https://lh5.googleusercontent.com/dplghwJq6X4fhzS5H6mFhAFj9x6vI-Y8xCT8NFOTS1m1Xqxiq7nkadVUnCPhdF0ePu4loIUkqVjtvmt0NXfO2k9ohAj4vSqxuecZS-EBDoWiRGD-hgPkQa4QEs6nQaUoqsWtkTeeVLr0namIZbmEyQhttps://lh3.googleusercontent.com/BTBdPiSJjxGslQH3BeZD4BaoJZ39HCgQmAhUsT_pMmuCBkQpXF4Oufxkc29xElrbY7UOC_t-XYD8wCe8-xr0WMFCu3DhySoqaYXxkDd4zDvRd6uFglNfbvwNH7fYiWW7sNqHblYmu1wrAZV9wwFdXA

**GHARDA FOUNDATION**

**GHARDA INSTITUTE OF TECHNOLOGY, LAVEL**

Department of Computer Engineering

**Evaluation Sheet**

Class: TE-Computer Engineering Sem: VI Subject: **Artificial Intelligence Lab(CSL604)**

Experiment No: 10

Title of Experiment: Study the implementation Bayesian belief network in AI using python.

Name of Student: Niraj Nitin Surve Roll No: 68

Date of Performance: 23/03/2023

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Evaluation Criteria | Max Marks | Marks Obtained |
| 1 | Practical Performance | 8 |  |
| 2 | Oral | 5 |  |
| 3 | Timely Submission | 2 |  |
|  | Total | 15 |  |

                   Signature of Subject Teacher

     (Prof. M. A. Khandke)

**Program Code –**

# Import the necessary libraries

from pgmpy.models import BayesianNetwork

from pgmpy.factors.discrete import TabularCPD

import networkx as nx

import matplotlib.pyplot as plt

# Define the structure of the Bayesian network

model = BayesianNetwork([('A', 'B'), ('C', 'B'), ('B', 'D')])

# Define the conditional probability tables (CPDs) for each node

cpd\_a = TabularCPD(variable='A', variable\_card=2, values=[[0.6], [0.4]])

cpd\_c = TabularCPD(variable='C', variable\_card=2, values=[[0.7], [0.3]])

cpd\_b = TabularCPD(variable='B', variable\_card=2,

values=[[0.9, 0.8, 0.5, 0.3], [0.1, 0.2, 0.5, 0.7]],

evidence=['A', 'C'], evidence\_card=[2, 2])

cpd\_d = TabularCPD(variable='D', variable\_card=2,

values=[[0.6, 0.1], [0.4, 0.9]],

evidence=['B'], evidence\_card=[2])

model.add\_cpds(cpd\_a, cpd\_c, cpd\_b, cpd\_d)

model.check\_model()

# Define the node positions

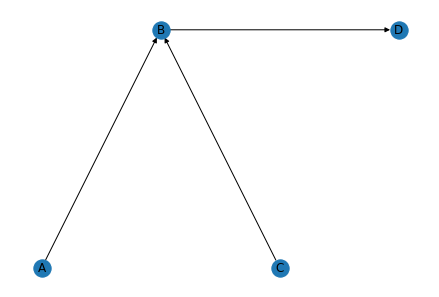
pos = {'A': (0, 0), 'C': (1, 0), 'B': (0.5, 0.5), 'D': (1.5, 0.5)}

# Plot the graph of the model

nx.draw(model, pos=pos, with\_labels=True)

plt.show()

**Output –**

****