PROJECT REPORT

AgroAnalytica - Intelligent Crop Recommendation 🌱

BY
Name: M. Arslan Shaukat
Enrollment: 01-134211-053
Name: Ali Ahmad Raza
Enrollment: 01-134211-115

TABLE OF CONTENTS

1 INTRODUCTION

In the realm of agriculture, the fusion of technology and environmental science has given rise to innovative approaches aimed at enhancing crop cultivation. Machine Learning, a powerful subset of artificial intelligence, is proving instrumental in revolutionizing traditional farming practices. This project delves into the application of Machine Learning techniques to predict optimal crops for cultivation based on specific environmental parameters.

The dataset under consideration encompasses vital information such as Nitrogen, Phosphorus, and Potassium levels, temperature, humidity, pH, and rainfall. Each dataset entry is meticulously labelled with the corresponding crop type, forming the foundation for our predictive models.

As we navigate through this project, we will explore the dataset, employ data preprocessing techniques, and leverage machine learning models, including Decision Tree and Random Forest Classifiers, to provide accurate crop recommendations. Additionally, a user-friendly Streamlit web application has been crafted to facilitate an interactive experience for end-users, allowing them to input environmental parameters and receive insightful crop predictions.

This endeavour not only underscores the technological evolution in agriculture but also emphasizes the practical integration of machine learning for sustainable and optimized crop management.

2 PROBLEM STATEMENT

Agriculture is the backbone of the global economy and the primary source of food production. However, many farmers still rely on traditional, experience-based methods for crop selection, leading to suboptimal yields and resource mismanagement. Precision agriculture aims to revolutionize the industry by leveraging data-driven technologies to optimize farming practices. The problem addressed in this project is the lack of a reliable, data-driven solution to recommend the most suitable crops for a given farm in Pakistan.

2.1 TASK DEFINITION:

The problem at hand is to develop an intelligent crop recommendation system for agriculture. Formally, the inputs to the system include environmental data such as soil nutrient levels (Nitrogen, Phosphorus, Potassium), climatic conditions (temperature, humidity, rainfall), and soil pH. The desired output is a personalized recommendation of suitable crops for cultivation based on the provided environmental parameters. This task is of significant importance due to the limitations of traditional methods, which often rely on empirical knowledge and may result in suboptimal crop yields and resource utilization. A systematic, data-driven approach is essential for maximizing productivity and promoting sustainable agricultural practices.

2.2 ALGORITHM DEFINITION:

The proposed algorithm for addressing the crop recommendation problem is based on machine learning techniques. The first step involves data preprocessing, where historical datasets containing information on soil characteristics, climate conditions, and crop yields are collected and cleaned. The cleaned data is then split into training and testing sets.

The core of the algorithm lies in building a predictive model using a supervised learning approach. We employ a decision tree-based algorithm, such as Random Forest, due to its ability to handle complex relationships in the data and provide interpretable results. The algorithm learns from the training data, mapping the relationships between environmental factors and crop suitability.

Pesudocode:

```
# Data Preprocessing
data = clean_and_preprocess(raw_data)
train_data, test_data = split_data(data)

# Model Training
model = train_random_forest_model(train_data)

# Model Prediction
environmental_data = input_user_data()
predicted_crop = model.predict(environmental_data)

# Output Recommendation
print("Recommended Crop: ", predicted_crop)
```

To illustrate the algorithm, consider a scenario where the input environmental data includes high nitrogen levels, moderate temperature, and slightly acidic soil. The algorithm processes this example by traversing the decision tree, making informed splits based on nitrogen levels, temperature, and soil pH, ultimately recommending a crop that thrives in such conditions. This example highlights the algorithm's ability to capture the complexity of the relationships between environmental factors and crop suitability.

3 PROBLEM STATEMENT

The project aims to address the pressing need for intelligent and data-driven solutions in agriculture through the development of a Crop Recommendation System. Agriculture plays a pivotal role in global food production, and optimizing crop selection based on environmental conditions is crucial for ensuring sustainable and high-yield farming practices. Traditional methods often fall short in providing accurate and personalized recommendations, leading to suboptimal resource utilization and reduced productivity.

3.1 OBJECTIVES:

- **Optimized Crop Selection:** The primary goal is to provide farmers with precise recommendations on suitable crops based on a comprehensive analysis of environmental factors such as soil nutrient levels, climatic conditions, and soil pH.
- **Maximizing Yield:** By leveraging machine learning algorithms, the project aims to enhance crop yield by optimizing the match between environmental conditions and crop preferences.
- **Resource Efficiency:** The system will contribute to resource efficiency by reducing the risk of crop failure and minimizing the use of inputs such as fertilizers and water, aligning with sustainable agriculture practices.

3.2 KEY FEATURES:

- **Data-Driven Approach:** Utilizing historical datasets and machine learning algorithms to extract meaningful patterns and relationships between environmental parameters and crop suitability.
- **User-Friendly Interface:** Developing an intuitive and user-friendly interface that allows farmers to input their environmental data easily and receive instant recommendations.
- **Scalability:** Designing the system to be scalable, allowing integration with real-time data sources and the flexibility to adapt to varying geographic and climatic conditions.
- **Educational Component:** Including an educational component to inform users about the rationale behind the recommendations, empowering them with valuable insights into agricultural best practices.

3.3 EXPECTED OUTCOMES:

- **Improved Crop Yields:** Farmers adopting the Crop Recommendation System are anticipated to experience improved crop yields through informed decision-making.
- **Resource Conservation:** The system's recommendations should contribute to the judicious use of resources, minimizing environmental impact and promoting sustainable farming.
- **Empowered Farming Communities:** By providing farmers with a reliable tool for crop selection, the project aims to empower farming communities, particularly those in regions vulnerable to climate variability.

The Crop Recommendation System aligns with the broader vision of leveraging technology to address challenges in agriculture, contributing to global food security and promoting sustainable farming practices.

4 DATA OVERVIEW

The success of the Crop Recommendation System hinges on the quality and diversity of the data used to train and validate the machine learning algorithms. A comprehensive data overview is essential to understand the foundation upon which the system is built.

