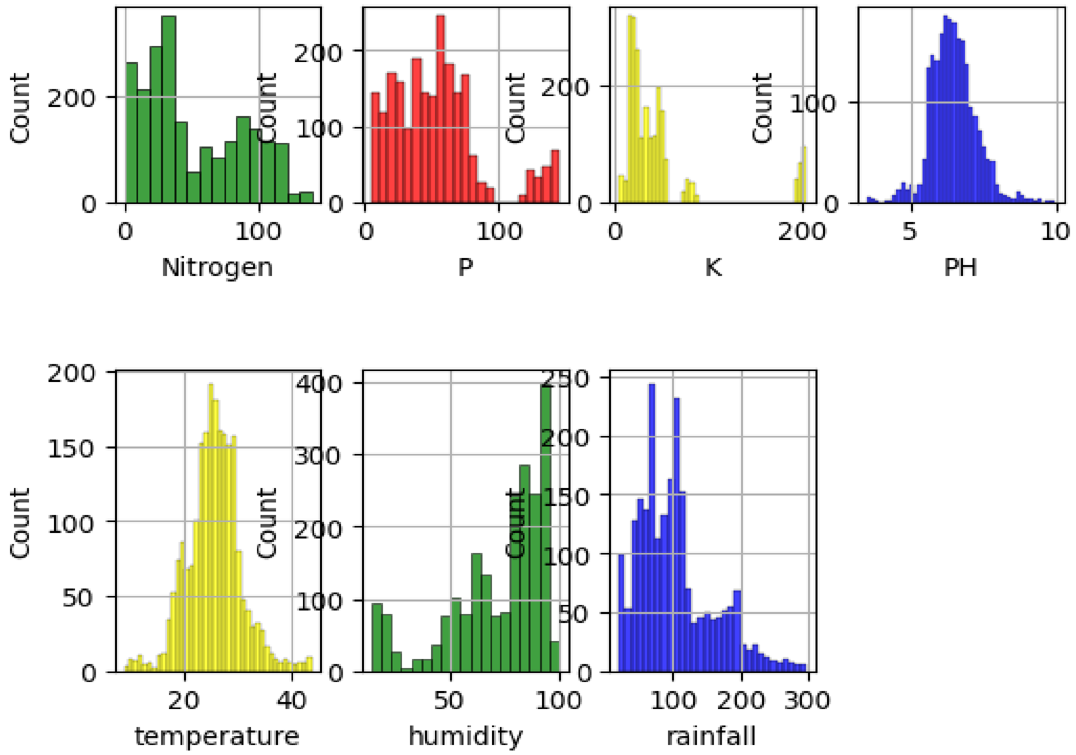
```
[ ]  print(data[(data['temperature']<20) & (data['humidity']<50)]['label'].unique())
```

```
['chickpea' 'kidneybeans' 'pigeonpeas']
```

Checking distribution for each crop

```python
plt.subplot(3,4,1)
sns.histplot(data['N'],color="green")
plt.xlabel("Nitrogen")
plt.grid()

plt.subplot(3,4,2)
sns.histplot(data['P'],color="red")
plt.xlabel("P")
plt.grid()

plt.subplot(3,4,3)
sns.histplot(data['K'],color="yellow")
plt.xlabel("K")
plt.grid()

plt.subplot(3,4,4)
sns.histplot(data['ph'],color="blue")
plt.xlabel("PH")
plt.grid()

plt.subplot(2,4,5)
sns.histplot(data['temperature'],color="yellow")
plt.xlabel("temperature")
plt.grid()

plt.subplot(2,4,6)
sns.histplot(data['humidity'],color="green")
plt.xlabel("humidity")
plt.grid()
```

