**Academic Year: 2025-26** **Name of Student: Amit Rajegore**
**Semester: VII** **Student ID:22104025**
**Class / Branch/ Div: BE- IT C** **Roll No.-18**
**Subject: Data Science Lab** **Date of Submission:7-7-25**
**Name of Instructor: Prof. Shraddha Birje**

## Experiment No.1

**Aim:** To implement Inferencing with Bayesian Network in Python.

+ Code + Text

```
pip install pgmpy
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pgmpy in /usr/local/lib/python3.7/dist-packages (0.1.19)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.7/dist-packages (from pgmpy) (0.10.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from pgmpy) (4.64.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from pgmpy) (1.21.6)
Requirement already satisfied: torch in /usr/local/lib/python3.7/dist-packages (from pgmpy) (1.12.0+cu113)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from pgmpy) (1.3.5)
Requirement already satisfied: networkx in /usr/local/lib/python3.7/dist-packages (from pgmpy) (2.6.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-packages (from pgmpy) (1.0.2)
Requirement already satisfied: pyparsing in /usr/local/lib/python3.7/dist-packages (from pgmpy) (3.0.9)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from pgmpy) (1.7.3)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from pgmpy) (1.1.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas->pgmpy) (2022.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas->pgmpy) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas->pgmpy) (1.15.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn->pgmpy) (3.1.0)
Requirement already satisfied: patsy>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from statsmodels->pgmpy) (0.5.2)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from torch->pgmpy) (4.1.1)
```

```
from pgmpy.models import BayesianNetwork
from pgmpy.factors.discrete import TabularCPD
import networkx as nx
import pylab as plt
```

### Defining Bayesian Structure

```
model = BayesianNetwork([('Guest', 'Host'), ('Price', 'Host')])
```

=

### Defining the CPDs:

```
cpd_guest = TabularCPD('Guest', 3, [[0.33], [0.33], [0.33]])
cpd_price = TabularCPD('Price', 3, [[0.33], [0.33], [0.33]])
cpd_host = TabularCPD('Host', 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=['Guest', 'Price'], evidence_card=[3, 3])
```

### Associating the CPDs with the network structure

```
model.add_cpds(cpd_guest, cpd_price, cpd_host)
model.check_model()
```

```
True
```

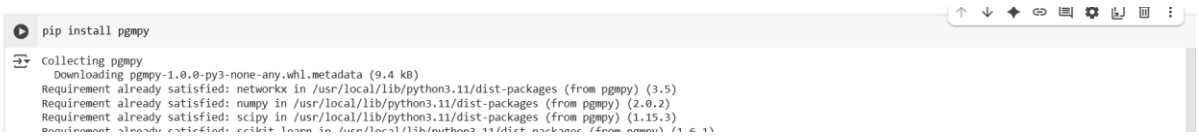
### Infering the posterior probability

```
from pgmpy.inference import VariableElimination
```

```
infer = VariableElimination(model)
posterior_p = infer.query(['Host'], evidence={'Guest':2 , 'Price': 2})
print(posterior_p)
nx.draw(model, with_labels=True)
plt.savefig('model.png')
plt.close()
```

**Academic Year: 2025-26** **Name of Student: Amit Rajegore**
**Semester: VII** **Student ID:22104025**
**Class / Branch/ Div: BE- IT C** **Roll No.-18**
**Subject: Data Science Lab** **Date of Submission:7-7-25**
**Name of Instructor: Prof. Shraddha Birje**


## Output Screenshots :




**Academic Year: 2025-26** **Name of Student: Amit Rajegore**
**Semester: VII** **Student ID:22104025**
**Class / Branch/ Div: BE- IT C** **Roll No.-18**
**Subject: Data Science Lab** **Date of Submission:7-7-25**
**Name of Instructor: Prof. Shraddha Birje**

```
[5] cpd_guest = TabularCPD(variable='Guest', variable_card=3, values=[[0.33], [0.33], [0.34]])
    cpd_price = TabularCPD(variable='Price', variable_card=3, values=[[0.33], [0.33], [0.34]])

    cpd_host = TabularCPD(
        variable='Host',
        variable_card=3,
        values=[
            [0, 0, 0, 0.5, 1, 0, 1, 0.5],
            [0.5, 0, 0, 0, 0, 0, 0, 0.5],
            [0.5, 1, 1, 0.5, 0, 1, 0, 0]
        ],
        evidence=['Guest', 'Price'],
        evidence_card=[3, 3]
    )
```

```
model.add_cpds(cpd_guest, cpd_price, cpd_host)
model.check_model()
```

```
True
```

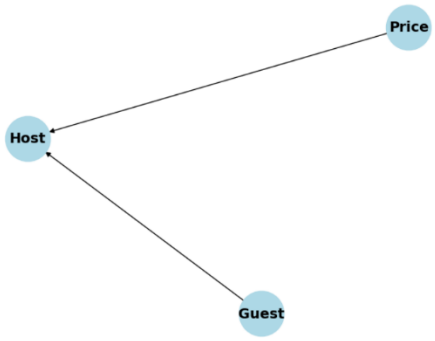
```
[7] from pgmpy.inference import VariableElimination
```

```
infer = VariableElimination(model)
posterior_p = infer.query(['Host'], evidence={'Guest': 2, 'Price': 2})
print(posterior_p)
```

```
+---------+-------------+
| Host    |   phi(Host) |
+=========+=============+
| Host(0) |      0.5000 |
+---------+-------------+
| Host(1) |      0.5000 |
+---------+-------------+
| Host(2) |      0.0000 |
+---------+-------------+
```

```
G = nx.DiGraph(model.edges())

nx.draw(G, with_labels=True, node_size=2000, node_color='lightblue', font_size=14, font_weight='bold')
plt.savefig('model.png')
plt.show()
```




**Conclusion:** In conclusion, we implemented the Monty Hall problem using a Bayesian network with the Python library pgmpy. We also visualized the model by plotting a Directed Acyclic Graph (DAG) using networkx and pylab. This helped us understand the probability shifts when switching doors.