Artificial Intelligence Lab
KCA 351: Session 2021-22

Experiment – No-8

1.Problem Statement : **Write a program to demonstrate the working of Bayesian network using heart disease dataset "https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data". Calculate the probability of model = BayesianModel([("A","B"),("B","C"),("C","RESULT")]) while considering the following 14 attributes**

A. #3 (age)
B. #4 (sex)
C. #9 (cp)
D. #10 (trestbps)
E. #12 (chol)
F. #16 (fbs)
G. #19 (restecg)
H. #32 (thalach)
I. #38 (exang)
J. #40 (oldpeak)
K. #41 (slope)
L. #44 (ca)
M. #51 (thal)
N. #58 (num) (the predicted attribute)

Program:

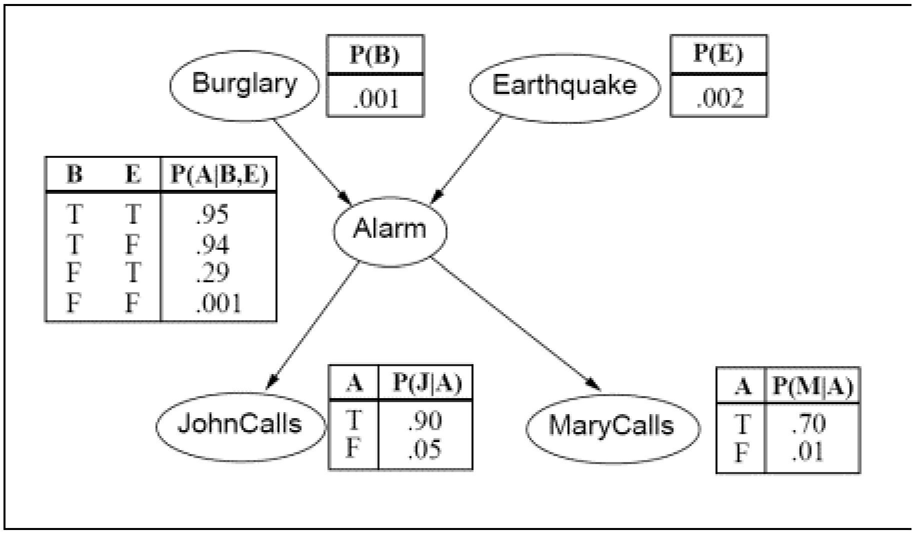
```
import numpy as np
import pandas as pd
import csv
from pgmpy.estimators import MaximumLikelihoodEstimator
from pgmpy.models import BayesianModel
from pgmpy.inference import VariableElimination

heartDisease = pd.read_csv('heart.csv')
heartDisease = heartDisease.replace('?',np.nan)

print('Sample instances from the dataset are given below')
print(heartDisease.head())

print('\n Attributes and datatypes')
print(heartDisease.dtypes)

model= BayesianModel([('age','heartdisease'),('sex','heartdisease'),('exan
g','heartdisease'),('cp','heartdisease'),('heartdisease','restecg'),('hear
tdisease','chol')])
print('\nLearning CPD using Maximum likelihood estimators')
model.fit(heartDisease,estimator=MaximumLikelihoodEstimator)

print('\n Inferencing with Bayesian Network:')
HeartDiseasetest_infer = VariableElimination(model)

print('\n 1. Probability of HeartDisease given evidence= restecg')
q1=HeartDiseasetest_infer.query(variables=['heartdisease'],evidence={'rest
ecg':1})
print(q1)

print('\n 2. Probability of HeartDisease given evidence= cp ')
q2=HeartDiseasetest_infer.query(variables=['heartdisease'],evidence={'cp':
2})
print(q2)
```