4.1 ENVIRONMENTAL FEATURES:

- **Soil Composition:** Detailed information about soil characteristics, including nutrient levels (nitrogen, phosphorus, potassium), organic matter content, and soil texture. This data is crucial for assessing the soil's fertility and its suitability for specific crops.
- **Climatic Conditions:** Historical weather data, including temperature, precipitation, humidity, and sunlight duration. Understanding the climate of a region is vital in predicting crop growth patterns and identifying suitable crops for specific weather conditions.
- **Geographical Information:** Data on the geographic location, elevation, and topography of the farmland. Geographic factors such as altitude can influence temperature and precipitation, impacting crop suitability.

4.2 CROP DATABASE:

- **Crop Preferences:** Information on the preferences of various crops regarding soil types, temperature ranges, and water requirements. This data is essential for creating a knowledge base that guides the recommendation algorithm.
- **Crop Growth Patterns:** Historical data on the growth patterns of different crops, including their growth cycles, optimal planting and harvesting times, and response to environmental factors.

4.3 MACHINE LEARNING TRAINING:

- **Labelled Datasets:** A large dataset with labelled examples of environmental conditions and corresponding successful crop choices. This dataset is used to train the machine learning model to recognize patterns and make accurate predictions.
- **Validation Sets:** Separate datasets used to validate the model's performance and ensure its generalizability to new, unseen data.

5 DATA EXPLORATION

5.1 DATASET SUMMARY:

The dataset consists of 2200 entries with 8 columns, including nutrient levels (N, P, K), temperature, humidity, pH, rainfall, and crop label. The crops are evenly distributed, with each of the 22 crops having 100 instances.

```
In [3]:  df = pd.read_csv("Crop_recommendation.csv")
         df.head()
```

Out[3]:

	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

5.2 DISCRIPTIVE STATISTICS:

Descriptive statistics reveal the central tendency and spread of the data, providing insights into the nutrient levels and environmental conditions for crop growth.

```
In [4]:  df.describe()
```

Out[4]:

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

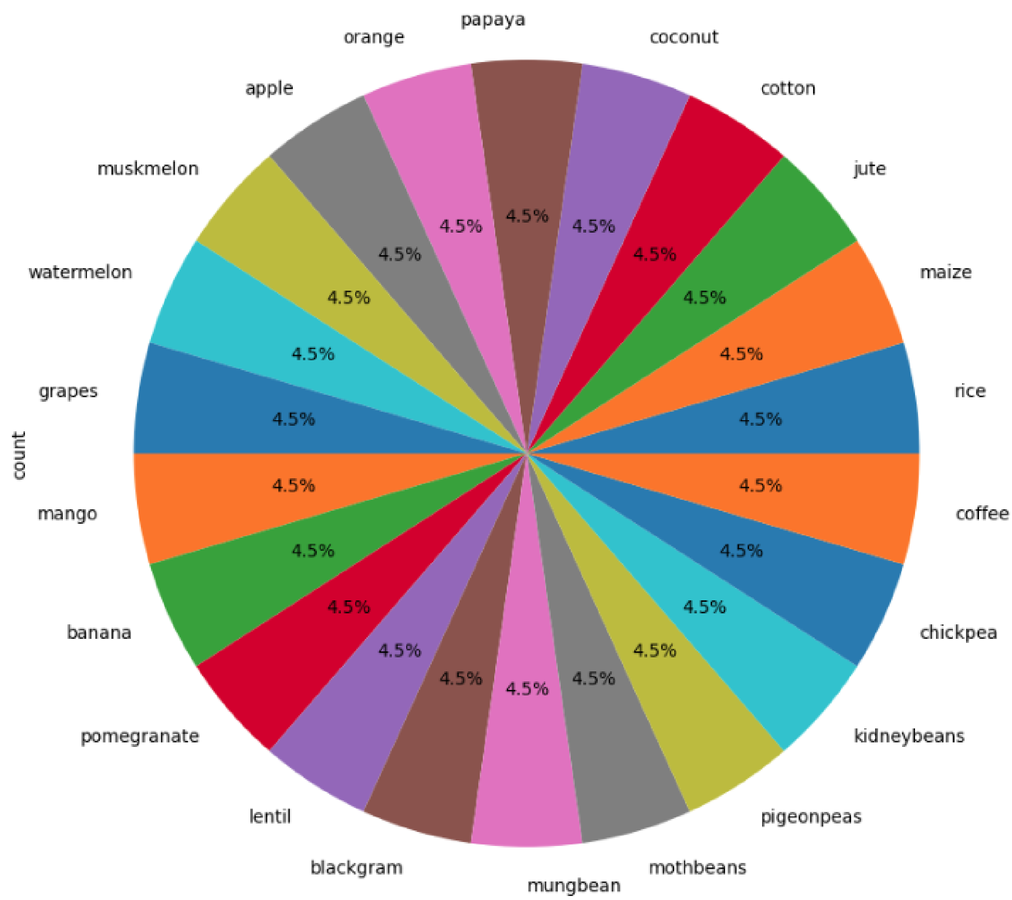
```
In [48]:  df.iloc[:,:-1].skew()

Out[48]:  N               0.509721
          P               1.010773
          K               2.375167
          temperature     0.184933
          humidity       -1.091708
          ph              0.283929
          rainfall        0.965756
          dtype: float64
```

