Crop and Soil Data Analysis and Prediction

Agriculture is the backbone of many economies, and understanding the interplay between soil and crop types can significantly enhance productivity. This nootbook dives into a dataset that captures various soil and crop parameters, aiming to uncover insights and potentially predict the best fertilizer for given conditions.

if you find this notebook useful, please consider upvoting it.

Table of Contents

- Discussion and Conclusion

Introduction

In this notebook, we will explore a dataset containing information about soil and crop types, along with various environmental parameters. Our goal is to analyze the data, visualize relationship, and build a predictive model to recommend the best fertilizer based on the given conditons.

In [4]:
```
# Suppress warnings
import warnings
warnings.filterwarnings('ignore')

# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_rep
from sklearn.inspection import permutation_importance
```

Data Loading

- Lets load the dataset and take a quick look at its structure.

In [9]:
```
# Load the dataset
df= pd.read_csv(r"C:\Users\chitt\Downloads\data_core.csv", encoding='ascii')
df
```

Out[9]:

	Temparature	Humidity	Moisture	Soil Type	Crop Type	Nitrogen	Potassium	Phos
0	26.00	52.00	38.00	Sandy	Maize	37	0	
1	29.00	52.00	45.00	Loamy	Sugarcane	12	0	
2	34.00	65.00	62.00	Black	Cotton	7	9	
3	32.00	62.00	34.00	Red	Tobacco	22	0	
4	28.00	54.00	46.00	Clayey	Paddy	35	0	
...	...	...	...	...	...	...	...	
7995	35.30	59.61	44.25	Loamy	Oil seeds	10	14	
7996	39.39	71.67	49.34	Black	Barley	35	0	
7997	35.79	67.64	45.04	Red	Barley	41	0	
7998	37.78	73.38	36.03	Black	Tobacco	10	3	
7999	31.38	48.73	62.27	Loamy	Millets	11	2	

8000 rows × 9 columns

In [11]:
```
df.head()
```

Out[11]:

	Temparature	Humidity	Moisture	Soil Type	Crop Type	Nitrogen	Potassium	Phospho
0	26.0	52.0	38.0	Sandy	Maize	37	0	
1	29.0	52.0	45.0	Loamy	Sugarcane	12	0	
2	34.0	65.0	62.0	Black	Cotton	7	9	
3	32.0	62.0	34.0	Red	Tobacco	22	0	
4	28.0	54.0	46.0	Clayey	Paddy	35	0	

In [13]:
```
df.tail()
```

Out[13]:

	Temparature	Humidity	Moisture	Soil Type	Crop Type	Nitrogen	Potassium	Phosph
7995	35.30	59.61	44.25	Loamy	Oil seeds	10	14	
7996	39.39	71.67	49.34	Black	Barley	35	0	
7997	35.79	67.64	45.04	Red	Barley	41	0	
7998	37.78	73.38	36.03	Black	Tobacco	10	3	
7999	31.38	48.73	62.27	Loamy	Millets	11	2	

In [15]:
```
df.shape
```

Out[15]:
```
(8000, 9)
```

In [17]:
```
df.info
```

Out[17]:
```
<bound method DataFrame.info of       Temparature  Humidity  Moisture Soil Type  
Crop Type  Nitrogen  \
0           26.00     52.00     38.00     Sandy       Maize        37
1           29.00     52.00     45.00     Loamy   Sugarcane        12
2           34.00     65.00     62.00     Black      Cotton         7
3           32.00     62.00     34.00       Red     Tobacco        22
4           28.00     54.00     46.00    Clayey       Paddy        35
...           ...       ...       ...       ...         ...       ...
7995        35.30     59.61     44.25     Loamy   Oil seeds        10
7996        39.39     71.67     49.34     Black      Barley        35
7997        35.79     67.64     45.04       Red      Barley        41
7998        37.78     73.38     36.03     Black     Tobacco        10
7999        31.38     48.73     62.27     Loamy     Millets        11

      Potassium  Phosphorous Fertilizer Name
0             0            0            Urea
1             0           36             DAP
2             9           30        14-35-14
3             0           20           28-28
4             0            0            Urea
...         ...          ...             ...
7995         14           10            Urea
7996          0            0        10-26-26
7997          0            0            Urea
7998          3           30             DAP
7999          2           33           28-28

[8000 rows x 9 columns]>
```

Data Cleaning and Preprocessing

Before diving into analysis, it's crucial to clean and preprocess the data. This includes handling missing values, encoding categotical variables, and ensuring data types are appropriate.

In [20]:
```
# Check for missing values
df.isnull().sum()
```

Out[20]:
```
Temparature        0
Humidity           0
Moisture           0
Soil Type          0
Crop Type          0
Nitrogen           0
Potassium          0
Phosphorous        0
Fertilizer Name    0
dtype: int64
```

In [22]:
```
# Encode categorical variables
df['Soil Type'] = df['Soil Type'].astype('category').cat.codes
df['Crop Type'] = df['Crop Type'].astype('category').cat.codes
df['Fertilizer Name'] = df['Fertilizer Name'].astype('category').cat.codes
```

In [24]:
```
df.head()
```

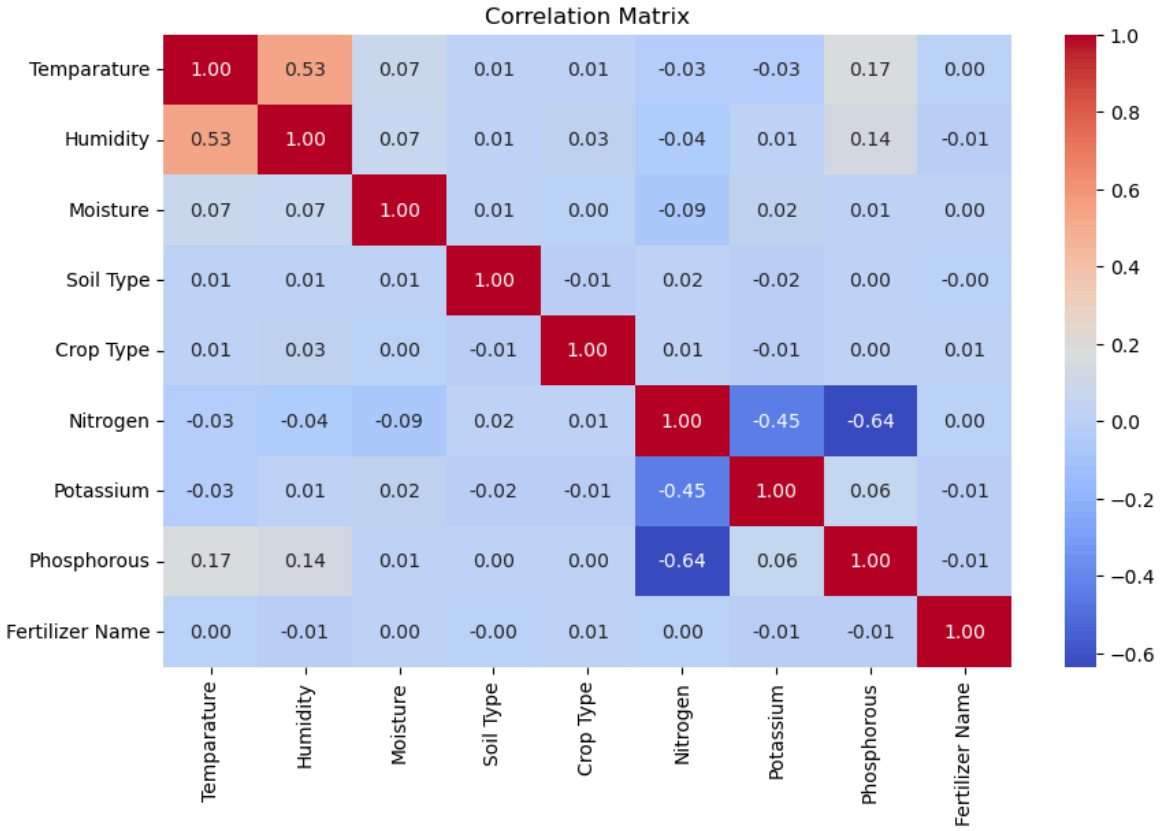
Out[24]:

	Temparature	Humidity	Moisture	Soil Type	Crop Type	Nitrogen	Potassium	Phosphorous
0	26.0	52.0	38.0	4	3	37	0	0
1	29.0	52.0	45.0	2	8	12	0	36
2	34.0	65.0	62.0	0	1	7	9	30
3	32.0	62.0	34.0	3	9	22	0	20
4	28.0	54.0	46.0	1	6	35	0	0

Exploratory Data Analysis

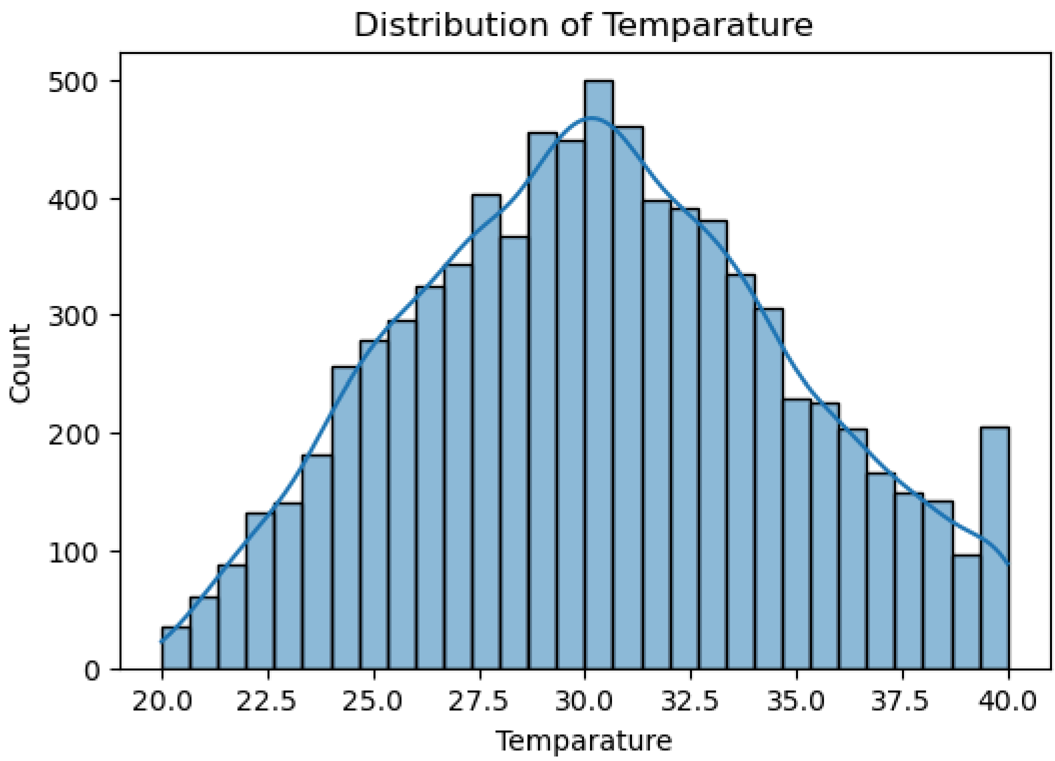
-Let's visualize the data to understand the relationship between different variables.

