CL-II 3 IR

July 21, 2025

Title: Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using the standard Heart Disease Data Set (You can use Python ML library classes/API.

[ ]:
```
'''NAME:Aher Swami Sandip
ROLL NO.01
COURSE: AI&DS
CLASS: BE
SUB:Computer Laboratory-II (Information Retrival)'''
```

[ ]:
```
pip install pgmpy==0.1.23
```

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
df=pd.read_csv("heart.csv")
```

[3]:
```
df.head()
```

[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

[4]:
```
df.tail()
```

[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
1020	59	1	1	140	221	0	1	164	1	0.0
1021	60	1	0	125	258	0	0	141	1	2.8

```
1022   47    1   0       110   275    0        0      118      1      1.0
1023   50    0   0       110   254    0        0      159      0      0.0
1024   54    1   0       120   188    0        1      113      0      1.4

      slope  ca  thal  target
1020      2   0     2       1
1021      1   1     3       0
1022      1   1     2       0
1023      2   0     2       1
1024      1   1     3       0
```

[9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
 12  thal      1025 non-null   int64
 13  target    1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

[11]:
```
df.describe()
```

[11]:
```
                age          sex           cp     trestbps        chol  \
count  1025.000000  1025.000000  1025.000000  1025.000000  1025.00000
mean     54.434146     0.695610     0.942439   131.611707   246.00000
std       9.072290     0.460373     1.029641    17.516718    51.59251
min      29.000000     0.000000     0.000000    94.000000   126.00000
25%      48.000000     0.000000     0.000000   120.000000   211.00000
50%      56.000000     1.000000     1.000000   130.000000   240.00000
75%      61.000000     1.000000     2.000000   140.000000   275.00000
max      77.000000     1.000000     3.000000   200.000000   564.00000

               fbs      restecg      thalach        exang      oldpeak  \
```

```
count  1025.000000  1025.000000  1025.000000  1025.000000  1025.000000
mean      0.149268     0.529756   149.114146     0.336585     1.071512
std       0.356527     0.527878    23.005724     0.472772     1.175053
min       0.000000     0.000000    71.000000     0.000000     0.000000
25%       0.000000     0.000000   132.000000     0.000000     0.000000
50%       0.000000     1.000000   152.000000     0.000000     0.800000
75%       0.000000     1.000000   166.000000     1.000000     1.800000
max       1.000000     2.000000   202.000000     1.000000     6.200000

             slope           ca         thal       target
count  1025.000000  1025.000000  1025.000000  1025.000000
mean      1.385366     0.754146     2.323902     0.513171
std       0.617755     1.030798     0.620660     0.500070
min       0.000000     0.000000     0.000000     0.000000
25%       1.000000     0.000000     2.000000     0.000000
50%       1.000000     0.000000     2.000000     1.000000
75%       2.000000     1.000000     3.000000     1.000000
max       2.000000     4.000000     3.000000     1.000000
```

[13]:
```
df.columns
```

[13]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

[15]:
```
df.isnull().sum()
```

[15]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

[17]:
```
from pgmpy.models import BayesianNetwork
from pgmpy.estimators import MaximumLikelihoodEstimator, HillClimbSearch,␣
 ↪BicScore
from pgmpy.inference import VariableElimination
```

[18]:
```
model = BayesianNetwork([
    ('age', 'trestbps'),
    ('age', 'fbs'),
    ('sex', 'trestbps'),
    ('sex', 'chol'),
    ('trestbps', 'target'),
    ('chol', 'target'),
    ('fbs', 'target')
])
```

[19]:
```
model.nodes()
```

[19]:
```
NodeView(('age', 'trestbps', 'fbs', 'sex', 'chol', 'target'))
```

[20]:
```
model.edges()
```

[20]:
```
OutEdgeView([('age', 'trestbps'), ('age', 'fbs'), ('trestbps', 'target'),
('fbs', 'target'), ('sex', 'trestbps'), ('sex', 'chol'), ('chol', 'target')])
```

[21]:
```
model.fit(df, estimator=MaximumLikelihoodEstimator)
```

[22]:
```
for cpd in model.get_cpds():
    print(cpd)
```