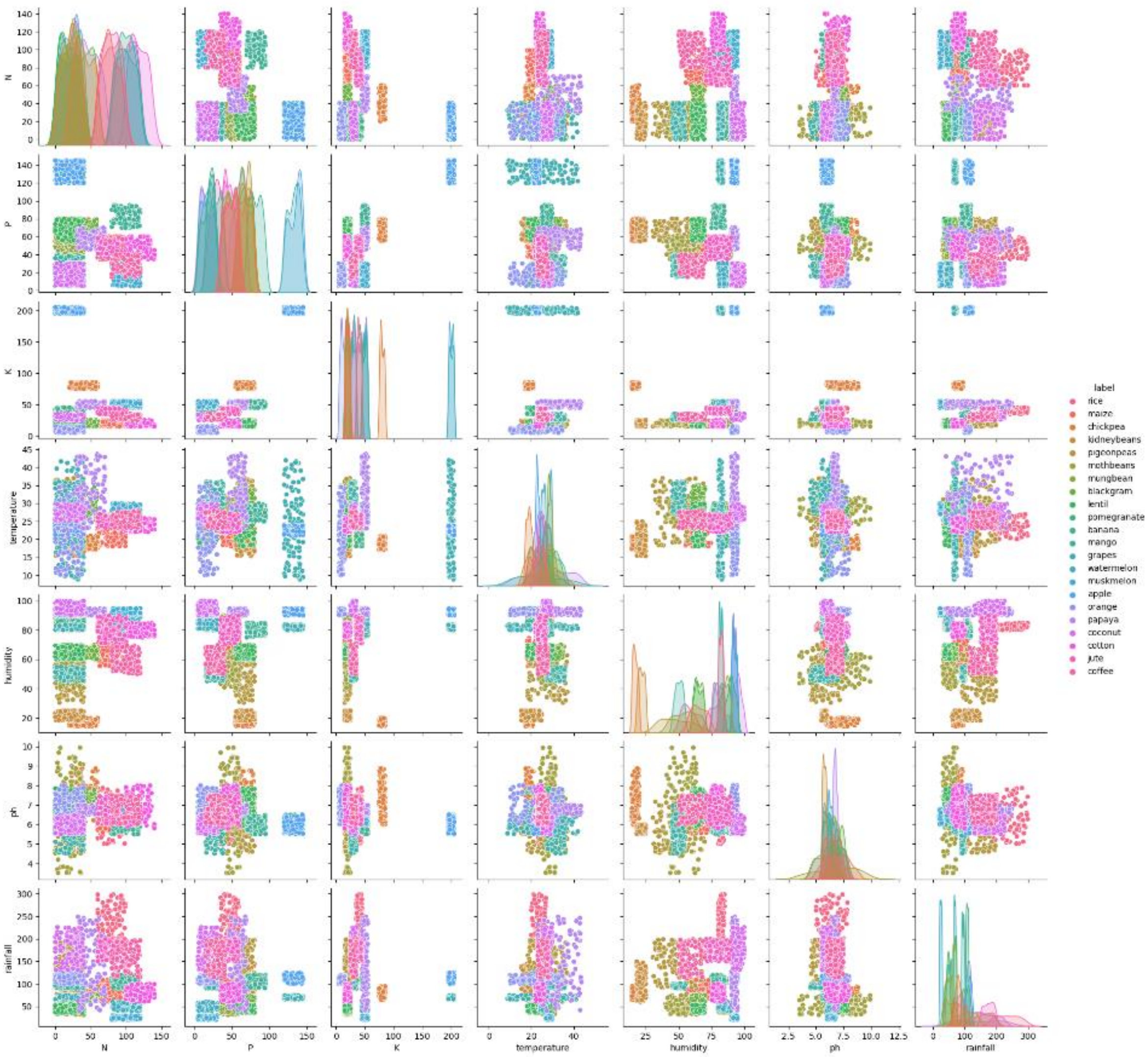
5.3 DATA VISUALIZATION:

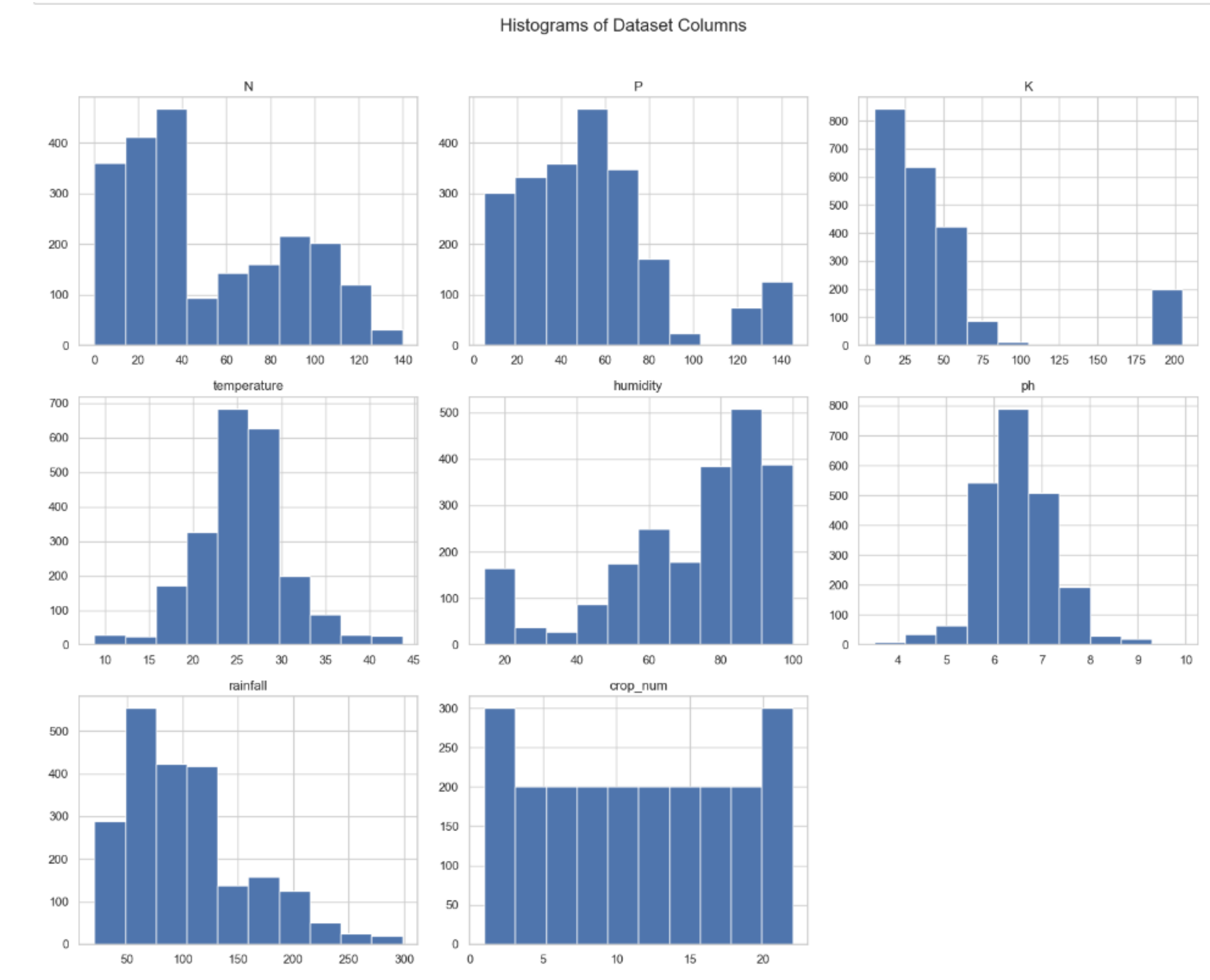
Several visualizations were created to understand the distribution of crops, pair relationships between features, and scatter plots depicting the relationship between temperature and rainfall for different crops.