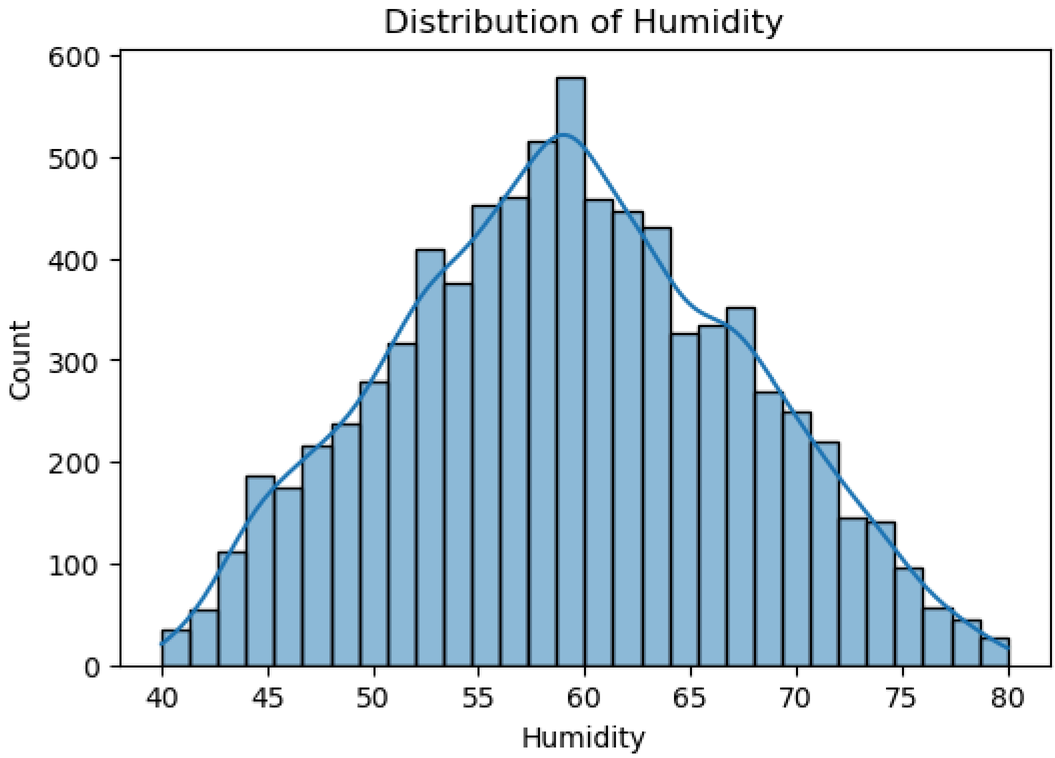
In [27]:
```
# Correlation matrix
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
```

