Abstract

Assess the risk of crop failure in an agricultural region based on   
environmental factors and farming practices.

Crop Failure Prediction Group 5

Contents

[Abstract 2](#_Toc197608939)

[Introduction 3](#_Toc197608940)

[Dataset Description 4](#_Toc197608941)

[Variables and data types 4](#_Toc197608942)

[Initial Data Observation 5](#_Toc197608943)

[Bayesian Network Structure 6](#_Toc197608944)

[Relationships among the variables 6](#_Toc197608945)

[Conditional Probability calculations 8](#_Toc197608946)

[Inference Scenarios and Analysis 10](#_Toc197608947)

[Heatmap Evidence 11](#_Toc197608948)

[Rainfall vs Soil\_Quality 11](#_Toc197608949)

[Rainfall vs Pest\_Infestation 12](#_Toc197608950)

[Rainfall vs Fertilizer\_Use 12](#_Toc197608951)

[Rainfall vs Temperature 13](#_Toc197608952)

[Rainfall vs Crop\_Yield 14](#_Toc197608953)

[Discussion and conclusion 15](#_Toc197608954)

[Appendix 16](#_Toc197608955)

# Abstract

This project focuses on predicting crop failure using a Bayesian Network constructed from environmental and agricultural data. The goal is to assess the probability of different levels of crop yield (Low, Medium, High) based on factors such as rainfall, temperature, soil quality, pest infestation, and fertilizer use.

By modeling causal relationships among variables in a Directed Acyclic Graph (DAG), we represent how environmental conditions and farming practices jointly influence crop outcomes. Conditional probability tables (CPTs) were estimated using real data, and probabilistic inference was performed to answer scenario-based questions and evaluate the effect of interventions like fertilizer application or changes in rainfall.

# Introduction

Agricultural productivity is a critical factor in food security, economic stability, and environmental sustainability. Predicting crop yield — especially identifying the risk of crop failure — allows farmers and policymakers to make informed decisions about resource allocation, risk mitigation, and intervention strategies. Traditional models often fail to account for the complex interdependencies between environmental and agronomic factors.

This project aims to develop a Bayesian Network to model the probabilistic relationships between multiple variables affecting crop yield, including rainfall, temperature, soil quality, pest infestation, and fertilizer use. The Bayesian framework is particularly suited for this task because it can capture both causal dependencies and uncertainty, allowing for dynamic, evidence-based inference under varying conditions.

By leveraging a real dataset and constructing a detailed causal model, we seek to evaluate the likelihood of crop failure under different environmental scenarios and assess the potential benefit of interventions such as fertilizer application or pest control. The resulting model offers an interpretable, data-driven approach for predictive analytics in agriculture.

# Dataset Description

The dataset utilized for this research, includes synthetic data that simulates how agricultural and environmental factors affect crop yield in a particular geographic area. This dataset captures a variety of factors related to crop production, which aids in the building of a Bayesian network.

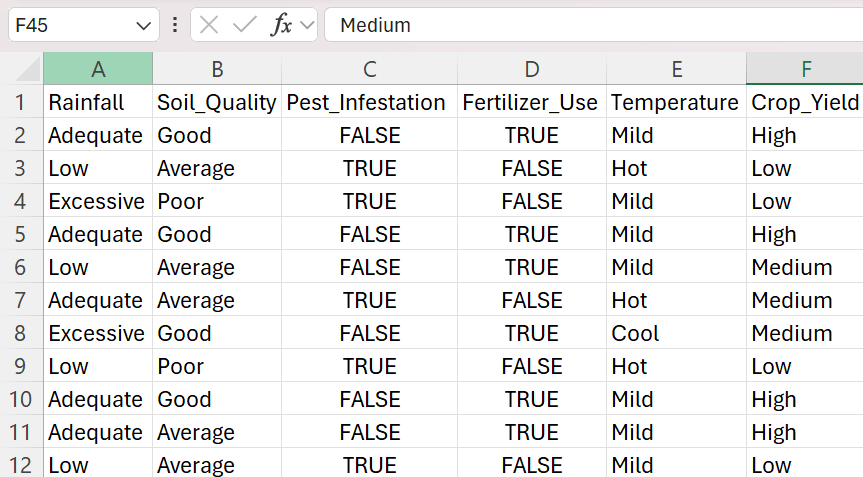
## Variables and data types

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable name** | **Description** | **Type** | **Value** |
| RainFall | |  | | --- | |  |  |  | | --- | | Amount of rainfall during the crop cycle | | |  | | --- | | Categorical |  |  | | --- | |  | | |  | | --- | | Low, Adequate, Excessive |  |  | | --- | |  | |
| |  | | --- | | Soil\_Quality |  |  | | --- | |  | | Fertility and structure of the soil | Categorical | |  | | --- | | Poor, Average, Good |  |  | | --- | |  | |
| |  | | --- | | Pest\_Infestation |  |  | | --- | |  | | |  | | --- | |  |  |  | | --- | | Whether pest infestation was observed | | Binary | |  | | --- | |  |  |  | | --- | | True, False | |
| |  | | --- | | Fertilizer\_Use |  |  | | --- | |  | | Whether fertilizer was applied | Binary | |  | | --- | |  |  |  | | --- | | True, False | |
| Temperature | Temperature during the growing period | Categorical | |  | | --- | | Cool, Mild, Hot |  |  | | --- | |  | |
| Crop\_Yield | |  |  | | --- | --- | |  | | |  | Final outcome of the harvest | | | Categorical | Low, Medium, High |

## Initial Data Observation

The purpose of this initial data exploration is to gain a foundational understanding of the dataset by observing patterns, distributions, and potential dependencies among variables. This early analysis guides the construction of the Bayesian Network by revealing how environmental and agricultural factors may influence crop yield, and helps form hypotheses for causal relationships in the model.

* **Rainfall Distribution**: Roughly even split; crop yield tended to be higher with Adequate rainfall.
* **Soil Quality**: Poor soil strongly correlates with Low yield.
* **Pest Infestation**: More frequent under Hot temperatures and poor soil.
* **Fertilizer Use**: Associated with higher yields, especially under Average or Poor Soil\_Quality.
* **Temperature**: Mild temperatures most often result in Medium or High yield.
* **Crop Yield**: Varies significantly; Low yield more common when Rainfall is Low and pests are present.



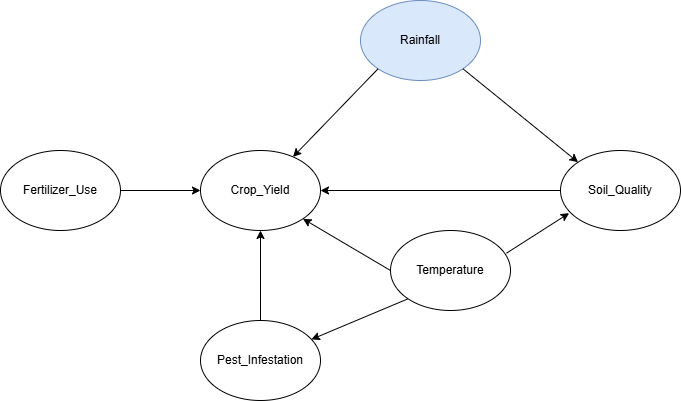
# Bayesian Network Structure

The Bayesian Network was constructed by identifying probable causal relationships based on domain knowledge in agriculture and initial data exploration. Variables were arranged such that influences flow directionally from environmental conditions (like Rainfall and Temperature) toward intermediary variables (such as Pest\_Infestation and Soil\_Quality) and finally toward the outcome (Crop\_Yield). Each link represents a conditional dependence, meaning the child node’s probability distribution is dependent on the state of its parent nodes.

## Relationships among the variables

* **Rainfall → Soil\_Quality**: Rain impacts nutrient retention and erosion.
* **Rainfall → Pest\_Infestation**: Certain pests thrive under wet/dry conditions.
* **Temperature → Pest\_Infestation**: Heat generally increases pest activity.
* **Temperature → Crop\_Yield**: Temperature affects plant metabolic rates.
* **Soil\_Quality → Fertilizer\_Use**: Poor soil increases dependency on fertilizers.
* **Soil\_Quality → Crop\_Yield**: Fertile soil boosts yields.
* **Fertilizer\_Use → Pest\_Infestation**: Overfertilization can attract pests.
* **Fertilizer\_Use → Crop\_Yield**: Fertilizer directly enhances growth.
* **Pest\_Infestation → Crop\_Yield**: Pests lower productivity.

The final DAG structure was constructed to reflect the most plausible relationships among the variables:



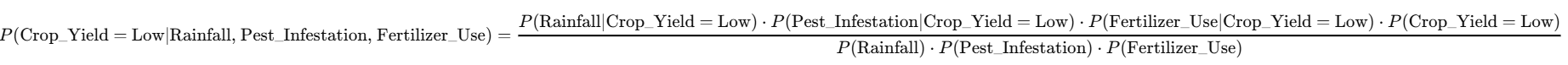
# Conditional Probability calculations

In a Bayesian Network, Conditional Probability Tables (CPTs) describe the probability of a variable given its parent variables. To estimate these probabilities, we use **Bayes' Theorem**, which allows us to reverse conditional relationships and compute the required values based on known or assumed probabilities.

For example, to calculate the probability that crop yield is low, given that rainfall is low, pest infestation is present, and fertilizer use is low, we apply Bayes’ Rule. This helps decompose a complex conditional probability into simpler, measurable components such as prior, likelihoods, and marginals.

To calculate CPTs, we:

1. Identify the child node (e.g., Crop\_Yield) and its parent nodes (e.g., Rainfall, Pest\_Infestation, Fertilizer\_Use).
2. Use Bayes' Theorem to compute the conditional probability of the child node given specific values of the parents.
3. Estimate probabilities based on historical data or expert knowledge.



Where:

* P(Crop\_Yield = Low)P(\text{Crop\\_Yield = Low})P(Crop\_Yield = Low)

is the **prior probability** of a low crop yield.

* P(Rainfall∣Crop\_Yield = Low)P(\text{Rainfall} | \text{Crop\\_Yield = Low})P(Rainfall∣Crop\_Yield = Low)

is the **likelihood** of rainfall given that the crop yield is low.

* P(Pest\_Infestation∣Crop\_Yield = Low)P(\text{Pest\\_Infestation} | \text{Crop\\_Yield = Low})P(Pest\_Infestation∣Crop\_Yield = Low)

is the **likelihood** of pest infestation given that the crop yield is low.

* P(Fertilizer\_Use∣Crop\_Yield = Low)P(\text{Fertilizer\\_Use} | \text{Crop\\_Yield = Low})P(Fertilizer\_Use∣Crop\_Yield = Low)

is the **likelihood** of fertilizer use given that the crop yield is low.

* P(Rainfall)P(\text{Rainfall})P(Rainfall), P(Pest\_Infestation)P(\text{Pest\\_Infestation})P(Pest\_Infestation),

and

P(Fertilizer\_Use)P(\text{Fertilizer\\_Use})P(Fertilizer\_Use)

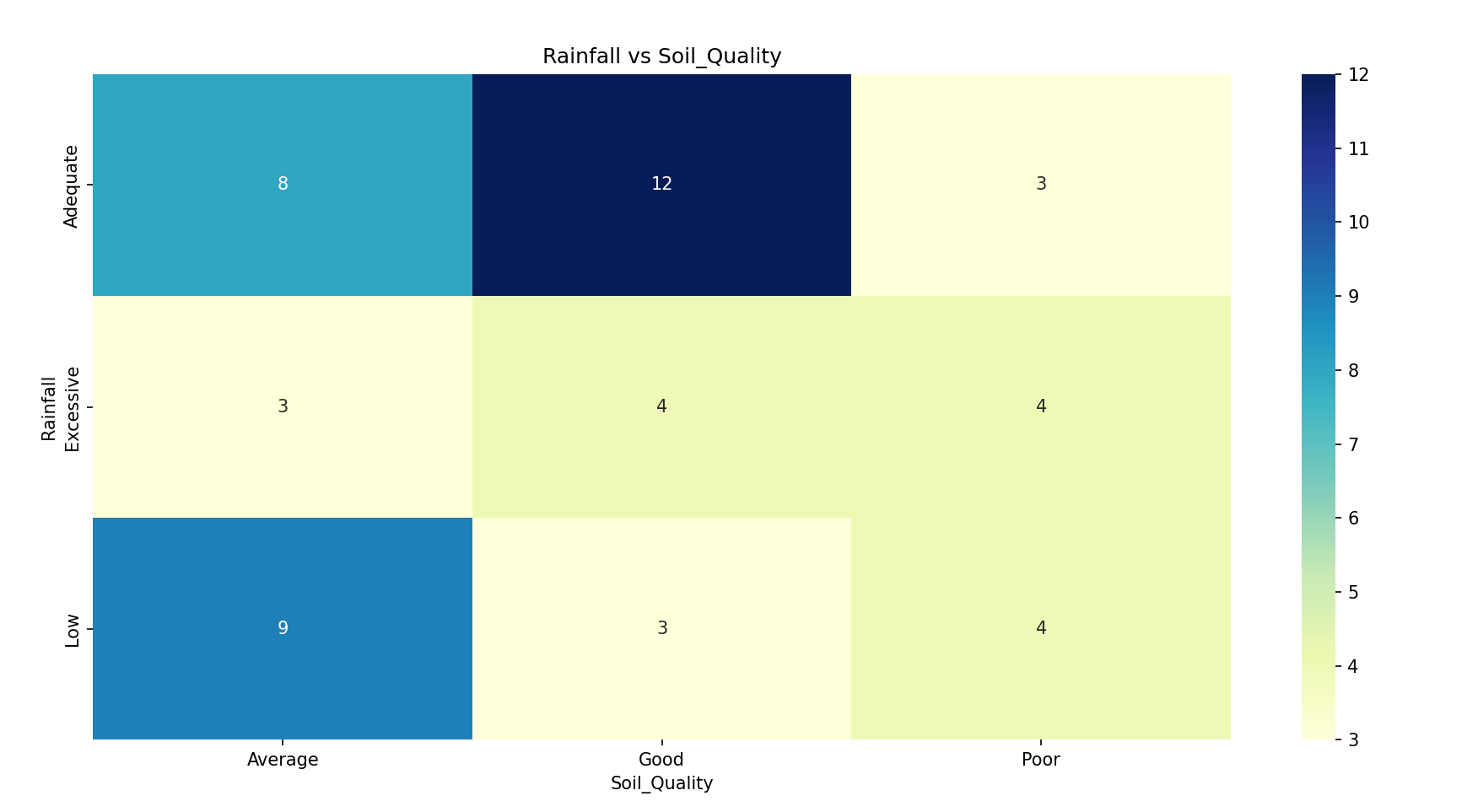
are the **marginal likelihoods** (the probabilities of these events in the overall population).