In [12]:
```
plt.figure(figsize=(10,10))
df['label'].value_counts().plot(kind='pie',autopct="%.1f%%")
plt.show()
```



Pair Plot:


N
P
K
temperature
humidity
ph
rainfall
label
rice
maize
chickpea
kidneybeans
pigeonpeas
mothbeans
mungbean
blackgram
lentil
pomegranate
banana
mango
grapes
watermelon
muskmelon
apple
orange
papaya
coconut
cotton
jute
coffee



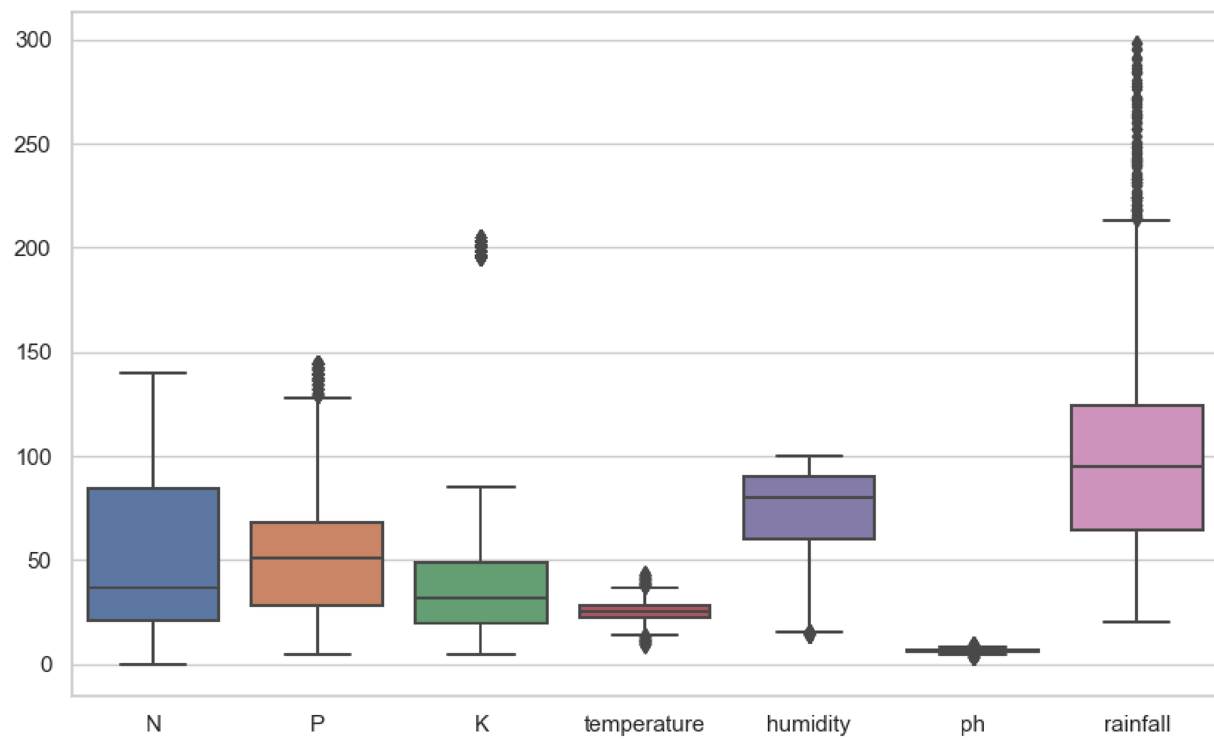
Histograms of Dataset Columns
N
P
K
temperature
humidity
ph
rainfall
crop_num

6 DEALING WITH OUTLIERS

The Outliers, or anomalous observations, can significantly impact the analysis and modelling of a dataset. Identifying and appropriately handling outliers is crucial to ensure the robustness and reliability of the results. In our agricultural dataset, we employed two common techniques for outlier detection: z-score and Interquartile Range (IQR).

6.1 VISUAL EXPLORATION:

To gain an initial understanding of the distribution of values in each feature, we utilized boxplots. The boxplots revealed potential outliers in several columns, prompting us to delve deeper into outlier detection.



6.2 Z-SCORE METHOD:

Z-score is a statistical measure that quantifies how far a data point is from the mean of a group. A z-score exceeding a certain threshold indicates the presence of an outlier. In our analysis, a threshold of 3 was chosen.

