Problema Monty Hall

Problema Monty Hall este o problemă celebră din teoria probabilităților. Situația este următoarea:

Ești la un concurs și ți se oferă să alegi între trei uși. În spatele uneia dintre ele se află o mașină, iar în spatele celorlalte două se află capre. Alegi o ușă, să zicem ușa 1, iar prezentatorul, care știe ce este în spatele fiecărei uși, deschide o altă ușă (cu o capră), de exemplu ușa 3. Apoi te întreabă dacă vrei să schimbi alegerea către ușa 2. Întrebarea este: Este avantajos pentru tine să schimbi ușa?

Intuitiv, pare că schimbarea ușii nu ar trebui să aibă niciun efect asupra șanselor tale. Cu toate acestea, folosind teorema lui Bayes, se poate demonstra că șansele de a câștiga cresc dacă schimbi ușa.

Interpretare Probabilistică:

Avem trei variabile aleatorii:

- **C**: ușa aleasă de concurent ({1, 2, 3}),
- **H**: ușa deschisă de prezentator ({1, 2, 3}),
- **P**: ușa cu premiul ({1, 2, 3}).

Premiul este plasat aleatoriu în spatele uneia dintre uși, deci probabilitatea ca premiul să fie în spatele oricărei uși este de 1/3. Aceeași probabilitate este valabilă și pentru alegerea concurentului:

```
P(C):
+----------+----------+----------+----------+
|    C     |     0    |     1    |     2    |
+----------+----------+----------+----------+
|          |    0.33  |    0.33  |    0.33  |
+----------+----------+----------+----------+

P(P):
+----------+----------+----------+----------+
|    P     |     0    |     1    |     2    |
+----------+----------+----------+----------+
|          |    0.33  |    0.33  |    0.33  |
+----------+----------+----------+----------+
```

Prezentatorul va deschide o ușă diferită de cea aleasă de concurent, care ascunde o capră. Matricea de probabilități condiționate pentru acțiunea prezentatorului, ținând cont de alegerea concurentului și de locația premiului, arată astfel:

```
P(H | P, C):
+------+------+------+------+------+------+------+------+------+------+
|   C  |         0         |         1         |         2         |
+------+------+------+------+------+------+------+------+------+------+
|   P  |   0  |   1  |   2  |   0  |   1  |   2  |   0  |   1  |   2  |
+------+------+------+------+------+------+------+------+------+------+
|  H=0 |   0  |   0  |   0  |   0  |  0.5 |   1  |   0  |   1  |  0.5 |
+------+------+------+------+------+------+------+------+------+------+
|  H=1 |  0.5 |   0  |   1  |   0  |   0  |   0  |   1  |   0  |  0.5 |
+------+------+------+------+------+------+------+------+------+------+
|  H=2 |  0.5 |   1  |   0  |   1  |  0.5 |   0  |   0  |   0  |   0  |
+------+------+------+------+------+------+------+------+------+------+
```

Dacă concurentul a ales ușa 0, iar prezentatorul a deschis ușa 2, trebuie să calculăm probabilitatea ca premiul să fie în spatele unei uși având aceste date:

```
P(P | H=2, C=0)
```

Implementare în Python:

```
from pgmpy.models import BayesianNetwork
from pgmpy.factors.discrete import TabularCPD

# Definirea structurii rețelei bayesiene
model = BayesianNetwork([("C", "H"), ("P", "H")])

# Definirea CPD-urilor (probabilități condiționate)
cpd_c = TabularCPD("C", 3, [[0.33], [0.33], [0.33]])
cpd_p = TabularCPD("P", 3, [[0.33], [0.33], [0.33]])
cpd_h = TabularCPD(
    "H",
    3,
    [
        [0, 0, 0, 0, 0.5, 1, 0, 1, 0.5],
        [0.5, 0, 1, 0, 0, 0, 1, 0, 0.5],
        [0.5, 1, 0, 1, 0.5, 0, 0, 0, 0],
    ],
```

```
    evidence=["C", "P"],
    evidence_card=[3, 3],
)

# Asocierea CPD-urilor cu structura rețelei
model.add_cpds(cpd_c, cpd_p, cpd_h)

# Verificarea modelului
model.check_model()

# Inferarea probabilităților posterioare
from pgmpy.inference import VariableElimination

infer = VariableElimination(model)
posterior_p = infer.query(["P"], evidence={"C": 0, "H": 2})
print(posterior_p)
```

Codul de mai sus folosește rețele Bayesiene pentru a calcula probabilitatea ca premiul să fie în spatele ușii 1, având în vedere că concurentul a ales ușa 0 și prezentatorul a deschis ușa 2.