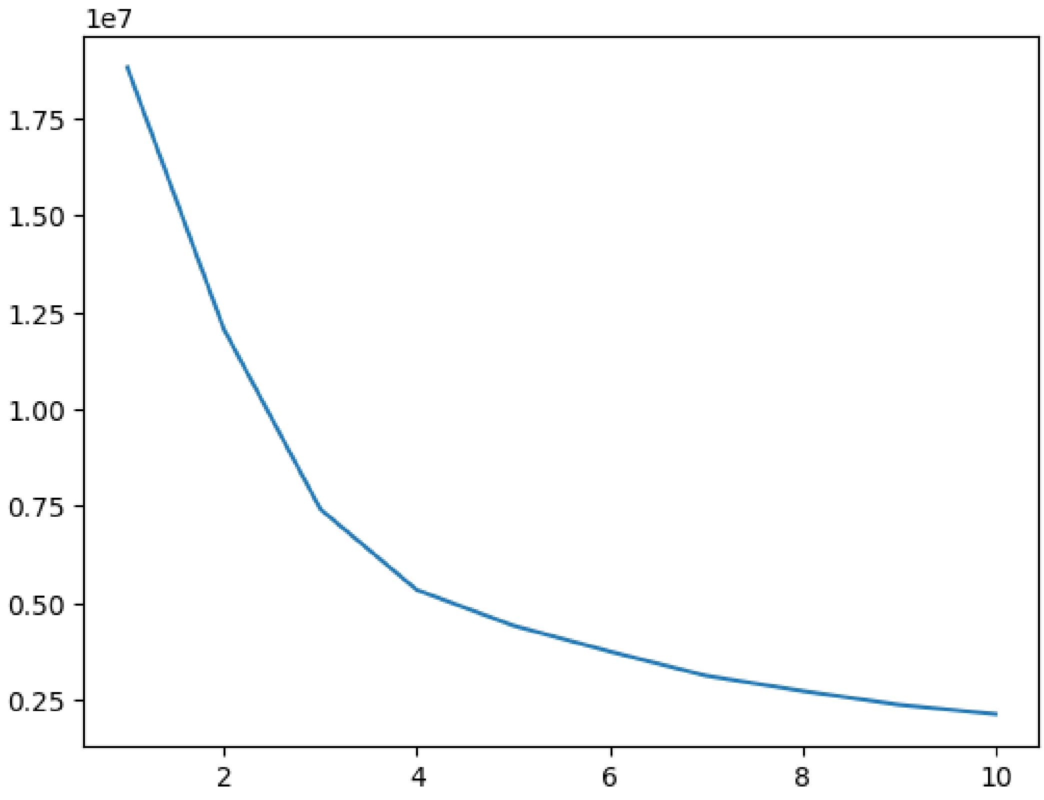
```
plt.subplot(2,4,7)
sns.histplot(data['rainfall'],color="blue")
plt.xlabel("rainfall")
plt.grid()
```



Clustering Analysis:

Determine the optimum number of clusters in data set:

```
from pandas.core.common import random_state
from sklearn.cluster import KMeans

x=data.drop(['label'],axis=1)
x=x.values
wcss=[]
for i in range(1,11):
  km=KMeans(n_clusters=i,init="k-means++", max_iter=2000,n_init=10,random_state=0)
  km.fit(x)
  wcss.append(km.inertia_)

plt.plot(range(1,11),wcss)
plt.show()
```



Implementing the k-means algorithm to perform clustering analysis

```
km=KMeans(n_clusters=4,init="k-means++", max_iter=2000,n_init=10,random_state=0)
y_means=km.fit_predict(x)

a=data['label']
y_means=pd.DataFrame(y_means)
z=pd.concat([y_means,a],axis=1)

z=z.rename(columns={0:'cluster'})

print("Cluster 1",z[z['cluster']==0]['label'].unique())
print("Cluster 2",z[z['cluster']==1]['label'].unique())
print("Cluster 3",z[z['cluster']==2]['label'].unique())
print("Cluster 4",z[z['cluster']==3]['label'].unique())
```

```
Cluster 1 ['maize' 'chickpea' 'kidneybeans' 'pigeonpeas' 'mothbeans' 'mungbean'
 'blackgram' 'lentil' 'pomegranate' 'mango' 'orange' 'papaya' 'coconut']
Cluster 2 ['maize' 'banana' 'watermelon' 'muskmelon' 'papaya' 'cotton' 'coffee']
Cluster 3 ['grapes' 'apple']
Cluster 4 ['rice' 'pigeonpeas' 'papaya' 'coconut' 'jute' 'coffee']
```

Building predictive model

```
y=data['label']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.2,random_state=0)
```

```
from sklearn.linear_model import LogisticRegression

model=LogisticRegression()
model.fit(x_train,y_train)
y_pred=model.predict(np.array([[40,40,40,40,100,7,200]]))

print(y_pred)
```

```
['coconut']
```

Model Performance

```
y_pred=model.predict(x_test)
from sklearn.metrics import classification_report
cr=classification_report(y_test,y_pred)
print(cr)
```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	18
banana	1.00	1.00	1.00	18
blackgram	0.86	0.82	0.84	22
chickpea	1.00	1.00	1.00	23
coconut	1.00	1.00	1.00	15
coffee	1.00	1.00	1.00	17
cotton	0.89	1.00	0.94	16
grapes	1.00	1.00	1.00	18
jute	0.84	1.00	0.91	21
kidneybeans	1.00	1.00	1.00	20
lentil	0.94	0.94	0.94	17
maize	0.94	0.89	0.91	18
mango	1.00	1.00	1.00	21
mothbeans	0.88	0.92	0.90	25
mungbean	1.00	1.00	1.00	17
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	23
papaya	1.00	0.95	0.98	21
pigeonpeas	1.00	1.00	1.00	22
pomegranate	1.00	1.00	1.00	23
rice	1.00	0.84	0.91	25
watermelon	1.00	1.00	1.00	17
accuracy			0.97	440
macro avg	0.97	0.97	0.97	440
weighted avg	0.97	0.97	0.97	440