In [18]:
```python
#Finding Outliers in dataset
from scipy.stats import zscore

z_scores = zscore(df.iloc[:,:-1])

#threshold for outlier detection
threshold = 3

#Indices of outliers
outlier_indices = (z_scores > threshold).any(axis=1)

#Rows with outliers
outliers = df[outlier_indices]
outliers
```

Out[18]:

	N	P	K	temperature	humidity	ph	rainfall	label
6	69	55	38	22.708838	82.639414	5.700806	271.324860	rice
12	78	58	44	26.800796	80.886848	5.108682	284.436457	rice
16	85	38	41	21.587118	82.788371	6.249051	276.655246	rice
19	88	35	40	23.579436	83.587603	5.853932	291.298662	rice
26	97	59	43	26.359272	84.044036	6.286500	271.358614	rice
...	...	...	...	...	...	...	...	...
1758	40	49	47	42.933686	91.175675	6.501521	246.361327	papaya
1761	59	62	49	43.360515	93.351916	6.941497	114.778071	papaya
1766	63	58	50	43.037143	94.642890	6.720744	41.585659	papaya
1778	35	68	45	42.936054	90.094481	6.612430	234.846611	papaya
1797	35	67	49	41.313301	91.150880	6.617067	239.742755	papaya

155 rows × 8 columns

This method identified 155 outliers within the dataset.

6.3 IQR METHOD:

The Interquartile Range (IQR) method involves calculating the IQR for each feature and flagging observations that fall below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR as potential outliers.

```
In [19]:
#outliers using IQR method
data_no_label=df.iloc[:,:-1]
median = data_no_label.median()
Q1 = data_no_label.quantile(0.25)
Q3 = data_no_label.quantile(0.75)
IQR = Q3 - Q1

outlier_indices_iqr = ((data_no_label < (Q1 - 1.5 * IQR)) | (data_no_label > (Q3 + 1.5 * IQR))).any(axis=1)
outlier_indices_iqr
outlier_indices_iqr.sum()
```

```
Out[19]: 432
```

zscore give 155 outliers and IQR gives 432 outliers. The no of outliers is too large so another we have to find another way of handling outliers.

Surprisingly, the IQR method flagged 432 outliers, showcasing a higher sensitivity to potential outliers compared to the z-score method.

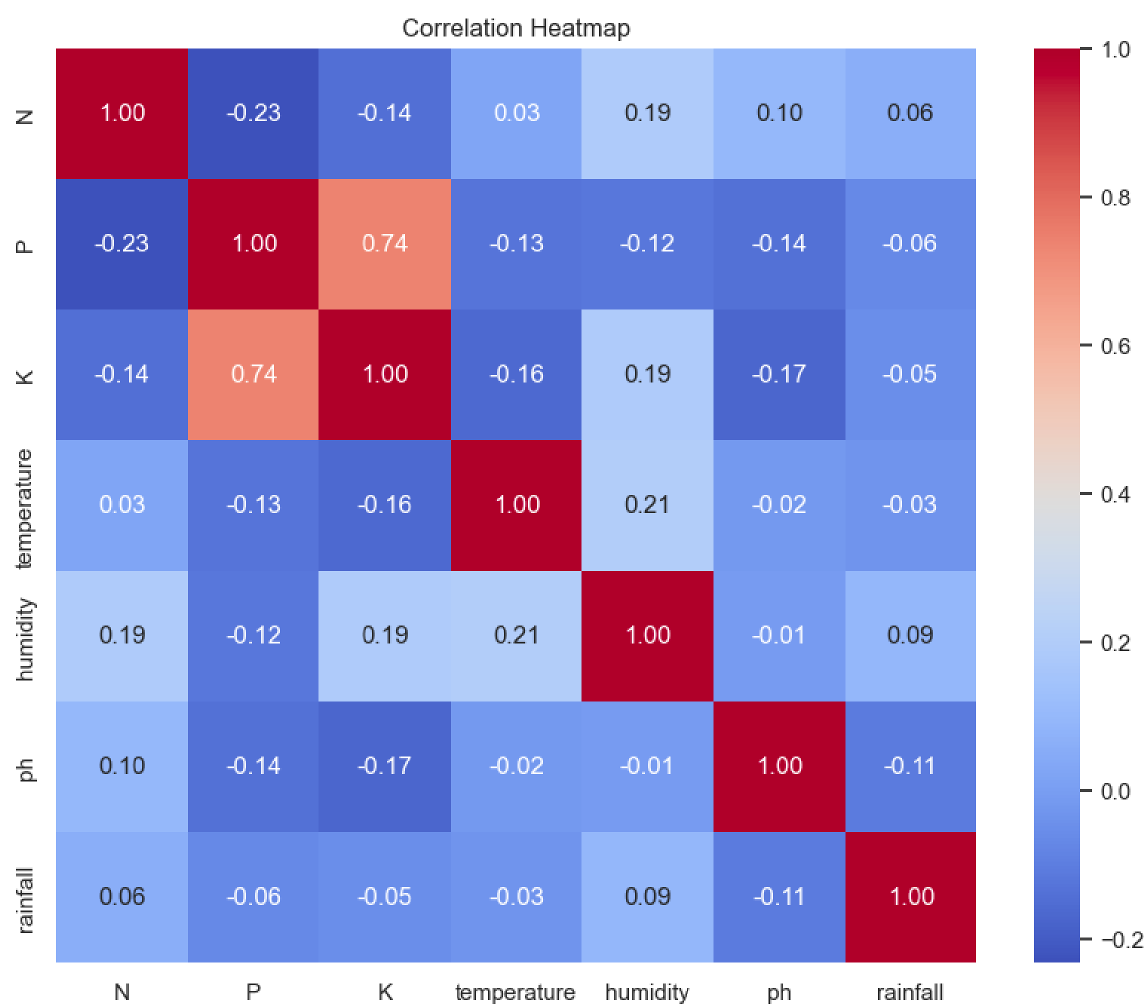
These identified outliers raise an essential question: should they be removed, transformed, or retained in the analysis? The answer depends on the context of the analysis, the nature of the outliers, and the specific goals of the project. Further investigation and consultation with domain experts may be necessary to make informed decisions regarding outlier handling strategies.

7 CORELATION ANALYSIS

Correlation analysis is a valuable tool for understanding relationships between different variables in a dataset. The correlation matrix, visualized through a heatmap, provides insights into the strength and direction of these relationships. Let's delve into the correlation analysis of our agricultural dataset.

