

✓ Exercise 1: Setting Up the Environment

```
# Install pgmpy library for Bayesian Network modeling
```

```
!pip install pgmpy
```

```

Requirement already satisfied: pgmpy in /usr/local/lib/python3.10/dist-packages (0.1.26)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from pgmpy) (3.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from pgmpy) (1.26.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from pgmpy) (1.13.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from pgmpy)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.2.2)
Requirement already satisfied: pyparsing in /usr/local/lib/python3.10/dist-packages (from pgmpy) (3.1.2)
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.4.1+cu121)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.10/dist-packages (from pgmpy) (0.14.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from pgmpy) (4.66.5)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from pgmpy) (1.4.2)
Requirement already satisfied: opt-einsum in /usr/local/lib/python3.10/dist-packages (from pgmpy) (3.3.0)
Requirement already satisfied: xgboost in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.1.3)
Requirement already satisfied: google-generativeai in /usr/local/lib/python3.10/dist-packages (from pgmpy) (0.6.6)
Requirement already satisfied: google-ai-generativelanguage==0.6.6 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (0.6.6)
Requirement already satisfied: google-api-core in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.19.1)
Requirement already satisfied: google-api-python-client in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.132.0)
Requirement already satisfied: google-auth>=2.15.0 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.27.0)
Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-packages (from pgmpy) (4.25.3)
Requirement already satisfied: pydantic in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.7.1)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from pgmpy) (4.11.0)
Requirement already satisfied: proto-plus<2.0.0dev,>=1.22.3 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (1.24.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.9.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2024.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (3.5.0)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (0.5.6)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (24.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from pgmpy) (3.13.1)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from pgmpy) (1.12.1)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (3.1.3)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2024.10.0)
Requirement already satisfied: nvidia-nccl-cu12 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.20.5)
Requirement already satisfied: googleapis-common-protos<2.0.dev0,>=1.56.2 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (1.63.0)
Requirement already satisfied: requests<3.0.0.dev0,>=2.18.0 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.32.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (5.5.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (0.4.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (4.9)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from pgmpy) (1.16.0)
Requirement already satisfied: httplib2<1.dev0,>=0.19.0 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (0.20.2)
Requirement already satisfied: google-auth-httplib2<1.0.0,>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (0.2.0)
Requirement already satisfied: uritemplate<5,>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (4.1.1)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.1.5)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (0.7.0)
Requirement already satisfied: pydantic-core==2.23.4 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.23.4)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (1.3.0)
Requirement already satisfied: grpcio<2.0dev,>=1.33.2 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (1.64.0)
Requirement already satisfied: grpcio-status<2.0.dev0,>=1.33.2 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (1.64.0)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (0.6.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from pgmpy) (2024.12.14)

```

✓ Exercise 2: Define the Bayesian Network Structure

```
# Import required libraries
import pandas as pd
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
from pgmpy.models import BayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination

# Step 1: Define the Bayesian Network structure for heart disease diagnosis
def define_network_structure():
    model = BayesianNetwork([
        ('Age', 'HeartDisease'), # Age influences HeartDisease
        ('Smoking', 'HeartDisease'), # Smoking influences HeartDisease
        ('Exercise', 'HeartDisease'), # Exercise influences HeartDisease
        ('Cholesterol', 'HeartDisease'), # Cholesterol influences HeartDisease
        ('BloodPressure', 'HeartDisease'), # BloodPressure influences HeartDisease
        ('HeartDisease', 'ChestPain'), # HeartDisease causes ChestPain
        ('HeartDisease', 'ECGResult') # HeartDisease affects ECGResult
    ])
    return model
```

✓ Exercise 3: Define Conditional Probability Tables (CPTs)

```
# Step 2: Define the Conditional Probability Tables (CPTs) for the Bayesian Network
def define_cpts(model):
    cpd_age = TabularCPD(variable='Age', variable_card=3, values=[[0.3], [0.5], [0.2]])
    cpd_smoking = TabularCPD(variable='Smoking', variable_card=2, values=[[0.6], [0.4]])
    cpd_exercise = TabularCPD(variable='Exercise', variable_card=2, values=[[0.7], [0.3]])
    cpd_cholesterol = TabularCPD(variable='Cholesterol', variable_card=2, values=[[0.8], [0.2]])
    cpd_bp = TabularCPD(variable='BloodPressure', variable_card=2, values=[[0.75], [0.25]])

# CPT for HeartDisease, considering its dependencies on multiple factors
cpd_hd = TabularCPD(
    variable='HeartDisease', variable_card=2,
    values=[
        [0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2,
         0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3,
         0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1],
        [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,
         0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,
         0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]],
    evidence=['Age', 'Smoking', 'Exercise', 'Cholesterol', 'BloodPressure'],
    evidence_card=[3, 2, 2, 2, 2]
)

# CPT for Chest Pain dependent on HeartDisease
cpd_cp = TabularCPD(
```

```

        variable='ChestPain', variable_card=2,
        values=[[0.7, 0.3], [0.3, 0.7]],
        evidence=['HeartDisease'], evidence_card=[2]

# CPT for ECGResult dependent on HeartDisease
)
cpd_ecg = TabularCPD(
    variable='ECGResult', variable_card=2,
    values=[[0.8, 0.2], [0.2, 0.8]],
    evidence=['HeartDisease'], evidence_card=[2]
)

model.add_cpds(cpd_age, cpd_smoking, cpd_exercise, cpd_cholesterol, cpd_bp, cpd_hd, cpd_cp, cpd_ecg)
return model

```

✓ Exercise 4: Inference in the Bayesian Network

```

# Step 3: Perform inference on the Bayesian Network
def perform_inference(model):
    infer = VariableElimination(model)

# Query for probability of HeartDisease given certain conditions
query_hd = infer.query(variables=['HeartDisease'], evidence={'Age': 1, 'Smoking': 1, 'Cholesterol':

# Query for probability of ECGResult given ChestPain
query_ecg = infer.query(variables=['ECGResult'], evidence={'ChestPain': 1})

# Query for probability of HeartDisease in non-exercising patients
query_no_exercise = infer.query(variables=['HeartDisease'], evidence={'Exercise': 1})

return query_hd, query_ecg, query_no_exercise

```

✓ Exercise 5: Parameter Learning from Simulated Data

```

# Step 4: Simulate healthcare dataset for parameter learning
def simulate_data():
    data = pd.DataFrame(np.random.randint(0, 2, size=(1000, 8)), columns=['Age', 'Smoking', 'Exercise',
    return data

# Step 5: Learn parameters (CPTs) from simulated data
def learn_parameters(model, data):
    from pgmpy.estimators import MaximumLikelihoodEstimator
    model.fit(data, estimator=MaximumLikelihoodEstimator)
    return model

```

✓ Exercise 6: Network Visualization and Analysis

```
# Step 6: Visualize the Bayesian Network structure
def visualize_network(model):
    G = nx.DiGraph()
    G.add_edges_from(model.edges())
    pos = nx.spring_layout(G)
    nx.draw(G, pos, with_labels=True, node_size=2000, font_size=10, node_color='lightblue', font_weight=
plt.show())

# Step 7: Perform sensitivity analysis (Smoking vs Non-Smoking) on HeartDisease
def sensitivity_analysis(model, infer):
    smoking_on_hd = infer.query(variables=['HeartDisease'], evidence={'Smoking': 1})
    non_smoking_on_hd = infer.query(variables=['HeartDisease'], evidence={'Smoking': 0})

    return smoking_on_hd, non_smoking_on_hd
```