Artificial Intelligence Lab
KCA 351: Session 2021-22

Output :

```
Learning CPD using Maximum likelihood estimators

 Inferencing with Bayesian Network:

 1. Probability of HeartDisease given evidence= restecg

    +-----------------+---------------------+
    | heartdisease    |   phi(heartdisease) |
    +=================+=====================+
    | heartdisease(0) |              0.1012 |
    +-----------------+---------------------+
    | heartdisease(1) |              0.0000 |
    +-----------------+---------------------+
    | heartdisease(2) |              0.2392 |
    +-----------------+---------------------+
    | heartdisease(3) |              0.2015 |
    +-----------------+---------------------+
    | heartdisease(4) |              0.4581 |
    +-----------------+---------------------+

2. Probability of HeartDisease given evidence= cp

    +-----------------+---------------------+
    | heartdisease    |   phi(heartdisease) |
    +=================+=====================+
    | heartdisease(0) |              0.3610 |
    +-----------------+---------------------+
    | heartdisease(1) |              0.2159 |
    +-----------------+---------------------+
    | heartdisease(2) |              0.1373 |
    +-----------------+---------------------+
    | heartdisease(3) |              0.1537 |
    +-----------------+---------------------+
    | heartdisease(4) |              0.1321 |
    +-----------------+---------------------+
```

2.Problem Statement : Write a program to demonstrate the working of Bayesian network for the following graph:



- Calculate the probability of a burglary if John and Mary calls (0: True, 1: False)
- Calculate the probability of alarm starting if there is a burglary and an earthquake (0: True, 1: False)

Calculate the probability of alarm starting if there is a burglary and an earthquake (0: True, 1: False)

Program:

```
pip install pgmpy
import pgmpy. inference
import pgmpy. models
import networkx as nx
import pylab as plt
model = pgmpy. models . BayesianNetwork ([( ’ Burglary ’ , ’Alarm ’) , ( ’
 Earthquake ’ , ’Alarm ’) ,
                                          ( ’Alarm ’ , ’ JohnCalls ’) , (
’Alarm ’ , ’MaryCalls ’ ) ] )
# Define conditional probability distributions (CPD)
# Probability of burglary (True , False )
cpd burglary = pgmpy. factors . discrete .TabularCPD(” Burglary ” , 2 , [[
0.001] , [0.999]])
# Probability of earthquake (True , False )
cpd earthquake = pgmpy. factors . discrete .TabularCPD(”Earthquake” , 2 ,
[[0.002] , [0.998]])
# Probability of alarm going of (True , False ) given a burglary and/or ea
rthquake
cpd alarm = pgmpy. factors . discrete .TabularCPD( ’Alarm ’ , 2 , [[0.95 ,
 0.94 , 0.29 , 0.001] , [0.05 , 0.06 , 0.71 , 0.999]] ,
                                                  evidence =[’Burglary ’ ,
 ’Earthquake ’] , evidence card =[2, 2])
# Probability that John calls (True , False ) given that the alarm has sou
nded
cpd john = pgmpy. factors . discrete .TabularCPD( ’ JohnCalls ’ , 2 , [[0.
90 , 0.05] , [0.10 , 0.95]] , evidence =[’Alarm ’] , evidence card =[2])
# Probability that Mary calls (True , False ) given that the alarm has sou
nded
cpd mary = pgmpy. factors . discrete .TabularCPD( ’ MaryCalls ’ , 2 , [[0.
70 , 0.01] , [0.30 , 0.99]] , evidence =[’Alarm ’] , evidence card =[2])
# Add CPDs to the network structure
model . add cpds ( cpd burglary , cpd earthquake , cpd alarm , cpd john ,
cpd mary)
# Check i f the model is valid , throw an exception otherwise
model . check model ()
# Print probability distributions
print ( ’ Probability distribution , P( Burglary ) ’)
print ( cpd burglary )
print ()
```

```
print ( ’ Probability distribution , P(Earthquake ) ’)
print ( cpd earthquake )
print ()
print ( ’ Joint probability distribution , P(Alarm | Burglary , Earthquake
 ) ’)
print ( cpd alarm )
print ()
print ( ’ Joint probability distribution , P( JohnCalls | Alarm) ’)
print ( cpd john )
print ()
print ( ’ Joint probability distribution , P(MaryCalls | Alarm) ’)
print (cpd mary) print () # Plot the model
nx. draw(model , with labels=True) plt . savefig ( ’ alarm1 .png ’) plt .
close ()
# Perform variable elimination for inference
# Variable elimination (VE) is a an exact inference algorithm in bayesian
networks
infer = pgmpy. inference . VariableElimination (model)
# Calculate the probability of a burglary i f John and Mary calls (0: True
 , 1: False )
posterior probability = infer . query ([ ’ Burglary ’] , evidence={’JohnCa
lls ’ : 0 , ’MaryCalls ’ : 0})
# Print posterior probability
print ( ’ Posterior probability of Burglary i f JohnCalls (True) and MaryC
alls(True ) ’)
print ( posterior probability )
print ()
# Calculate the probability of alarm starting i f there is a burglary and
an earthquake (0: True , 1: False )
posterior probability = infer . query ([ ’ Alarm ’] , evidence= {’ Burglar
y ’ : 0 , ’Earthquake ’ : 0})
# Print posterior probability
print ( ’ Posterior probability of Alarm sounding i f Burglary (True) and
Earthquake(True ) ’)
print ( posterior probability )
print ()
```