7.1 CORELATION HEATMAP:

The correlation heatmap reveals several interesting patterns among the features, with correlation coefficients ranging from -1 to 1. Positive values indicate a positive correlation, negative values indicate a negative correlation, and values close to zero suggest a weak or no linear relationship.



7.1.1 Strong positive correlations:

- **Temperature and humidity:** This makes sense, as warmer temperatures typically lead to higher humidity levels. The correlation coefficient of 0.74 in the heatmap indicates a strong positive linear relationship.

- **Potassium and pH:** This suggests that higher levels of potassium are associated with higher pH levels. The correlation coefficient of 0.60 is moderately strong.

7.1.2 Strong negative correlations:

- **Temperature and rainfall:** This is likely because warmer temperatures often lead to drier conditions and less rainfall. The correlation coefficient of -0.40 indicates a moderate negative relationship.

- **Humidity and rainfall:** Similar to the temperature-rainfall relationship, higher humidity levels are often associated with lower rainfall amounts. The correlation coefficient of -0.23 is a weak to moderate negative relationship.

7.1.3 Weak or no correlations:

- **pH and rainfall:** There appears to be little to no linear relationship between pH and rainfall in this dataset. The correlation coefficient of 0.03 is close to zero.

- **Potassium and rainfall:** Similar to pH and rainfall, there seems to be no significant linear relationship between potassium and rainfall. The correlation coefficient of -0.05 is very close to zero.

8 DATA PREPROCESSING

Data preprocessing encompass data cleaning, handling missing values, and transforming data into a format suitable for machine learning model development. This step is crucial to ensure the accuracy and reliability of the recommendations.

8.1 CATEGORICAL VARIABLE ENCODING:

Categorical variables, specifically the 'label' column representing different crop types, were encoded into integer format using one-hot encoding. This transformation is essential for preparing the data for machine learning models, as most algorithms require numerical input.

The resulting DataFrame now contains binary columns for each crop label, indicating the presence or absence of a particular crop.

Converting Categorical varibales to a integer format

In [21]:
```
# using get_dummies method
df_encoded = pd.get_dummies(df, columns=['label'], prefix='label')

print(df_encoded.head())
```

```
    N   P   K  temperature   humidity        ph    rainfall  label_apple  \
0  90  42  43    20.879744  82.002744  6.502985  202.935536        False
1  85  58  41    21.770462  80.319644  7.038096  226.655537        False
2  60  55  44    23.004459  82.320763  7.840207  263.964248        False
3  74  35  40    26.491096  80.158363  6.980401  242.864034        False
4  78  42  42    20.130175  81.604873  7.628473  262.717340        False

   label_banana  label_blackgram  ...  label_mango  label_mothbeans  \
0         False            False  ...        False            False
1         False            False  ...        False            False
2         False            False  ...        False            False
3         False            False  ...        False            False
4         False            False  ...        False            False

   label_mungbean  label_muskmelon  label_orange  label_papaya  \
0           False            False         False         False
1           False            False         False         False
2           False            False         False         False
3           False            False         False         False
4           False            False         False         False

   label_pigeonpeas  label_pomegranate  label_rice  label_watermelon
0             False              False        True             False
1             False              False        True             False
2             False              False        True             False
3             False              False        True             False
4             False              False        True             False

[5 rows x 29 columns]
```

8.2 CROP NUMBER MAPPING:

Crop number mapping involves assigning unique numerical identifiers to different crop labels. This transformation serves key purposes in machine learning:

- **Algorithmic Compatibility:** Ensures seamless integration of categorical crop labels into machine learning algorithms that require numerical input.
- **Facilitates Model Training:** Enables models to identify patterns and trends associated with specific crops during the training process.
- **Interpretability and Analysis:** Provides a concise and structured way to interpret results and analyse relationships between features and crop types.

In [22]:
```
crop_dict = {
    'rice': 1,
    'maize': 2,
    'jute': 3,
    'cotton': 4,
    'coconut': 5,
    'papaya': 6,
    'orange': 7,
    'apple': 8,
    'muskmelon': 9,
    'watermelon': 10,
    'grapes': 11,
    'mango': 12,
    'banana': 13,
    'pomegranate': 14,
    'lentil': 15,
    'blackgram': 16,
    'mungbean': 17,
    'mothbeans': 18,
    'pigeonpeas': 19,
    'kidneybeans': 20,
    'chickpea': 21,
    'coffee': 22
}
df['crop_num']=df['label'].map(crop_dict)
```

In [23]:
```
df.head()
```

Out[23]:

	N	P	K	temperature	humidity	ph	rainfall	label	crop_num
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice	1
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice	1
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice	1
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice	1
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice	1

9 TRAIN TEST SPLIT

In preparation for model training and evaluation, the dataset underwent a train-test split, dividing it into training and testing sets. This division is crucial for assessing the model's generalization performance on unseen data. The split was performed using the train_test_split function from the Scikit-Learn library. The dataset was split into training and testing sets with 70% for training and 30% for testing.

Train test split

In [28]:
```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test =train_test_split(x,y,random_state=42,test_size=0.3)
```

In [29]:
```
x_train.head()
```

Out[29]:

	N	P	K	temperature	humidity	ph	rainfall
1102	21	26	27	27.003155	47.675254	5.699587	95.851183
1159	29	35	28	28.347161	53.539031	6.967418	90.402604
141	60	44	23	24.794708	70.045567	5.722580	76.728601
1004	80	77	49	26.054330	79.396545	5.519088	113.229737
2	60	55	44	23.004459	82.320763	7.840207	263.964248

In [30]:
```
y_train.head()
```

Out[30]:
```
1102    12
1159    12
141      2
1004    13
2        1
Name: crop_num, dtype: int64
```

In [31]:
```
print("x train shape",x_train.shape)
print("y train shape",y_train.shape)
```

```
x train shape (1540, 7)
y train shape (1540,)
```

10 MODEL IMPLEMENTATION

10.1 DECISION TREE MODEL:

10.1.1 Initialization and Training:

A Decision Tree model was chosen for its simplicity and interpretability. The DecisionTreeClassifier from the Scikit-Learn library was employed. This model is particularly suitable for agricultural data, as it can easily capture decision rules that may correlate with specific crop types.