```
+---------+------------+
| age(29) | 0.00390244 |
+---------+------------+
| age(34) | 0.00585366 |
+---------+------------+
| age(35) | 0.0146341  |
+---------+------------+
| age(37) | 0.00585366 |
+---------+------------+
| age(38) | 0.0117073  |
+---------+------------+
| age(39) | 0.0136585  |
+---------+------------+
| age(40) | 0.0107317  |
+---------+------------+
| age(41) | 0.0312195  |
+---------+------------+
| age(42) | 0.0253659  |
+---------+------------+
| age(43) | 0.0253659  |
+---------+------------+
| age(44) | 0.035122   |
+---------+------------+
| age(45) | 0.0243902  |
```

age(46)	0.022439
age(47)	0.017561
age(48)	0.022439
age(49)	0.0165854
age(50)	0.0204878
age(51)	0.0380488
age(52)	0.0419512
age(53)	0.0253659
age(54)	0.0517073
age(55)	0.0292683
age(56)	0.0380488
age(57)	0.0556098
age(58)	0.0663415
age(59)	0.044878
age(60)	0.0360976
age(61)	0.0302439
age(62)	0.0360976
age(63)	0.0312195
age(64)	0.0331707
age(65)	0.0263415
age(66)	0.0243902
age(67)	0.0302439
age(68)	0.0117073
age(69)	0.00878049

```
+---------+------------+
| age(70) | 0.0136585  |
+---------+------------+
| age(71) | 0.0107317  |
+---------+------------+
| age(74) | 0.00292683 |
+---------+------------+
| age(76) | 0.00292683 |
+---------+------------+
| age(77) | 0.00292683 |
+---------+------------+
```