Output :

```
Probability distribution, P(Burglary)
+-------------+-------+
| Burglary(0) | 0.001 |
+-------------+-------+
| Burglary(1) | 0.999 |
+-------------+-------+

Probability distribution, P(Earthquake)
+---------------+-------+
| Earthquake(0) | 0.002 |
+---------------+-------+
| Earthquake(1) | 0.998 |
+---------------+-------+

Joint probability distribution, P(Alarm | Burglary, Earthquake)
+------------+---------------+---------------+---------------+---------------+
| Burglary   | Burglary(0)   | Burglary(0)   | Burglary(1)   | Burglary(1)   |
+------------+---------------+---------------+---------------+---------------+
| Earthquake | Earthquake(0) | Earthquake(1) | Earthquake(0) | Earthquake(1) |
+------------+---------------+---------------+---------------+---------------+
| Alarm(0)   | 0.95          | 0.94          | 0.29          | 0.001         |
+------------+---------------+---------------+---------------+---------------+
| Alarm(1)   | 0.05          | 0.06          | 0.71          | 0.999         |
+------------+---------------+---------------+---------------+---------------+

Joint probability distribution, P(JohnCalls | Alarm)
+--------------+----------+----------+
| Alarm        | Alarm(0) | Alarm(1) |
+--------------+----------+----------+
| JohnCalls(0) | 0.9      | 0.05     |
+--------------+----------+----------+
| JohnCalls(1) | 0.1      | 0.95     |
+--------------+----------+----------+

Joint probability distribution, P(MaryCalls | Alarm)
+--------------+----------+----------+
| Alarm        | Alarm(0) | Alarm(1) |
+--------------+----------+----------+
| MaryCalls(0) | 0.7      | 0.01     |
+--------------+----------+----------+
| MaryCalls(1) | 0.3      | 0.99     |
+--------------+----------+----------+

Posterior probability of Burglary if JohnCalls(True) and MaryCalls(True)
+-------------+-----------------+
| Burglary    |   phi(Burglary) |
+=============+=================+
| Burglary(0) |          0.2842 |
+-------------+-----------------+
| Burglary(1) |          0.7158 |
+-------------+-----------------+

Finding Elimination Order: :    0/0 [00:00<?, ?it/s]

   0/0 [00:00<?, ?it/s]
Posterior probability of Alarm sounding if Burglary(True) and Earthquake(True)
+----------+--------------+
| Alarm    |   phi(Alarm) |
+==========+==============+
| Alarm(0) |       0.9500 |
+----------+--------------+
| Alarm(1) |       0.0500 |
+----------+--------------+
```