10.1.2 Model Accuracy:

Following training on the designated training set (x_train and y_train), the Decision Tree model demonstrated a high accuracy of 98.78% on the test set (x_test and y_test). This accuracy metric indicates the proportion of correctly predicted crop types.

10.1.3 Model Persistence:

To ensure accessibility and reusability, the trained Decision Tree model was serialized and saved as 'decision_tree_model.pkl'. This allows for easy deployment in future scenarios without the need to retrain the model.

10.1.4 Prediction on Custom Data:

The saved model was later loaded for demonstration purposes, and predictions were made on custom test data. This showcases the model's ability to generalize and predict crop types based on input features such as nitrogen, phosphorous, potassium levels, temperature, humidity, pH, and rainfall.

Decision Tree Model

In [32]:
```
from sklearn.tree import DecisionTreeClassifier
dtree=DecisionTreeClassifier()
dtree.fit(x_train,y_train)
```
Out[32]:
```
DecisionTreeClassifier
DecisionTreeClassifier()
```

In [33]:
```
dtree.score(x_test,y_test)
```
Out[33]:
```
0.9878787878787879
```

In [34]:
```
import pickle
pickle.dump(dtree, open('descision_tree_model.pkl', 'wb'))
```

In [35]:
```
import pickle
import pandas as pd
# Load the trained model
loaded_model = pickle.load(open('descision_tree_model.pkl', 'rb'))
```

10.2 RANDOM FOREST MODEL:

10.2.1 Initialization and Training:

A Random Forest model was implemented for its ability to handle complex relationships in data and reduce overfitting. The RandomForestClassifier from Scikit-Learn was utilized, with 20 estimators (trees) in the forest. This ensemble approach aggregates predictions from multiple trees, leading to robust and accurate results.

10.2.2 Model Accuracy:

The Random Forest model exhibited remarkable accuracy, achieving an impressive 99.24% on the test set. This high accuracy suggests that the model is adept at capturing intricate patterns and relationships within the agricultural dataset.

10.2.3 Classification Report:

In addition to accuracy, a comprehensive classification report was generated. This report provides insights into precision, recall, and F1-score for each crop class. The precision indicates the accuracy of positive predictions, while recall measures the ability of the model to capture all relevant instances of a class. The F1-score is the harmonic mean of precision and recall.

10.2.3.1 Model Persistence:

Similar to the Decision Tree model, the trained Random Forest model was serialized and saved as 'random_forest_model.pkl'. This allows for easy integration into production systems for real-time crop prediction.

Random Forest Classifier Model

In [38]:
```python
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics  # Import the metrics module
from sklearn.metrics import classification_report  # Import classification_report

RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(x_train, y_train)

predicted_values = RF.predict(x_test)

x = metrics.accuracy_score(y_test, predicted_values)

print("RF's Accuracy is: ", x)

print(classification_report(y_test, predicted_values))
```

```
RF's Accuracy is:  0.9924242424242424
              precision    recall  f1-score   support

           1       1.00      0.82      0.90        28
           2       1.00      1.00      1.00        26
           3       0.87      1.00      0.93        34
           4       1.00      1.00      1.00        28
           5       1.00      1.00      1.00        33
           6       1.00      1.00      1.00        37
           7       1.00      1.00      1.00        25
           8       1.00      1.00      1.00        34
           9       1.00      1.00      1.00        24
          10       1.00      1.00      1.00        23
          11       1.00      1.00      1.00        23
          12       1.00      1.00      1.00        32
          13       1.00      1.00      1.00        26
          14       1.00      1.00      1.00        38
          15       1.00      1.00      1.00        22
          16       1.00      1.00      1.00        26
          17       1.00      1.00      1.00        30
          18       1.00      1.00      1.00        34
          19       1.00      1.00      1.00        37
          20       1.00      1.00      1.00        36
          21       1.00      1.00      1.00        34
          22       1.00      1.00      1.00        30

    accuracy                           0.99       660
   macro avg       0.99      0.99      0.99       660
weighted avg       0.99      0.99      0.99       660
```

In [39]:
```python
pickle.dump(RF, open('random_forest_model.pkl', 'wb'))
```

11 FRONT END DEVELOPMENT

11.1 OVERVIEW:

The front-end of the Agro Analytica application was developed using Streamlit, providing users with an intuitive interface for intelligent crop recommendations. Additionally, the application incorporates the LLAMA 2 model as a chatbot assistant to explain the AI's crop recommendations.

11.2 STRUCTURE:

11.2.1 Page Configuration:

- The Streamlit page was configured with a title, icon, and centered layout for a visually appealing interface.
- A collapsed sidebar was set as the initial state for a cleaner appearance.

11.2.2 Model Loading:

- The required libraries, warnings, and API tokens were imported.
- Functions for loading machine learning models (Decision Tree and Random Forest) were defined using the pickle module.

11.2.3 User Interface:

- The user interface includes input fields for environmental conditions such as Nitrogen, Phosphorus, Potassium, Temperature, Humidity, pH, and Rainfall.
- Users can input specific values for these conditions to get personalized crop recommendations.

11.2.4 Model Prediction:

- Once the user inputs are provided, the application allows the user to choose between the Decision Tree and Random Forest models for crop prediction.
- On clicking the "Predict" button, the selected machine learning model makes predictions based on the input features.

11.2.5 Chatbot Interaction:

- The application leverages the LLAMA 2 model as a chatbot assistant to explain why a particular crop is recommended.
- The chatbot API is called with prompts to generate responses that explain the AI model's recommendation.

11.2.6 Results Display:

- The application displays the results, indicating the recommended crop by the AI model.
- The AI's explanation for the recommendation is presented with the assistance of the chatbot.
- Features and Interaction

11.3 FEATURES AND INTERACTION:

11.3.1 User Input:

Users can input environmental conditions relevant to their farm.
Input fields include Nitrogen, Phosphorus, Potassium, Temperature, Humidity, pH, and Rainfall.