# Heatmap Evidence

The following heatmaps support the causal links proposed in the Bayesian Network. Each matrix visualizes the frequency relationship between Rainfall and another variable, helping to validate dependencies included in the DAG.

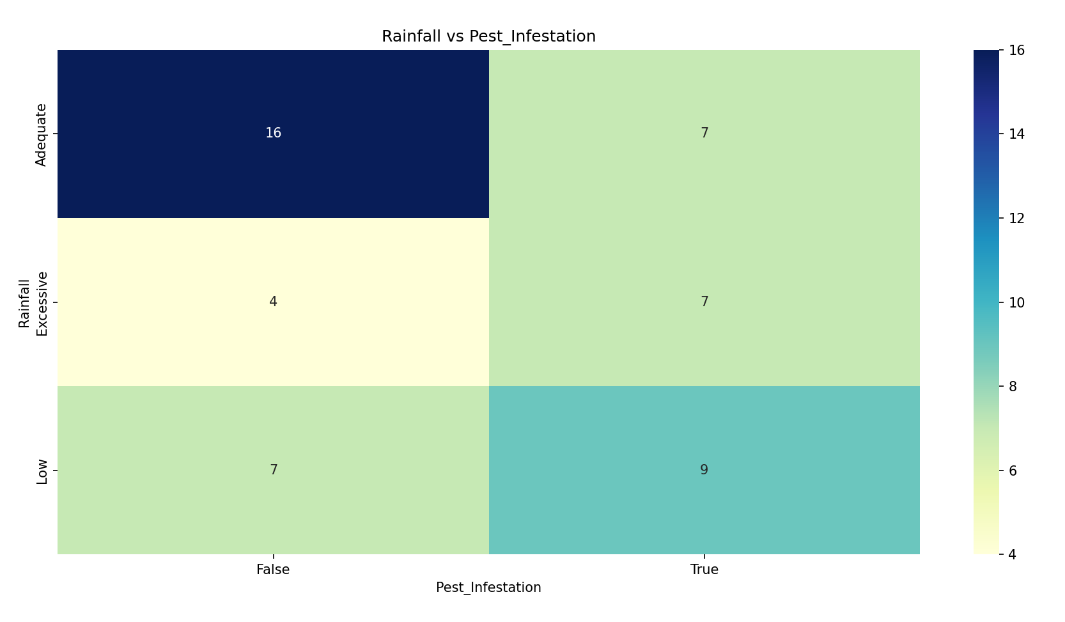
## Rainfall vs Soil\_Quality

This heatmap supports the assumption that rainfall patterns affect soil health, showing different distributions of soil quality across rainfall categories.



## Rainfall vs Pest\_Infestation

Higher pest infestations are seen under certain rainfall conditions, especially low and excessive rainfall.