✓ Results

```
# Main function to run all steps
def main():
    model = define_network_structure()
    model = define_cpds(model)

    query_hd, query_ecg, query_no_exercise = perform_inference(model)

    print("Probability of Heart Disease (Middle-aged, Smoker, High Cholesterol, High BP):")
    print(query_hd)

    print("\nProbability of Abnormal ECG given Chest Pain:")
    print(query_ecg)

    print("\nProbability of Heart Disease in Non-Exercising Patients:")
    print(query_no_exercise)

    data = simulate_data()
    model = learn_parameters(model, data)

    print("\nModel parameters learned from simulated data:")
    for cpd in model.get_cpds():
        print(cpd)

    visualize_network(model)

    infer = VariableElimination(model)
    smoking_on_hd, non_smoking_on_hd = sensitivity_analysis(model, infer)

    print("\nProbability of Heart Disease (Smoker):")
    print(smoking_on_hd)

    print("\nProbability of Heart Disease (Non-Smoker):")
    print(non_smoking_on_hd)

# Run the main function
main()
```



```
WARNING:pgmpy:Replacing existing CPD for Age
WARNING:pgmpy:Replacing existing CPD for HeartDisease
WARNING:pgmpy:Replacing existing CPD for Smoking
WARNING:pgmpy:Replacing existing CPD for Exercise
WARNING:pgmpy:Replacing existing CPD for Cholesterol
WARNING:pgmpy:Replacing existing CPD for BloodPressure
WARNING:pgmpy:Replacing existing CPD for ChestPain
WARNING:pgmpy:Replacing existing CPD for ECGResult
Probability of Heart Disease (Middle-aged, Smoker, High Cholesterol, High BP):
```

```
+-----+-----+
| HeartDisease | phi(HeartDisease) |
+=====+=====+
| HeartDisease(0) | 0.2500 |
+-----+-----+
| HeartDisease(1) | 0.7500 |
+-----+-----+
```

Probability of Abnormal ECG given Chest Pain:

```
+-----+-----+
| ECGResult | phi(ECGResult) |
+=====+=====+
| ECGResult(0) | 0.4281 |
+-----+-----+
| ECGResult(1) | 0.5719 |
+-----+-----+
```

Probability of Heart Disease in Non-Exercising Patients:

```
+-----+-----+
| HeartDisease | phi(HeartDisease) |
+=====+=====+
| HeartDisease(0) | 0.4284 |
+-----+-----+
| HeartDisease(1) | 0.5716 |
+-----+-----+
```

