

SAMPLE CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

data = {
    'pH': np.random.uniform(4.5, 8.5, 100),
    'moisture': np.random.uniform(10, 60, 100),
    'N': np.random.randint(0, 100, 100),
    'P': np.random.randint(0, 100, 100),
    'K': np.random.randint(0, 100, 100)
}

df = pd.DataFrame(data)

def classify(row):
    if row['N'] < 20:
        return 'Nitrogen Deficiency'
    elif row['P'] < 15:
        return 'Phosphorus Deficiency'
    elif row['K'] < 15:
        return 'Potassium Deficiency'
    elif row['pH'] < 5.5:
        return 'Low pH - Add lime'
    elif row['pH'] > 7.5:
        return 'High pH - Add sulfur'
    else:
        return 'Optimal'

df['recommendation'] = df.apply(classify, axis=1)

# Train ML model to simulate AI prediction
X = df[['pH', 'moisture', 'N', 'P', 'K']]
y = df['recommendation']
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X, y)

new_sample = {
    'pH': 5.2,
```

```

'moisture': 35.0,
'N': 18,
'P': 10,
'K': 25
}
sample_df = pd.DataFrame([new_sample])

prediction = model.predict(sample_df)[0]
print(f"\n ↴ Real-Time Soil Reading:")
print(sample_df)
print(f"\n 📈 AI Recommendation: {prediction}")

def plot_nutrient_levels(sample):
    nutrients = ['N', 'P', 'K']
    values = [sample[n] for n in nutrients]
    thresholds = [20, 15, 15]

    plt.figure(figsize=(8, 6))
    bars = plt.bar(nutrients, values, color=['#4CAF50', '#2196F3', '#FFC107'], label='Current Level')

    for idx, threshold in enumerate(thresholds):
        plt.axhline(y=threshold, color='r', linestyle='--', label='Threshold' if idx == 0 else "")

    plt.ylim(0, 100)
    plt.title('Soil Nutrient Levels')
    plt.xlabel('Nutrient')
    plt.ylabel('Concentration')
    plt.legend()
    plt.grid(True)
    for bar, value in zip(bars, values):
        plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height() + 2, f'{value}', ha='center')
    plt.tight_layout()
    plt.show()

plot_nutrient_levels(new_sample)

def show_trend(dataframe):
    plt.figure(figsize=(10, 5))
    sns.lineplot(data=dataframe[['N', 'P', 'K']])
    plt.title("Nutrient Trends Over Time (Simulated)")
    plt.xlabel("Sample Index")
    plt.ylabel("Nutrient Levels")

```

```
plt.grid(True)
plt.legend(["Nitrogen", "Phosphorus", "Potassium"])
plt.tight_layout()
plt.show()

show_trend(df)
```

OUTPUT:

