Name: **Harsh Pramod Padyal**

Roll No : **40**

Div: **D20B**

Experiment 01

AIM: To Implement Inferencing with Bayesian Network in python

---

Theory:

---

1. Introduction to Bayesian Networks

A **Bayesian Network**, also called a **Belief Network**, is a graphical model that represents uncertain knowledge and helps in reasoning under uncertainty. It visualizes how different factors (variables) are interrelated and helps predict outcomes based on available data.

For example, deciding whether to take a **bus** or **walk** to work may depend on:

- **Traffic conditions**
- **Time of day**
- **Urgency of your schedule**

Bayesian Networks help calculate the probabilities of such decisions systematically.

> **Simple Analogy**: A Bayesian Network is like a **map** showing how events affect each other, where the **connections** are drawn using **probabilities**.

Bayesian Networks are widely used in:

- Decision-making systems
- Medical diagnosis
- System troubleshooting
- Artificial Intelligence applications

---

2. Key Components of Bayesian Networks

Bayesian Networks are made up of two main parts:

2.1 Directed Acyclic Graph (DAG)

- **Nodes** represent variables (e.g., Time, Traffic, Schedule, Transport).
- **Edges (arrows)** represent dependency relationships.
- The graph is **acyclic** — it doesn't have loops or cycles.

> For example:
>
> - **Time → Traffic**: Time of day affects traffic conditions.
> - **Traffic → Transport**: Traffic affects the mode of transport.

> - **Schedule → Transport**: Urgency of schedule affects transport choice.

2.2 Conditional Probability Distributions (CPDs)

Each node (variable) has a **Conditional Probability Table (CPT)** that defines the likelihood of different outcomes **given its parent nodes**.

Example:

- Probability of **Heavy Traffic** is higher in the **Morning** than in the **Evening**:

$$P(\text{Traffic} = \text{Heavy} \mid \text{Time} = \text{Morning}) = 0.7$$

---

3. Example Scenario: Transport Decision

Let's consider a scenario where your transport decision depends on:

- **Time** (Morning/Evening)
- **Traffic** (Heavy/Light)
- **Schedule** (Urgent/Relaxed)

The relationships:

- **Time → Traffic**
- **Traffic → Transport**
- **Schedule → Transport**

---

4. Mathematical Concepts

4.1 Joint Probability

The joint probability of all variables:

$$P(\text{Time}, \text{Traffic}, \text{Schedule}, \text{Transport})$$

Calculating this directly is complex due to many combinations. Bayesian Networks simplify it using **conditional probabilities**.

4.2 Conditional Probability

The probability of one event **given** another:

$$P(\text{Traffic} = \text{Heavy} \mid \text{Time} = \text{Morning}) = 0.7$$

These are stored in **Conditional Probability Distributions (CPDs)**.

4.3 Chain Rule for Bayesian Networks

The joint probability is computed using the product of conditional probabilities:

$$P(\text{Time}, \text{Traffic}, \text{Schedule}, \text{Transport}) = P(\text{Time}) \cdot P(\text{Traffic} \mid \text{Time}) \cdot P(\text{Schedule}) \cdot P(\text{Transport} \mid \text{Traffic},$$

Each term is easy to calculate using CPDs.

---

5. CPD Table Example

Table: Conditional Probability for Transport node

Traffic	Schedule	P(Transport = Bus)	P(Transport = Walk)
Heavy	Urgent	0.8	0.2

Traffic	Schedule	P(Transport = Bus)	P(Transport = Walk)
Heavy	Relaxed	0.2	0.8
Light	Urgent	0.5	0.5
Light	Relaxed	0.1	0.9

---

6. Inference in Bayesian Networks

Inference is the process of calculating the probability of a certain variable given known evidence.

> Example: Given:

- Time = Morning
- Schedule = Urgent

We compute:

$$P(\text{Transport} = \text{Bus} \mid \text{Time} = \text{Morning}, \text{Schedule} = \text{Urgent})$$

This is done using inference algorithms like **Variable Elimination**, available in libraries such as **pgmpy**.

---

7. How Bayesian Networks Work

Bayesian Networks integrate **DAGs** and **CPDs** to:

- **Model dependencies** between variables
- **Compute probabilities** efficiently
- **Answer queries** based on new evidence

> **Example**: If it's morning and schedule is urgent, the network might predict a **high probability of choosing the bus** due to expected heavy traffic.

