

```
!pip install pgmpy -q
```

2.0/2.0 MB 25.7 MB/s eta 0:00:00  
756.0/756.0 kB 40.4 MB/s eta 0:00:00

```
import pandas as pd
import numpy as np
```

```
from pgmpy.models import DiscreteBayesianNetwork
from pgmpy.estimators import MaximumLikelihoodEstimator
from pgmpy.inference import VariableElimination
```

```
data = {
    'age': [63, 67, 67, 37, 41, 56, 62, 57, 63, 53],
    'sex': [1, 1, 1, 1, 0, 1, 0, 0, 1, 1],
    'cp': [1, 4, 4, 3, 2, 2, 4, 4, 4, 4],
    'trestbps': [145, 160, 120, 130, 130, 120, 140, 120, 130, 140],
    'chol': [233, 286, 229, 204, 268, 236, 268, 354, 254, 203],
    'fbs': [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    'thalach': [150, 108, 129, 172, 160, 178, 160, 163, 147, 155],
    'heartdisease': [0, 2, 1, 0, 3, 0, 3, 0, 2, 1]
}
```

```
df = pd.DataFrame(data)
```

```
print("Sample Heart Disease Dataset:")
print(df.head())
print(f"\nDataset shape: {df.shape}")
```

Sample Heart Disease Dataset:

	age	sex	cp	trestbps	chol	fbs	thalach	heartdisease
0	63	1	1	145	233	1	150	0
1	67	1	4	160	286	0	108	2
2	67	1	4	120	229	0	129	1
3	37	1	3	130	204	0	172	0
4	41	0	2	130	268	0	160	3

Dataset shape: (10, 8)

```
model = DiscreteBayesianNetwork([
    ('age', 'trestbps'),
    ('age', 'heartdisease'),
    ('sex', 'heartdisease'),
    ('cp', 'heartdisease'),
    ('trestbps', 'heartdisease'),
    ('chol', 'heartdisease'),
    ('fbs', 'heartdisease')
])
```

```
print("Training Bayesian Network...")
model.fit(df, estimator=MaximumLikelihoodEstimator)
print("Model trained successfully!")
```

Training Bayesian Network...  
Model trained successfully!

```
inference = VariableElimination(model)
print("Inference engine created!")
```

Inference engine created!

```
query1 = inference.query(['heartdisease'], evidence={'age': 37})
print("Heart Disease Probability for Age=30:")
print(query1)
```

Heart Disease Probability for Age=30:

heartdisease	phi(heartdisease)
heartdisease(0)	0.2542
heartdisease(1)	0.2486
heartdisease(2)	0.2486
heartdisease(3)	0.2486

```
query2 = inference.query(['heartdisease'], evidence={'age': 56, 'sex': 1})
print("\nHeart Disease Probability for Male, Age=50:")
print(query2)
```

Heart Disease Probability for Male, Age=50:

heartdisease	phi(heartdisease)
heartdisease(0)	0.2620
heartdisease(1)	0.2460
heartdisease(2)	0.2460
heartdisease(3)	0.2460

```
def predict_heart_disease(**evidence):
    try:
        result = inference.query(['heartdisease'], evidence=evidence)
        probs = result.values
        max_idx = np.argmax(probs)
        confidence = probs[max_idx]
        return max_idx, confidence
```

```
except:  
    return "Unable to predict", 0.0
```

```
test_cases = [  
    {'age': 37, 'sex': 1, 'cp': 2},  
    {'age': 56, 'sex': 0, 'trestbps': 140},  
    {'age': 62, 'chol': 233}  
]
```

```
for i, case in enumerate(test_cases):  
    pred, conf = predict_heart_disease(**case)  
    print(f"Case {i+1}: {case}")  
    print(f"Predicted class: {pred}, Confidence: {conf:.3f}\n")
```

Case 1: {'age': 37, 'sex': 1, 'cp': 2}  
Predicted class: 0, Confidence: 0.250

Case 2: {'age': 56, 'sex': 0, 'trestbps': 140}  
Predicted class: 0, Confidence: 0.250

Case 3: {'age': 62, 'chol': 233}  
Predicted class: 0, Confidence: 0.250