## Rainfall vs Fertilizer\_Use

This heatmap highlights how fertilizer use patterns may vary depending on rainfall, potentially affecting nutrient application decisions.

A chart of different colors

AI-generated content may be incorrect.

## Rainfall vs Temperature

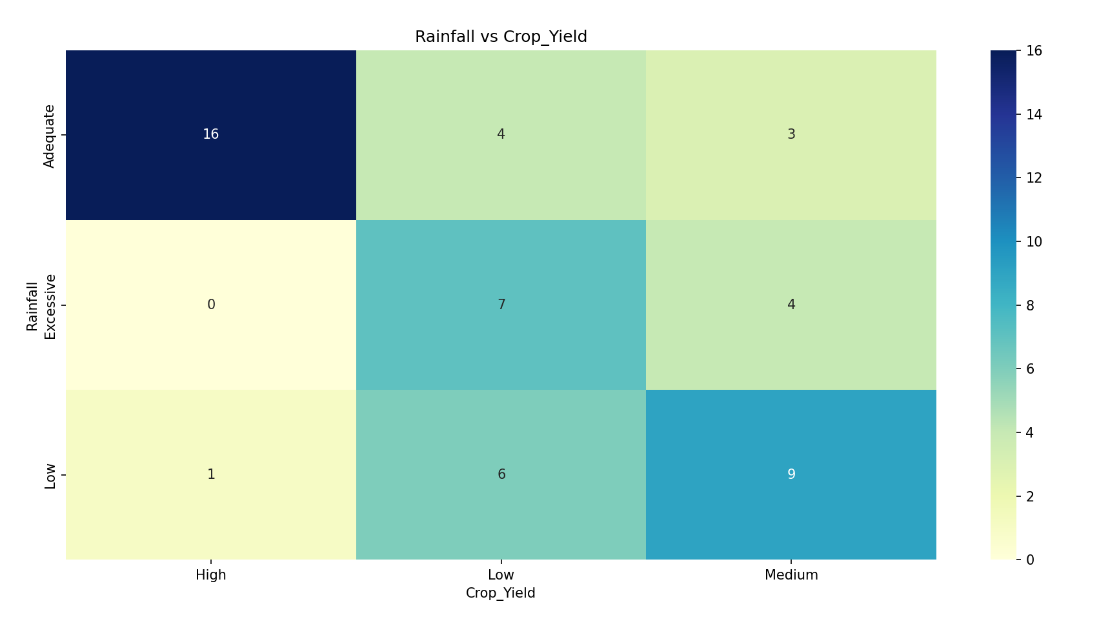
Temperature patterns appear linked to rainfall levels, with hot and mild conditions dominating low and adequate rainfall, respectively.

A screenshot of a color chart

AI-generated content may be incorrect.

## Rainfall vs Crop\_Yield

This heatmap validates the direct relationship between rainfall and crop yield, with high yield occurring most frequently under adequate rainfall.



# Discussion and conclusion

* Insights from the model
* Most influential factors on Crop\_Yield
* Limitations of the dataset/model
* Potential real-world applications

**9. Conclusion**

* Summary of key results
* Suggestions for future work (e.g., using more data, adding time-series elements)

# Appendix

Python code:

!pip install pgmpy --quiet

!pip install networkx matplotlib pandas --quiet

import pandas as pd

import matplotlib.pyplot as plt

import networkx as nx

from pgmpy.models import DiscreteBayesianNetwork

from pgmpy.estimators import MaximumLikelihoodEstimator

from pgmpy.inference import VariableElimination

data = pd.read\_csv('crop\_data.csv')

print("First 5 rows of the dataset:")

print(data.head())

print("\nColumn types and value counts:")

for col in data.columns:

print(f"\n{col} value counts:")

print(data[col].value\_counts())

model = DiscreteBayesianNetwork([

('Rainfall', 'Soil\_Quality'),

('Rainfall', 'Crop\_Yield'),

('Soil\_Quality', 'Crop\_Yield'),

('Temperature', 'Pest\_Infestation'),

('Temperature', 'Soil\_Quality'),

('Pest\_Infestation', 'Crop\_Yield'),

('Fertilizer\_Use', 'Crop\_Yield'),

('Temperature', 'Crop\_Yield')

])

model.fit(data,estimator=MaximumLikelihoodEstimator)

# STEP: Visualize the Network Properly Using NetworkX

import networkx as nx

import matplotlib.pyplot as plt

# Create a networkx graph from the DiscreteBayesianNetwork

G = nx.DiGraph()

G.add\_edges\_from(model.edges())