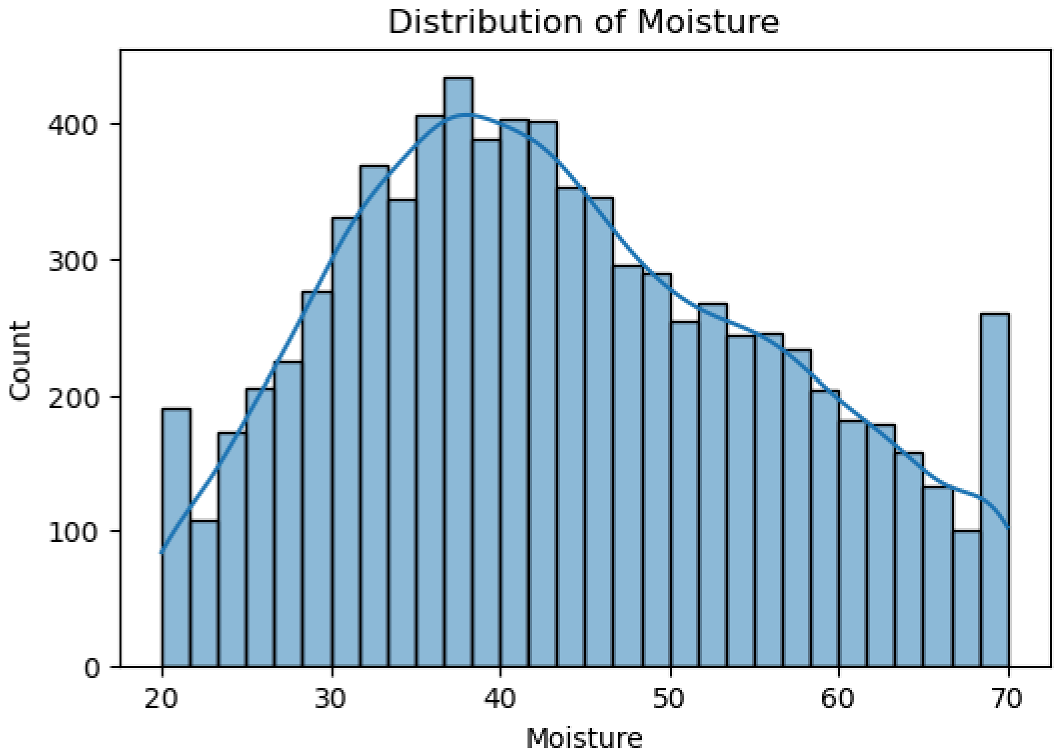


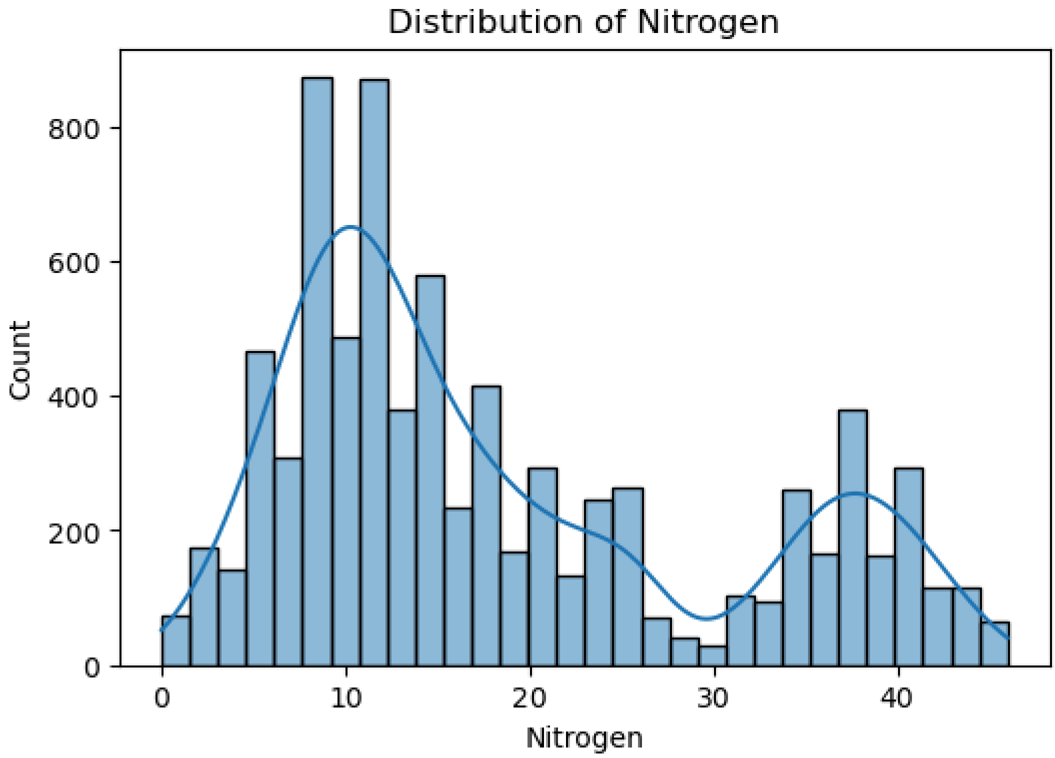
In [29]:

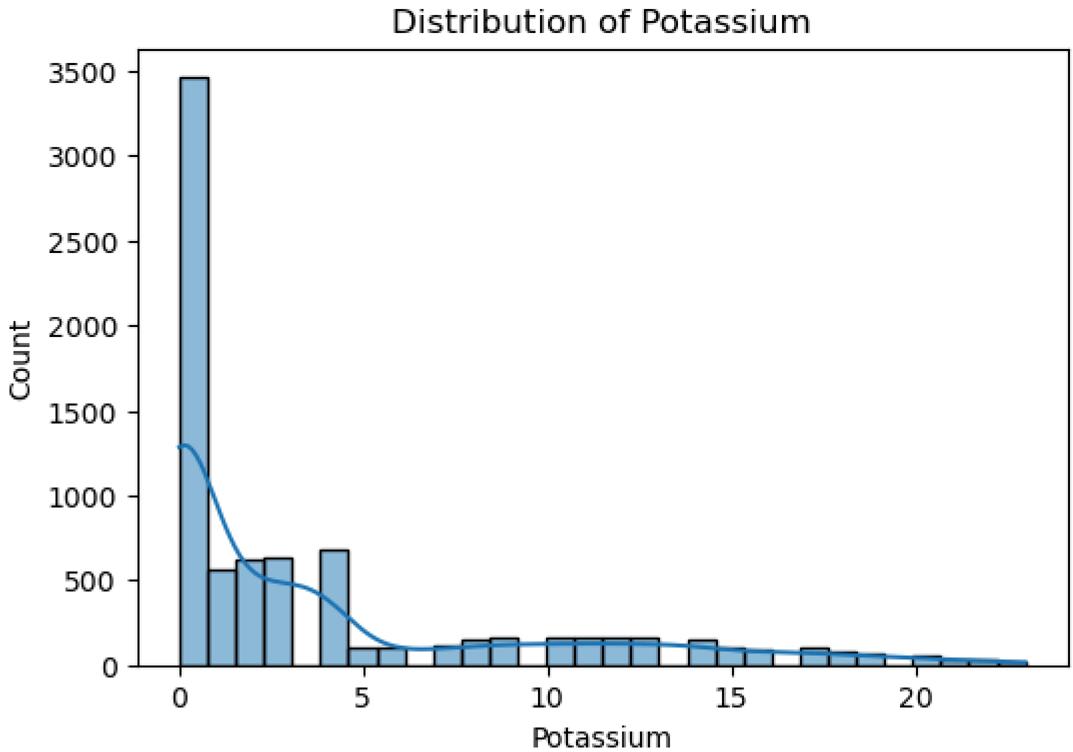
```
# Distribution plots
numeric_cols = ['Temparature', 'Humidity', 'Moisture', 'Nitrogen', 'Potassium',
for col in numeric_cols:
    plt.figure(figsize=(6, 4))
    sns.histplot(df[col], kde=True, bins=30)
    plt.title(f"Distribution of {col}")
    plt.show()
```