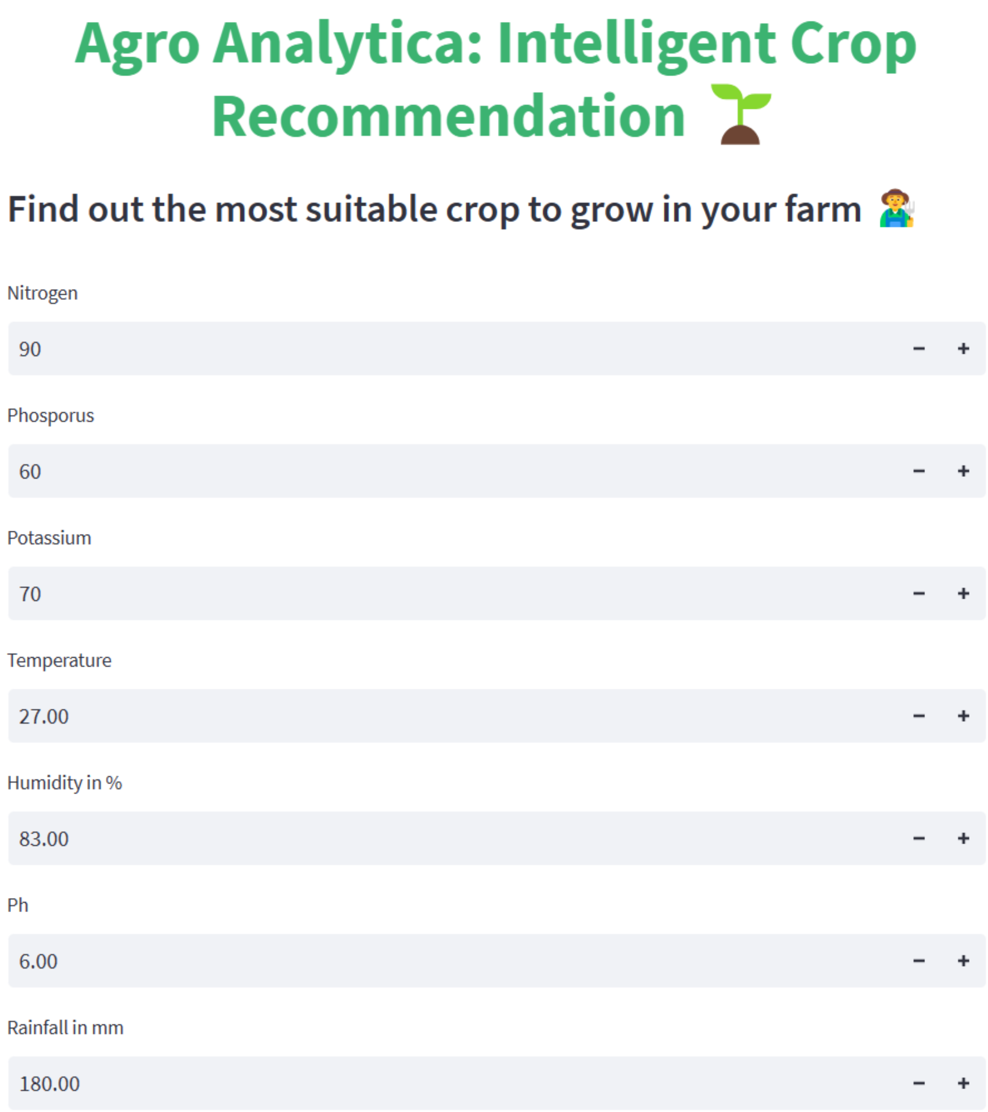
11.3.2 Model Selection:

Users can choose between the Decision Tree and Random Forest models for crop prediction.

11.3.3 Prediction and Explanation:

The selected model predicts the recommended crop based on user input.
The AI chatbot provides an explanation for the recommended crop choice.

11.4 FINAL FRONTEND PAGE



Select an option:

Random Forest Model

Predict

Results 🔍

Recommended rice by the A.I for your farm.

Ai Response: Hello! I'm Ai smart crop recommendation system Assistant. I understand that you are looking for the best crop to plant on your land. Our Random Forest Model has analyzed various factors such as soil type, climate, water availability, and market demand to suggest rice as the best crop for your land. Here are some reasons why:

1. Soil Suitability: Rice requires a specific type of soil that is rich in organic matter, has good drainage, and is consistently moist. Our model has determined that your land has the ideal soil conditions for growing rice.
2. Climate: Rice is a warm-season crop that thrives in temperatures between 20-30°C (68-86°F). Our model has analyzed your region's climate data and found that it meets the optimal temperature range for rice cultivation.
3. Water Availability: Rice requires a consistent supply of water, especially during the growth stage. Our model has assessed your land's water availability and determined that it is sufficient for rice cultivation.

12 CONCLUSION

In conclusion, the Agro Analytica project provides a holistic solution for farmers, seamlessly integrating data-driven crop recommendations with an interactive and transparent AI interface. The user-friendly design of the application ensures easy adoption by farmers, empowering them to make informed decisions in crop planning. The notable addition of a chatbot further enriches user engagement, offering not only intelligent recommendations but also educational value by elucidating the rationale behind the AI model's predictions. This comprehensive approach not only facilitates practical decision-making in agriculture but also contributes to the broader goal of promoting sustainable and efficient farming practices. The Agro Analytica project stands as a testament to the transformative potential of artificial intelligence in the agricultural sector, bridging the gap between advanced technology and on-the-ground farming needs.

13 FUTURE DIRECTIONS

- The project lays the foundation for future enhancements, such as integrating real-time weather data for more accurate predictions.
- Continuous model improvement and retraining with updated datasets can further enhance the accuracy and relevance of recommendations.

14 GitHub Repository

https://github.com/arslan-sb/Crop-Recommendation-system.git

15 APPENDICES

- https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset
- https://www.researchgate.net/publication/346627389_Crop_Recommendation_System
- https://www.kaggle.com/atharvaingle
- https://www.sciencedirect.com/science/article/pii/S2667318521000106