```
+---------------+-----+--------------------+---------+
| age           | … | age(77)            | age(77) |
+---------------+-----+--------------------+---------+
| sex           | … | sex(0)             | sex(1)  |
+---------------+-----+--------------------+---------+
| trestbps(94)  | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(100) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(101) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(102) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(104) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(105) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(106) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(108) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(110) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(112) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(114) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(115) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(117) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(118) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(120) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
| trestbps(122) | … | 0.02040816326530612 | 0.0     |
+---------------+-----+--------------------+---------+
```

trestbps(123)	…	0.02040816326530612	0.0
trestbps(124)	…	0.02040816326530612	0.0
trestbps(125)	…	0.02040816326530612	1.0
trestbps(126)	…	0.02040816326530612	0.0
trestbps(128)	…	0.02040816326530612	0.0
trestbps(129)	…	0.02040816326530612	0.0
trestbps(130)	…	0.02040816326530612	0.0
trestbps(132)	…	0.02040816326530612	0.0
trestbps(134)	…	0.02040816326530612	0.0
trestbps(135)	…	0.02040816326530612	0.0
trestbps(136)	…	0.02040816326530612	0.0
trestbps(138)	…	0.02040816326530612	0.0
trestbps(140)	…	0.02040816326530612	0.0
trestbps(142)	…	0.02040816326530612	0.0
trestbps(144)	…	0.02040816326530612	0.0
trestbps(145)	…	0.02040816326530612	0.0
trestbps(146)	…	0.02040816326530612	0.0
trestbps(148)	…	0.02040816326530612	0.0
trestbps(150)	…	0.02040816326530612	0.0
trestbps(152)	…	0.02040816326530612	0.0
trestbps(154)	…	0.02040816326530612	0.0
trestbps(155)	…	0.02040816326530612	0.0
trestbps(156)	…	0.02040816326530612	0.0
trestbps(160)	…	0.02040816326530612	0.0

trestbps(164)	…	0.02040816326530612	0.0
trestbps(165)	…	0.02040816326530612	0.0
trestbps(170)	…	0.02040816326530612	0.0
trestbps(172)	…	0.02040816326530612	0.0
trestbps(174)	…	0.02040816326530612	0.0
trestbps(178)	…	0.02040816326530612	0.0
trestbps(180)	…	0.02040816326530612	0.0
trestbps(192)	…	0.02040816326530612	0.0
trestbps(200)	…	0.02040816326530612	0.0

age	age(29)	age(34)	…	age(74)	age(76)	age(77)
fbs(0)	1.0	1.0	…	1.0	1.0	1.0
fbs(1)	0.0	0.0	…	0.0	0.0	0.0

sex(0)	0.30439
sex(1)	0.69561

sex	sex(0)	sex(1)
chol(126)	0.0	0.004207573632538569
chol(131)	0.0	0.004207573632538569
chol(141)	0.009615384615384616	0.0
chol(149)	0.01282051282051282	0.005610098176718092
chol(157)	0.0	0.005610098176718092
chol(160)	0.009615384615384616	0.0
chol(164)	0.009615384615384616	0.0
chol(166)	0.0	0.005610098176718092

chol(167)	0.0	0.005610098176718092
chol(168)	0.0	0.004207573632538569
chol(169)	0.0	0.005610098176718092
chol(172)	0.0	0.004207573632538569
chol(174)	0.0	0.005610098176718092
chol(175)	0.0	0.015427769985974754
chol(176)	0.0	0.004207573632538569
chol(177)	0.009615384615384616	0.015427769985974754
chol(178)	0.009615384615384616	0.0
chol(180)	0.0	0.005610098176718092
chol(182)	0.0	0.004207573632538569
chol(183)	0.01282051282051282	0.0
chol(184)	0.0	0.004207573632538569
chol(185)	0.0	0.004207573632538569
chol(186)	0.0	0.005610098176718092
chol(187)	0.0	0.005610098176718092
chol(188)	0.0	0.009817671809256662
chol(192)	0.0	0.009817671809256662
chol(193)	0.0	0.008415147265077139
chol(195)	0.009615384615384616	0.0
chol(196)	0.009615384615384616	0.004207573632538569
chol(197)	0.03205128205128205	0.012622720897615708
chol(198)	0.009615384615384616	0.005610098176718092
chol(199)	0.009615384615384616	0.008415147265077139

chol(200)	0.0	0.004207573632538569
chol(201)	0.009615384615384616	0.008415147265077139
chol(203)	0.0	0.016830294530154277
chol(204)	0.019230769230769232	0.021037868162692847
chol(205)	0.01282051282051282	0.004207573632538569
chol(206)	0.0	0.011220196353436185
chol(207)	0.0	0.009817671809256662
chol(208)	0.0	0.008415147265077139
chol(209)	0.022435897435897436	0.0
chol(210)	0.009615384615384616	0.0
chol(211)	0.009615384615384616	0.014025245441795231
chol(212)	0.0	0.025245441795231416
chol(213)	0.009615384615384616	0.004207573632538569
chol(214)	0.009615384615384616	0.004207573632538569
chol(215)	0.009615384615384616	0.0
chol(216)	0.009615384615384616	0.004207573632538569
chol(217)	0.0	0.005610098176718092
chol(218)	0.0	0.011220196353436185
chol(219)	0.009615384615384616	0.009817671809256662
chol(220)	0.01282051282051282	0.011220196353436185
chol(221)	0.0	0.009817671809256662
chol(222)	0.0	0.009817671809256662
chol(223)	0.009615384615384616	0.009817671809256662
chol(224)	0.0	0.005610098176718092

chol(225)	0.02564102564102564	0.0
chol(226)	0.009615384615384616	0.014025245441795231
chol(227)	0.0	0.011220196353436185