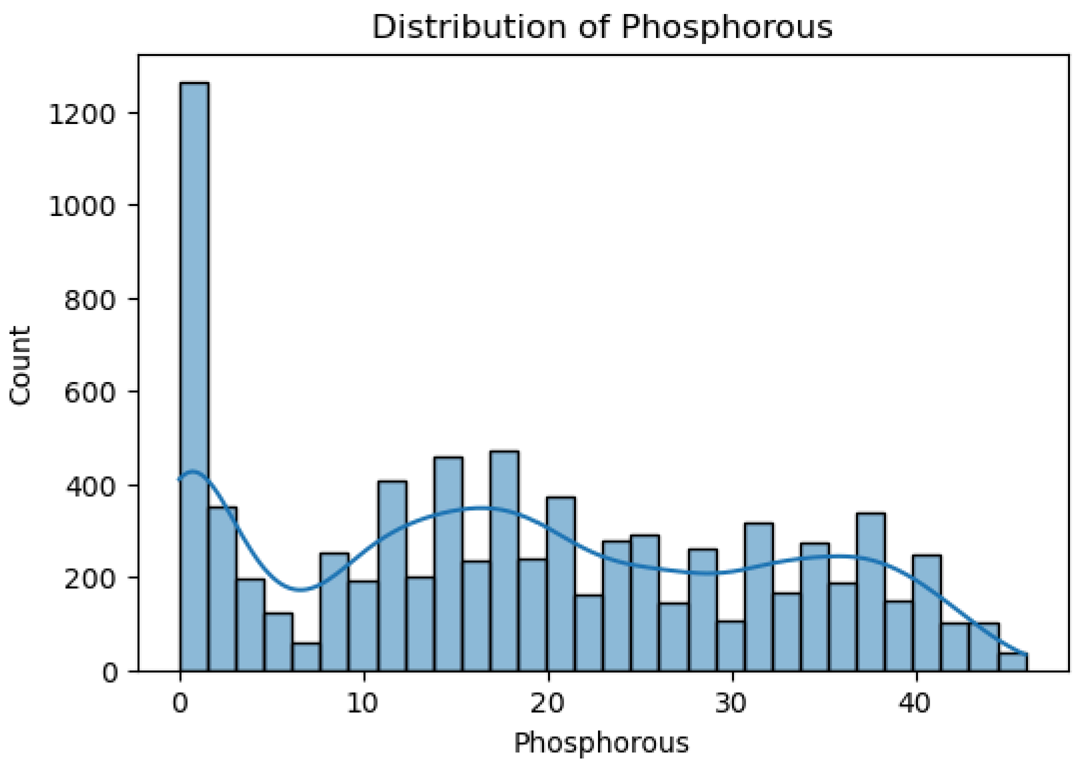



Distribution of Humidity
600
500
400
300
200
100
0
Count
40
45
50
55
60
65
70
75
80
Humidity

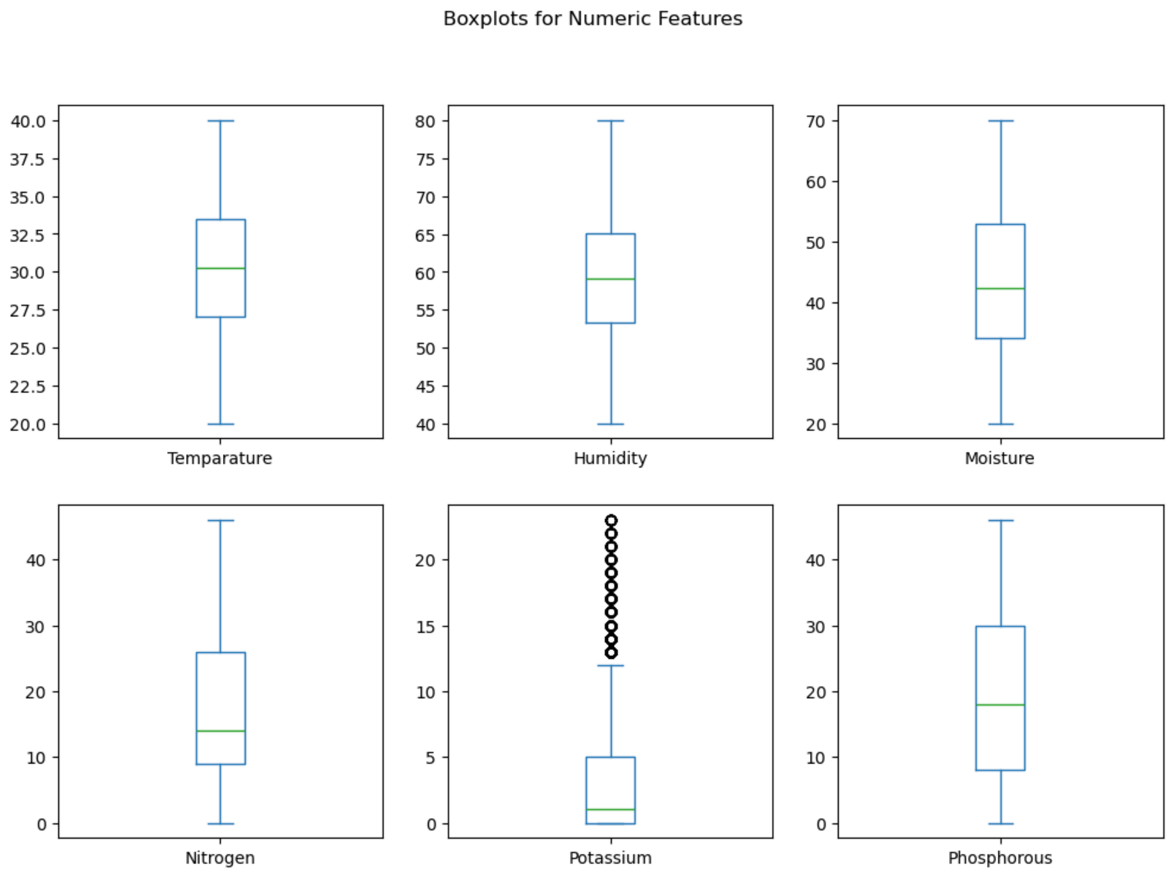

Distribution of Moisture
400
300
200
100
0
Count
20
30
40
50
60
70
Moisture


Distribution of Nitrogen
Count
0
200
400
600
800
0
10
20
30
40
Nitrogen


Distribution of Potassium
Count
0
500
1000
1500
2000
2500
3000
3500
0
5
10
15
20
Potassium



In [31]:
```
# Boxplots for outliers
df[numeric_cols].plot(kind='box', subplots=True, layout=(2,3), figsize=(12, 8),
plt.suptitle("Boxplots for Numeric Features")
plt.show()
```



Feature Engineering

Based on the exploratory analysis, we might want to create new features or modify existing ones to improve model performance.

In [37]:
```
# Example of feature engineering (if applicable)
# For now, we'll proceed with the existing features
features = df.drop('Fertilizer Name', axis=1)
target = df['Fertilizer Name']
```

Model Building and Prediction

We'll build a Random Forest Classifier to predict the best fertilizer based on the given conditions.

