CROP PREDICTION ANALYSIS

Team Memners:
DEVYANSH - 20BBS0155 (VIT Vellore)
BHAVESH - 20BBS0200 (VIT Vellore)
NANCY - 20MID0115 (VIT Vellore)
RUFAIDA - 20MIS7001 (VIT -AP)

Guide Name:
Siddharth Awasthi

1 INTRODUCTION

1.1 Overview A brief description about your project

India occupies an important position globally as a major producer in the agricultural sector. Agriculture is not only the largest economic sector but also plays an important role in the overall Indian economy. The agricultural industry is characterized by diverse crop production and is influenced by various environmental and economic factors. By regularly advising farmers on improvements in cultivation techniques and advances in factors affecting crop production, the country's agricultural sector can be strengthened. Yield forecasting is one of the advances with great potential for advancing agriculture.

Traditionally, farmers have predicted crop yields based on experience. However, the advent of digitalization in agriculture has raised awareness of the need for timely and accurate information on crop growth. This perception is especially true for young farmers who want to adopt modern farming methods. Such advances in agriculture require the use of machine learning. Crop forecasting systems have emerged as a solution to meet this need for accurate forecasting.

1.2 Purpose The use of this project. What can be achieved using this.

Crop forecasting is the process of predicting crop yields using machine learning techniques. Harness the power of digitization and data-driven insights to provide farmers with valuable information on projected crop production at the right time and place. Crop forecasting systems integrate historical data, environmental factors and economic indicators to help farmers make informed decisions and optimize agricultural practices.

This project focuses on developing a crop forecasting system utilizing machine learning technology. The system aims to predict future crop yields by analyzing historical data such as past crop yields, environmental conditions, and economic variables. The use of machine learning plays a key role in identifying patterns, relationships and factors that influence crop production. These insights provide farmers with valuable information that can improve their decision-making process and lead to increased productivity.

The integration of machine learning into the agricultural sector is part of a broader digitization trend transforming various industries. Using machine learning, farmers can optimize resource allocation, implement precision farming techniques, reduce risk, and improve overall farm management. The availability of accurate yield forecasts helps farmers with planning, resource allocation, market analysis and risk management.

The implementation of machine learning-based crop forecasting systems could revolutionize the agricultural sector in India. These systems use historical data, environmental factors and economic indicators to provide farmers with valuable insight into future crop yields. This knowledge enables farmers to make informed decisions, optimize agricultural practices, and ultimately contribute to increased productivity and profitability. The integration of machine learning into agriculture is an important step in modernising the sector and providing farmers with the tools they need to thrive in a dynamic and competitive environment.

2 LITERATURE SURVEY

2.1 Existing problem Existing approaches or method to solve this problem

Various approaches are being pursued to address the challenges in agricultural production. Traditional farming practices passed down through generations provide valuable empirical knowledge and skills. Advisory services play an important role in providing farmers with guidance and information on improving agricultural practices. With the help of weather forecasts, farmers can plan their activities accordingly and optimize resource allocation. Soil testing and nutrient management can help you understand soil conditions and apply appropriate fertilizers. Integrated pest management technology reduces reliance on pesticides through the use of natural enemies and surveillance systems. Precision agriculture uses advanced technologies such as remote sensing, GPS, and IoT devices to collect data for site-specific crop management.

2.2 Proposed solution What is the method or solution suggested by you?

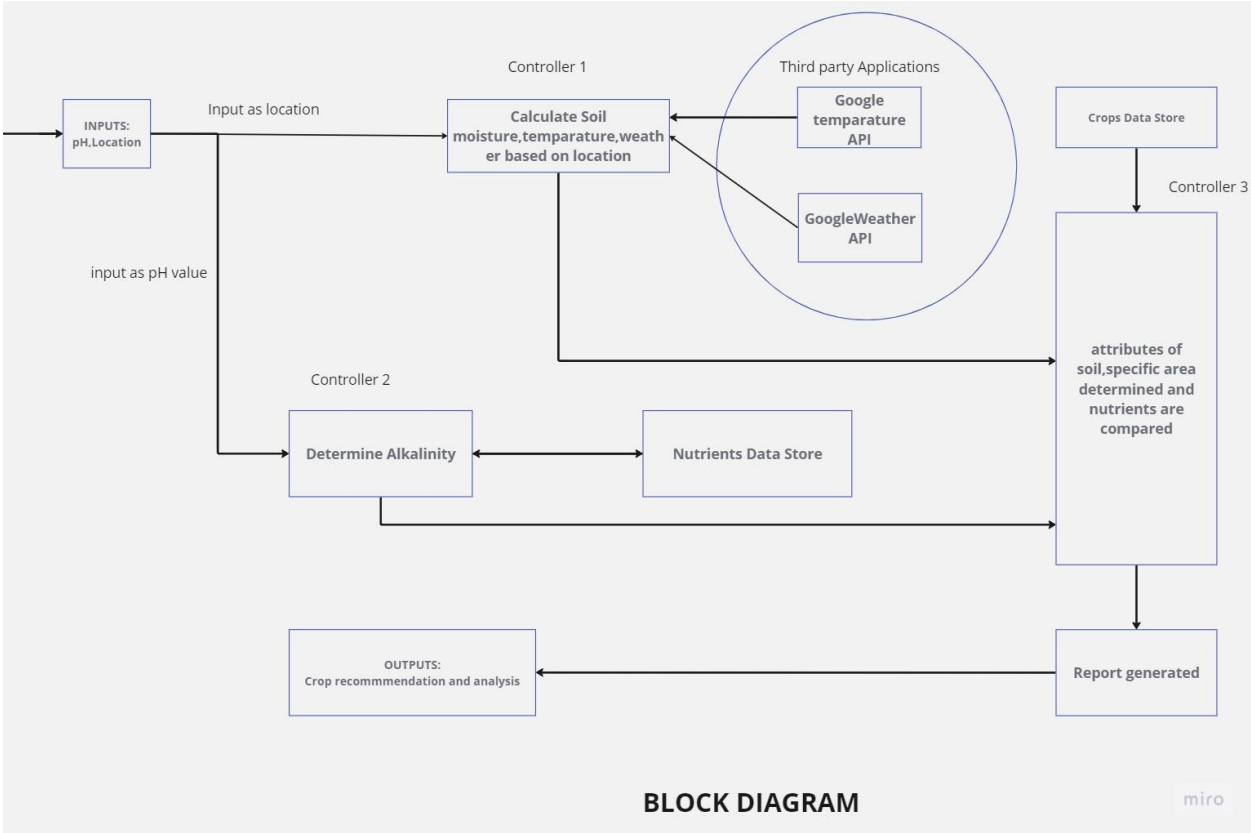
The proposed solution for addressing the challenges in agricultural crop production involves the utilization of two main methods: K-Nearest Neighbors (KNN) Classifier and Random Forests.

KNN is a machine learning algorithm used for both regression and classification tasks. It determines the class or value of a new data point by identifying its k-nearest neighbors in the training set based on a distance metric. The majority class among these neighbors is used for classification, while the average value is used for regression. KNN requires no training time but can be computationally expensive for large datasets and requires careful selection of the optimal value of k.
Random Forests, on the other hand, is an ensemble learning method that combines multiple decision trees to improve model accuracy and generalization. It randomly selects subsets of

features and creates decision trees based on those features. The final prediction for a new data point is made by aggregating the predictions of all the trees through voting or averaging. Random Forests can handle high-dimensional datasets, reduce overfitting, and provide insights into feature importance.
The proposed solution aims to enhance the accuracy and reliability of crop yield prediction in agriculture.

3 THEORETICAL ANALYSIS

3.1 Block diagram Diagrammatic overview of the project.



The suggested system will assess soil quality, forecast crop output in accordance, and, if necessary, offer fertiliser based on soil quality.

The architecture's functioning is shown in Figure 1 as follows:

The user must supply the system with the pH value (calculated depending on the percentage of nutrition) and the location. The processing of results is handled by two controllers.
Using third-party apps like APIs for weather and temperature, type of soil, nutrient value of the soil in that region, amount of rainfall in that region, and soil composition, location is used as an input to controller 1.

Controller 2 receives a pH value as input, and uses that value to calculate the soil's alkalinity. Along with it, percentages of nutrients such organic matter, iron, manganese, phosphorus, potassium, sulphur, magnesium, and calcium can be acquired.
A predetermined "nutrients" data storage is compared to the output of controllers 1 and 2. These results are then given to controller 3, who compares the combination of the aforementioned outcomes with the specified data set stored in the crop data store

3.2 Hardware / Software designing Hardware and software requirements of the project

Hardware requirements:

- Computer:

A computer system capable of running the necessary software tools and handling the computational requirements of the project.

- Processor:

A modern processor with enough processing power to efficiently handle data analysis, modeling, and forecasting tasks.

- Storage:

Sufficient RAM to accommodate the size of your dataset and the computational requirements of your machine learning algorithms.

- Sensors and data collection devices:

For real-time data collection, suitable sensors and data collection equipment may be required to collect relevant environmental and crop-related data.

Software requirements:

- Python:

Since The project is based on the Python programming language, so you need a compatible Python interpreter installed on your system.

- Jupyter Notebook, Spyder, or Integrated Development Environment (IDE):

Environments like Jupyter Notebook or Spyder or IDEs like PyCharm or Anaconda Navigator can be used for code development, experimentation, and analysis.

- Data analytics and machine learning libraries:

Python libraries such as NumPy, Pandas are essential for data analysis, preprocessing, and implementing machine learning algorithms.

- Using the Flask web framework:

to implement the project's web-based functionality enables user interaction and exposes predictive models via an API.

- Visualization libraries:

Matplotlib or Seaborn can be used for data visualization to get insights from datasets and display results effectively.

- Required Libraries and Packages:

Specific implementations may require additional libraries or packages such as TensorFlow, Keras, or SciPy that support advanced machine learning techniques or specific algorithms.

The hardware and software requirements mentioned above provide a foundation for setting up the project environment and ensuring compatibility and functionality throughout the development and execution phases. It is crucial to ensure that the hardware resources meet the computational demands, while the software components, including Spyder and Flask, are correctly installed and configured to enable seamless execution of the project tasks.

4 EXPERIMENTAL INVESTIGATIONS

(Analysis or the investigation made while working on the solution.)

During the course of implementing the proposed solution, a series of rigorous experimental investigations were conducted to analyze and evaluate the efficacy of the KNN Classifier and Random Forests in predicting crop yields. These investigations encompassed the following key areas:

=>Data Analysis: The acquired dataset, comprising essential features such as temperature, moisture, rainfall, and historical crop yields, underwent meticulous analysis to identify inherent patterns, correlations, and any potential anomalies. Statistical tests, descriptive statistics, and data visualization techniques were employed to gain comprehensive insights into the dataset.

=>Model Training and Tuning: The KNN Classifier and Random Forests models were trained utilizing the preprocessed dataset. Extensive exploration of hyperparameter configurations, including the number of neighbors (k) for KNN and the number of trees and their depth for Random Forests, was undertaken. Cross-validation techniques were employed to assess model performance and identify the optimal hyperparameter settings.

=>Performance Evaluation: The trained models underwent thorough evaluation using diverse performance metrics such as accuracy, precision, recall, and F1-score. Comparisons between the models` predictions and the actual crop yields were made to ascertain their predictive capabilities and uncover any potential discrepancies or errors.

=>Comparative Analysis: A meticulous comparative analysis was conducted to assess the performance of the KNN Classifier and Random Forests. Key aspects examined included accuracy, computational efficiency, robustness against noise or outliers, and the models' ability to handle high-dimensional data. This analysis aimed to determine the most suitable model for accurately predicting crop yields.

The experimental investigations followed an iterative process of model training, evaluation, and refinement. The results derived from these investigations provided valuable insights into the strengths and limitations of the models, enabling the refinement of the solution to enhance its accuracy and effectiveness in predicting crop yields.

5 FLOWCHART Diagram showing the control flow of the solution

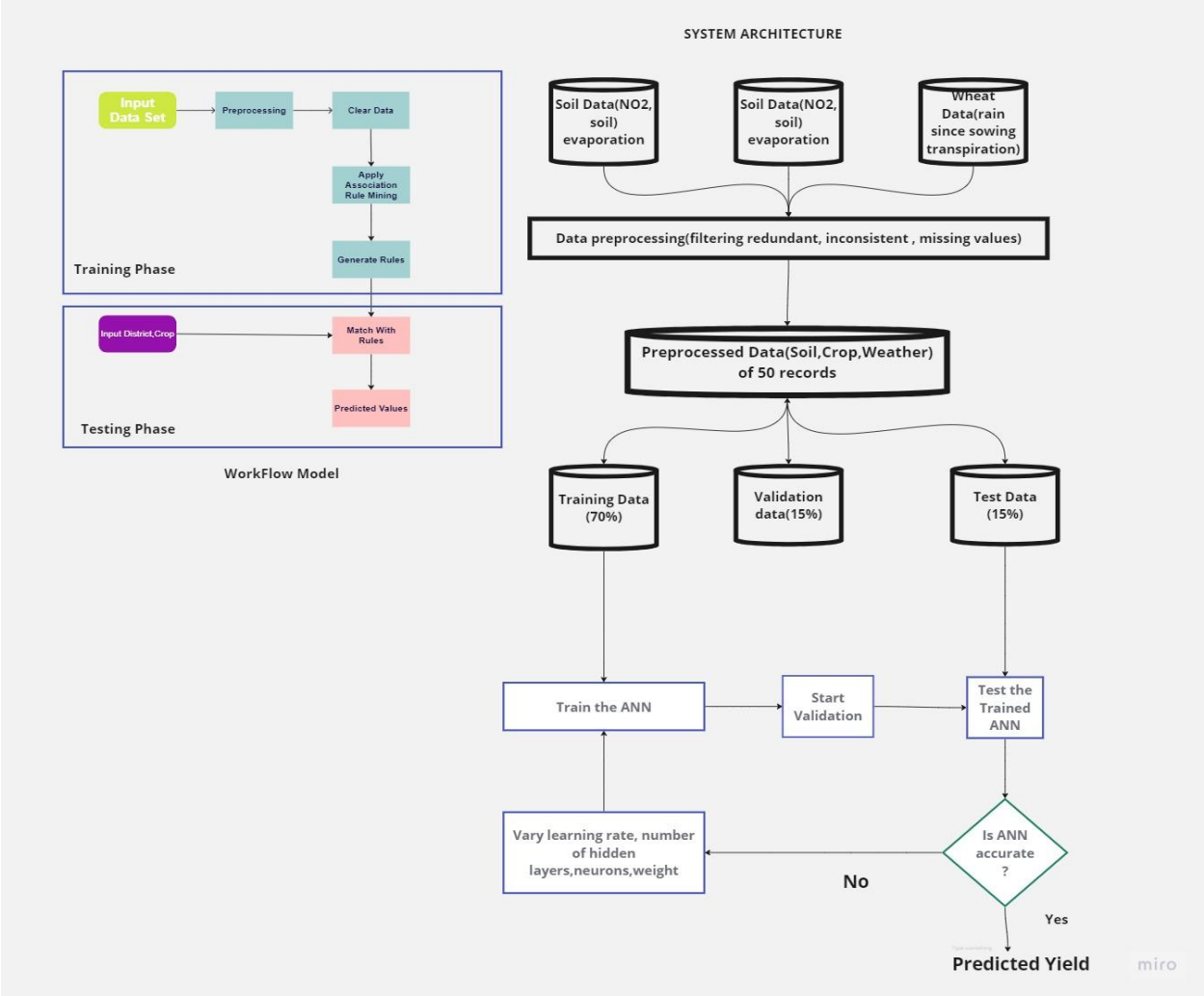


Figure 3 depicts a workflow paradigm with two phases, namely the training phase and the testing phase.

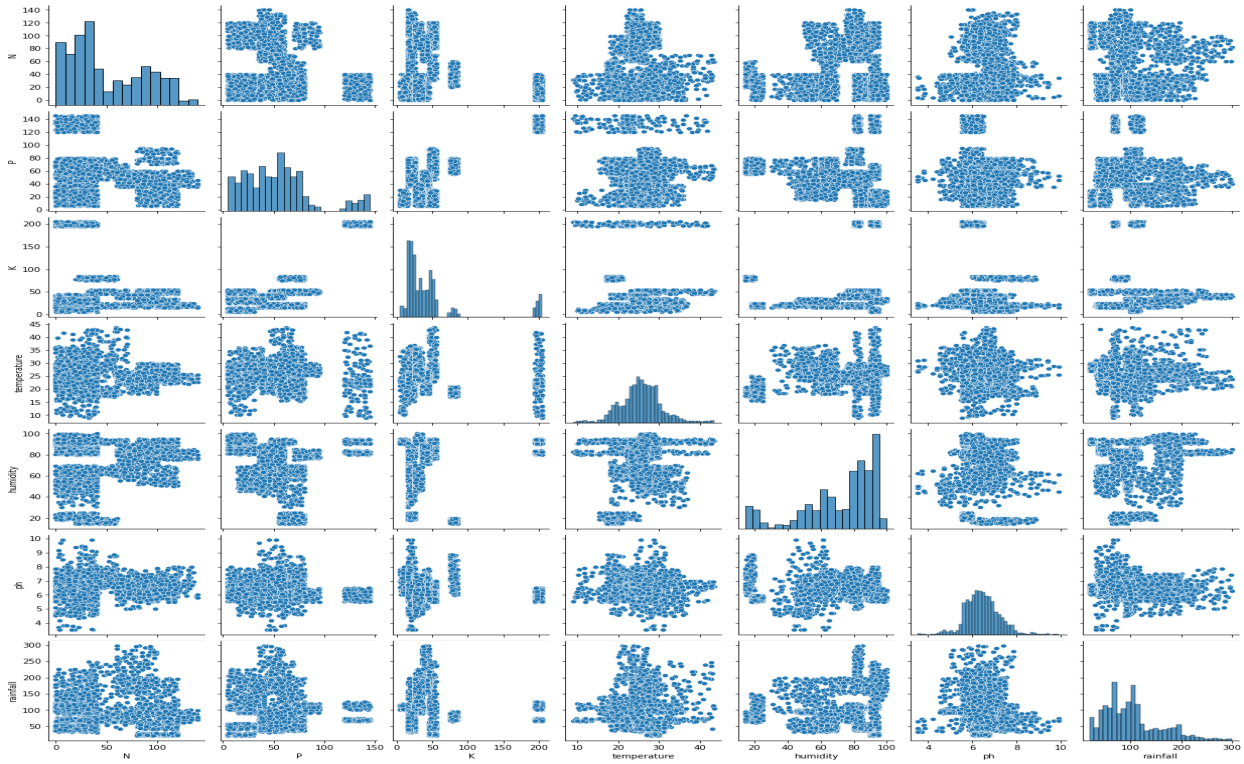
The input layer receives the input data during the training phase, and the hidden layer does preprocessing. The data cluster is taken by the hidden layer, which processes it and feeds the output to the input of the output layer. Input and hidden layers are included in the training phase, while the output layer is only present in the testing phase. Testing data is never larger than training data. The training phase includes all of the key features (such as data preparation and clustering). Redundancy checks, data filters, outlier detection, correlation tests, feature extraction, and other operations are included in the preprocessing of data. Clustering is the act of grouping data with comparable properties. The data is fed into the testing phase when the training phase is finished. The majority of the training phase involves the classification and forecasting of data. Data is classified using a relevant method, and prediction is utilised to foretell the value of the data.

The flow diagram of an artificial neural network for crop prediction is shown in Figure 4. Three datasets are fed into the input layer of the ANN technique: the soil data (such as soil fertility, soil evaporation, and soil moisture), the crop data (such as biomass, previous year crop production), and the specific grain data (to be planted). The input layer provides the preprocessed data to the hidden layer after preparing the data. ANN does Exploratory Data Analysis (EDA) and filters redundant data for inconsistent and missing values after collecting the dataset via input layer. The primary stage in model construction is data preparation. The learning rates of the model and, consequently, the accuracy of the model are impacted by redundant and missing values in the dataset.

Following preprocessing, the dataset is divided into training and testing data using an ANN, with training data making up 70% of the dataset and testing and testing and validation data making up 15% of the dataset. The model was then created using ANN and trained using a training set after that. When training is finished, data should be categorised and evaluated using a testing set to gauge the model's learning rate and correctness. The validation dataset will check whether the predicted ANN is accurate in the interim.
If the outcome is accurate, we can forecast how much a sample crop will produce, which will provide us with a list of the most suited crops to cultivate.

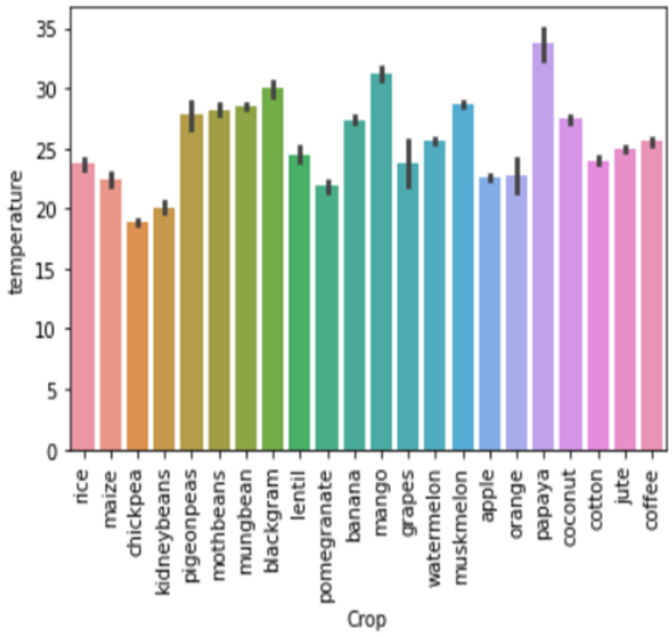
6 RESULT

Multivariate Visualization using Pairplot

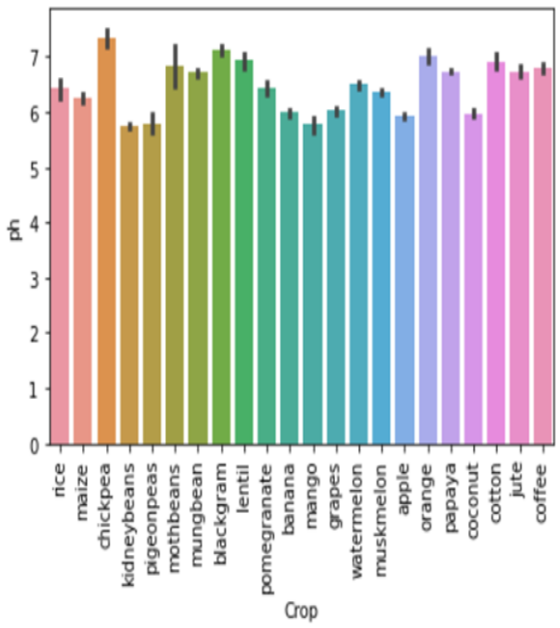


Bivariate Visualization using Bar Plots

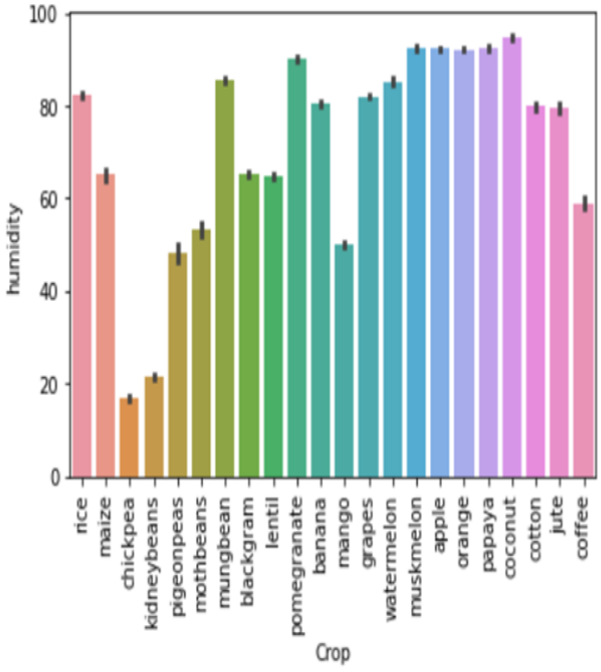
1. Crop and Temperature- Papaya requires highest temperature whereas Chickpea requires lowest temperature



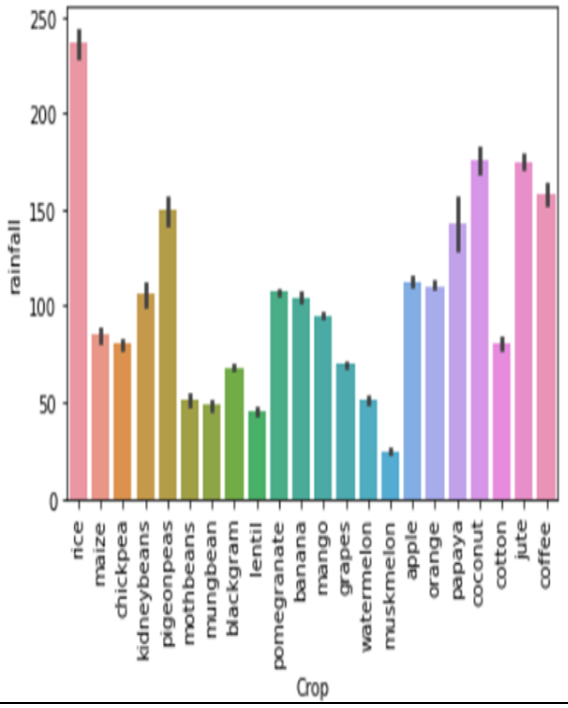
2. Crop and PH - Chickpea has the highest ph whereas mango has the lowest ph.



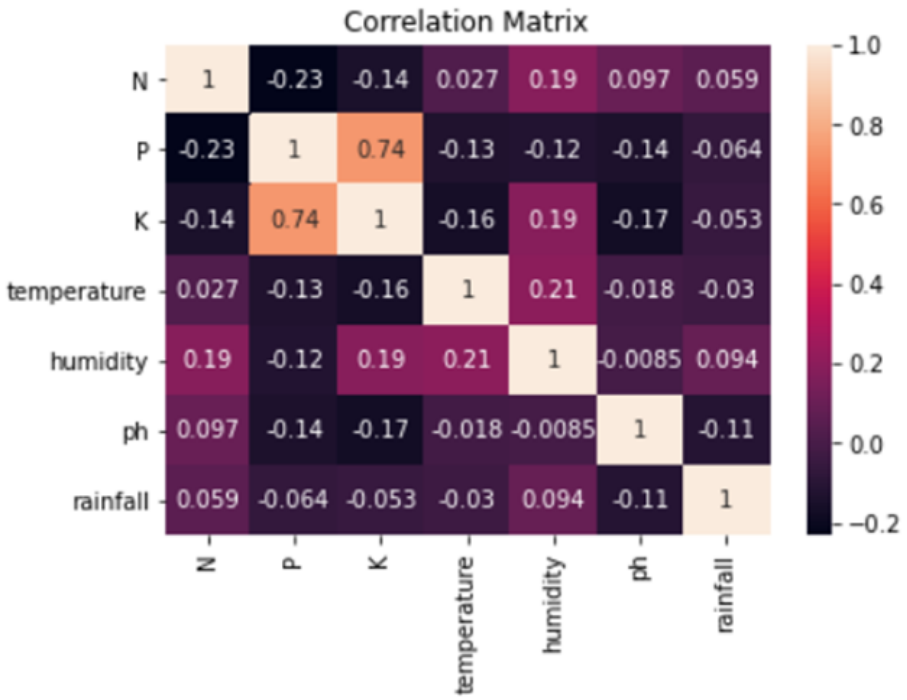
3. Crop and Humidity: Coconut requires the highest humidity and lowest for Chickpea.



4. Crop and Rainfall- Rice requires highest rainfall and Muskmelon requires lowest rainfall.



CORRELATION MATRIX

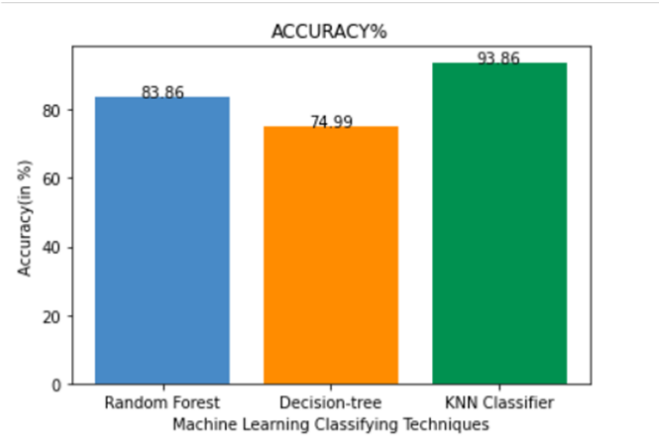


K, P are having highest correlation

TESTING ACCURACY

KNN Classifier has the highest accuracy.

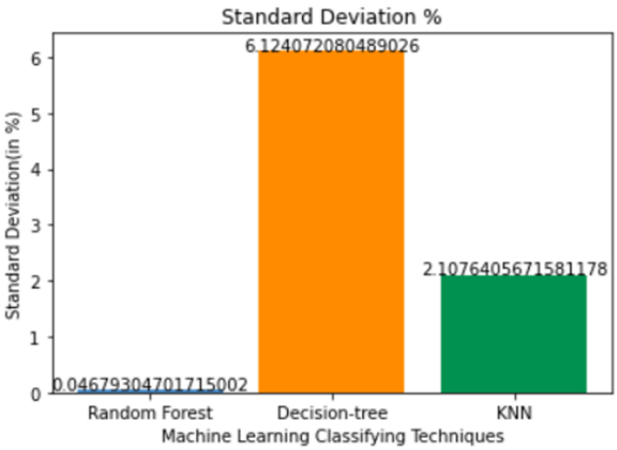
	Algorithms	Accuracy	Standard Deviation
0	Random Forest	83.86	0.046793
1	Decision-tree	74.99	6.124072
2	KNN Classifier	93.86	2.107641

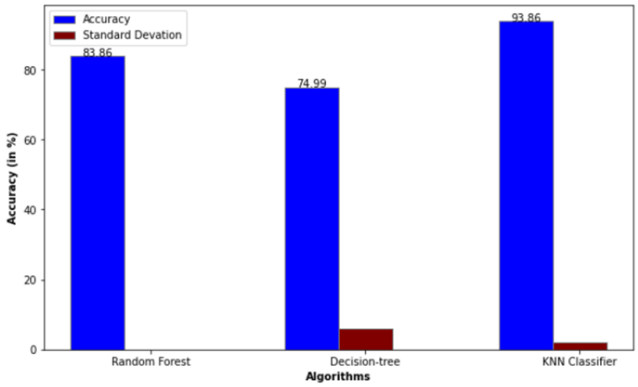


STANDARD DEVIATION

Decision tree has the highest SD and Random Forest has the lowest SD

Best Model





As we are predicting the crop we need the highest accuracy so according to that we are selecting KNN Classifier as it is giving the highest accuracy of 93.86%

7 ADVANTAGES & DISADVANTAGES

Advantages:

1. Improved accuracy: Machine learning models can provide more accurate predictions of crop yield by analyzing various factors.
2. Timely decision-making: Predictions before harvest enable farmers to make informed decisions and optimize agricultural practices.
3. Precision agriculture: Machine learning models integrated with sensor technologies enable precise resource allocation and reduce waste.
4. Enhanced productivity and profitability: Accurate predictions help farmers optimize operations, maximize productivity, and increase profitability.
5. Support for sustainable farming practices: Machine learning models identify challenges and recommend sustainable practices.

Disadvantages:

1. Data dependency: Machine learning models require large amounts of accurate and relevant data.
2. Complexity and technical expertise: Developing and deploying models requires specialized skills.
3. Limited interpretability: Some models provide predictions without clear explanations.
4. Lack of context awareness: Models may not account for sudden changes in conditions or market dynamics.

8 APPLICATIONS

(The areas where this solution can be applied)

1.Precision Irrigation: By integrating the model with sensor technologies, farmers can optimize water usage based on crop water requirements, reducing water wastage and mitigating the impact of drought.

2.Market Forecasting and Pricing: Agribusinesses and farmers can utilize the model to predict crop yields and market conditions. This helps in making informed decisions about pricing, marketing strategies, and crop diversification, leading to improved profitability.

3.Sustainable Resource Management: By analyzing factors such as soil quality, nutrient levels, and historical yield data, the model can recommend optimal resource management practices. This includes efficient fertilizer use, crop rotation, and soil conservation techniques, promoting long-term sustainability.

4.Agricultural Research and Development: Researchers can leverage crop prediction models to study the impact of environmental changes, new technologies, or crop varieties on yield. This knowledge can drive innovation in agricultural practices, resulting in improved productivity and resilience.

5.Government Policy and Planning: Crop prediction models can assist policymakers in formulating effective agricultural policies. By understanding crop production trends, policymakers can allocate resources, subsidies, and support systems to regions facing challenges such as low yields or frequent droughts.

In summary, crop prediction models using machine learning have the potential to revolutionize agriculture by enabling precision farming, mitigating risks from natural calamities, improving profitability, guiding sustainable practices, supporting research and development, and informing policy decisions.

9 CONCLUSION

In the model with the help of train, validation, and test split the model can ignore the overfitting issues and give better accuracy and prediction values.
As Decision Tree, KNN Classifier and Random Forest regression techniques are implemented on the input data to assess the best performance yielding method. These methods are compared using performance metrics. According to the analyses of metrics all the algorithms work well but k-Nearest Neighbour (kNN) gives a better accuracy score on test data with minimal standard deviation than Decision Tree Classifier. The proposed work can also be extended to analyse the climatic conditions and other factors for the crop and to increase the crop production.

10 FUTURE SCOPE

The future scope of crop prediction models using machine learning is vast and holds significant potential for transforming agriculture. Some key areas of future development and application include:

1. Climate Change Adaptation: As climate change continues to impact agricultural systems, crop prediction models can play a crucial role in helping farmers adapt to changing conditions. Models can be trained to consider climate change projections and provide recommendations for resilient crop varieties and adaptive farming practices.

2. Integration with IoT and AI Technologies: Combining crop prediction models with Internet of Things (IoT) devices and artificial intelligence (AI) technologies, such as drones and robotics, can enable real-time data collection and automated decision-making processes. This integration can lead to more efficient and precise agricultural practices.

3. Expansion to Small-Scale Farming: Currently, crop prediction models are predominantly applied to large-scale commercial farming. In the future, there is scope for adapting and tailoring these models to address the unique challenges faced by small-scale farmers, who make up a significant portion of the global agricultural sector.

4. Mobile Applications: Increasing accessibility and user-friendliness of crop prediction models through mobile based applications can empower farmers with easy-to-use tools for obtaining personalized recommendations and real-time updates.

5. Collaboration and Knowledge Sharing: Creating platforms for collaboration and knowledge sharing among farmers, researchers, and industry experts can facilitate the continuous improvement and refinement of crop prediction models. This can lead to the development of more robust and accurate models that cater to specific regional and crop-specific requirements.

6. Policy Support and Government Initiatives: Governments and policymakers can play a vital role in promoting the adoption and implementation of crop prediction models. By incorporating these models into agricultural policies, governments can provide necessary support, incentives, and resources to ensure widespread adoption and benefit for farmers.

11 BIBILOGRAPHY

References of previous works or websites visited/books referred for analysis about the project, solution previous findings etc.

APPENDIX A.

Source Code Attach the code for the solution built.

```
import numpy as np
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

crop_data=pd.read_csv("Crop_recommendation.csv")
crop_data
crop_data.size
crop_data.shape
crop_data.info()
crop_data.columns
crop_data.rename(columns = {'label':'Crop'}, inplace = True)
crop_data
crop_data.describe()
crop_data.isnull().sum()

df = crop_data.dropna()
df

df.isnull().values.any()

g = sns.pairplot(crop_data)
g

df.Crop.unique()

# get top 5 most frequent growing crops
n = 5
df['Crop'].value_counts()[:5].index.tolist()

sns.barplot(df["Crop"], df["temperature"])
plt.xticks(rotation = 90)
sns.barplot(df["Crop"], df["ph"])
plt.xticks(rotation = 90)
sns.barplot(df["Crop"], df["humidity"])
plt.xticks(rotation = 90)
sns.barplot(df["Crop"], df["rainfall"])
plt.xticks(rotation = 90)

df.corr()

sns.heatmap(df.corr(), annot =True)
plt.title('Correlation Matrix')

# Shuffling data to remove order effects
from sklearn.utils import shuffle

data  = shuffle(df,random_state=5)
data.head()

data=data.reset_index(drop=True)
data.head()

data.shape

# # Original dataset Known dataset
```

```
df1 = data.iloc[:1760]
df1

# # Unknown dataset

df2=data.iloc[1760:,0:7]
df2


# # Applying methods on original dataset

# Selection of Feature and Target variables.

x = df1[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]
target = df1['Crop']
x

from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

y=pd.get_dummies(df1['Crop'])

y

from sklearn.model_selection import train_test_split


x_train,x_val,y_train,y_val = train_test_split(x,y,test_size=0.25, random_state= 0)


print("x_train :",x_train.shape)
print("x_val :",x_val.shape)
print("y_train :",y_train.shape)
print("y_val :",y_val.shape)

x_test=df2

x_test

# # RandomForestClassifier

from sklearn.datasets import make_classification
from sklearn.multioutput import MultiOutputClassifier
from sklearn.ensemble import RandomForestClassifier

forest = RandomForestClassifier(random_state=1)
multi_target_forest = MultiOutputClassifier(forest, n_jobs=-1)
multi_target_forest.fit(x_train, y_train)


multi_target_forest.predict(x_val)
```

```
multi_score=multi_target_forest.score(x_val,y_val)
multi_score_accuracy = round(multi_score*100,2)
multi_score_accuracy

y_pred = multi_target_forest.predict(x_val)
y_pred


from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

from sklearn.metrics import accuracy_score
a1 = accuracy_score(y_val, y_pred)
print('Accuracy score:', accuracy_score(y_val, y_pred))

y_pred.shape


# # Testing accuracy
from sklearn.model_selection import cross_val_score
score = cross_val_score(multi_target_forest,X = x_val, y = y_pred,cv=7)
score

b1 = "{:.2f}".format(score.mean()*100)
b1 = float(b1)
b1

c1 = (score.std())
c1

print("Accuracy : {:.2f}%".format (score.mean()*100))
print("Standard Deviation : {:.2f}%".format(score.std()*100))

print('RFC Report')
print('=======================================')
print()
print(classification_report(y_val, y_pred))


# # Decision Tree

from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier(random_state=6)
multi_target_decision = MultiOutputClassifier(clf, n_jobs=-1)
multi_target_decision.fit(x_train, y_train)

y_pred = multi_target_decision.predict(x_val)
y_pred


# # Training Accuracy

# Calculating Accuracy

from sklearn.metrics import accuracy_score
```

```
a2 = accuracy_score(y_val,y_pred)
print('Accuracy score:', accuracy_score(y_val,y_pred))
a2


# # Testing Accuracy

from sklearn.model_selection import cross_val_score
score = cross_val_score(multi_target_decision,X = x_val, y = y_pred,cv=7)
score

b2 = "{:.2f}".format(score.mean()*100)
b2 = float(b2)
b2

c2 = (score.std()*100)
c2


# # KNN Classifier

from sklearn.neighbors import KNeighborsClassifier

knn_clf=KNeighborsClassifier()
model = MultiOutputClassifier(knn_clf, n_jobs=-1)
model.fit(x_train, y_train)

knn_pred = model.predict(x_val)
knn_pred

knn_pred.shape

# # Trainig accuracy

# Calculating Accuracy

from sklearn.metrics import accuracy_score
a3 = accuracy_score(y_val,knn_pred)
print('Accuracy score:', accuracy_score(y_val,knn_pred))
a3

# # Testing Accuracy

from sklearn.model_selection import cross_val_score
score = cross_val_score(model,X = x_val, y = knn_pred,cv=5)
score

b3 = "{:.2f}".format(score.mean()*100)
b3 = float(b3)
b3

c3 = (score.std()*100)
c3

import pandas as pd
```

```
data = {'Algorithms':['Random Forest', 'Decision-tree', 'KNN Classifier'],
        'Accuracy':[b1, b2, b3],
        'Standard Deviation':[c1,c2,c3]}

df = pd.DataFrame(data)

df

import numpy as np
import matplotlib.pyplot as plt

Algorithms = ['Random Forest', 'Decision-tree','KNN']
Accuracy = [c1, c2, c3]

def addlabels(x,y):
    for i in range(len(x)):
        plt.text(i,y[i],y[i],ha = 'center')

x_pos = np.arange(len(Accuracy))

plt.bar(x_pos, Accuracy, color= ['#488AC7','#ff8c00','#009150'])
plt.title("Standard Deviation %")
plt.xticks(x_pos, Algorithms)
plt.ylabel('Standard Deviation(in %)')
plt.xlabel('Machine Learning Classifying Techniques')
addlabels(Algorithms, Accuracy)
plt.show()

import numpy as np
import matplotlib.pyplot as plt

Algorithms = ['Random Forest', 'Decision-tree','KNN Classifier']
Accuracy = [b1, b2, b3]

x_pos = np.arange(len(Accuracy))

plt.bar(x_pos, Accuracy, color=['#488AC7','#ff8c00','#009150'])
plt.title("ACCURACY%")
addlabels(Algorithms, Accuracy)
plt.xticks(x_pos, Algorithms)
plt.ylabel('Accuracy(in %)')
plt.xlabel('Machine Learning Classifying Techniques')

plt.show()

import numpy as np
import matplotlib.pyplot as plt

barWidth = 0.25
fig = plt.subplots(figsize =(10, 6))

Algorithms = ['Random Forest', 'Decision-tree', 'KNN Classifier']
Accuracy = [b1, b2, b3]
Standard_Deviation = [c1,c2,c3]

br1 = np.arange(len(Accuracy))
```

```
br2 = [x + barWidth for x in br1]
br3 = [x + barWidth for x in br2]

addlabels(Algorithms, Accuracy)

plt.bar(br1, Accuracy, color ='blue', width = barWidth,
        edgecolor ='grey', label ='Accuracy')

plt.bar(br2, Standard_Deviation, color ='maroon', width = barWidth,
        edgecolor ='grey', label ='Standard Devation')

plt.xlabel('Algorithms', fontweight ='bold', fontsize = 10)
plt.ylabel('Accuracy (in %)', fontweight ='bold', fontsize = 10)
plt.xticks([r + barWidth for r in range(len(Accuracy))],
        Algorithms)

plt.legend()
plt.show()


crop_data.columns


# Get input from the user for the new data point
N = float(input('Enter the value for N: '))
P = float(input('Enter the value for P: '))
K = float(input('Enter the value for K: '))
temperature = float(input('Enter the temperature (in Celsius): '))
humidity = float(input('Enter the humidity (in %): '))
ph = float(input('Enter the pH value: '))
rainfall = float(input('Enter the rainfall (in mm): '))

# Create a new data point for making crop recommendations
new_data = pd.DataFrame({'N': [N], 'P': [P], 'K': [K], 'temperature': [temperature], 'humidity':
[humidity], 'ph': [ph], 'rainfall': [rainfall]})



# Use the trained model to make a crop recommendation for the new data point
prediction = multi_target_forest.predict(new_data)
prediction


predicted_crop_index = prediction.argmax()
predicted_crop_name = y.columns[predicted_crop_index]
print('\n\n\t\t\033[1mThe recommended crop is:\033[0m', predicted_crop_name)
```

—----THE END-------