chol(228)	0.01282051282051282	0.005610098176718092
chol(229)	0.0	0.016830294530154277
chol(230)	0.0	0.015427769985974754
chol(231)	0.0	0.014025245441795231
chol(232)	0.0	0.009817671809256662
chol(233)	0.0	0.016830294530154277
chol(234)	0.022435897435897436	0.019635343618513323
chol(235)	0.0	0.008415147265077139
chol(236)	0.019230769230769232	0.004207573632538569
chol(237)	0.0	0.005610098176718092
chol(239)	0.009615384615384616	0.014025245441795231
chol(240)	0.01282051282051282	0.014025245441795231
chol(241)	0.009615384615384616	0.0
chol(242)	0.009615384615384616	0.0
chol(243)	0.009615384615384616	0.014025245441795231
chol(244)	0.019230769230769232	0.004207573632538569
chol(245)	0.0	0.012622720897615708
chol(246)	0.0	0.014025245441795231
chol(247)	0.0	0.008415147265077139
chol(248)	0.009615384615384616	0.004207573632538569
chol(249)	0.009615384615384616	0.011220196353436185

chol(250)	0.009615384615384616	0.008415147265077139
chol(252)	0.009615384615384616	0.0
chol(253)	0.0	0.009817671809256662
chol(254)	0.009615384615384616	0.019635343618513323
chol(255)	0.0	0.008415147265077139
chol(256)	0.009615384615384616	0.011220196353436185
chol(257)	0.0	0.004207573632538569
chol(258)	0.009615384615384616	0.009817671809256662
chol(259)	0.0	0.004207573632538569
chol(260)	0.0	0.009817671809256662
chol(261)	0.0	0.009817671809256662
chol(262)	0.0	0.004207573632538569
chol(263)	0.01282051282051282	0.008415147265077139
chol(264)	0.009615384615384616	0.004207573632538569
chol(265)	0.022435897435897436	0.0
chol(266)	0.0	0.008415147265077139
chol(267)	0.009615384615384616	0.004207573632538569
chol(268)	0.022435897435897436	0.0
chol(269)	0.041666666666666664	0.004207573632538569
chol(270)	0.0	0.008415147265077139
chol(271)	0.009615384615384616	0.004207573632538569
chol(273)	0.0	0.008415147265077139
chol(274)	0.0	0.012622720897615708
chol(275)	0.009615384615384616	0.005610098176718092

chol(276)	0.0	0.005610098176718092
chol(277)	0.009615384615384616	0.004207573632538569
chol(278)	0.01282051282051282	0.0
chol(281)	0.0	0.005610098176718092
chol(282)	0.0	0.019635343618513323
chol(283)	0.009615384615384616	0.009817671809256662
chol(284)	0.0	0.005610098176718092
chol(286)	0.0	0.011220196353436185
chol(288)	0.022435897435897436	0.005610098176718092
chol(289)	0.0	0.011220196353436185
chol(290)	0.0	0.004207573632538569
chol(293)	0.0	0.005610098176718092
chol(294)	0.019230769230769232	0.0
chol(295)	0.009615384615384616	0.004207573632538569
chol(298)	0.0	0.008415147265077139
chol(299)	0.0	0.009817671809256662
chol(300)	0.0	0.005610098176718092
chol(302)	0.009615384615384616	0.004207573632538569
chol(303)	0.019230769230769232	0.004207573632538569
chol(304)	0.009615384615384616	0.004207573632538569
chol(305)	0.009615384615384616	0.0
chol(306)	0.009615384615384616	0.0
chol(307)	0.01282051282051282	0.0
chol(308)	0.009615384615384616	0.004207573632538569

chol(309)	0.0	0.015427769985974754
chol(311)	0.0	0.005610098176718092
chol(313)	0.009615384615384616	0.0
chol(315)	0.0	0.009817671809256662
chol(318)	0.009615384615384616	0.005610098176718092
chol(319)	0.01282051282051282	0.0
chol(321)	0.0	0.004207573632538569
chol(322)	0.0	0.005610098176718092
chol(325)	0.009615384615384616	0.004207573632538569
chol(326)	0.0	0.004207573632538569
chol(327)	0.01282051282051282	0.0
chol(330)	0.01282051282051282	0.005610098176718092
chol(335)	0.0	0.011220196353436185
chol(340)	0.009615384615384616	0.0
chol(341)	0.01282051282051282	0.0
chol(342)	0.01282051282051282	0.0
chol(353)	0.0	0.005610098176718092
chol(354)	0.009615384615384616	0.0
chol(360)	0.009615384615384616	0.0
chol(394)	0.009615384615384616	0.0
chol(407)	0.01282051282051282	0.0
chol(409)	0.009615384615384616	0.0
chol(417)	0.009615384615384616	0.0
chol(564)	0.009615384615384616	0.0

```
+-----------+----------------------+----------------------+
+-----------+--------------+-----+--------------+---------------+
| chol      | chol(126)    | … | chol(564)     | chol(564)     |
+-----------+--------------+-----+--------------+---------------+
| fbs       | fbs(0)       | … | fbs(1)        | fbs(1)        |
+-----------+--------------+-----+--------------+---------------+
| trestbps  | trestbps(94) | … | trestbps(192) | trestbps(200) |
+-----------+--------------+-----+--------------+---------------+
| target(0) | 0.5          | … | 0.5           | 0.5           |
+-----------+--------------+-----+--------------+---------------+
| target(1) | 0.5          | … | 0.5           | 0.5           |
+-----------+--------------+-----+--------------+---------------+
```

[23]:
```python
# Perform inference
inference = VariableElimination(model)

# Example query: Probability of having heart disease given specific conditions
query_result = inference.query(variables=['target'], evidence={'age': 55, 'sex':
 ↪ 1, 'trestbps': 140, 'chol': 240, 'fbs': 0})

print(query_result)
```

```
+-----------+---------------+
| target    |   phi(target) |
+===========+===============+
| target(0) |        0.5000 |
+-----------+---------------+
| target(1) |        0.5000 |
+-----------+---------------+
```

[ ]: