

AAI EXPERIMENT NO 2 : Design And Implement Bayesian Network For Outcome Prediction In Python

```
pip install pgmpy
```

```
Requirement already satisfied: pgmpy in /usr/local/lib/python3.12/dist-packages (1.0.0)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from pgmpy) (3.6.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from pgmpy) (2.0.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.12/dist-packages (from pgmpy) (1.16.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.12/dist-packages (from pgmpy) (1.6.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (from pgmpy) (2.2.2)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (from pgmpy) (2.9.0+cpu)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.12/dist-packages (from pgmpy) (0.14.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from pgmpy) (4.67.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from pgmpy) (1.5.3)
Requirement already satisfied: opt-einsum in /usr/local/lib/python3.12/dist-packages (from pgmpy) (3.4.0)
Requirement already satisfied: pyro-ppl in /usr/local/lib/python3.12/dist-packages (from pgmpy) (1.9.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas->pgmpy) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas->pgmpy) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas->pgmpy) (2025.3)
Requirement already satisfied: pyro-api>=0.1.1 in /usr/local/lib/python3.12/dist-packages (from pyro-ppl->pgmpy) (0.1.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (3.20.2)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (1.14.0)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (3.1.6)
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (2025.3.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn->pgmpy) (3.6.0)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.12/dist-packages (from statsmodels->pgmpy) (1.0.2)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.12/dist-packages (from statsmodels->pgmpy) (25.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas->pgmpy) (1.17.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch->pgmpy) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from Jinja2->torch->pgmpy) (3.0.3)
```

```
from pgmpy.models import DiscreteBayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination
```

```
# Define the network structure
model = DiscreteBayesianNetwork([
    ('Pain', 'OrthopedicDisease'),
    ('Swelling', 'OrthopedicDisease'),
    ('OrthopedicDisease', 'LossOfMovement')
])
```

```
#Define the Structure
#Suppose you have the following relationships:
# Pain and Swelling can both influence the presence of Orthopedic Disease.
# Loss of Movement can be influenced by Orthopedic Disease.

#Add Probability Tables (CPDs)
# Pain
cpd_pain = TabularCPD(variable='Pain', variable_card=2, values=[[0.7], [0.3]]) # 0: no pain, 1: pain# Swelling
cpd_swelling = TabularCPD(variable='Swelling', variable_card=2, values=[[0.8], [0.2]]) # 0: no, 1: yes# Orthopedic Disease depends on Pain and Swelling
cpd_disease = TabularCPD(
    variable='OrthopedicDisease',
    variable_card=2,
    values=[
        [0.99, 0.7, 0.8, 0.1], # No disease
        [0.01, 0.3, 0.2, 0.9] # Disease
    ],
    evidence=['Pain', 'Swelling'],
    evidence_card=[2, 2]
)

# Loss of Movement depends on Disease
cpd_loss = TabularCPD(
    variable='LossOfMovement',
    variable_card=2,
    values=[
        [0.7, 0.1], # No loss
        [0.3, 0.9] # Loss
    ],
    evidence=['OrthopedicDisease'],
    evidence_card=[2]
)

# Add CPDs to model
model.add_cpds(cpd_pain, cpd_swelling, cpd_disease, cpd_loss)
model.check_model()
```

```
True
```

```
#Make Predictions (Inference)
#Suppose a patient has pain and swelling, what's the chance they have the disease?
```

```
infer = VariableElimination(model)
result = infer.query(variables=['OrthopedicDisease'], evidence={'Pain': 1, 'Swelling': 1})
print(result)
```

```
+-----+-----+
| OrthopedicDisease | phi(OrthopedicDisease) |
+-----+-----+
| OrthopedicDisease(0) | 0.1000 |
+-----+-----+
| OrthopedicDisease(1) | 0.9000 |
+-----+-----+
```

```
# -----
```

```

..
# Design and Implementation of Bayesian Network
# For Outcome Prediction (Pass / Fail)
# =====

# Install required package before running:
!pip install pgmpy

from pgmpy.models import DiscreteBayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination

# -----
# Step 1: Define Bayesian Network Structure
# -----

model = DiscreteBayesianNetwork([
    ('Attendance', 'InternalMarks'),
    ('StudyHours', 'InternalMarks'),
    ('InternalMarks', 'Result')
])

# -----
# Step 2: Define Conditional Probability Distributions (CPDs)
# -----

# Attendance Probability
cpd_attendance = TabularCPD(
    variable='Attendance',
    variable_card=2,
    values=[[0.7], [0.3]],
    state_names={'Attendance': ['High', 'Low']}
)

# Study Hours Probability
cpd_studyhours = TabularCPD(
    variable='StudyHours',
    variable_card=2,
    values=[[0.6], [0.4]],
    state_names={'StudyHours': ['Sufficient', 'Insufficient']}
)

# Internal Marks depends on Attendance and Study Hours
cpd_internalmarks = TabularCPD(
    variable='InternalMarks',
    variable_card=2,
    values=[
        [0.9, 0.6, 0.7, 0.2], # Good
        [0.1, 0.4, 0.3, 0.8] # Poor
    ],
    evidence=['Attendance', 'StudyHours'],
    evidence_card=[2, 2],
    state_names={
        'InternalMarks': ['Good', 'Poor'],
        'Attendance': ['High', 'Low'],
        'StudyHours': ['Sufficient', 'Insufficient']
    }
)

# Result depends on Internal Marks
cpd_result = TabularCPD(
    variable='Result',
    variable_card=2,
    values=[
        [0.95, 0.3], # Pass
        [0.05, 0.7] # Fail
    ],
    evidence=['InternalMarks'],
    evidence_card=[2],
    state_names={
        'Result': ['Pass', 'Fail'],
        'InternalMarks': ['Good', 'Poor']
    }
)

# -----
# Step 3: Add CPDs to the Model
# -----

model.add_cpds(
    cpd_attendance,
    cpd_studyhours,
    cpd_internalmarks,
    cpd_result
)

# -----
# Step 4: Validate the Model
# -----

print("Model is valid:", model.check_model())

# -----
# Step 5: Perform Inference (Outcome Prediction)
# -----

inference = VariableElimination(model)

# Case 1: Attendance = High, StudyHours = Sufficient
query1 = inference.query(
    variables=['Result'],
    evidence={'Attendance': 'High', 'StudyHours': 'Sufficient'}
)

```

```

print("\nPrediction when Attendance = High and StudyHours = Sufficient")
print(query1)

# Case 2: Attendance = Low, StudyHours = Insufficient
query2 = inference.query(
    variables=['Result'],
    evidence={'Attendance': 'Low', 'StudyHours': 'Insufficient'})

print("\nPrediction when Attendance = Low and StudyHours = Insufficient")
print(query2)

# -----
# End of Program
# -----

```

```

Requirement already satisfied: pgmpy in /usr/local/lib/python3.12/dist-packages (1.0.0)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from pgmpy) (3.6.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from pgmpy) (2.0.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.12/dist-packages (from pgmpy) (1.16.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.12/dist-packages (from pgmpy) (1.6.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (from pgmpy) (2.2.2)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (from pgmpy) (2.9.0+cpu)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.12/dist-packages (from pgmpy) (0.14.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from pgmpy) (4.67.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from pgmpy) (1.5.3)
Requirement already satisfied: opt-einsum in /usr/local/lib/python3.12/dist-packages (from pgmpy) (3.4.0)
Requirement already satisfied: pyro-ppl in /usr/local/lib/python3.12/dist-packages (from pgmpy) (1.9.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas->pgmpy) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas->pgmpy) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas->pgmpy) (2025.3)
Requirement already satisfied: pyro-api>=0.1.1 in /usr/local/lib/python3.12/dist-packages (from pyro-ppl->pgmpy) (0.1.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (3.20.2)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (1.14.0)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (3.1.6)
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (2025.3.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn->pgmpy) (3.6.0)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.12/dist-packages (from statsmodels->pgmpy) (1.0.2)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.12/dist-packages (from statsmodels->pgmpy) (25.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas->pgmpy) (1.17.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch->pgmpy) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch->pgmpy) (3.0.3)
Model is valid: True

```

Prediction when Attendance = High and StudyHours = Sufficient

Result	phi(Result)
Result(Pass)	0.8850
Result(Fail)	0.1150

Prediction when Attendance = Low and StudyHours = Insufficient

Result	phi(Result)
Result(Pass)	0.4300
Result(Fail)	0.5700