CONCLUSION

The agriculture production optimization engine is a powerful tool that can help farmers optimize various factors related to agricultural production, including crop selection, planting density, irrigation, and fertilization. By analysing a wide range of data sources, such as weather patterns, soil quality, crop yield history, and market demand, the engine provides farmers with personalized recommendations tailored to their specific needs and circumstances. This enables farmers to maximize their yields and profits while minimizing waste, conserving resources, and reducing the environmental impact of farming.

One of the most significant benefits of the agriculture production optimization engine is that it can help farmers reduce their reliance on traditional agricultural practices that may be inefficient and unsustainable. By providing farmers with data-driven insights, the engine can help them make informed decisions about which crops to plant, how much water and fertilizer to use, and when to harvest their crops. This can help farmers reduce their costs and increase their profits, while also contributing to the sustainability of agriculture.

Another key advantage of the agriculture production optimization engine is its ability to adapt to changing conditions. By continuously analysing data from a variety of sources, the engine can adjust its recommendations in real-time to reflect changes in weather patterns, market demand, and other factors that may affect agricultural production. This can help farmers stay ahead of the curve and maximize their yields and profits even in unpredictable conditions.

In conclusion, the agriculture production optimization engine is a valuable tool that can help farmers improve their productivity, profitability, and sustainability. By providing personalized recommendations based on comprehensive data analysis, the engine can help farmers optimize their production processes, reduce waste, conserve resources, and minimize the environmental impact of farming. As such, the agriculture production optimization engine represents an important step forward in the ongoing effort to make agriculture more efficient, sustainable, and profitable.

FUTURE SCOPE

The agriculture production optimization engine is a powerful tool that has the potential to transform agriculture by providing farmers with personalized recommendations based on comprehensive data analysis. However, there are several areas where the engine could be further improved and expanded to enhance its effectiveness and impact.

One area of future development for the agriculture production optimization engine is the integration of new data sources. Currently, the engine uses data on weather patterns, soil quality, crop yield history to generate its recommendations. However, there may be additional data sources, such as satellite imagery and remote sensing data, that could be integrated to provide even more detailed and accurate insights into agricultural production processes.

Another area of future development for the agriculture production optimization engine is the expansion of its capabilities beyond crop selection and management. For example, the engine could be expanded to provide recommendations for livestock management, pest and disease control, and water management. This would enable farmers to optimize all aspects of their production processes, resulting in even greater efficiency and sustainability.

The agriculture production optimization engine could also be further developed to include more advanced modeling techniques, such as machine learning and artificial intelligence. These techniques could be used to identify patterns and trends in agricultural data that may not be immediately apparent to human analysts. This could lead to even more accurate and valuable recommendations for farmers.

Finally, the agriculture production optimization engine could be integrated with other agricultural technologies, such as precision agriculture tools and autonomous farming equipment. This would create a comprehensive system for optimizing agricultural production processes, from crop selection to harvesting.

In conclusion, the agriculture production optimization engine represents a powerful tool for improving agricultural productivity, profitability, and sustainability. However, there is significant potential for future development and expansion of the engine's capabilities, including the integration of new data sources, the expansion of its capabilities beyond crop selection and management, the use of advanced modeling techniques, and the integration with other agricultural technologies. By continuing to develop and expand the engine, we can create a more efficient, sustainable, and profitable agricultural industry for the future.

BIBLIOGRAPHY

1) Bouwman, A. F., Beusen, A. H. W., & Billen, G. (2019). Human alteration of the global nitrogen and phosphorus soil balances for the period 1970–2050. Global Biogeochemical Cycles, 33(4), 428-448.

2) Chen, X., Zhu, Y., & Zhou, Z. (2020). An agriculture production optimization model based on data mining and machine learning. Computers and Electronics in Agriculture, 170, 105248.

3) Food and Agriculture Organization of the United Nations. (2018). The future of food and agriculture: Alternative pathways to 2050.

4) Gómez-López, M. T., Ortiz-Miranda, D., Hernández-López, D., & Rodríguez-Pérez, J. E. (2021). Predicting yield, water, and fertilizer requirements in greenhouses using machine learning. Computers and Electronics in Agriculture, 181, 105952.

5) Kumar, S., Singh, D., Sharma, D., & Singh, R. (2020). Precision farming for sustainable agriculture: A review. Journal of Cleaner Production, 249, 119319