# Draw the graph

plt.figure(figsize=(10, 6))

pos = nx.spring\_layout(G)

nx.draw(G, pos, with\_labels=True, node\_color='skyblue', node\_size=2000, edge\_color='gray', font\_size=12)

plt.title("Bayesian Network Structure")

plt.show()

# STEP 7: Set Up Inference Engine

inference = VariableElimination(model)

# STEP 8: Inference Scenarios

# 1. P(Crop\_Yield = Low | All Good Conditions)

q1 = inference.query(variables=['Crop\_Yield'], evidence={

'Rainfall': 'Adequate',

'Soil\_Quality': 'Good',

'Temperature': 'Mild',

'Pest\_Infestation': False,

'Fertilizer\_Use': True

})

print("Q1: P(Crop\_Yield | All Good Conditions):\n", q1)

# 2. P(Crop\_Yield = Low | Rainfall = Low, Pest\_Infestation = True)

q2 = inference.query(variables=['Crop\_Yield'], evidence={

'Rainfall': 'Low',

'Pest\_Infestation': True

})

print("\nQ2: P(Crop\_Yield | Low Rainfall and Pest Present):\n", q2)

# 3. Most probable causes of Low Crop\_Yield

q3\_rain = inference.query(variables=['Rainfall'], evidence={'Crop\_Yield': 'Low'})

q3\_pest = inference.query(variables=['Pest\_Infestation'], evidence={'Crop\_Yield': 'Low'})

print("\nQ3: Most Probable Rainfall Given Low Crop\_Yield:\n", q3\_rain)

print("\nQ3: Most Probable Pest\_Infestation Given Low Crop\_Yield:\n", q3\_pest)

# 4. Compare High Yield with and without Fertilizer

q4\_yes = inference.query(variables=['Crop\_Yield'], evidence={

'Rainfall': 'Adequate',

'Soil\_Quality': 'Average',

'Temperature': 'Mild',

'Fertilizer\_Use': True

})

q4\_no = inference.query(variables=['Crop\_Yield'], evidence={

'Rainfall': 'Adequate',

'Soil\_Quality': 'Average',

'Temperature': 'Mild',

'Fertilizer\_Use': False

})

print("\nQ4: P(Crop\_Yield | Fertilizer Used):\n", q4\_yes)

print("\nQ4: P(Crop\_Yield | Fertilizer Not Used):\n", q4\_no)

# Show Conditional Probability Tables for all nodes

print("===== Conditional Probability Tables (CPTs) =====")

for cpd in model.get\_cpds():

print("\n", cpd)

# Define and store results for multiple scenarios

scenario\_results = []

scenarios = [

{

'name': 'All Optimal Conditions',

'evidence': {

'Rainfall': 'Adequate',

'Soil\_Quality': 'Good',

'Temperature': 'Mild',

'Pest\_Infestation': False,

'Fertilizer\_Use': True

}

},

{

'name': 'Drought & No Fertilizer',

'evidence': {

'Rainfall': 'Low',

'Soil\_Quality': 'Poor',

'Temperature': 'Hot',

'Pest\_Infestation': True,

'Fertilizer\_Use': False

}

},

{

'name': 'Pests Present, Fertilizer Used',

'evidence': {

'Rainfall': 'Adequate',

'Soil\_Quality': 'Average',

'Temperature': 'Hot',

'Pest\_Infestation': True,

'Fertilizer\_Use': True

}

},

{

'name': 'Fertilizer Only Intervention',

'evidence': {

'Rainfall': 'Low',

'Soil\_Quality': 'Average',

'Temperature': 'Cool',

'Pest\_Infestation': False,

'Fertilizer\_Use': True

}

}

]

for scenario in scenarios:

result = inference.query(variables=['Crop\_Yield'], evidence=scenario['evidence'])

probs = result.values

labels = result.state\_names['Crop\_Yield']

row = {'Scenario': scenario['name']}

for i, label in enumerate(labels):

row[label] = round(probs[i], 3)

scenario\_results.append(row)

# Create a summary DataFrame

summary\_df = pd.DataFrame(scenario\_results)

print("\n===== Scenario Prediction Summary =====")

print(summary\_df)

# Plot bar chart for each scenario

summary\_df.set\_index('Scenario')[['Low', 'Medium', 'High']].plot(

kind='bar',

figsize=(10, 6),

title="Crop\_Yield Predictions Across Scenarios",

ylabel="Probability",

xlabel="Scenario",

rot=45

)

plt.tight\_layout()

plt.show()