Experiment:-6

Aim:- Write a program to Implement Bayesian Network for the given data

Bayesian Network:- A Bayesian Network is a probabilistic graphical model that represents a set of variables and their conditional dependencies using a directed acyclic graph (DAG). It is widely used in decision-making, risk analysis, and predictive modeling.

Key Concepts of Bayesian Network

1. Graph Structure

- Nodes represent random variables.
- Edges represent probabilistic dependencies between variables.
- The network is acyclic (no loops).

2. Conditional Probability Table (CPT)

- Each node is associated with a CPT, which quantifies the effect of its parent nodes.

3. Joint Probability Distribution

- The joint probability of the variables can be calculated as the product of the conditional probabilities.

4. Inference in Bayesian Networks

- Prior Probability: The probability of an event before any evidence is considered.
- Posterior Probability: The probability of an event given some evidence.

How Bayesian Network Works

1. Model Construction

- Define the variables and their dependencies.
- Create a directed acyclic graph.

2. Probability Assignment

- Assign probabilities to each node's CPT.

3. Inference

- Use Bayes' theorem to calculate the probability of unknown variables given evidence.

Advantages of Bayesian Network

1. Graphical Representation: Easy to visualize relationships between variables.
2. Handles Uncertainty: Provides a systematic approach to deal with probabilistic data.
3. Modular: Updating probabilities is simple when new data is available.

Disadvantages of Bayesian Network

1. Scalability: Becomes complex with a large number of variables.
2. Dependency on Data Quality: Inaccurate data can lead to incorrect probabilities.

Applications of Bayesian Network

1. Medical Diagnosis: Determine the probability of diseases based on symptoms.
2. Risk Analysis: Assess potential risks in various scenarios.
3. Machine Learning: Feature selection and classification tasks.

Program Implementation

```
from pgmpy.models import BayesianNetwork
from pgmpy.inference import VariableElimination
from pgmpy.factors.discrete import TabularCPD
# Define the structure of the Bayesian Network
model = BayesianNetwork([
   ('Rain', 'Traffic'),
   ('Accident', 'Traffic'),
```

```
    ('Rain', 'Umbrella')
])

# Define the Conditional Probability Distributions (CPDs)
cpd_rain = TabularCPD(variable='Rain', variable_card=2, values=[[0.7], [0.3]])
cpd_accident = TabularCPD(variable='Accident', variable_card=2, values=[[0.8], [0.2]])
cpd_traffic = TabularCPD(
    variable='Traffic', variable_card=2,
    values=[[0.9, 0.4, 0.5, 0.1],
            [0.1, 0.6, 0.5, 0.9]],
    evidence=['Rain', 'Accident'],
    evidence_card=[2, 2]
)
cpd_umbrella = TabularCPD(
    variable='Umbrella', variable_card=2,
    values=[[0.8, 0.2],
            [0.2, 0.8]],
    evidence=['Rain'],
    evidence_card=[2]
)

# Add CPDs to the model
model.add_cpds(cpd_rain, cpd_accident, cpd_traffic, cpd_umbrella)
# Verify the model
if model.check_model():
    print("The Bayesian Network model is valid.")
```

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

# Query 1: Probability of Traffic given Rain and Accident
query_result = inference.query(variables=['Traffic'], evidence={'Rain': 1, 'Accident': 1})
print("\nProbability of Traffic given Rain and Accident:")
print(query_result)

# Query 2: Probability of Umbrella given Rain
query_result = inference.query(variables=['Umbrella'], evidence={'Rain': 1})
print("\nProbability of Umbrella given Rain:")
print(query_result)
```

Output :

```
PS C:\Users\Achin\OneDrive\Desktop\college work> python bayesian_network.py
The Bayesian Network model is valid.

Probability of Traffic given Rain and Accident:
+------------+----------------+
| Traffic    |   phi(Traffic) |
+============+================+
| Traffic(0) |         0.1000 |
+------------+----------------+
| Traffic(1) |         0.9000 |
+------------+----------------+

Probability of Umbrella given Rain:
+-------------+-----------------+
| Umbrella    |   phi(Umbrella) |
+=============+=================+
| Umbrella(0) |          0.2000 |
+-------------+-----------------+
| Umbrella(1) |          0.8000 |
+-------------+-----------------+
```