---

8. Applications of Bayesian Networks

Bayesian Networks are used in:

- **Decision Support** (e.g., transport planning)
- **Medical Diagnosis** (e.g., predict disease from symptoms)
- **Fault Diagnosis** (e.g., detect hardware issues)
- **Risk Assessment** (e.g., finance, marketing)
- **Bioinformatics** (e.g., gene prediction)

---

9. Advantages of Bayesian Networks

- **Intuitive**: Easy-to-understand visual representation
- **Efficient**: Simplifies complex joint probability computations
- **Flexible**: Supports inference and learning from new evidence
- **Robust**: Handles uncertainty effectively

Code:

Install Required Packages

```
!pip install pgmpy networkx matplotlib
```

Show hidden output

Import Libraries and Define the Network

```
from pgmpy.models import DiscreteBayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination
import networkx as nx
import matplotlib.pyplot as plt

# Define the Bayesian Network structure
model = DiscreteBayesianNetwork([
    ('Time', 'Traffic'),
    ('Traffic', 'Transport'),
    ('Schedule', 'Transport')
])
```

Define and Add Conditional Probability Distributions (CPDs)

```
# CPD for Time of Day (prior probability)
cpd_time = TabularCPD(
    variable='Time',
    variable_card=2,
    values=[[0.5], [0.5]],  # P(Time = Morning) = 0.5, P(Time = Evening) = 0.5
    state_names={'Time': ['Morning', 'Evening']}
)

# CPD for Traffic given Time of Day
cpd_traffic = TabularCPD(
    variable='Traffic',
    variable_card=2,
    values=[[0.3, 0.7],  # P(Traffic = Heavy | Time = Morning, Evening)
            [0.7, 0.3]], # P(Traffic = Light | Time = Morning, Evening)
    evidence=['Time'],
    evidence_card=[2],
    state_names={'Traffic': ['Heavy', 'Light'], 'Time': ['Morning', 'Evening']}
)

# CPD for Schedule (prior probability)
cpd_schedule = TabularCPD(
    variable='Schedule',
    variable_card=2,
    values=[[0.4], [0.6]],  # P(Schedule = Urgent) = 0.4, P(Schedule = Relaxed) = 0.6
    state_names={'Schedule': ['Urgent', 'Relaxed']}
)

# CPD for Transport given Traffic and Schedule
cpd_transport = TabularCPD(
    variable='Transport',
    variable_card=2,
    values=[[0.2, 0.8, 0.5, 0.9],  # P(Transport = Bus | Traffic, Schedule)
            [0.8, 0.2, 0.5, 0.1]], # P(Transport = Walk | Traffic, Schedule)
    evidence=['Traffic', 'Schedule'],
    evidence_card=[2, 2],
    state_names={'Transport': ['Bus', 'Walk'], 'Traffic': ['Heavy', 'Light'], 'Schedule': ['Urgent', 'Relaxed']}
)

# Add CPDs to the model
model.add_cpds(cpd_time, cpd_traffic, cpd_schedule, cpd_transport)
```

```
# Verify the model
print("Model is valid:", model.check_model())
```

```
Model is valid: True
```

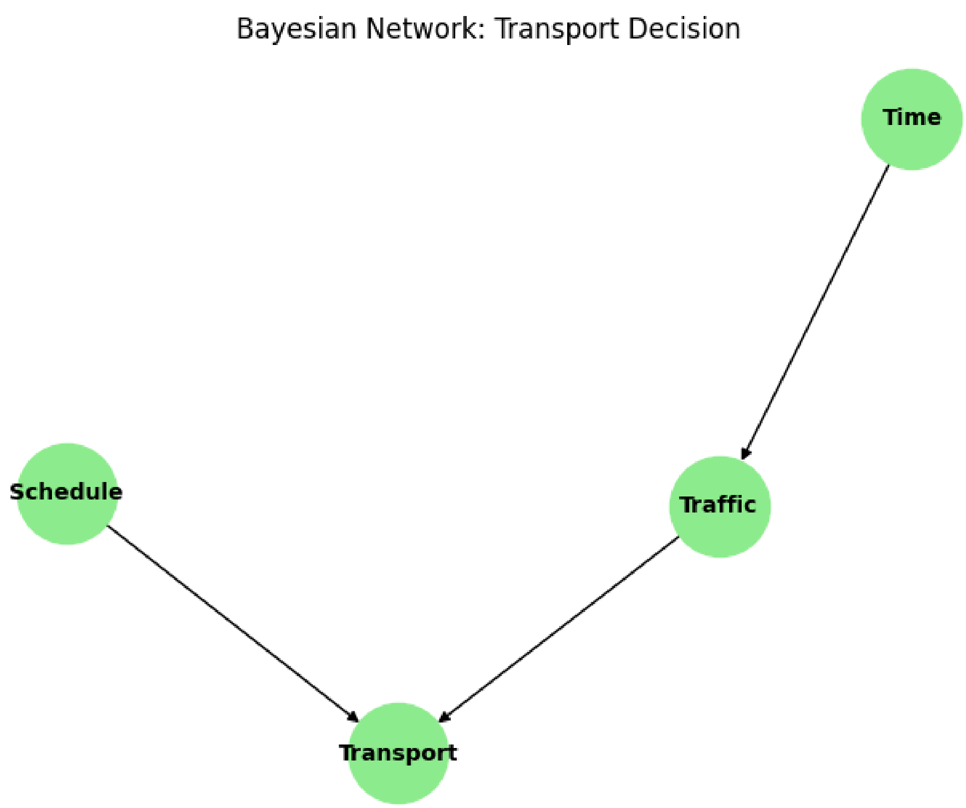
Perform Inference with Different Inputs

```
# Initialize inference
inference = VariableElimination(model)

# Query 1: Probability of taking the bus without any evidence
print("Query 1: P(Transport)")
result1 = inference.query(variables=['Transport'])
print(result1)

# Query 2: Probability of taking the bus given it's morning
print("\nQuery 2: P(Transport | Time = Morning)")
result2 = inference.query(variables=['Transport'], evidence={'Time': 'Morning'})
print(result2)

# Query 3: Probability of taking the bus given an urgent schedule
print("\nQuery 3: P(Transport | Schedule = Urgent)")
result3 = inference.query(variables=['Transport'], evidence={'Schedule': 'Urgent'})
print(result3)

# Query 4: Probability of taking the bus given morning and urgent schedule
print("\nQuery 4: P(Transport | Time = Morning, Schedule = Urgent)")
result4 = inference.query(variables=['Transport'], evidence={'Time': 'Morning', 'Schedule': 'Urgent'})
print(result4)
```

```
Query 1: P(Transport)
+-----------------+------------------+
| Transport       |   phi(Transport) |
+=================+==================+
| Transport(Bus)  |           0.6500 |
+-----------------+------------------+
| Transport(Walk) |           0.3500 |
+-----------------+------------------+

Query 2: P(Transport | Time = Morning)
+-----------------+------------------+
| Transport       |   phi(Transport) |
+=================+==================+
| Transport(Bus)  |           0.6860 |
+-----------------+------------------+
| Transport(Walk) |           0.3140 |
+-----------------+------------------+

Query 3: P(Transport | Schedule = Urgent)
+-----------------+------------------+
| Transport       |   phi(Transport) |
+=================+==================+
| Transport(Bus)  |           0.3500 |
+-----------------+------------------+
| Transport(Walk) |           0.6500 |
+-----------------+------------------+

Query 4: P(Transport | Time = Morning, Schedule = Urgent)
+-----------------+------------------+
| Transport       |   phi(Transport) |
+=================+==================+
| Transport(Bus)  |           0.4100 |
+-----------------+------------------+
| Transport(Walk) |           0.5900 |
+-----------------+------------------+
```

Visualize the Network (Fixed)

```
# Convert pgmpy model to networkx DiGraph for visualization
nx_graph = nx.DiGraph()
nx_graph.add_nodes_from(model.nodes())
nx_graph.add_edges_from(model.edges())

# Visualize the Bayesian Network
pos = nx.spring_layout(nx_graph)
nx.draw(nx_graph, pos, with_labels=True, node_color='lightgreen', node_size=2000, font_size=10, font_weight='bold')
plt.title("Bayesian Network: Transport Decision")
plt.show()
```



---

10. Conclusion

Bayesian Networks offer a powerful way to represent and reason under uncertainty. Using **Directed Acyclic Graphs (DAGs)** and **Conditional Probability Distributions (CPDs)**, they simplify complex relationships into manageable structures.

In our transport decision example, the network efficiently predicts whether you'll take a **bus** or **walk**, based on the **time of day**, **traffic conditions**, and **schedule urgency**.

This makes Bayesian Networks valuable tools for decision-making, diagnostics, and intelligent systems.