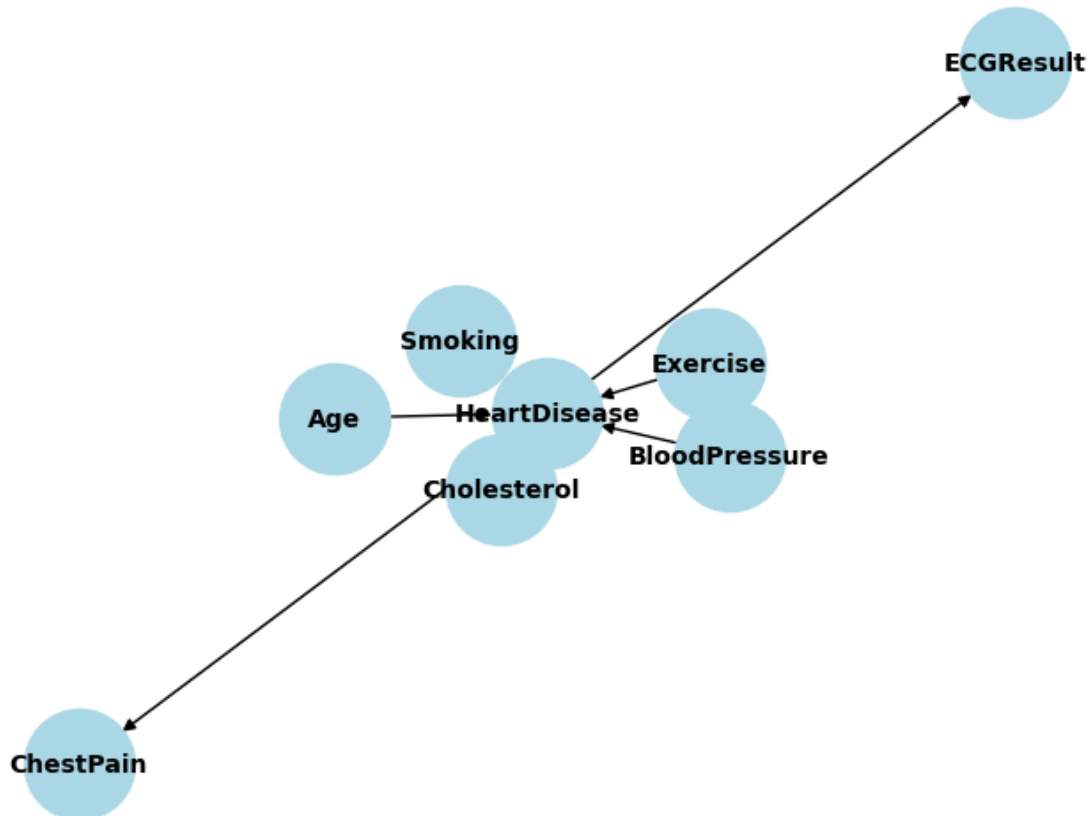
Model parameters learned from simulated data:

```
+-----+-----+
| Age(0) | 0.508 |
+-----+-----+
| Age(1) | 0.492 |
+-----+-----+
+-----+-----+
| Smoking(0) | 0.499 |
+-----+-----+
| Smoking(1) | 0.501 |
+-----+-----+
+-----+-----+
| Exercise(0) | 0.522 |
+-----+-----+
| Exercise(1) | 0.478 |
+-----+-----+
+-----+-----+
| Cholesterol(0) | 0.488 |
+-----+-----+
| Cholesterol(1) | 0.512 |
+-----+-----+
+-----+-----+
| BloodPressure(0) | 0.491 |
+-----+-----+
| BloodPressure(1) | 0.509 |
+-----+-----+
+-----+-----+-----+-----+-----+
| Age | Age(0) | ... | Age(1) | Age(1) |
```

```

+-----+-----+-----+-----+
| BloodPressure | BloodPressure(0) | ... | BloodPressure(1) | BloodPressure(1) |
+-----+-----+-----+-----+
| Cholesterol    | Cholesterol(0)   | ... | Cholesterol(1)   | Cholesterol(1)   |
+-----+-----+-----+-----+
| Exercise       | Exercise(0)      | ... | Exercise(1)      | Exercise(1)      |
+-----+-----+-----+-----+
| Smoking        | Smoking(0)       | ... | Smoking(0)       | Smoking(1)       |
+-----+-----+-----+-----+
| HeartDisease(0) | 0.5384615384615384 | ... | 0.4333333333333335 | 0.5135135135135135 |
+-----+-----+-----+-----+
| HeartDisease(1) | 0.46153846153846156 | ... | 0.5666666666666667 | 0.4864864864864865 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| HeartDisease | HeartDisease(0) | HeartDisease(1) |
+-----+-----+-----+-----+
| ChestPain(0) | 0.45233265720081134 | 0.49506903353057197 |
+-----+-----+-----+-----+
| ChestPain(1) | 0.5476673427991886 | 0.504930966469428 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| HeartDisease | HeartDisease(0) | HeartDisease(1) |
+-----+-----+-----+-----+
| ECGResult(0) | 0.5070993914807302 | 0.5207100591715976 |
+-----+-----+-----+-----+
| ECGResult(1) | 0.49290060851926976 | 0.47928994082840237 |
+-----+-----+-----+-----+

```



Probability of Heart Disease (Smoker):

```

+-----+-----+
| HeartDisease | phi(HeartDisease) |
+=====+=====+
| HeartDisease(0) | 0.5119 |
+-----+-----+
| HeartDisease(1) | 0.4881 |
+-----+-----+

```