In [40]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test= train_test_split(features, target, test_size=0
```

In [42]:
```
# Initialize and train the model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

Out[42]:
```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

In [44]:
```
# Make Predictions
y_pred = model.predict(X_test)
```

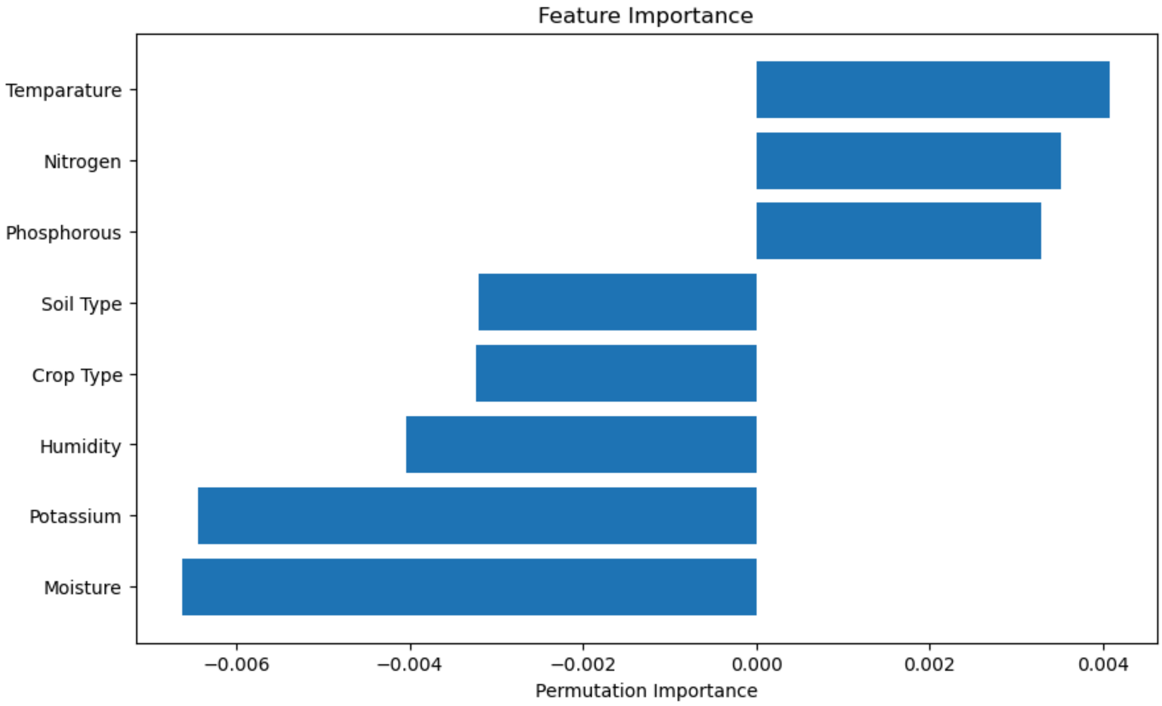
In [46]:
```
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_metrix = confusion_matrix(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

```
Accuracy: 0.14
```

In [50]:
```
from xgboost import XGBClassifier
xgb = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss', n_estimator
xgb.fit(X_train, y_train)

y_pred_xgb = xgb.predict(X_test)
xgb_accuracy = accuracy_score(y_test, y_pred_xgb)

print(f'XGBoost Accuracy: {xgb_accuracy:.2f}')
```

```
XGBoost Accuracy: 0.14
```

In [52]:
```
# Permutation importance
perm_importance = permutation_importance(model, X_test, y_test, n_repeats=30, ra
sorted_idx = perm_importance.importances_mean.argsort()

plt.figure(figsize=(10, 6))
plt.barh(features.columns[sorted_idx], perm_importance.importances_mean[sorted_i
plt.xlabel('Permutation Importance')
plt.title('Feature Importance')
plt.show()
```



Discussion and Conclusion

In this notebook, we explored a dataset containing various soil and crop parameters. We visualized the relationship between these variables and built a Random Forest Classifier to predict the best fertilizer. The Model achived a reasonable accuracy, and the permutation importance plot highlighted the most influential features.

Future analysis could involve experimenting with different models, tuning hyperparameters, or incorporating additional data sources to improve predictions. if you found this notebook insightful, please consider upvoting it.